# Using Linguistic Knowledge for Improving Automatic Speech Recognition Accuracy in Air Traffic Control

Master's Thesis in Computer Science

Van Nhan Nguyen

May 18, 2016
Halden, Norway

Østfold University College www.hiof.no

# Abstract

Recently, a lot of research has been conducted to bring Automatic Speech Recognition (ASR) into various areas of Air Traffic Control (ATC), such as air traffic control simulation and training, monitoring live operators for with the aim of safety improvements, air traffic controller workload measurement and conducting analysis on large quantities controller-pilot speech. However, due to the high accuracy requirements of the ATC context and its unique challenges such as call sign detection, the problem of poor input signal quality, the problem of ambiguity, the use of non-standard phraseology and the problem of dialects, accents and multiple languages, ASR has not been widely adopted in this field. In this thesis, in order to take advantage of the availability of linguistic knowledge, particularly syntactic and semantic knowledge, in the ATC domain, I aim at using different levels of linguistic knowledge to improve the accuracy of ASR systems via three steps: language modeling, n-best list re-ranking using syntactic knowledge and n-best list re-ranking using semantic knowledge.

Firstly, I propose a context-dependent class n-gram language model by combining the hybrid class n-gram and context-dependent language modeling approaches to address the two main challenges of language modeling in ATC, which are the lack of ATC-related corpora for training and the location-based data problem. Secondly, I use the first level of linguistic knowledge, syntactic knowledge to perform n-best list re-ranking. To facilitate this, I propose a novel feature called syntactic score and a WER-Sensitive Pairwise Perceptron algorithm. I use the perceptron algorithm to combine the proposed feature with the speech decoder's confidence score feature to re-rank the n-best list. Thirdly, I combine syntactic knowledge with the next level of linguistc knowledge, semantic knowledge to re-rank the n-best list. To do this, I propose a feature called semantic relatedness. I use the WER-Sensitive Pairwise Perceptron algorithm to combine the proposed feature with the syntactic score and speech decoder's confidence score features perform n-best list re-ranking. Finally, I build a baseline ASR system based on the Pocketsphinx recognizer from the CMU Sphinx framework, the CMUSphinx US English generic acoustic model and the generic cmudict_SPHINX_40 pronunciation dictionary and the three above-mentioned approaches.

I evaluate the baseline ASR system in terms of Word Error Rate (WER) on the well known ATCOSIM Corpus of Non-prompted Clean Air Traffic Control Speech (ATCOSIM) and my own Air Traffic Control Speech Corpus (ATCSC). The evaluation results show that the combination of the three proposed approaches reduces the WER of the baseline ASR system by 20.95% compared with traditional n-gram language models in recognizing general clearances from the ATCSC corpus.

This thesis makes three main contributions. Firstly, It addresses the two main challenges of language modeling in ATC, which are the lack of ATC-related corpora for training and the problem of location-based data, by proposing a novel language model called context-dependent class n-gram language model. The second contribution is the use of linguistic knowledge in post-processing, particularly n-best list re-ranking using syntactic and semantic knowledge, to improve the accuracy of ASR systems in ATC. Finally, it demonstrates that linguistic knowledge has great potential in addressing the existing challenges of ASR in ATC and facilitating the integration of ASR technologies into the ATC domain.

**Keywords:** Language Modeling, N-gram, Class N-gram, N-best List Re-ranking, Syntactic Knowledge, Semantic Knowledge, Automatic Speech Recognition, Air Traffic Control.

# Acknowledgments

After an intensive period of ten months, today is the day: writing this note of thanks is the finishing touch on my thesis. It has been a period of intense learning for me, not only in the scientific arena, but also on a personal level. Writing this thesis has had a big impact on me. I would like to reflect on the people who have supported and helped me so much throughout this period.

I would first like to express my sincere gratitude to my thesis advisor Assoc. Prof. Harald Holone for the continuous support of my Master's study and related research, for his patience, motivation, and immense knowledge. The door to Assoc. Prof. Holone office was always open whenever I ran into a trouble spot or had a question about my research or writing.

I would also like to thank The Institute for Energy Technology (John E. Simensen and Christian Raspotnig), Edda Systems AS, and WP3 of "Smart Buildings for Welfare (SBW)" at Østfold University College for support in the work with this thesis and related research.

I would also like to thank all my friends, classmates and labmates, especially Tien Tai Huynh and Jonas Nordström, for the stimulating discussions, for helping me through the many hours spent collecting data, for the sleepless nights we were working together before deadlines, and for all the fun we have had in the last ten months.

Finally, I must express my very profound gratitude to my parents, sisters and brothers for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

# Contents

**Abstract** **ii**

**Acknowledgments** **iii**

**List of Figures** **vii**

**List of Tables** **ix**

**1 Introduction** **1**

1.1 Background and motivation . . . . . . . . . . . . . . . . . . . . . . . . . . 1

1.2 Research statement and method . . . . . . . . . . . . . . . . . . . . . . . 2

1.3 Report Outline . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

**2 Theory and Related Work** **5**

2.1 Air Traffic Control (ATC) . . . . . . . . . . . . . . . . . . . . . . . . . . 5

2.2 Automatic Speech Recognition (ASR) . . . . . . . . . . . . . . . . . . . . 12

2.3 Related Work . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

**3 ASR Frameworks and Existing ATC-Related Corpora** **21**

3.1 ASR Frameworks . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21

3.2 Existing ATC-Related Corpora . . . . . . . . . . . . . . . . . . . . . . . . 25

**4 Case and Experimental Settings** **29**

4.1 Case . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29

4.2 Experimental Settings . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30

4.3 Air Traffic Control Speech Corpus (ATCSC) . . . . . . . . . . . . . . . . 32

**5 Research Findings** **33**

5.1 Language Modeling . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33

5.2 N-best List Re-ranking Using Syntactic Knowledge . . . . . . . . . . . . . 34

5.3 N-best List Re-ranking Using Semantic Knowledge . . . . . . . . . . . . . 35

5.4 Findings in summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36

**6 Discussion** **37**

6.1 Research Questions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37

6.2 Possibilities of Linguistic Knowledge in ASR in ATC . . . . . . . . . . . . 38

6.3 Linguistic Knowledge and Challenges of ASR in ATC . . . . . . . . . . . 39

**7 Conclusion and Further Work** **43**
7.1 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43
7.2 Further Work . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 44

**Bibliography** **50**

**A Language Modeling** **51**

**B N-best List Re-ranking Using Syntactic Knowledge** **63**

**C N-best List Re-ranking Using Semantic Knowledge** **70**

**D Possibilities, Challenges and the State of the Art of ASR in ATC** **77**

**E ATC Phraseology** **88**

# List of Figures

2.1 Structure of speech recognition system . . . . . . . . . . . . . . . . . . . . . . 13

4.1 Automated pilot system for air traffic control simulation and training . . . 30

# List of Tables

2.1 Examples of ICAO standrad phraseologies . . . . . . . . . . . . . . . . . . . . 6
2.2 Aviation spelling alphabet . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
2.3 Aviation numbers . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
2.4 Examples of syntactic knowledge in ATC . . . . . . . . . . . . . . . . . . . . 11

3.1 ASR open source frameworks/projects . . . . . . . . . . . . . . . . . . . . . . 25
3.2 Summary of features of ATC-related corpora . . . . . . . . . . . . . . . . . . 27

# Chapter 1

# Introduction

## 1.1 Background and motivation

In the past few years, the steadily increasing levels of air traffic world wide poses corresponding capacity challenges for air traffic control (ATC) services [45]. According to the “Outlook for Air Transport to the Year 2025” report [47] of International Civil Aviation Organization (ICAO), passenger traffic on the major international route group and aircraft movements in terms of aircraft departures and aircraft kilometers flown are expected to increase at average annual rates of 3 to 6 per cent and 3.6 to 4.1 per cent respectively through to the year 2025. Thus, ATC operations have to be investigated, reviewed and improved in order to be able to meet with the increasing demands

In ATC operations, most of the tasks of air traffic controllers involve verbal communications with pilots. This means that, the safety and performance of ATC operations depend heavily on the quality of these communications. Recently, with the aim of improving both safety and performance of ATC operations, many attempts have been made to integrate Automatic Speech Recognition (ASR) technologies into the ATC domain to facilitate applications such as air traffic control simulation and training, air traffic control workload measurement and balancing, analysis on large quantities control-pilot speech.

However, ASR technologies have not been successfully adopted in the ATC domain because of it high accuracy requirements and unique challenges. In my previous work [45], I pointed out that there are five major challenges to overcome in order to successfully apply ASR in ATC. The challenges are call sign detection, the problem of poor input signal quality, the problem of ambiguity, the use of non-standard phraseology and the problem of dialects, accents and multiple languages. I also identified four main approaches which can be used to improve the accuracy of ASR systems in the ATC domain. The approaches are syntactic analysis, semantic analysis, pragmatic analysis and dialects, accents and languages detection. While the first three approaches focus on integrating linguistic knowledge into ASR systems via language modeling or post-processing, the last approach adapts ASR systems based on speakers accent, dialect and language. In this thesis, in order to take advantage of the availability of linguistic knowledge in ATC, I aim at using linguistic knowledge, particularly syntactic and semantic knowledge, to improve the accuracy of ASR systems by performing language modeling and post-processing.

## 1.2 Research statement and method

### 1.2.1 Research questions

As stated above, the primary goal of this thesis is to use linguistic knowledge to improve the accuracy of ASR systems in ATC. To achieve this goal, I first carefully study the use of linguistic knowledge in the ATC domain and language modeling approaches. Thus, having a general view and good understanding of the possibilities of linguistic knowledge in ASR in ATC. I then address the existing challenges of ASR in ATC and improve the accuracy of ASR systems by integrating linguistic knowledge, particularly syntactic and semantic knowledge into language modeling and post-processing. Basically, at the end of this thesis, I need to answer following research questions:

**RQ** How can linguistic knowledge be used to improve automatic speech recognition accuracy in air traffic control?
Secondary relevant research questions are:

**RQ1.1** *Which type of language model is well suited for use in automatic speech recognition system in air traffic control domain?*

**RQ1.2** *To what extent can syntactic analysis improve the accuracy of speech recognition in air traffic control domain?*

**RQ1.3** *To what extent can semantic analysis improve the accuracy of speech recognition in air traffic control domain?*

The research questions I introduce here are aimed for facilitating the integration of ASR technologies into the ATC field in general. However, since the special case of this project is to develop an ASR system for ATC simulation and training, I narrow down the scope of this project to take advantage of the opportunities offered by the ATC simulation and training context. More details about the special case can be found in Chapter 4. In Chapter 6 I will revisit these research questions and discuss how can the findings from this project be adapted for use in both ATC live operations and ATC simulation and training.

### 1.2.2 Method

To answer the research questions, following steps are needed to be followed. While the first four steps are for addressing the three secondary research questions, RQ1.1, RQ1.2 and RQ1.3, the last step is for tackling the main research question, RQ1.

- Select an ASR framework and an ATC-related corpus for training - I first review ten well-known ASR open source frameworks including Bavieca, CMU Sphinx, Hidden Markov Model Toolkit (HTK), Julius, Kaldi, RWTH ASR, SPRAAK, CSLU Toolkit, The transLectures-UPV toolkit (TLK) and iATROS in order to select a framework for developing a baseline ASR system. I then review five existing ATC-related corpora including ATCOSIM, LDC94S14A, HIWIRE, Air Traffic Control Communication Speech Corpus and Air Traffic Control Communication corpus in order to select a corpus for training. More details about the frameworks and the corpora can be found in Chapter 3.

- Utilize linguistic knowledge in language modeling in ATC *(RQ1.1)* - I first evaluate different language models (n-gram, class n-gram) in terms of Word Error Rate (WER) and Real Time Factor (RTF) on the baseline ASR system in order to select a well-suited language model for use in ATC. I then improve the selected language model by integrating linguistic knowledge into the language modeling process. Finally, I use the baseline ASR system to evaluate the language model on the well known ATCOSIM Corpus of Non-prompted Clean Air Traffic Control Speech (ATCOSIM) and my own Air Traffic Control Speech Corpus (ATCSC).

- Integrate syntactic knowledge into post-processing *(RQ1.2)* - I first study different approaches (e.g., language modeling, post-processing) for using syntactic knowledge in improving the accuracy of ASR systems in general. I then analyze the use of syntactic knowledge in the ATC domain in order to select a well-suited approach for facilitating the integration of syntactic knowledge into post-processing. Finally, I use the baseline ASR system to evaluate the selected approach on the ATCOSIM and ATCSC corpora.

- Integrate semantic knowledge into post-processing *(RQ1.3)* - I first look into different approaches (e.g., language modeling, post-processing) for combining syntactic and semantic knowledge in post-processing to improve the accuracy of ASR systems in general. I then analyze the use of syntactic and semantic knowledge in the ATC domain in order to select a well-suited approach for facilitating the integration of semantic knowledge into post-processing. Finally, I use the baseline ASR system to evaluate the selected approach on the ATCOSIM and ATCSC corpora.

- Discuss the possibilities and challenges of linguistic knowledge in improving the accuracy ASR systems in ATC (RQ1). Firstly, I build a Proof-of-Concept (POC) ASR system based on the selected framework and the above-mentioned three approaches. Secondly, I evaluate the system in terms of WER on the ATCOSIM and ATCSC corpora. Finally, I conduct a detailed analysis of the evaluation results and discuss the possibilities and challenges of linguistic knowledge in ASR in ATC to answer the main research question of this thesis *"How can linguistic knowledge be used to improve automatic speech recognition accuracy in air traffic control?"*.

More details about the research questions and their corresponding methods can be found in Chapter 5, as well as the three included papers in Appendix A, Appendix B and Appendix C.

## 1.3 Report Outline

The remainder of this thesis is structured as follows: Chapter 2 presents background knowledge covering the ATC field in general, ASR technologies, as well as relevant related work, before I present a brief review of ten ASR open source frameworks and five existing ATC-related corpora in Chapter 3. In Chapter 4, I describe the special case that forms the basic of this project, four experiments designed to address the above-mentioned research questions, together with a brief summary of how the case affects the design of the experiments. The end of the chapter contains a description of my own Air Traffic Control Speech Corpus (ATCSC) which is recorded with the aim of simulating a training and simulation setting. Chapter 5 summarizes the research findings from each of the three included papers. In Chapter 6 and Chapter 7, I discuss and conclude my work, as well as present suggestions for further work. Following that, the three paper included in thesis, my previous work and a full list of ICAO standard phraseologies can be found as appendices.

# Chapter 2

# Theory and Related Work

This chapter has three main purposes. Firstly, it presents a brief description of the Air Traffic Control (ATC) field in general, with special attention paid to cover standard phraseology recommend by International Civil Aviation Organization (ICAO), ATC control units and sources of knowledge in speech in ATC. The second purpose of this chapter is to describe the structure of an Automatic Speech Recognition (ASR) system and its modules, together with methods for measuring ASR systems performance, as well as language modeling approaches. The end of this chapter contains a summary of relevant related work covering ASR in ATC.

## 2.1 Air Traffic Control (ATC)

According to the Oxford English Dictionary [61], Air Traffic Control (ATC) is “the ground-based personnel and equipment concerned with controlling and monitoring air traffic within a particular area”. The main purpose of ATC systems is to prevent collisions, provide safety, organize aircraft operating in the system and expedite air traffic [1]. With the steady increase in air traffic over the past few years, ATC has become more and more important. This increase has also resulted in more complex procedures, regulations and technical systems [54]. Thus, ATC systems have to be continuously improved to meet the evolving demands in air traffic.

In ATC, air traffic controller have an incredibly large responsibility for maintaining the safe, orderly and expeditious conduct of air traffic. Given the important roles of air traffic control and air traffic controllers, there is an ongoing need to strengthen training and testing of the operators. Further, being able to simulate the working environment of controllers enables increased safety through the use of support systems that can assist controllers and improve procedures, and by analyzing controller-pilot communications [45].

### 2.1.1 ICAO Standard Phraseologies

In ATC, air traffic controllers and pilots are usually recommended to use ICAO standard phraseologies in their communications. However, when the circumstances differ, air traffic controllers and pilots will be expected to use plain language. In order to avoid possible confusion and misunderstandings in communication, the plain language should be clear and concise as possible [29][26]. The phraseologies recommended by ICAO can be grouped based on types of air traffic control services as follows:

- ATC Phraseologies
  - General
  - Area control services
  - Approach control services
  - Phraseologies for us on and in the vicinity of the aerodrome
  - Coordination between ATS units
  - Phraseologies to be used related to CPDLC
- ATS Surveillance Service Phraseologies
  - General ATS surveillance service phraseologies
  - Radar in approach control service
  - Secondary surveillance radar (SSR) and ADS-B phraseologies
- Automatic Dependent Surveillance - Contract (ADS-C) Phraseologies
- Alerting Phraseologies
- Ground Crew/Flight Crew Phraseologies

Examples of the ICAO standard phraseologies in three different circumstances, description of levels, level changes and vectoring instructions, as well as how air traffic controllers and pilots use the phraseologies in their communication are shown in Table 2.1.

Table 2.1: Examples of ICAO standrad phraseologies

| Circumstancs | Phraseologies | Examples |
|---|---|---|
| Description of levels | FLIGHT LEVEL (number); or<br>(number) METERS; or<br>(number) FEET. | FLIGHT LEVEL 120<br>3000 METERS<br>6000 FEET |
| Level changes | (callsign) CLIMB (or DESCEND);<br>followed as necessary by:<br>TO (level); | CLIMB TO 6000 FEET |
| Vectoring instructions | FLY HEADING (three digits);<br>TURN LEFT HEADING (three digits) | FLY HEADING 120<br>TURN LEFT HEADING 120 |

In ATC operations, word spelling and pronouncing numbers are very common tasks. However, the pronunciation of letters in the alphabet and numbers may vary according to the language habit, accent and dialect of the speakers. Thus, these tasks frequently cause misunderstandings in communication between controllers and pilots. In order to eliminate wide variations in pronunciation and avoid the misunderstandings, ICAO recommends new ways of pronouncing numbers and letters in the alphabet [26]. Table 2.2 and Table 2.3 contain pronunciations of the aviation alphabet and numbers which are provided by ICAO. The syllables printed in capital letters in the tables are the indications of word stresses. For example, in the word ECKO (Eck oh), the primary emphasis is ECK. By using the pronunciation tables, “WTO 98.54” can be pronounced as “WISSkey TANGgo OSScar NINer AIT DAYSEEMAL FIFE FOWer”.

Table 2.2: Aviation spelling alphabet

| Word | Pronunciation | Word | Pronunciation |
|---|---|---|---|
| A - ALFA | AL fah | N - NOVEMBER | no VEM ber |
| B - BRAVO | BRAH voh | O - OSCAR | OSS car |
| C - CHARLIE | CHAR lee OR SHAR lee | P - PAPA | pah PAH |
| D - DELTA | DELL tah | Q - QUEBEC | keh BECK |
| E - ECHO | ECK oh | R - ROMEO | ROW me oh |
| F - FOXTROT | FOKS trot | S - SIERRA | see AIR rah |
| G - GOLF | golf | T - TANGO | TANG go |
| H - HOTEL | hoh TEL | U - UNIFORM | YOU nee form OR OO nee form |
| I - INDIA | IN dee ah | V - VICTOR | VIK tah |
| J - JULIET | JEW lee ETT | W - WHISKEY | WISS key |
| K - KILO | KEY loh | X - X-RAY | ECKS ray |
| L - LIMA | LEE mah | Y - YANKEE | YANG key |
| M - MIKE | mike | Z - ZULU | ZOO loo |

Table 2.3: Aviation numbers

| Term | Pronunciation | Term | Pronunciation |
|---|---|---|---|
| 0 | ZE RO | 7 | SEV en |
| 1 | WUN | 8 | AIT |
| 2 | TOO | 9 | NIN er |
| 3 | THREE | decimal | DAY SEE MAL |
| 4 | FOW er | hundred | HUN dred |
| 5 | FIFE | thousand | TOU SAND |
| 6 | SIX | | |

In order to conduct a detailed analysis of ICAO standard phraseologies, I extract a full list of phraseologies from “Chapter 12 - Phraseologies, Doc 4444/510: Procedures for Air Navigation Services - Air Traffic Management 15th Edition” [29]. The list can be found in Appendix E. The number of phraseologies without call signs, unit names and navigational aids/fixes is 538 words. Thus, the size of vocabulary used in the ATC domain including the aviation spelling alphabet and aviation numbers is about 577 words.

With the advances in modern ASR technologies, recognizing 577 words is not a difficult task. However, in ATC live operations, the number of phraseologies used by controllers and pilots is much larger than 577 words. For example, in the ATCOSIM corpus [33] the total number of words used by controllers and pilots is more than 850 words. In live ATC operations, with the large number of call signs (about 6000) [28], as well as a huge number of unit names and navigational aids/fixes, the size of vocabulary will be dramatically increased.

### 2.1.2 Air Traffic Control Units

ATC units are designed to give one or more of the following services [27]:

- Air traffic control service, which is to prevent collisions, provide safety, organize aircraft and expedite air traffic. Based on the control areas where air traffic control services are provided, the services can be categorized into three groups as follows:

  - Aerodrome control service, which is responsible for preventing collisions and organizing air traffic on taxiways, runways and in Control Zone (CTR).
  - Approach control service, which is to prevent collisions and organize air traffic between arriving and departing aircraft in Terminal Control Area (TMA).
  - Area control service, which is responsible for preventing collisions and organizing air traffic between en-route aircraft in Control Areas (CTA) and along Airways (AWY).

- Flight information service, which provides useful information (e.g., status of navigation ads, weather information, closed airfields, status of airports) for conducting safe and efficient flights.
- Alerting service, which provides services to all known aircraft. The main responsibility of alerting service is to assist aircraft in difficulties, for example, by initiating Search and Rescue (SAR) when accidents occur.

ATC units can be classified based on their responsibilities as follows:

- Aerodrome Tower Control (TWR) unit, which provides aerodrome control services. This unit usually has three different positions:
  - Delivery or clearance delivery, which is responsible for two main tasks: Give IFR departure clearances prior to start-up and push-back and give special IFR instructions in cooperation with approach controller. This position only gives air traffic control service and alerting service if the airfield is closed.
  - Ground control, which is responsible for four main tasks: Give VFR flight plan clearances, give push-back clearances, give taxi clearance to departure runways and give taxi clearance to the terminal gate. In addition to air traffic control service, the ground control position also gives traffic information service (e.g., traffic information on ground to prevent collisions) and alerting service if the airfield is closed.
  - Tower control, which is responsible for five main tasks: Give take-off clearances, give landing clearances, give runway crossing and back-track clearances, give VFR integration clearances in circuit and give VFR orbit clearances to delay the integration clearance. This position gives all three types of services: Air traffic control service (e.g., landing and take-off clearances, entering runway clearances), traffic information service (e.g., traffic information between VFR/VFR and IFR/VFR) and alerting service (e.g., in the control zone).
- Approach Control (APP) unit, which provides approach control services. This unit usually has two different positions:
  - Approach control, which is responsible for five main tasks: Give IFR initial, intermediate and final approach clearances, give radar vectoring and separate traffic using altitude, heading and speed parameters, make regulation clearances, assure adequate separation between all traffic and give VFR transit

clearances. This position gives all three types of services: air traffic control service (e.g., IFR clearances and instructions), traffic information services (traffic information between VFR/VFR and IFR/VFR) and alerting services (e.g., in the terminal area).

  - Departure control, which is responsible for four main tasks: Give IFR clearances, give radar vectoring using altitude, heading and speed parameters, make departure regulation clearances and assure adequate separation between all traffic. This position gives all three types of services: Air traffic control service (e.g., IFR clearances and instructions), traffic information service (e.g., traffic information between VFR/VFR and IFR/VFR) and alerting services (e.g., in the terminal area).

- En-route, Center, Or Area Control Center (ACC) unit, which provides area control services. This unit is responsible for four main tasks: Give STAR/arrival route clearances, give directs and regulation clearances, give radar vectoring using altitude, heading and speed parameters and assure adequate separation between all traffic. This unit gives all three types of services: Air traffic control service (e.g., en-route clearances, give IFR clearance and instructions), traffic information service (e.g., traffic information between VFR/VFR and IFR/VFR, traffic information between VFR/IFR and IFR/IFR) and alerting service (e.g., in the FIR Area).

In ATC operations, all the ATC units are needed to be continuously improved to meet the evolving demands in air traffic. However, there are three main reasons why ASR technologies should be integrated into either en-route control or approach control units first. Firstly, en-route and approach controllers usually use more standardized phraseologies in their communications with pilots than tower and ground controllers. This happens because the en-route and approach control positions usually involve more standardized tasks such as give radar vectoring, give STAR/arrival route clearances and give approach/departure clearances. On the other hand, tower and ground control positions usually have to deal with less standardized tasks, for example, control vehicles on the maneuvering area at the airport, receive and provide weather information and status of the airport, answer questions and requests from pilots about parking of aircraft. The use of standardized phraseologies and limited vocabulary of en-route and approach controllers facilitates the integration of post-processing approaches, particularly syntactic analysis and semantic analysis, into ASR systems. Secondly, air traffic in en-route and terminal control areas, which are controlled by en-route and approach controllers, are usually less variety in general compared with other control areas. The less variability in air traffic of the en-route and approach control areas leads to the less variability in speech of the controllers, which offers a great opportunity for ASR systems to archive higher accuracy. Finally, most of existing ATC-related corpora have been recorded either from en-route control or approach control units (e.g., ATCOSIM [33], Air Traffic Control Complete LDC94S14A [20]). In the development of ASR systems, selecting a corpus for training and testing is a very important task. Because both performance and accuracy of the ASR systems depend heavily on the quality of the training corpus.

### 2.1.3 Sources of Knowledge in Speech in ATC

Speech recognition comes naturally to human being. We can easily listen to others and understand them even with people we never met before. In some cases, we can understand speech even when we mishear some words. We can also understand ungrammatical utterances or new expressions. These happens because we use not only acoustic information but also linguistic and contextual information to interpret speech.

On the other hand, speech recognition has been considered a difficult task for machines. Because unlike humans, machines typically use only acoustic information to perform speech recognition. In addition, ASR systems have to deal with tremendous amount of variability present in a speech signal (e.g., speaker properties, co-articulation, allophonic variants and phoneme variations, environment) [5]. In order to improve the accuracy of ASR systems, many attempts have been made to use linguistic knowledge in assisting the recognition process of the systems [67, 3, 40, 55, 16]. According to [30], there are seven levels of linguistic knowledge which can be used by speech recognizers to resolve the uncertainties and ambiguities resulted from the speech recognition process:

1. Acoustic analysis, which extracts features from speech input signal.
2. Phonetic analysis, which identifies basic units of speech (e.g., vowels, consonants, phonemes).
3. Prosodic analysis, which identifies linguistic structures by using intonation, rhythm, or stress.
4. Lexical analysis, which compares extracted features with reference templates to match words.
5. Syntactic analysis, which tests the grammatically correctness of sentences.
6. Semantic analysis, which tests the meaningfulness of sentences.
7. Pragmatic analysis, which predicts future words based on the previous words and the state of the system.

While the first four steps are the basis of general ASR systems, the last three steps can be found in domain-specific ASR systems such as call centers and voice-based navigation systems.

#### Syntactic Knowledge

In general, syntactic knowledge is the knowledge about how words combine to form phrases, phrases combine to form clauses and clauses join to make sentences. In other words, syntactic knowledge is the knowledge which can be used to test if a sentence is grammatically correct.

However, in ATC, the language used by controllers and pilots in their communications is based on the ICAO standard phraseologies instead of natural language. Thus, syntactic knowledge in ATC is the knowledge about how words combine to form a valid ATC clearance. In other words, syntactic knowledge in ATC is the knowledge which can be used to test if an ATC clearance is well formatted. Some examples of syntactic knowledge in ATC can be found in Table 2.4.

Table 2.4: Examples of syntactic knowledge in ATC

| **Type of Clearance** | **Phraseology** |
|---|---|
| Vectoring Clearance | *<Callsign>, TURN LEFT (or RIGHT) HEADING (three digits)* |
| Taxi Procedures | *<Callsign>, TAXI VIA RUNWAY (runway code)* |
| Descend Clearance | *<Callsign>, DESCEND TO FLIGHT LEVEL <FL>* |

### Semantic Knowledge

In general, semantic knowledge is the knowledge about words and sentences that are meaningful in a specific domain. In other words, semantic knowledge is the knowledge which can be used to test if a sentence is meaningful.

Scene controllers and pilots use ICAO standard phraseologies in their communications instead of natural language, semantic knowledge in ATC is slightly different from general semantic knowledge. In ATC, semantic knowledge is the knowledge which can be used to test if an ATC clearance is meaningful without contextual information (e.g., valid runway codes, flight levels). Some examples of semantic knowledge in ATC are:

- According to [65], runways are named by a number between 01 and 36, which is generally the magnetic azimuth of the runway's heading in decadegrees. If there are more than one runway pointing in the same direction (parallel runways), each runway is identified by appending Left (L), Center (C) and Right (R) to the number to identify its position (when facing its direction). Thus, valid runway codes are 01[L|C|R], 02[L|C|R],...,36[L|C|R], for example:

  *<Callsign>, TAXI VIA RUNWAY <01[L|C|R], 02[L|C|R],...,36[L|C|R]>*

- IFR Flight levels with magnetic route figure of merit (FOM) from 180 degrees to 359 degrees are in steps of 20 from FL 020 to FL 280, and in steps of 40 from FL 310 to FL 51, for example:

  *<Callsign>, DESCEND TO FLIGHT LEVEL <020|040|060|...|280|310|350|...|510>*

### Pragmatic Knowledge

Pragmatic knowledge is the knowledge about context and state of the system. In ATC, pragmatic knowledge is the knowledge which can be used to test if a clearance is meaningful in a specific context or a specific state of the system, for example:

- If the present airport is Oslo Airport, Gardermoen, the valid runway codes are only 01L/19R and 01R/19L. Because the Oslo Airport, Gardermoen has only two parallel runways:
  - 01L/19R: 11,811 x 148 ft (3,600 x 45 m);
  - 01R/19L: 9,678 x 148 ft (2,950 x 45 m).

  *An example of a taxi procedure:*
  *<Callsign>, TAXI VIA RUNWAY <01L/19R | 01R/19L>)*

- If the present airport is Oslo Airport, Gardermoen, the valid units and radio frequencies are limited to the following list:
  - TWR (Gardermoen Tower): 118.300, 118.700, 120.100, 123.325, 257.800, 121.500, 243.000 (MHZ);
  - CLR (Gardermoen Delivery): 121.675, 121.925 (MHZ);
  - SMC (Gardermoen Ground): 121.600, 121.900, 121.725 (MHZ);
  - ATIS (Gardermoen Arrival Information): 126.125 (MHZ);
  - ATIS (Gardermoen Departure Information): 127.150 (MHZ);
  - ARO (Gardermoen Briefing/Handling): 134.175 (MHZ).

  When a unit call sign is detected, the number of valid frequencies can be limited to the unit's frequencies. For example, if the unit call sign is "Gardermoen Delivery", valid frequencies are only: 121.675 MHz and 121.925 (MHz).

  *An example of a transfer of control and/or frequency change clearance:*
  *<Callsign>, CONTACT Gardermoen Delivery < 121.675 | 121.925 > [NOW]*

- If the present flight level is 150, descends are valid to only flight levels which are lower than 150 (e.g., 100, 110, 120, 130, 140), for example:

  *<Callsign>, DESCEND TO FLIGHT LEVEL <100|110|120|130|140>*

I have presented a detailed introduction to the ATC field in general. In the following section, I focus on describing the general structure of an Automatic Speech Recognition (ASR) system and its modules, as well as summarize some of the well-known language modeling approaches.

## 2.2 Automatic Speech Recognition (ASR)

According to [45], "speech recognition is the process of converting a speech signal into a sequence of words. It also called Automatic Speech Recognition (ASR) or Speech-to-Text (STT)". In recent years, the technology and performance of ASR systems have been improving steadily. This has resulted in their successful use in many application areas such as in-car systems or environments in which users are busy with their hands (e.g., voice user interfaces), hospital-based health care applications (e.g., systems for dictation into patient records, speech-based interactive voice response systems, systems to control medical equipment and language interpretation systems), home automation (e.g., voice command recognition systems), speech-to-text processing (e.g., word processors or emails), and personal assistants on mobile phones (e.g., Siri on iOS, Cortana on Window Phone, Google Now on Android) [45].

The general goal of speech recognition can be described as follows: Given an acoustic observation $X = X_1, X_2, ..., X_n$, find the corresponding word sequence $W = W_1, W_2, ..., W_n$ that has the maximum posterior probability $P(W \mid X)$ [24], expressed using Bayes theorem in Equation 2.1.

$$W = \underset{w}{\operatorname{argmax}} P(W \mid X) = \underset{w}{\operatorname{argmax}} \frac{P(W)P(X \mid W)}{P(X)} \tag{2.1}$$

Since the observation X is fixed and P(X) is independent of W, the maximization is equivalent to maximization of the following equation:

$$W = \underset{w}{\operatorname{argmax}} P(W \mid X) = \underset{w}{\operatorname{argmax}} P(W)P(X \mid W) \tag{2.2}$$

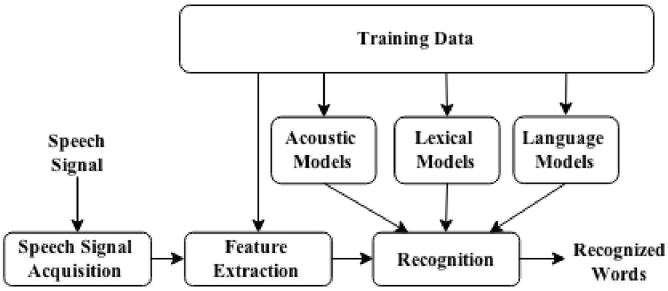

Figure 2.1: Structure of speech recognition system

Figure 2.1 shows the general structure of a speech recognition system. The general process of a speech recognition system can be briefly described as follows: A speaker utters an original word sequences $W' = W'_1, W'_2, ..., W'_n$ and produces a corresponding speech signal $I$. The Speech Signal Acquisition module obtains the speech signal $I$, for example by using a microphone, before the Feature Extraction module converts the signal to a feature vector $X = X_1, X_2, ..., X_n$. Finally, the Recognition module solves the maximization described in Equation 2.2 based on the feature vector $X$, acoustic model $P(X \mid W)$, language model $P(W)$ and lexical model in order to find a word sequence $W = W_1, W_2, ..., W_n$ that perfectly approximates the original word sequence $W'$.

### 2.2.1 Modules of Speech Recognition Systems

ASR systems typically contain six main modules: Speech Signal Acquisition, Feature Extraction, Acoustic Model, Language Model, Lexical Model and Recognition.

1. Speech Signal Acquisition, which is responsible for acquiring speech signal from speakers, for example by using microphones. In ATC, the speech signal acquisition module is typically advantaged by a special device called “push to talk (PTT)” button. Thus, besides acquiring speech signal from speakers, the module is also responsible for detecting boundaries of the input clearances.

2. Feature Extraction, which is the process of converting a speech signal into a feature vector in order to reduce the dimensionality of the input vector while maintaining relevant information of the signal. In addition, the feature extraction process also eliminates unwanted variability from different sources (e.g., speaker variations, pronunciation variations and environment variations) and noise in speech signal [58]. Many feature extraction techniques have been proposed. Some examples are Principal Component Analysis (PCA), Mel Frequency Cepstral Coefficients (MFCC), Independent Component Analysis (ICA), Linear Predictive Coding (LPC), Autocorrelation Mel Frequency Cepstral Coefficients (AMFCCs), Relative Autocorrelation Sequence (RAS), Perceptual Linear Predictive Analysis (PLP) and a new scope of

this field, Hybrid Features (HF). Studies have shown that MFCC, PLP and LPC are techniques that have been used extensively in speech recognition [12, 14]. Recently, Hybrid Features are overcoming the existing features and becoming an active research area in ASR [14].

3. Acoustic Model, which is responsible for representing the relationship between audio signals and linguistic units that make up speech such as words, syllables and phonemes. Acoustic models are usually trained by using audio recordings and their corresponding transcripts. In Equation. 2.2, $P(X \mid W)$ represents the acoustic model, which is the probability of acoustic observation X given that the word sequence W is uttered. Many types of acoustic models have been proposed, for example, Hidden Markov Model (HMM), Dynamic Time Warping (DTW), Artificial Neural Networks (ANNs). Studies have shown that HMM is the most successful method for acoustic modeling [24].

4. Language Model, which is responsible for assigning probability to a given word sequence $W = W_1, W_2, ..., W_n$. The probability assigned to a specific word sequence $W$ is the indication of how likely the word sequence occurs as a sentence in the language that described by the language model. With the ability to assign probability to word sequences, language models narrow down the search space of ASR systems to only valid word sequences and bias the outputs of the systems toward "grammatical" word sequences based on the grammars defined by the language model [24].

5. Lexical Model, which is also known as pronunciation dictionary, is responsible for representing the relationships between acoustic-level representations and the word sequences output by the speech recognizer. Lexical models are developed to provide pronunciations of words or short phrases in a given language. The development process of lexical models typically includes two main steps: First, word list development, which is a process of defining and selecting the basis units of written language - the recognition vocabulary (the word list). While the word list is usually obtained from training corpora in large-vocabulary speech recognition, it can be determined manually by the word occurrences in small-vocabulary and domain-specific speech recognition. Second, pronunciation development, which includes phone set definition and pronunciation generation. Typically, the pronunciations may be taken from existing pronunciation dictionaries. However, if the word list includes words that feature unusual spelling, the pronunciations can be created manually or generated by automatic grapheme to phoneme (g2p) conversion softwares such as Phonetisaurus and sequitur-g2p.

6. Recognition Module, which is also known as speech decoder or search module, is responsible for recognizing which words were spoken based on inputs from the feature extraction module, acoustic model, language model and lexical model. The recognition process of a speech recognizer is usually referred to as a search process with the main goal is to find a word sequence $W = W_1, W_2, ..., W_n$ that has maximum posterior probability $P(W \mid X)$ as represented in Equation 2.2. Studies have shown that Viterbi and A* stack decoders are the two most accurate decoders for performing the search in speech recognition. Recently, with the help of efficient pruning techniques, Viterbi beam search has becoming the predominant search method for speech recognition [24].

### 2.2.2 Performance of Speech Recognition Systems

In ASR, accuracy and speed are the two most common metrics that have been used for measuring system performance. While speed is usually rated with Real Time Factor (RTF), Word Error Rate (WER) is usually used for measuring accuracy [45]. WER can be computed by using Equation 2.3:

$$WER = \frac{S + D + I}{N} \tag{2.3}$$

where S is the number of substitutions, D is the number of deletions, I is the number of insertions and N is the number of words in the reference.

If I is the duration of an input and P is the time required to process the input. RTF can be computed by using Equation 2.4:

$$RTF = \frac{P}{I} \tag{2.4}$$

WER is usually used for measuring the accuracy of ASR systems in general. On the other hand, Concept Error Rate (CER) and Command Success Rate (CSR) are usually used for measuring the accuracy of domain-specific ASR systems such as command and control ASR systems. If M is the number of misrecognized concepts and N is the total number of concepts, CER can be computed by using Equation 2.5:

$$CER = \frac{M}{N} \tag{2.5}$$

In ATC, it is not important that ASR systems can recognize every single word, but it is important that the conveyed concepts are correctly detected [45]. Therefore, CER is usually used for measuring the accuracy of ASR systems in ATC instead of WER.

### 2.2.3 Language Model

Language models play a critical role in ASR because they describe the language that the system recognize and bias the outputs of the system toward “grammatical” sentences based on the grammars defined by the language models. This means that, the accuracy of an ASR system depends heavily on the quality of its language model. In Equation 2.2, P(W) represents the language model, which is the probability of word sequence $W = W_1, W_2, ..., W_n$ uttered. Many types of language models have been proposed. Some well-known examples are grammars (e.g., regular grammar, context-free grammar) and stochastic language models (e.g., n-gram language model, class n-gram language model, adaptive language model).

#### Grammars

According to the Chomsky hierarchy (also known as Chomsky-Schützenberger hierarchy) [8, 24], there are four types of formal grammars:

- Type 0 - Phrase structure grammars, which are unrestricted grammars that include all formal grammars. The phrase structure grammars generate languages which can be recognized by Turing machines.

- Type 1 - Context-sensitive grammars, which is a subset of phrase structure grammars. The Context-sensitive grammars generate languages which can be recognized by Linear Bounded Automaton (LBA).
- Type 2 - Context-free grammars (CFGs), which is a subset of context-sensitive grammars. The context-free grammars generate languages which can be recognized by non-deterministic pushdown automaton, which is also known as Recursive Transition Network (RTN).
- Type 3 - Regular grammars, which is a subset of context-free grammars. The regular grammars generate languages which can be recognized by Finite State Machines (FSMs).

Context-free grammars have been widely use in Natural Language Processing (NLP) and domain-independent ASR systems because of its compromise between parsing efficiency and power in representing the structure of languages. On the other hand, regular grammars are commonly found in more restricted and domain-specific ASR systems [24]. This happens because of the limited power in representing the structures of languages of regular grammars.

In ATC, grammars can be created by hand or by generating from codes with the JSpeech Grammar Format (JSGF) [25]. Below is an example of grammars which are written in the JSGF format:

```
#JSGF V1.0;
/**
 * JSGF Grammars for description of flight levels
 */
grammar level;
public <Levels> = FLIGHT LEVEL <Number>+ | <Number>+ METERS | <Number>+ FEET
```

### Stochastic Language Models

The main idea of stochastic language models is to estimate the probability of word sequences $W = W_1, W_2, ..., W_n$ occur as sentences based on training corpora. The main goal of stochastic language models is to assign higher probability to the likely word sequences. There are four main types of stochastic language models, Probabilistic Context-Free Grammars (PCFGs), n-gram language model, class n-gram language model and adaptive language model.

**Probabilistic Context-Free Grammars (PCFGs)** , which extend the context-free grammars by augmenting each production rule with probability. Because of the augmented probability in production rules, the training process requires one extra step compared with the context-free grammars training process. In addition to determine a set of rules for grammar G based on a training corpus, estimating the probability of each rule in G based on the corpus is also required. The recognition process of PCFGs is similar to other stochastic language models (e.g., n-gram language model, class n-gram language model), which involves the computation of the probability P(W) of word sequences $W = W_1, W_2, ..., W_n$ generated by the start symbol S. Unlike context-free grammar parser which produces a list of all possible parses for an input, PCFGs parser produces the most probable parse or a ranking of possible parses based on the probability P(W).

**N-gram Language Models** , which are responsible for representing the probability of word sequences $W = W_1, W_2, ..., W_n$ occur as sentences in a given language. For example, for a language model describing the language that air traffic controllers and pilots use in their communications, we might have $P(REPORTSPEED) = 0.0001$, which means that one out of every ten thousands clearances a controller may say “REPORT SPEED”. On the other hand, $P(Ilovedogs) = 0$, because it is very unlikely that controllers or pilots would utter such a strange clearance or respond. However, it is impractical to calculate the probability of every possible word sequences $W$ (see Equation 2.6).

$$P(W) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1w_2)...P(w_n \mid w_2, ..., w_n - 1) \tag{2.6}$$

Because even with moderate values of n there are a huge number of different word sequences $W$ which have size n. To deal with this problem, we assume that the probability of the *ith* word $w_i$ depends only on its n-1 previous words. With that assumption, we have n-gram language model. If $n = 1$, 2 and 3 we have unigram language model: $P(w_i)$, bigram language model: $P(w_i|w_{i-1})$ and trigram language model: $P(w_i|w_{i-2}, w_{i-1})$ respectively. Although n-gram language models typically require very big training corpora (e.g., millions of words corpora) for training, they have been widely used for many domain-independent speech recognition systems because of their high accuracy and performance [49, 51, 2, 35].

**Class N-gram Language Models** , which extend n-gram language models by grouping words that exhibit similar semantic or grammatical behavior. For example, different call signs such as Speedbird, Swissair, Jetblue, Norstar can be grouped into a broad class [CALLSIGN], different airports names such as Gardermoen, Frankfurt am Main International, Hartsfield Jackson Atlanta International can be grouped into a broad class [AIRPORT]. According to [24], if we assume that a word $w_i$ can be uniquely mapped to only one class $c_i$, then the class n-gram model can be computed based on the previous n-1 classes as follow:

$$P(w_i \mid c_{i-n+1}...c_{i-1}) = P(w_i \mid c_i)P(c_i|c_{i-n+1}...c_{i-1}) \tag{2.7}$$

where $P(w_i \mid c_i)$ is the probability of word $w_i$ given class $c_i$ in the current position, and $P(c_i|c_{i-n+1}...c_{i-1})$ denotes the probability of class $c_i$ given n-1 previous classes. Typically, there are two main types of class n-gram language models:

- Rule-based class n-gram, which is based on syntactic and semantic information that exist in the given language to cluster words together, for example, class [DIGIT] which includes ten words,“zero, one, two, three, four, five, six, seven, eight, nine”.
- Data-driven class n-gram, which is based on data-driven clustering algorithms to generalize the concept of word similarities. Output of clustering algorithms are different clusters which are equivalent with manually defined classes in Rule-based class n-gram.

Since the classes in class n-gram language models have the ability to encode syntactic and semantic information, class n-gram language models have been widely used for many domain-specific ASR systems [43, 66, 42].

**Adaptive Language Model** focuses on using knowledge about the topic of conversation to dynamically adjust the language model parameters (e.g., n-gram probabilities, vocabulary size) to improve the quality of the model [13, 37, 34, 52]. Many adaptive language models have been proposed, for example, cache language models, topic adaptive models and maximum entropy models.

### N-Gram Smoothing

N-gram language models suffer from a very well-known problem called zero probability, $P(W) = 0$, which is also known as "dealing with unseen data". This problem occurs when the training corpus is not big enough. Sentences which occur in test corpus but do not occur in training corpus will be given zero probabilities by the n-gram language model, $P(W) = 0$. When $P(W)$ is zero, no matter how unambiguous the acoustic signal is, the word sequence $W$ will never be considered as a possible transcription, thus an error will be made.

In order to deal with the zero probability problem many n-gram smoothing techniques have been applying to the n-gram modeling process. The main purpose of n-gram smoothing is to assign all word sequences non-zero probabilities by adjusting low probabilities such as zero probabilities upward, and high probabilities downward in order to prevent errors in the recognition process.

Many n-gram smoothing techniques have been proposed, for example, Additive smoothing (Laplace smoothing), Deleted interpolation smoothing, Backoff smoothing, Good-Turing Estimates, Katz smoothing and Kneser-Ney smoothing. According to [24], Kneser-Ney smoothing, Katz smoothing and Deleted interpolation smoothing slightly outperform Additive smoothing, Backoff smoothing and Good-Turing Estimates.

### Complexity Measurement of Language Models

In general, a good language model "prefers" grammatical sentences than ungrammatical sentences. There are two main metrics that have been using for evaluating language model performance [24]:

- Word Error Rate (WER), which requires the integration of the language model into an ASR system and measurement of WER on test sets. Language model A is better than language model B, if the ASR system that uses the language model A produces lower WER than the one that uses the language model B.

- Perplexity, which is the probability of the test set, normalized by the number of words. Perplexity can also be roughly interpreted as the average branching factor of the text [24]. For example, the perplexity of the task of recognizing digits "0, 1, 2, 3, 4, 5, 6, 7, 8, 9" is 10. Language model A is better than language model B, if the language model A can assign lower perplexity to the test corpus then the language model B. Perplexity can be computed by using Equation 2.8 as follows:

$$PP(W) = \hat{P}(w_1, w_2, \ldots, w_N)^{-\frac{1}{N}} \tag{2.8}$$

where $\hat{P}(w_1, w_2, \ldots, w_N)$ is the probability estimate assigned to the word sequence $(w_1, w_2, \ldots, w_N)$ by a language model and N is the number of words of the sequence.

I have presented a detailed introduction to the ATC field in general and ASR technologies. In the next section, I review some related work covering ASR in ATC, as well as different approaches for improving the accuracy of ASR systems in the ATC domain.

## 2.3 Related Work

Since the 80s (or earlier), researchers have started to introduce ASR technologies into ATC [62, 23, 21]. Since then, continuous efforts have been made to improve the accuracy of ASR systems in order to facilitate applications such as ATC workload measurement and balancing [10, 11], analysis of ATC speech [48, 17], speech interfaces [18], and ATC simulation and training [22, 36, 15]. In addition, continuous attempts have also been made to apply ASR technologies in reducing ATC communication errors. One example is the work of Geacăr Claudiu-Mihai [19], who converted spoken clearances into machine-usable data for text clearances broadcast which is considered as a backup channel for the verbal communications.

However, due to the high accuracy requirements of the ATC context and its unique challenges such as call sign detection, poor input signal quality, the problem of ambiguity, the use of non-standard phraseology, and the problem of dialects, accents and multiple languages [45], ASR technologies have not been widely adopted in this field.

In order to address the above-mentioned challenges and improve the accuracy of ASR systems in ATC, a few efforts have been made to integrate higher levels of knowledge sources, which are usually not available for standard ASR systems, such as linguistic knowledge, situation knowledge and dialog contextual information into ASR systems. For example, Karen Ward et al. [64] proposed a speech act model of ATC speech in order to improve the accuracy of speech recognition and understanding in ATC. The main idea of the model is to focus on using two dialog models, speech act and the collaborative view of conversation, to predict the form and content of the next utterance in order to reduce the size of grammar and vocabulary that the system has to deal with. Another example is the work of D. Schaefer [55], who proposed a cognitive model of air traffic controller in order to use situation knowledge as a mean to improve the accuracy of ASR systems. According to the author, the model can continuously observe the present situation and generate a prediction of the next clearances that the controller is most likely to say. In addition, studies have shown that the acquisition and processing of higher levels of knowledge sources is a very promising approach for improving the accuracy of ASR systems in ATC [31]. Unfortunately, none of the above-mentioned approaches can address completely the existing challenges of ASR in ATC.

In this thesis, in order to take advantage of the availability of linguistic knowledge in the ATC domain, I aim at using linguistic knowledge to address the existing challenges of ASR in ATC. The approaches which facilitate the integration linguistic knowledge into ASR systems can be categorized into three groups: language modeling, N-best filtering and re-ranking, and word lattice filtering and re-ranking.

The main idea of the language modeling approach is to integrate linguistic knowledge into decoding to guide the search process. The main advantage of this approach is that it can reduce the search space in decoding which increases both accuracy and performance of the system. For example, L. Miller et al. used context-free grammars as language model to integrate linguistic knowledge in to ASR systems [40].

N-best list re-ranking have been widely used for improving ASR systems accuracy. The main ideal of this approach is to re-score N-best hypotheses and then use the scores to perform re-ranking. The hypothesis that ranked highest will be the output of the system. There are many different methods that can be used to perform N-best list re-ranking. For example, Z. Zhou et al. conducted a comparative study of discriminative methods: perceptron, boosting, ranking support vector machine (SVM) and minimum sample risk (MSR) for N-best list re-ranking in both domain adapting and generalizing tasks [68]. Another example is the work of T. Oba et al [46]. The authors compared three methods; Reranking Boosting (ReBst), Minimum Error Rate Training (MERT) and the Weighted Global Log-Linear Model (W-GCLM) for training discriminative n-gram language models for a large vocabulary speech recognition task. With regard to N-best filtering, the main idea is to verify the list of N-best hypotheses which are already sorted by score with a verifier. The first hypothesis accepted by the verifier will be the output of the system. One approach that have been widely used to perform N-best filtering is using a natural language processing (NLP) module as a verifier [69].

Lattices is a directed graph which represents a set of hypothesized words with different starting and ending positions in the input signal. Lattices are typically used to represent search results and served as intermediate format between recognition passes. The main idea of lattices filtering and re-ranking is to first generate lattices and then use post-processing parser to filter or re-rank the lattices [5]. One example is the work of Ariya Rastrow et al [50]. The authors proposed an approach for re-scoring speech lattices based on hill climbing via edit-distance based neighborhoods.

# Chapter 3

# ASR Frameworks and Existing ATC-Related Corpora

This chapter focuses two main purposes. First, it presents a detailed review of ten well-known open source Automatic Speech Recognition (ASR) frameworks which are selected based on their popularity and community size, documentation, supported features and customers reviews. For the sake of completeness, a list of other relevant frameworks/projects is also included.

Second, it describes five main existing ATC-related corpora. In the development of ASR systems, selecting a good speech corpus for training is a crucial task because both accuracy and performance of the ASR systems depend heavily on the quality of the corpus.

## 3.1 ASR Frameworks

In this section, I first review ten well-known open source ASR frameworks including Bavieca, CMU Sphinx, Hidden Markov Model Toolkit (HTK), Julius, Kaldi, RWTH ASR, SPRAAK, CSLU Toolkit, The transLectures-UPV toolkit (TLK) and iATROS. I then select a framework for developing a baseline ASR system.

### 3.1.1 Bavieca

Bavieca is a very well-known open source framework for speech recognition which is distributed under the Apache 2.0 license. With the core technology is Continuous Density Hidden Markov Models (CD-HMMs), Bavieca supports acoustic modeling, adaption techniques and also discriminative training. The framework is written in C++ programming language, however, in addition to C++ native APIs, the framework also supports Java APIs (a wrapper of the native APIs), which makes incorporating speech recognition capabilities to Java applications become easier. Bavieca is a well-documented framework which provides many examples, tutorials and API references. The framework was evaluated using the WSJ Nov'92 database [6], the result was quite impressive at 2.8% Word Error Rate (WER), which is achieved by using trigram language model on a 5000-words corpus.

Bavieca's website: `http://www.bavieca.org/index.html`<br>
Bavieca's source code: `http://sourceforge.net/projects/bavieca/`

### 3.1.2 CMU Sphinx

CMU Sphinx is a collection of speech recognition systems developed by Carnegie Mellon University (CMU) research group, which also collects over 20 years of the CMU research. The systems are distributed under the BSD-like license which allows commercial distribution. CMU Sphinx has a very large and active community with more than 400 users, active development and release schedule. According to [60], the CMU Sphinx toolkit includes a number of packages for different task and applications:

- Pocketsphinx - speech recognizer library written in C;
- Sphinxtrain - acoustic model training tools;
- Sphinxbase - support library required by Pocketsphinx and Sphinxtrain;
- Sphinx4 - adjustable, modifiable recognizer written in Java.

In addition to C library, CMU Sphinx also supports Java library (Sphinx4) which makes incorporating speech recognition capabilities to Java applications become easier. The main technology of the CMU Sphinx framework is Hidden Markov Models (HMMs). In addition to English, CMU Sphinx also supports many other languages such as French, German, Dutch and Russian.

CMU Sphinix's website: `http://cmusphinx.sourceforge.net/`
CMU Sphinix's source code: `http://sourceforge.net/projects/cmusphinx/`

### 3.1.3 Hidden Markov Model Toolkit (HTK)

The Hidden Markov Model Toolkit (HTK), which is written in C programming language, is a toolkit for building and manipulating hidden Markov models. HTK has been using for both speech recognition and speech synthesis research (mainly for speech recognition). The toolkit is distributed under their own license (HTK End User License Agreement), which does not allow to distribute or sub-license to any third party to any form. Although this project has been inactive since April 2009, it has still been used extensively because of its sophisticated tools for HMM training, testing and results analysis, as well as its extensive documentation, tutorials and examples. The toolkit was evaluated using the well-known WSJ Nov'92 database [6], the result was quite impressive at 3.2% WER, which is achieved by using trigram language model on a 5000-words corpus.

HTK's website (including HTK's source code and book): `http://htk.eng.cam.ac.uk/`

### 3.1.4 Julius

Julius, which is written in C programming language, is an open source, large vocabulary, continuous speech recognition framework. The framework is distributed under the BSD-like license, which allows commercial distribution. The main technologies of Julius are n-gram language models and context-dependent HMMs. Julius is a well-documented framework, which provides many sample programs, full source code documentation and manual. Unfortunately, most of the documents are in Japanese. Julius has a large and active community. Currently, Julius provides free language models for both Japanese and

English. However, the English language model cannot be used in any commercial product or for any commercial purpose.

Julius's website: `http://julius.sourceforge.jp/en_index.php`
Julius' source code: `http://sourceforge.jp/cvs/view/julius/`

### 3.1.5 Kaldi

Kaldi, which is written in C++ programming language, is a toolkit for speech recognition distributed under the Apache License v2.0. Kaldi is a very well-documented toolkit, which provides many tutorials, examples, API references, as well as descriptions of its modules, namespaces, classes and files. Kaldi supports many advanced technologies such as Deep Neural Network (the latest hot topic in speech recognition), Hidden Markov Models and a set of sophisticated tools (e.g., estimate LDA, train decision trees) and libraries (e.g., matrix library). Kaldi was evaluated using the well-known WSJ Nov'92 database [6], the evaluation result on a 20000-words corpus using bigram language model was 11.8% WER.

Kaldi's webpage: `http://kaldi.sourceforge.net/index.html`
Kaldi's source code : `https://svn.code.sf.net/p/kaldi/code/`

### 3.1.6 RWTH ASR

RWTH ASR, which is written in C++ programming language, is a set of tools and libraries for speech recognition decoding and developing of acoustic models. RWTH ASR is distributed under their own license (RWTH ASR License), which allows for non-commercial use only. Although RWTH ASR is not a well-documented toolkit, it has still been used widely because of its advanced technologies and sophisticated tools such as neural networks (deep feed-forward networks), speaker adaption, HMMs and Gaussian mixture model (GMM) for acoustic modeling, Mel-frequency cepstral coefficients (MFCCs) and Perceptual Linear Predictive Analysis (PLP) for feature extraction. The RWTH ASR community is quite small, however, there is a RWTH ASR System Support forum where we can discuss and ask for help from RWTH ASR's developers and active users. In addition, RWTH ASR provides a demonstration of large vocabulary speech recognition system which includes triphones acoustic model and 4-gram language model. The demo models can be downloaded directly from their website.

RWTH ASR website : `http://www-i6.informatik.rwth-aachen.de/rwth-asr/manual/index.php/Main_Page`

### 3.1.7 SPRAAK

SPRAAK, which is written in C and Python programming languages, is a speech recognition toolkit distributed under an academic license, which is free for academic usage and at moderate cost for commercial usage. The main technology of the toolkit is HMMs. SPRAAK is a quite well-documented toolkit which provides many examples, tutorials and API references. Unfortunately, SPRAAK has been inactive since 2010 (the latest version is V1.0 released on December 7, 2010).

SPRAAK's website: `http://www.spraak.org/`

### 3.1.8 CSLU Toolkit

CSLU Toolkit, which is written in C/C++ programming languages, is a comprehensive suite of tools for speech recognition and human-computer interaction research. The toolkit is distributed under OHSU CSLU Toolkit Non-commercial license. However, there are also several options for evaluating and licensing CSLU Toolkit for commercial use . CSLU Toolkit is a very well-known toolkit because of its advanced technologies (e.g., HMMs and hybrid HMM/Artificial Neural Networks (ANN)), full and detailed documentation for users, developers and researchers. Unfortunately, this project has been inactive since 2010.

CSLU Tookit's website: `http://www.cslu.ogi.edu/toolkit/`

### 3.1.9 The transLectures-UPV toolkit (TLK)

The transLectures-UPV toolkit (TLK) , which is written in C programming language, is a toolkit for automatic speech recognition distributed under the Apache License 2.0. The main technology of toolkit is HMMs. The transLectures-UPV toolkit is a very well-documented toolkit which provides many examples and tutorials. Currently, TLK only supports Linux and Mac OS X.

TLL's website: `https://www.translectures.eu//doctools/manpages/tlk.1.html`
TLK source code: `http://bazaar.launchpad.net/~translectures/tlk/trunk/files`

### 3.1.10 iATROS

iATROS, which is written in C programming language, is a framework for both speech recognition and handwritten text recognition distributed under the the GNU General Public License v3.0. Although iATROS lacks of documentation and has been inactive since 2006, it has still been a quite popular framework because of its advanced technologies such as HMMs, MFCC),LDA and Viterbi-like search.

iATROS's website: `https://www.prhlt.upv.es/page/projects/multimodal/idoc/iatros`

### 3.1.11 Summary

Among the reviewed frameworks, the CMU Sphinix framework is the best option for this project because of the following reasons: Firstly, CMU Sphinix is a cross-platform framework which supports both desktop operating systems (e.g., Windows, Linux, Mac OS) and mobile operating systems (e.g., Android, iOS, Window Phone). Secondly, CMU Sphinix provides toolkit for training acoustic and language models, as well as toolkits which can facilitate post-processing approaches (e.g., syntactic analysis, semantic analysis). Thirdly, CMU Sphinix has a very large and active community, as well as active development and release schedule. Finally, CMU Sphinix is distributed under the BSD-like license which allows both academic and commercial distributions.

### 3.1.12 Other Frameworks/Projects

For the sake of completeness, I also include a list of other relevant frameworks/projects. Although some of these frameworks/projects are quite small compared with the reviewed frameworks/projects, they are still worth mentioned because of their interesting technologies and applications.

| ID | Frameworks/Projects | Descriptions |
|---|---|---|
| 1 | AaltoASR | `https://github.com/aalto-speech/AaltoASR` |
| 2 | Palaver speech recognition | `https://github.com/JamezQ/Palaver` |
| 3 | SCARF | `http://research.microsoft.com/en-us/projects/scarf/` |
| 4 | SHoUT speech recognition toolkit | `http://shout-toolkit.sourceforge.net/` |
| 5 | Barista | `https://github.com/usc-sail/barista` |
| 6 | Juicer | `https://github.com/idiap/juicer` |
| 7 | OpenDcd | `http://opendcd.org/` |
| 8 | SailAlign | `https://github.com/nassosoassos/sail_align` |
| 9 | SRTk | `https://bitbucket.org/yotaro/srtk` |
| 10 | Speechlogger | `https://speechlogger.appspot.com/en/` |
| 11 | The Edinburgh Speech Tools Library | `http://www.cstr.ed.ac.uk/projects/speech_tools/` |
| 12 | FreeSpeech | `http://thenerdshow.com/freespeech.html` |
| 13 | OpenEars | `http://www.politepix.com/openears/` |
| 14 | Simon | `https://simon.kde.org/` |
| 15 | Xvoice | `http://xvoice.sourceforge.net/` |
| 16 | SphinxKeys | `https://code.google.com/p/sphinxkeys/` |
| 17 | Platypus | `http://thenerdshow.com/platypus.html` |

Table 3.1: ASR open source frameworks/projects

I have reviewed ten well-known open source ASR frameworks and selected the CMU Sphinx framework for developing the baseline ASR system. In the next section, I review five existing ATC-related corpora in order to select a corpus for training and testing.

## 3.2 Existing ATC-Related Corpora

In the last few years, many speech corpora have been created by using Web crawling and TV recording technologies. Unfortunately, very few of the corpora are related to ATC. In the this section, with the aim of selecting a speech corpus for training and testing ASR systems in ATC, I review five well-known ATC-related corpora including The ATCOSIM Corpus of Non-Prompted Clean Air Traffic Control Speech, Air Traffic Control Complete LDC94S14A corpus, HIWIRE corpus, Air Traffic Control Communication Speech Corpus and Air Traffic Control Communication corpus.

### 3.2.1 The ATCOSIM Corpus of Non-Prompted Clean Air Traffic Control Speech

The ATCOSIM Corpus of Non-Prompted Clean Air Traffic Control Speech (ATCOSIM) [33] is a speech database of ATC operators speech. The ATCOSIM corpus consists of recordings of en-route controllers speech recorded in typical ATC control room condition during ATC real-time simulations. The ATCOSIM corpus contains ten hours of speech data, which were recorded from six male and four female controllers who were either German or Swiss nationality. Their native languages are German, Swiss German or Swiss French. The ATCOSIM corpus is available online to public and can be obtained for free of charge at `https://www.spsc.tugraz.at/tools/atcosim`

### 3.2.2 Air Traffic Control Complete LDC94S14A

The Air Traffic Control Complete LDC94S14A corpus [20] is a speech database of voice communications between various controllers and pilots in approach control unit. The speech data was recorded from three different airports in the United States: Dallas Fort Worth (DFW), Logan International (BOS) and Washington National (DCA). The corpus contains approximately 70 hours of both male and female controllers and pilots speech. Most of the controllers and pilots are native English speakers. The corpus was published in 1994 and only available for commercial. However, a sample version of the corpus can be obtained for free of charge at `https://catalog.ldc.upenn.edu/LDC94S14A`.

### 3.2.3 HIWIRE

The HIWIRE database [57] is a noisy and non-native English speech corpus of communications between controllers and pilots in military air traffic control. According to [57], the database contains a total of 8099 English utterances which were recorded from 81 non-native English speakers (31 French, 20 Greek, 20 Italian, and 10 Spanish speakers). The HIWIRE database has no usage restrictions. However, it is only available on request at `http://catalog.elra.info/product_info.php?products_id=1088`.

### 3.2.4 Air Traffic Control Communication Speech Corpus

The Air Traffic Control Communication Speech corpus [63] is a speech database of voice communications between controllers and pilots at four different control units:

- GRP (ground control) - 19.2 hours of data;
- TWR (tower control) - 22.5 hours of data;
- APP (approach control) - 25.5 hours of data;
- ACC (area control) - 71.3 hours of data.

The speech data was recorded mostly from the Air Navigation Services of the Czech Republic in Jeneč. The rest of the speech data was recorded from Lithuania and Philippines airspace.

### 3.2.5 Air Traffic Control Communication

According to [59], the Air Traffic Control Communication corpus contains 20 hours of recordings of communications between air traffic controllers and pilots. The corpus is publicly available and licensed under the "Attribution-NonCommercial-NoDerivs 3.0 Unported (CC BY-NC-ND 3.0)" license.

### 3.2.6 Other ATC-related Corpora

For the sake of completeness, I also include other small relevant ATC-related corpora:

- English TTS speech corpus of air traffic (pilot) messages - Serbian accent [38];
- English TTS speech corpus of air traffic (pilot) messages - Taiwanese accent [39].

### 3.2.7 Summary

Among the five reviewed ATC-related speech corpora, which are summarized in Table 3.2, the ATCOSIM corpus is the best option for this project because of the following reasons. Firstly, the ATCOSIM corpus consists of recordings of en-route controllers speech which perfectly matches with the scope of this thesis. Secondly, the ATCOSIM corpus contain only air traffic controllers speech without silence periods which is a good fit for training and testing ASR systems in ATC. Finally, the corpus is publicly available for free of charge with no usage restrictions.

Table 3.2: Summary of features of ATC-related corpora

| | **ATCOSIM** | **LDC94S14A** | **HIWIRE** | **ATCC Speech Corpus** | **ATCC** |
|---|---|---|---|---|---|
| **Control Unit** | en-route | approach | N/A | mixed | mixed |
| **Number of Speakers** | 10 | unknown (large) | 81 | unknown (large) | unknown (large) |
| **Gender** | mixed | mixed | mixed | mixed | mixed |
| **Level of English** | non-native | mostly native | non-native | non-native | non-native |
| **Native Language** | German<br>Swiss German<br>Swiss French | English | French<br>Greek<br>Italian<br>Spanish | N/A | N/A |
| **Duration** | 10 hours<br>10078 utterances | 70 hours | 8099 utterances | GRP: 19.2 hours<br>TWR: 22.5 hours<br>APP: 25.5 hours<br>ACC: 71.3 hours | 20 hours |
| **Free of Charge** | yes | no | no | no(?) | yes |

In addition to the ATCOSIM corpus that I chose, I also create a corpus for further testing called Air Traffic Control Speech Corpus (ATCSC). More details about the corpus can be found in Section 4.3.

# Chapter 4

# Case and Experimental Settings

This chapter serves three main purposes. First, it describes the special case that forms the basic of this project, which is developing an “automated pilot” system for Air Traffic Control (ATC) simulation and training. Second, It presents four experiments designed to answer the research questions introduced in Chapter 1, together with a brief summary of how the case affects the design of the experiments. The end of the chapter contains a short description of my own Air Traffic Control Speech Corpus (ATCSC) which is recorded with the aim of simulating an ATC simulation and training setting.

## 4.1 Case

This project is in collaboration with Edda systems AS and Institute for Energy Technology (IFE). The primary goal of this project is to develop an “automated pilot” system for ATC simulation and training.

ATC simulation provides facilities for testing and evaluation of new systems and concepts, and training of air traffic controller students to handle realistic scenarios. Current ATC simulation systems typically require “pseudo-pilots” who will act as real pilots in the simulation of controller-pilot communications with air traffic controller students. The use of “pseudo-pilots” makes ATC simulators less flexible and comes at a relatively high cost.

The main goal of this project is to introduce Automated Speech Recognition (ASR) technologies into ATC simulation and training in order to replace the “pseudo pilots” by so-called “automated pilots”. The “automated pilot”, which is showed in Figure 4.1, will interpret and process air traffic controllers speech using a combination of an ASR module and a Natural Language Processing (NLP) module, and generate responses that are sent back to the controllers using a Speech Synthesis (SS) module. The use of “automated pilots” instead of “pseudo-pilots” can dramatically reduce the cost of ATC simulation systems and make the systems more flexible.

In this thesis, I focus on the first step which is developing an ASR module for ATC simulation and training. The natural language processing and speech synthesis modules will be considered in future work.

Although the primary goal of this project is to develop an “automated pilot” system for ATC simulation and training, I aim at developing the ASR module in a way that it can be easily adapted for use in other types of ATC-related applications. Some examples are air traffic controllers workload measurement, controller-pilot speech analysis and transcription, and backup controller, which is a system that combines an ASR module

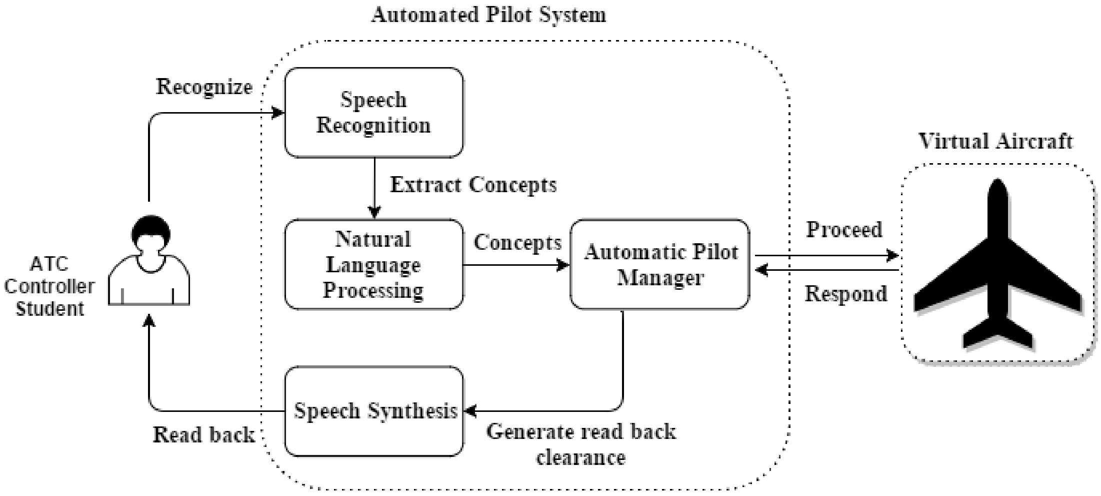

Figure 4.1: Automated pilot system for air traffic control simulation and training

with other information sources in the ATC context (e.g., radar information, minimum safe altitudes, restricted zones, and weather information) to catch potentially dangerous situations that might be missed by the controller as well as provide suggestions and safety information to the controller in real time.

In addition, since the ASR module is a command-and-control-like speech recognition module, the approaches and algorithms proposed in this thesis can also be easily adapted for use in other command-and-control-like ASR systems. Some examples are in-car ASR systems, ASR for smart homes, call centers and voice-controlled robots.

## 4.2 Experimental Settings

To answer the research questions introduced in Chapter 1, I design four experiments. The first three experiments, which can be found in Section 4.2.1, Section 4.2.2 and Section 4.2.3, are for addressing the three secondary research questions. Experiment concerning the main research question is presented in Section 4.2.4.

Although evaluating the ASR module in a real training and simulation setting is not in the scope of this thesis, my understanding of the setting together with Edda Systems AS and IFE have affected my choice of method in designing the experiments. In training and simulation, air traffic controller students are usually required to use ICAO standard phraseologies. Thus, the amount of linguistic knowledge, particularly syntactic and semantic knowledge, in their communications with pilots is relatively high which is a good fit for syntactic and semantic analysis. In addition, since signal quality in training and simulation setting is typically higher than in ATC live operations, existing acoustic models, for example, the CMU Sphinx US English generic acoustic model provided by CMU, can be reused with a very little effort in adaptation.

### 4.2.1 Language Modeling

To answer the first secondary research question, I design an experiment as follows: Firstly, I build a baseline ASR system based on the Pocketsphinx recognizer from the CMU

Sphinx framework, the CMUSphinx US English generic acoustic model and the generic cmudict_SPHINX_40 pronunciation dictionary. Secondly, I evaluate different language models (n-gram, class n-gram) in terms of Word Error Rate (WER) and Real Time Factor (RTF) on the baseline system in order to select a well-suited language model for use in ASR systems in ATC. Thirdly, I improve the selected language model by integrating linguistic knowledge into the language modeling process. To facilitate this, I propose a context-dependent class n-gram language model by combining the hybrid class n-gram language modeling and context-dependent language modeling approaches. Fourthly, I use the baseline system to evaluate the proposed model on the well known ATCOSIM Corpus of Non-prompted Clean Air Traffic Control Speech (ATCOSIM) and my own Air Traffic Control Speech Corpus (ATCSC). I use 85% of the data from the ATCOSIM corpus for training and the renaming 15% of the data for evaluations. In order to evaluate the ability of the proposed model in recognizing general ATC clearances, I also evaluate the proposed model on the ATCSC corpus. I use k-fold cross-validation to increase the reliability of the evaluations. Finally, I compare the evaluation results of the proposed model with traditional n-gram models (unigram, bigram and trigram).

### 4.2.2 N-best List Re-ranking Using Syntactic Knowledge

To address the second secondary research question, I design an experiment as follows: Firstly, I integrate syntactic knowledge into the baseline ASR system by performing n-best list re-ranking. To do this, I propose a novel feature called syntactic score. I compute the syntactic score using syntactic rules which are created by replacing expansions of word classes with their corresponding class labels. I propose a WER-Sensitive Pairwise Perceptron algorithm and use the perceptron to combine the proposed feature with the speech decoder's confidence score feature. Secondly, I evaluate the proposed approach on the ATCOSIM and ATCSC corpora. I use 85% of the data from the ATCOSIM corpus for training and the renaming 15% of the data for evaluations. In order to evaluate the ability of the proposed approach in recognizing general ATC clearances, I also evaluate the proposed approach on the ATCSC corpus. I use k-fold cross-validation to increase the reliability of the evaluations. Finally, I compare the evaluation results of the proposed approach with traditional n-gram language models (unigram, bigram and trigram) and the model proposed in the previous experiment which is the context-dependent class n-gram language model.

### 4.2.3 N-best List Re-ranking Using Semantic Knowledge

To tackle the third secondary research question, I design an experiment as follows: Firstly, I combine syntactic and semantic knowledge to re-rank the n-best list. To facilitate this, I propose a feature called semantic relatedness. I measure the semantic relatedness using the Pointwise Mutual Information approach. I use the WER-Sensitive Pairwise Perceptron algorithm to combine the proposed feature with the syntactic score and speech decoder's confidence score features. Secondly, I evaluate the proposed approach on the ATCOSIM and ATCSC corpora. I use 85% of the data from the ATCOSIM corpus for training and the renaming 15% of the data for evaluations. In order to evaluate the ability of the proposed approach in recognizing general ATC clearances, I also evaluate the proposed approach on the ATCSC corpus. I use k-fold cross-validation to increase the reliability of the evaluations. Finally, I compare the evaluation results of the proposed approach with the context-dependent class n-gram language model, traditional n-gram language models (unigram, bigram and trigram) and the approach proposed in the previous experiment which is n-best list re-ranking using syntactic score and speech decoder's confidence score features.

### 4.2.4 The Proof-of-Concept Automatic Speech Recognition System

To answer the main research question, I design an experiment as follows: I first build a Poof-of-Concept (POC) ASR system by combing the baseline ASR system, the context-dependent class n-gram language model and n-best list re-ranking using semantic relatedness, syntactic score and speech decoder's confidence score features. I then evaluate the system on the ATCOSIM and ATCSC corpora. Finally, I conduct a detailed analysis of the evaluation results, and discuss the possibilities and challenges of linguistic knowledge in improving the accuracy of ASR systems in ATC.

## 4.3 Air Traffic Control Speech Corpus (ATCSC)

In order to simulate an ATC simulation and training setting for evaluating the proposed approaches, I create a corpus called Air Traffic Control Speech Corpus (ATCSC) based on the following criteria:

- High signal quality;
- Low level background noise;
- ICAO standardized clearances;
- High quality transcriptions.

I first generate 4800 ICAO standardized clearances using 28 most frequently used templates extracted from "Doc 4444/510: Procedures for Air Navigation Services Air Traffic Management" [29]. In order to evaluate the ability of the proposed approaches in recognizing general clearances, I use a different set of location-based data (e.g., call signs, units name and navigational aids/fixes) with the ATCOSIM corpus's. I then use the generated clearances to record 4800 clearances from 12 speakers (11 male, 1 female) of three nationalities Norwegian, Swedish and Vietnamese, each reading 400 clearances. The speakers speech were picked up by the Zoom H2n Handy Recorder in a quiet room. The signals were recorded onto Waveform Audio File Format (WAV) with a sampling frequency of 44.1 kHz and a resolution 16 bit. Since the speakers were asked to read from pre-generated clearances, it is guaranteed that the quality of the corpus transcriptions is high.

# Chapter 5

# Research Findings

This chapter summarizes the research findings of the three included papers which can be found in Appendix A, Appendix B and Appendix C. The papers are presented together with the research questions that they answer. The first paper focuses on analyzing the use of linguistic knowledge in Air Traffic Control (ATC) and different language modeling approaches in order to propose a language model that is well suited for use in Automatic Speech Recognition (ASR) systems in ATC. The second and the third papers look in to the possibilities of using linguistic knowledge, particularly syntactic and semantic knowledge, in n-best list re-ranking to improve the accuracy of ASR systems.

The three included papers partly represent my journey of using linguistic knowledge to improve the accuracy of ASR systems in the ATC domain via three major steps: language modeling, n-best list re-ranking using syntactic knowledge and n-best list re-ranking using semantic knowledge.

## 5.1 Language Modeling

### 5.1.1 Research question

**RQ 1.1** Which type of language model is well suited for use in automatic speech recognition system in air traffic control domain?

### 5.1.2 Abstract

This paper, which can be found in Appendix A, describes my first step in using linguistic knowledge to improve the accuracy of ASR systems in ATC. To facilitate this, I integrate linguistic knowledge into the language modeling process, with special attention paid to address the two main challenges of language modeling in ATC, which are the lack of ATC-related corpora for training and the location-based data problem.

I propose a hybrid class n-gram language model by combining a rule-based class n-gram language model proposed by Brown et al. [7] with a n-gram language model. In order to deal with the above-mentioned two challenges, I improve the hybrid class n-gram language model by utilizing the context-dependent language modeling approach to propose a novel language model called context-dependent class n-gram. I train the proposed model via three steps: I first identify ten word classes (CALLSIGN, UNIT-NAME, FIX, NUMBER, LETTER, GREETING, NON-VERBAL-ARTICULATIONS, DIRECTION, POSITION and UNIT) based on my analysis of the ICAO standard phraseologies and the ATCOSIM corpus. I then generate a class-based training corpus by replacing words in the ATCOSIM training corpus with their corresponding class labels using the SRI Language Modeling Toolkit. Finally, I train the context-dependent class n-gram language model as a normal n-gram language model with the class-based training corpus using The CMU Statistical Language Modeling (SLM) Toolkit.

I build a baseline ASR system based on the Pocketsphinx recognizer from the CMU Sphinx framework, the CMUSphinx US English generic acoustic model and the generic cmudict_SPHINX_40 pronunciation dictionary. I integrate the proposed context-dependent class n-gram language model into the baseline system and evaluate the model in terms of Word Error Rate (WER) on the well known ATCOSIM Corpus of Non-prompted Clean Air Traffic Control Speech (ATCOSIM) and my own Air Traffic Control Speech Corpus (ATCSC). The proposed model outperforms traditional n-gram language models and shows 17.19% improvement in terms of WER on the ATCSC corpus.

Three main findings are presented in this paper. Firstly, the proposed model can address the lack of ATC-related corpora for training by adopting the hybrid class n-gram language modeling approach. Words are replaced by their corresponding class labels before training in order to reduce the amount of data required for training a high quality language model. Secondly, the proposed model can tackle the location-based data problem by enabling the integration of external data into the language model. In the training phase, location-based data (e.g., call signs, unit names and navigational aids/fixes) are replaced by their corresponding class labels in order to increase the generality of the model. In the running phase, the class members are loaded into the trained model via a class definition file, which is a file that contains class labels and their corresponding class members. The use of class labels in the training phrase and class definition file in the running phase has a great potential in addressing the location-based data problem because it facilitates the update of location-based data (class members) at run time. Finally, the significant improvement in terms of WER (17.19%) compared with traditional n-gram language models and the ability to address the two main challenges of language modeling in ATC demonstrate that the context-dependent class n-gram language model is a well suited model for use in ASR systems ATC.

## 5.2 N-best List Re-ranking Using Syntactic Knowledge

### 5.2.1 Research question

**RQ1.2** To what extent can syntactic analysis improve the accuracy of speech recognition in air traffic control domain?

### 5.2.2 Abstract

This paper, which can be found in Appendix B, summarizes my second step in using linguistic knowledge to improve the accuracy of ASR systems in ATC. To do this, I integrate the first level of linguistic knowledge, syntactic knowledge into post-processing by performing n-best list re-ranking.

I propose a novel feature called syntactic score. I compute the syntactic score by using syntactic rules which are created by replacing expansions of word classes with their corresponding class labels. I improve the syntactic score computation process by introducing context-dependent syntactic rules. First, different rule sets for different contexts are generated. Then, at the running phase, a corresponding rule set is selected based on contextual information of the system. By using different context-dependent syntactic rule sets for different contexts, the number of rules that are needed to compute syntactic score can be dramatically reduced, which can improve both accuracy and performance of the syntactic score computation process.

I propose a WER-Sensitive Pairwise Perceptron algorithm by improving the Average Perceptron algorithm in three ways: Firstly, I adopt the idea of the WER-Sensitive Perceptron algorithm presented in [53] to incorporate WER metric into the training of the perceptron. Secondly, I improve the algorithm by utilizing the pairwise ranking approach. Finally, I adopt the mini-batch gradient descent, momentum and the Bold Driver learning rate adaptation [4] approaches to optimize the perceptron training process.

I use the perceptron to combine the proposed feature with the speech decoder's confidence score feature. I use the baseline ASR system to evaluate the proposed approach in terms of WER on the ATCOSIM and ATCSC corpora. The evaluation results shows that the proposed approach reduces the WER by 1.21% and 0.21% compared with the context-dependent class n-gram language model on the ATCSC and ATCOSIM corpora respectively. In addition, the proposed approach together with the context-dependent class n-gram language model shows 18.40% improvement in terms of WER compared with traditional n-gram models on the ATCSC corpus.

This paper presents two main findings. First, the use of context-dependent syntactic rules allows the proposed approach to be easily adapted for use in new contexts without retraining, which makes the proposed approach a practical approach. Second, the difference in the evaluation results of the proposed approach on the ATCSC and ATCOSIM corpora demonstrates that the performance of n-best list re-ranking using syntactic knowledge on a corpus depends heavily on the amount of syntactic knowledge available in the corpus.

## 5.3 N-best List Re-ranking Using Semantic Knowledge

### 5.3.1 Research question

**RQ1.3** To what extent can semantic analysis improve the accuracy of speech recognition in air traffic control domain?

### 5.3.2 Abstract

This paper, which can be found in Appendix C, describes my third step in using linguistic knowledge to improve the accuracy of ASR systems in ATC. To facilitate this, I combine syntactic and semantic knowledge to re-rank the n-best list.

I propose a feature called semantic relatedness. I adopt the Pointwise Mutual Information (PMI) approach proposed in [9] to measure the semantic relatedness. The main reason that I choose the PMI approach is that it can capture long-span semantic relationships between words in ATC clearances, which are typically overlooked by n-gram language models. To address the lack of ATC-related corpora for training and the location-based data problem, I improve the PMI approach by estimating the association ratio on syntactic rules instead of original transcriptions from ATC-related corpora. I use the WER-Sensitive Pairwise Perceptron algorithm to combine the semantic relatedness, syntactic score and speech decoder's confidence score features to perform n-best list re-ranking.

I evaluate the proposed approach in terms of WER on the ATCOSIM and ATCSC corpora. The evaluations results show that the proposed approach reduces the WER by 0.31% and 1.53% compared with n-best list re-ranking using syntactic score and speech decoder's confidence score features on the ATCOSIM and ATCSC corpora respectively. In addition, the proposed approach together with the context-dependent class n-gram language model shows 20.95% improvement in terms of WER compared with traditional n-gram models on the ATCSC corpus.

This paper makes three main contributions. First, it demonstrates how can different levels of linguistic knowledge, particularly syntactic and semantic knowledge, be used together in post-processing to assist the recognition process of ASR systems in ATC. Second, it shows that the performance of n-best list re-ranking using syntactic and semantic knowledge on a corpus depends heavily on the amount of syntactic and semantic knowledge available in the corpus. Third, it reveals that the combination of the context-dependent class n-gram language model and n-best list re-ranking using syntactic and semantic knowledge has great potential in improving the accuracy of ASR systems in ATC.

## 5.4 Findings in summary

In this thesis, I take advantage of the availability of linguistic knowledge in the ATC domain to improve the accuracy of ASR systems via three steps: Firstly, I propose a hybrid language model called context-dependent class n-gram to address the two main challenges of language modeling in ATC, which are the lack of ATC-related corpora for training and the location-based data problem. Secondly, I integrate the first level of linguistic knowledge, syntactic knowledge into post-processing to improve the accuracy of ASR systems by performing n-best list re-ranking using syntactic knowledge. I propose a novel feature called syntactic score and a perceptron algorithm called WER-Sensitive Pairwise Perceptron. I use the perceptron algorithm to combine the syntactic score and speech decoder's confidence score features to re-rank the n-best list. Finally, I take this further by looking in to combining the next level of linguistic knowledge, semantic knowledge with syntactic knowledge in re-ranking the n-best list. I propose a feature called semantic relatedness. I use the WER-Sensitive Pairwise Perceptron algorithm to combine the semantic relatedness feature with the syntactic score and speech decoder's confidence score features to perform n-best list re-ranking.

The combination of the proposed approaches proposed reduces the WER by 20.95% compared with traditional n-gram language models in recognizing general clearances from the ATCSC corpus. The significant improvement in terms of WER of the proposed approaches indicates that language modeling and post-processing using linguistic knowledge have great potential in improving the accuracy of ASR systems in ATC. In addition, the difference in the evaluation results of the proposed approaches on the ATCOSIM and ATCSC corpora reveals that the performance of the proposed approaches on a corpus depends heavily on the amount of linguistic knowledge available in the corpus.

# Chapter 6

# Discussion

In this chapter, I focus on three main purposes. I first revisit the research questions introduced in Chapter 1, and point to where the relevant discussions can be found. For more details, see Chapter 5 and the three included papers which can be found in Appendix A, Appendix B and Appendix C. I then address the main research question by using the findings and contributions from the three included papers together with my understanding of Automatic Speech Recognition (ATC) technologies and the Air Traffic Control (ATC) field to discuss the possibilities of linguistic knowledge in ASR in ATC. Finally, I review five major challenges of ASR in ATC and reflect how the approaches proposed in this thesis may help to address the challenges in both ATC simulation and ATC live operations.

## 6.1 Research Questions

In the following sections, I present the research questions together with a brief summary of where the relevant findings and contributions can be found.

### 6.1.1 RQ1.1

> Which type of language model is well suited for use in automatic speech recognition system in air traffic control domain?

This research question is addressed primarily in the first paper which can be found in Appendix A. This paper presents a context-dependent class n-gram language model proposed to address the two main challenges of language modeling in ATC, which are the lack of ATC-related corpora for training and the location-based data problem. A summary of this paper can be found in Section 5.1. For more background knowledge covering language models and motivation for this research question, see Section 2.2.3 and Chapter 1.

### 6.1.2 RQ1.2

> To what extent can syntactic analysis improve the accuracy of speech recognition in air traffic control domain?

Background knowledge and related work relevant to this research question including syntactic knowledge in ATC and different approaches for performing syntactic analysis can be found in Section 2.1.3 and Section 2.3. This research question is tackled mainly in the second paper which can be found in Appendix B. This paper aims at performing n-best list re-ranking using syntactic knowledge. To facilitate this, a novel feature called syntactic score and a WER-Sensitive Pairwise Perceptron algorithm are proposed. The proposed feature is then combined with the speech decoder's confidence score feature using the perceptron algorithm to re-rank the n-best list. The findings and contributions of this paper are summarized in Section 5.2.

### 6.1.3 RQ1.3

> To what extent can semantic analysis improve the accuracy of speech recognition in air traffic control domain?

This research question is addressed primarily in the third paper which can be found in Appendix C. This paper looks into combining syntactic and semantic knowledge to re-rank the n-best list. To do this, a feature called semantic relatedness is proposed. The feature is then combined with the syntactic score and the speech decoder's confidence score features using the WER-Sensitive Pairwise Perceptron algorithm to re-rank the n-best list. A summary of this paper can be found in Section 5.3. For more background knowledge covering semantic knowledge and different approaches for performing semantic analysis, see Section 2.1.3, Section 2.3.

### 6.1.4 RQ1

> How can linguistic knowledge be used to improve automatic speech recognition accuracy in air traffic control?

I have revisited the three secondary research questions, and pointed to where the relevant findings and contributions can be found. In the following sections, I focus on answering this main research question by discussing the possibilities of linguistic knowledge in ASR in ATC, and arguing how the approaches proposed in this thesis may help to address the existing challenges of ASR in both ATC simulation and ATC live operations. The findings and contributions relevant to this research question can be found in Chapter 5 and the three included papers in Appendix A, Appendix B and Appendix C.

## 6.2 Possibilities of Linguistic Knowledge in ASR in ATC

Using linguistic knowledge in language modeling and post-processing is a potential approach for improving the accuracy of ASR systems in general. However, this approach has not been successfully applied in ASR because it is very challenging to obtain a significant amount of linguistic knowledge including syntactic, semantic and pragmatic knowledge from general speech. Fortunately, the ATC domain offers many great possibilities that can facilitate this approach.

Firstly, in order to avoid possible confusion and misunderstandings, air traffic controllers and pilots are usually required to use standard phraseologies in their communications. In addition, most of the standard procedures for air navigation services used by air traffic controllers are predefined by ICAO [29]. This means that, the amount of linguistic knowledge available in the controller-pilot communications is large which is a good fit for the above-mentioned approach.

Secondly, since air traffic controllers usually use standard pheaseologies in their communications with pilots, and follow the standard procedures provided by ICAO in most of their tasks, it is typically easy to obtain a significant amount of linguistic knowledge, particularly syntactic and semantic knowledge, in the ATC domain. For example, syntactic and semantic knowledge can be either obtained from ICAO Docs such as "Doc 4444/510: Procedures for Air Navigation Services - Air Traffic Management 15th Edition" [29] or extracted from ATC-related speech corpora. With ten word classes presented in the first paper, which can be found in Appendix A, syntactic knowledge can be extracted from a speech corpus by replacing words in the corpus with their corresponding class labels. In this thesis, with the aim of developing an ASR system that can be easily adapted for use in different contexts, I first utilize the later approach which is using the ten word classes to extract syntactic and semantic knowledge from the ATCOSIM speech corpus to generate syntactic rules. I then use the syntactic rules to compute the syntactic score and semantic relatedness features to re-rank the n-best list.

Finally, the findings presented in Chapter 5 and the three included papers reveal that pragmatic knowledge is a potential candidate for assisting syntactic and semantic knowledge in addressing the challenges of ASR in ATC. One of the main applications of pragmatic knowledge is that it can be used to limit the search space of ASR systems which can improve both systems accuracy and performance. In ASR in general, pragmatic knowledge has not been used widely because obtaining a significant amount of pragmatic knowledge is a very challenging task. Fortunately, pragmatic knowledge is typically easy to obtain in ATC. For example, location information of aircrafts can be obtained from radar information and flight plans. One possible solution to combine pragmatic knowledge with syntactic and semantic knowledge in ATC is to combine either the speech act model proposed by Karen Ward et al[64] or the cognitive model proposed by D. Schaefer [55] with syntactic and semantic analysis.

The findings presented in Chapter 5 and the three included papers show that using linguistic knowledge in language modeling reduces the WER of the baseline ASR system by 18.21% compared with traditional n-gram language models. Using linguistic knowledge in post-processing, particularly n-best list re-ranking using syntactic and semantic knowledge, reduces the WER of the system further by 2.74%. The above-mentioned possibilities and the significant improvements in terms of WER of the proposed approaches demonstrate that linguistic knowledge has great potential in addressing the two main challenges of language modeling in ATC and improving the accuracy of ASR systems in both ATC simulation and ATC live operations.

I have discussed the possibilities of linguistic knowledge in improving the accuracy of ASR systems in ATC. In the following section, I focus on arguing how the findings and contributions from the three included papers may help to address the existing challenges of ASR in both ATC simulation and ATC live operations.

## 6.3 Linguistic Knowledge and Challenges of ASR in ATC

In my previous work [45], I identified five major challenges to overcome in order to successfully apply ASR in ATC. Although the work is not a part of this thesis, I include it as Appendix D for convenience. The five major challenges are:

1. The problem of poor input signal quality;
2. Call sign detection;
3. The use of non-standard phraseology;
4. The problem of dialects, accents and multiple languages;
5. The problem of ambiguity.

The first challenge which is the problem of poor input signal quality and the fifth challenge which is the problem of ambiguity have been defined as out of the scope of this thesis. More details about the challenges can be found in Chapter 4 and Appendix D. The first challenge can be addressed by either using high quality microphones or adapting existing acoustic models. The fifth challenge can be tackled to some degree by the Natural Language Processing (NLP) module in the automated pilot system presented in Chapter 4. On the other hand, the remaining three challenges have not been successful addressed in ATC. In the following two sections, I discuss how can the approaches proposed in this thesis be used to tackle the three above-mentioned challenges, as well as how the proposed approaches facilitate the integration of ASR technologies into ATC.

### 6.3.1 Call Sign Detection

In ATC simulation and training, recognizing aircraft call signs is not a challenging task for ASR systems since the number of call signs used in a specific simulation and training session is quite small. On the other hand, because of the variety of ways to refer to the same flight call sign and the use of airline aliases, there are more than 6000 call signs that have been used in ATC live operations [28]. In addition, the call signs are usually not standard English, for instance, Speedbird, Norstar and Germanwings. This means that, call sign detection is an extremely challenging task of ASR systems in ATC live operations.

Fortunately, this challenge can be addressed to some degree by using the proposed context-dependent class n-gram language model together with n-best list re-ranking using syntactic and semantic knowledge. In the training phrase, call signs are replaced by a class label named [CALLSIGN]. In the running phrase, the class members of the [CALLSIGN] class are loaded into the trained model via a class definition file, which is a file that contains class labels and their corresponding class members. The use of the [CALLSIGN] class together with pragmatic knowledge, particularly location information of the system and aircrafts, can reduce the number of call signs that the system has to recognize. For example, radar information and flight plans could be used to reduce the list of likely aircraft call signs that a controller may refer to in a sector to only those in the sector or about to enter the sector. This means that, the proposed approaches together with pragmatic knowledge can address the call sign detection challenge of ASR in ATC live operations to some degree.

### 6.3.2 The Use of Non-Standard Phraseologies and Multiple Languages

In ATC simulation and training, air traffic controller are usually required to use standard pharesologies, thus the problems of non-standard phraseologies and multiple languages hardly occur. On the other hand, in ATC live operations, air traffic controllers frequently use non-standard phraseologies and multiple languages in their communications with pilots. For example, a controller may say:

**CL1: Guten morgen** Lufthansa one two three descend level one two zero
**CL2: Good morning** Speedbird one three three turn left to Oslo
**CL3:** Lufthansa **ah** one two three turn right to **hm** Paris

In the first clearance (CL1), the control uses two languages, German ("*Guten morgen*" is good morning in German) and English. In the second and the third clearances (CL2 and CL3), the controller uses non-standard phraseologies, which are "*good morning*", "*ah*" and "*hm*".

The two above-mentioned problems can be addressed to some degree by using the proposed context-dependent class n-gram language model together with n-best list reranking using syntactic and semantic knowledge. In the training phrase, non-standard phraseologies are replaced class labels. For example, "*Guten morgen*" and "*Good morning*" are replaced by a class label named [GREETINGS], '*ah*" and "*hm*" are replaced by a class label named [NON-VERBAL-ARTICULATIONS]. By using class labels in training instead of words, non-standard phraseologies including foreign words can be eliminated. In the running phrase, pragmatic knowledge can be used to identify which class members should be loaded into the trained model via a class definition file. For example, if the system is deployed in a center in Norway, it is likely that Norwegian controllers will use both Norwegian and English in their communications with pilots. Thus, Norwegian and English greeting phrases such as hallo, hei, god morgen, hello and good morning should be loaded into the [GREETINGS] class. By doing this, trained language models can be easily adapt to recognize non-standard phraseologies and foreign words. In other words, the proposed approaches have great potential in addressing the use of non-standard phraseologies and multiples languages challenges of ASR in ATC live operations.

# Chapter 7

# Conclusion and Further Work

## 7.1 Conclusion

In this thesis I have presented my work in using linguistic knowledge to improve the accuracy of Automatic Speech Recognition (ASR) systems in Air Traffic Control (ATC). In order to take advantage of the opportunities offered by the ATC domain such as the availability of linguistic knowledge, particularly syntactic, semantic and pragmatic knowledge, my aim has been to improve the accuracy of the ASR systems via three steps: language modeling, n-best list re-ranking using syntactic knowledge and n-best list re-ranking using semantic knowledge. The three above-mentioned steps are also the main steps that I use the address the main research question of this thesis, which is " *How can linguistic knowledge be used to improve automatic speech recognition accuracy in air traffic control?"*.

The main research question was addressed primarily in Chapter 5 and Chapter 6. The three secondary research questions were addressed mainly in Chapter 5, as well as in the three included papers which can be found in Appendix A, Appendix B and Appendix C. To answer the research questions, I first build a baseline ASR system based on the Pocketsphinx recognizer from the CMU Sphinx framework, the CMUSphinx US English generic acoustic model and the generic cmudict_SPHINX_40 pronunciation dictionary. I then improve the system by performing the above-mentioned three steps. Next, I evaluate the system in terms of Word Error Rate (WER) on the well known ATCOSIM Corpus of Non-prompted Clean Air Traffic Control Speech and my own Air Traffic Control Speech Corpus (ATCSC). Finally, I discuss the possibilities of using linguistic knowledge in improving the accuracy of ASR systems in ATC, and argue how the approaches proposed this thesis may help to address the existing challenges of ASR in both ATC simulation and ATC live operations.

This thesis makes four main contributions. Firstly, it proposes a novel language model called context-dependent class n-gram language model to address the two main challenges of language modeling in ATC, which are the lack of ATC-related corpora for training and the problem of location-based data. The second contribution is the use of the first level of linguistic knowledge, syntactic knowledge in post-processing to improve the accuracy of ASR systems. To facilitate this, I propose a novel feature called syntactic score and a WER-Sensitive Pairwise Perceptron algorithm. I use the algorithm to combine the proposed feature with the speech decoder's confidence score feature to perform n-best list re-ranking. Thirdly, it combines syntactic knowledge with the next level of linguistic knowledge, semantic knowledge to further improve the accuracy of the ASR systems.

To do this, I propose a feature called semantic relatedness. I combine the proposed feature with the syntactic score and speech decoder's confidence score features using the WER-Sensitive Pairwise Perceptron algorithm to re-rank the n-best list. The proposed approaches reduce the WER of the baseline ASR system by 20.95% compared with traditional n-gram language models in recognizing general ATC clearances from the ATCSC corpus. Finally, it demonstrates that linguistic knowledge has great potential in addressing the existing challenges of ASR in ATC and facilitating the integration of ASR technologies into the ATC domain.

## 7.2 Further Work

For further improvements, I suggest these following directions. First of all, I combine the last level of linguistic knowledge, pragmatic knowledge with syntactic and semantic knowledge to re-rank the n-best list. In ATC live operations, air traffic controllers are responsible for one or a very few specific sectors. Thus, the amount of pragmatic knowledge used by the controllers in their communications with pilots is relatively high. The use of pragmatic knowledge in post-processing not only can assist syntactic and semantic knowledge in addressing the existing challenges of ASR in ATC but aslo can improve both performnance and accuracy of the ASR systems. Secondly, I deploy and evaluate the proof-of-concept (POC) speech recognition system in terms of Word Error Rate (WER), as well as training and simulation quality in a real ATC training and simulation setting. Since ASR technologies have not been widely used in ATC, I aim at evaluating not only the accuracy of the POC speech recognition system but also how it affects the quality of ATC training and simulation. Finally, I take this further by adapting the POC speech recognition system for use in live ATC operations. Because of the special case of this project, in this thesis, I aim at developing an ASR system for ATC simulation and training. However, my final goal is to use ASR technologies to improve the performance of controller-pilot communications and increase the automation of ATC systems.

# Bibliography

[1] Federal Aviation Administration. Air traffic control - chapter 2. general control, faa 7110.65 2-1-1. Technical report, U.S. Department of Transportation, February 19, 2014.

[2] Ebru Arısoy, Brian Roark, Izhak Shafran, and Murat Saraçlar. Discriminative n-gram language modeling for turkish. In *Proc. of Interspeech*, 2008.

[3] M. Balakrishna, D. Moldovan, and E.K. Cave. N-best list reranking using higher level phonetic, lexical, syntactic and semantic knowledge sources. In *2006 IEEE International Conference on Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings.*, volume 1, pages I–I, May 2006.

[4] Roberto Battiti. Accelerated backpropagation learning: Two optimization methods. *Complex systems*, 3(4):331–342, 1989.

[5] René Beutler. *Improving speech recognition through linguistic knowledge.* PhD thesis, SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZURICH, 2007.

[6] Daniel Bolaños. The bavieca open-source speech recognition toolkit. In *SLT*, pages 354–359, 2012.

[7] Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479, 1992.

[8] Noam Chomsky. Three models for the description of language. *Information Theory, IRE Transactions on*, 2(3):113–124, 1956.

[9] Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29, 1990.

[10] José Manuel Cordero, Manuel Dorado, and José Miguel de Pablo. Automated speech recognition in atc environment. In *Proceedings of the 2nd International Conference on Application and Theory of Automation in Command and Control Systems*, pages 46–53. IRIT Press, 2012.

[11] José Manuel Cordero, Natalia Rodríguez, José Miguel, and Manuel Dorado. Automated speech recognition in controller communications applied to workload measurement. *Third SESAR Innovation Days*, 2013.

[12] Namrata Dave. Feature extraction methods lpc, plp and mfcc in speech recognition. *International Journal for Advance Research in Engineering and Technology*, 1(6):1–4, 2013.

[13] Stephen Della Pietra, Vincent Della Pietra, Robert L Mercer, and Salim Roukos. Adaptive language modeling using minimum discriminant estimation. In *Proceedings of the workshop on Speech and Natural Language*, pages 103–106. Association for Computational Linguistics, 1992.

[14] SB Dhonde and SM Jagade. Feature extraction techniques in speaker recognition: A review. *International Journal on Recent Technologies in Mechanical and Electrical Engineering (IJRMEE)*, 2(5):104–106, 2015.

[15] Loïc Dourmap and Philippe Truillet. Vocal interaction and air traffic management: The voice project. In *Int. Conf. Human-Computer Interaction in Aeronautics, Toulouse, France, Sep*, 2004.

[16] Hakan Erdogan, Ruhi Sarikaya, Stanley F Chen, Yuqing Gao, and Michael Picheny. Using semantic analysis to improve speech recognition performance. *Computer Speech & Language*, 19(3):321–343, 2005.

[17] F Fernández, J Ferreiros, JM Pardo, V Sama, R de Córdoba, J Marias-Guarasa, JM Montero, R San Segundo, LF d'Haro, M Santamaría, et al. Automatic understanding of atc speech. *Aerospace and Electronic Systems Magazine, IEEE*, 21(10):12–17, 2006.

[18] J. Ferreiros, J.M. Pardo, R. de Córdoba, J. Macias-Guarasa, J.M. Montero, F. Fernández, V. Sama, L.F. d'Haro, and G. González. A speech interface for air traffic control terminals. *Aerospace Science and Technology*, 21(1):7 – 15, 2012.

[19] Claudiu-Mihai Geacăr. Reducing pilot/atc communication errors using voice recognition. In *Proceedings of ICAS*, volume 2010, 2010.

[20] John Godfrey. Air traffic control complete ldc94s14a. web download. *Philadelphia: Linguistic Data Consortium*, 1994.

[21] Robert F Hall. Voice recognition and artificial intelligence in an air traffic control environment. Technical report, DTIC Document, 1988.

[22] H Hering. Stif interface (speech techniques for simulation facilities). *Signal*, 1(100p):2, December 1 1996.

[23] H Hering. Comparative experiments with speech recognizers for atc simulations. Technical report, EUROCONTROL, 1998.

[24] Xuedong Huang, Alex Acero, Hsiao-Wuen Hon, and Raj Foreword By-Reddy. *Spoken language processing: A guide to theory, algorithm, and system development.* Prentice Hall PTR, 2001.

[25] Andrew Hunt and Andrew Hunt. Jspeech grammar format. *W3C Note, June*, 2000.

[26] ICAO. Annex 10: Aeronautical telecommunications. volume ii - communication procedures including those with pans status. *International Civil Aviation Organization*, 2001.

[27] ICAO. Annex 11: Air traffic services. air traffic control service, flight information service, alerting service. *International Civil Aviation Organization*, 2001.

[28] ICAO. Doc 8585/155: Designators for aircraft operating agencies, aeronautical authorities and services. *International Civil Aviation Organization*, 2001.

[29] ICAO. Doc 4444/510: Procedures for air navigation services air traffic management. *International Civil Aviation Organization*, 2007.

[30] Karlsson Joakim. The integration of automatic speech recognition into the air traffic control system. Technical report, Cambridge, Mass.: Flight Transportation Laboratory, Dept. of Aeronautics and Astronautics, Massachusetts Institute of Technology,[1990], 1990.

[31] Karlsson Joakim. The integration of automatic speech recognition into the air traffic control system. Technical report, Cambridge, Mass.: Flight Transportation Laboratory, Dept. of Aeronautics and Astronautics, Massachusetts Institute of Technology,[1990], 1990.

[32] Daniel Jurafsky, Chuck Wooters, Jonathan Segal, Andreas Stolcke, Eric Fosler, G Tajchaman, and Nelson Morgan. Using a stochastic context-free grammar as a language model for speech recognition. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1, pages 189–192. IEEE, 1995.

[33] Stefan Petrik Konrad Hofbauer and Horst Hering. The atcosim corpus of non-prompted clean air traffic control speech. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may 2008. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2008/.

[34] Raymond Lau, Ronald Rosenfeld, and Salim Roukos. Adaptive language modeling using the maximum entropy principle. In *Proceedings of the workshop on Human Language Technology*, pages 108–113. Association for Computational Linguistics, 1993.

[35] Christian Mandery. *Distributed N-Gram Language Models: Application of Large Models to Automatic Speech Recognition.* PhD thesis, Informatics Institute, 2011.

[36] F Marque, SK Bennacef, F Neel, and S Trinh. Parole: a vocal dialogue system for air traffic control training. In *Applications of Speech Technology*, 1993.

[37] Sven C Martin, Jörg Liermann, and Hermann Ney. Adaptive topic-dependent language modelling using word-based varigrams. In *In Proc. Eurospeech'97.* Citeseer, 1997.

[38] Jindřich Matoušek and Daniel Tihelka. English TTS speech corpus of air traffic (pilot) messages - serbian accent, 2014. LINDAT/CLARIN digital library at Institute of Formal and Applied Linguistics, Charles University in Prague.

[39] Jindřich Matoušek and Daniel Tihelka. English TTS speech corpus of air traffic (pilot) messages - taiwanese accent, 2014. LINDAT/CLARIN digital library at Institute of Formal and Applied Linguistics, Charles University in Prague.

[40] LG Miller and SE Levinson. Syntactic analysis for large vocabulary speech recognition using a context-free covering grammar. In *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*, pages 271–274. IEEE, 1988.

[41] Mehryar Mohri, Fernando Pereira, and Michael Riley. Speech recognition with weighted finite-state transducers. In *Springer Handbook of Speech Processing*, pages 559–584. Springer, 2008.

[42] Welly Naptali, Masatoshi Tsuchiya, and Seiichi Nakagawa. Multi class-based n-gram language model for new words using web data. In *Proceedings of the 11th WSEAS international conference on robotics, control and manufacturing technology, and 11th WSEAS international conference on Multimedia systems & signal processing*, pages 125–131. World Scientific and Engineering Academy and Society (WSEAS), 2011.

[43] Welly Naptali, Masatoshi Tsuchiya, and Seiichi Nakagawa. Topic-dependent-class-based-gram language model. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(5):1513–1525, 2012.

[44] Hermann Ney. Dynamic programming speech recognition using a context-free grammar. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'87.*, volume 12, pages 69–72. IEEE, 1987.

[45] Van Nhan Nguyen and Harald Holone. Possibilities, challenges and the state of the art of automatic speech recognition in air traffic control. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 9(8):1742–1751, 2015.

[46] Takanobu Oba, Takaaki Hori, and Atsushi Nakamura. A comparative study on methods of weighted language model training for reranking lvcsr n-best hypotheses. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 5126–5129. IEEE, 2010.

[47] Organisation de l'aviation civile internationale. *Outlook for Air Transport to the Year 2025*, volume 313 of *ICAO circular*. International Civil Aviation Organization, 2007.

[48] JM Pardo, J Ferreiros, F Fernandez, Valentin Sama, R De Cordoba, Javier Macias-Guarasa, JM Montero, R San-Segundo, LF D'Haro, and Germán González. 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, 2011.

[49] Bartosz Rapp. N-gram language models for polish language. basic concepts and applications in automatic speech recognition systems. In *Computer Science and Information Technology, 2008. IMCSIT 2008. International Multiconference on*, pages 321–324. IEEE, 2008.

[50] Ariya Rastrow, Markus Dreyer, Abhinav Sethy, Sanjeev Khudanpur, Bhuvana Ramabhadran, and Mark Dredze. Hill climbing on speech lattices: A new rescoring framework. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5032–5035. IEEE, 2011.

[51] Brian Roark, Murat Saraclar, and Michael Collins. Discriminative n-gram language modeling. *Computer Speech & Language*, 21(2):373–392, 2007.

[52] Ronald Rosenfeld. A maximum entropy approach to adaptive statistical language modelling. *Computer Speech & Language*, 10(3):187–228, 1996.

[53] H. Sak, M. Saraclar, and T. Gungor. Discriminative reranking of asr hypotheses with morpholexical and n-best-list features. In *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*, pages 202–207, Dec 2011.

[54] D. Schaefer. Context-sensitive speech recognition in the air traffic control simulation. *EEC Technical/Scientific Report No. 2001-004*, 2001.

[55] Dirk Schäfer. *Context-sensitive speech recognition in the air traffic control simulation. Universität Der Bundeswehr Munchen Fakultät Fur Luft-Und Raumfahrttechnik.* PhD thesis, Ph. D. Thesis, 2001, and Eurocontrol Experimental Centre, EEC Note 02, 2001.

[56] Johan Schalkwyk, I Lee Hetherington, and Ezra Story. Speech recognition with dynamic grammars using finite-state transducers. In *INTERSPEECH*, 2003.

[57] JC Segura, T Ehrette, A Potamianos, D Fohr, I Illina, PA Breton, V Clot, R Gemello, M Matassoni, and P Maragos. The hiwire database, a noisy and non-native english speech corpus for cockpit communication. *Online. http://www. hiwire. org*, 2007.

[58] U. Sharma, S. Maheshkar, and A.N. Mishra. Study of robust feature extraction techniques for speech recognition system. In *Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), 2015 International Conference on*, pages 654–658, Feb 2015.

[59] Luboš Šmídl. Air traffic control communication, 2011. LINDAT/CLARIN digital library at Institute of Formal and Applied Linguistics, Charles University in Prague.

[60] CMU Sphinx. Cmu sphinx. open source toolkit for speech recognition. *Online. http://cmusphinx.sourceforge.net*, 2011.

[61] Stevenson. Oxford dictionary of english.

[62] Thanassis Trikas. Automated speech recognition in air traffic control. Technical report, Cambridge, Mass.: Massachusetts Institute of Technology, Dept. of Aeronautics and Astronautics, Flight Transportation Laboratory, 1987, 1987.

[63] Šmídl, Luboš and Pavel Ircing. Air traffic control communication (atcc) speech corpus. *CLARIN Annual Conference 2014 in Soesterberg, The Netherlands*, 2014.

[64] Karen Ward. *A Speech Act Model of Air Traffic Control Dialogue.* PhD thesis, Dept. of Computer Science and Engineering, Oregon Graduate Institute of Science & Technology, 1992.

[65] Wikipedia. Runway — wikipedia, the free encyclopedia, 2016. [Online; accessed 3-May-2016].

[66] Hirofumi Yamamoto, Shuntaro Isogai, and Yoshinori Sagisaka. Multi-class composite n-gram language model for spoken language processing using multiple word clusters. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 531–538. Association for Computational Linguistics, 2001.

[67] S. L. Young, A. G. Hauptmann, W. H. Ward, E. T. Smith, and P. Werner. High level knowledge sources in usable speech recognition systems. *Commun. ACM*, 32(2):183–194, February 1989.

[68] Zhengyu Zhou, Jianfeng Gao, F.K. Soong, and H. Meng. A comparative study of discriminative methods for reranking lvcsr n-best hypotheses in domain adaptation and generalization. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 1, pages I–I, May 2006.

[69] V. Zue, J. Glass, D. Goodine, H. Leung, M. Phillips, J. Polifroni, and S. Seneff. Integration of speech recognition and natural language processing in the mit voyager system. In *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*, pages 713–716 vol.1, Apr 1991.

# Appendix A

# Language Modeling

*Proceedings of the 2016 Winter Simulation Conference*
*T. M. K. Roeder, P. I. Frazier, R. Szechtman, and E. Zhou, eds.*

# USING CONTEXT-DEPENDENT CLASS N-GRAM LANGUAGE MODEL FOR IMPROVING SPEECH RECOGNITION ACCURACY IN AIR TRAFFIC CONTROL

Van Nhan Nguyen
Harald Holone

Faculty of Computer Sciences
Østfold University College
PO Box 700, 1757 Halden, NORWAY

## ABSTRACT

Recently, a lot of research has been conducted to bring Automatic Speech Recognition (ASR) into various areas of Air Traffic Control (ATC). Due to the high accuracy requirements of the ATC context and its unique challenges, ASR has not been widely adopted in this field. One of the main challenges of integrating ASR in ATC is language modeling. With the lack of ATC-related corpora for training, and the problem of location-based data, it is very difficult to train a robust language model for ATC. In this paper, we propose a context-dependent class n-gram language model. We integrate the model into the Pocketsphinx speech recognizer and evaluate the model in terms of Word Error Rate (WER) on the well known ATCOSIM and our own ATCSC corpora. Our proposed model outperforms the n-gram model and shows 18.21% WER relative improvement on the ATCSC corpus.

## 1 INTRODUCTION

Steadily increasing levels of air traffic world wide poses corresponding capacity challenges for air traffic control services. According to the "Outlook for Air Transport to the Year 2025" report of International Civil Aviation Organization (ICAO) (Organisation de l'aviation civile internationale 2007), passenger traffic on the major international routes is expected to grow about 3 to 6 percent each year through to the year 2025. Thus, ATC operations has to investigate, review and improve in order to be able to meet the increasing demands (Cordero, Dorado, and de Pablo 2012). In ATC operations, communication between controllers and pilots is one of the key components. The quality of this communication significantly affects the performance as well as the safety of ATC operations.

Integration of automatic speech recognition (ASR) technologies in the ATC domain has been investigated in order to improve the performance of controller-pilot communications and to increase the automation of ATC systems. The introduction of automatic speech recognition to ATC and the steadily improvement in accuracy and performance of ASR technologies have opened many potential opportunities to investigate, review and improve ATC operations. For example, facilitating applications such as simulating the work environment of controllers for testing and training, controller workload measurement and balancing, operational support systems for controllers which help to detect dangerous situations, and providing suggestions as well as safety information to the operators.

However, due to the high accuracy requirements of the ATC context and its unique challenges such as call sign detection, poor input signal quality, the problem of ambiguity, the use of non-standard phraseology, and the problem of dialects, accents and multiple languages (Nguyen and Holone 2015), automatic speech recognition has not been widely adopted in this field.

Therefore, in this paper, in order to take advantage of the opportunities offered by the ATC context to improve recognition accuracy of ASR systems, we aim at using different levels of linguistic knowledge to assist the recognition process of ASR systems. Our primary research question is "*how can linguistic knowledge be used to improve automatic speech recognition accuracy?*".

In ASR systems, there are many components that can be improved by using linguistic knowledge such as language models, speech decoders and post-processing modules. However, we decide to start with the language model since it is a fundamental component of ASR systems. The language model plays a critical role in ASR systems because it describes the language that the system recognizes, and biases the outputs of the system toward "grammatical" sentences based on the grammars defined by the model. Thus, the accuracy of ASR systems depends heavily on the quality of its language model.

However, creating a language model for the ATC domain is a very challenging task because of the lack of ATC-related corpora for training, and the problem of location-based data which exists in most existing ATC-related corpora. Example corpora are ATCOSIM (Konrad Hofbauer and Hering 2008), Air Traffic Control Complete LDC94S14A (Godfrey 1994), The HIWIRE database (Segura et al. 2007), and Air Traffic Control Communication (ATCC) corpus (Šmídl 2011). Since most of the existing ATC-related corpora are recorded from either only one or very few centers, the corpora typically contain a lot of location-based data such as callsigns, units name and navigational aids/fixes. The use of location-based data in training language models makes the models less general and less accurate when they are used for recognizing clearances which contain new location-based data (e.g., callsigns, unit names and navigational aids/fixes).

Therefore, in this paper, we focus on a secondary research question "*which type of language model is well suited for use in automatic speech recognition system?*". In order to answer the question and address the above-mentioned problems, we first analyze ATC context as well as different language modeling approaches. Secondly, we propose a context-dependent class n-gram language model by combining a hybrid language model and a context-dependent model. Finally, we build a baseline speech recognition system based on the Pocketsphinx recognizer from the CMU Sphinx framework (Sphinx 2011) and use the system for evaluating different types of language models in terms of Word Error Rate (WER).

Our proposed model offers two main features: Firstly, the context-dependent class n-gram language model which enables the integration of external data at run time, that can be used to solve the location-based data problem. Secondly, the proposed model can be trained using generalized data (class labels), which can address the problem of lack of ATC-related corpora for training to some degree. Since words are replaced by their corresponding class labels before training, we can train higher quality language models even with smaller corpora.

The remainder of the paper is structured as follows: Section 2 presents background and related work about n-gram and class n-gram language models, including and their existing problems, before we present descriptions about our proposed language mode in Section 3. In Section 4, we present our methodology for identifying classes, language modeling and evaluations. The evaluation results are presented in Section 5. Finally, in Section 6 and Section 7 we discuss about the properties of the proposed model and conclude the paper with a summary and further work.

## 2 BACKGROUND AND RELATED WORK

Speech recognition is the process of converting a speech signal into a sequence of words. It also called Automatic Speech Recognition (ASR) or Speech-to-Text (STT).

The general speech recognition approach can be described in two steps. 1) Given an acoustic observation, identify a feature vector sequence $X = X_1, X_2, ..., X_n$ using a feature extraction module. 2) Given this vector, find the corresponding word sequence $W = W_1, W_2, ..., W_n$ that has the maximum posterior probability $P(W \mid X)$ (Huang et al. 2001), expressed using Bayes theorem in (1).

$$W = \underset{w}{\operatorname{argmax}} P(W \mid X) = \underset{w}{\operatorname{argmax}} \frac{P(W)P(X \mid W)}{P(X)}. \tag{1}$$

In equation (1), P(W) represents the language model, which is the probability of word sequence $W = W_1, W_2, ..., W_n$ uttered. For example, for a language model describing the language that air traffic controllers and pilots use in their communication, we might have P(report speed) = 0.0001, which means that one out of every ten thousands clearances a controller may say “report speed”. On the other hand, P(I love dogs) $\approx 0$, because it is very unlikely controllers or pilots would utter such a strange clearance or response.

In ASR, there are many types of models which can used to describe language to recognize such as grammars, decision tree models and stochastic language models.

Grammars (e.g., regular grammar, context-free grammar) is a very basic approach for modeling language for ASR systems. Grammars have been widely used for modeling language for domain-dependent speech recognition systems such as call centers and command and control systems (Chomsky 1956). One example of using grammars for language modeling is the work of Jurafsky et al. (1995). The authors used a stochastic context-free grammar (SCFG). They claim that the SCFG improved the WER from 34.6% (bigram) to 29.6% (SCFG) and 28.8% (mix between bigram and SCFGLMs).

Decision tree models are binary decision trees designed to estimate the probability that a given word will be the next word uttered. According to Bahl et al. (1989), “at each node of the tree there is a yes/no question relating to the words already spoken, and at each leaf there is a probability distribution over the allowable vocabulary”. Bahl et al. (1989) used a tree-based statistical language model for natural language speech recognition. The authors claim that the tree is comparable to an equivalent trigram model on 5000-word vocabulary and is shown to be superior.

Stochastic language models (e.g., probabilistic context-free grammars, n-gram language model, class n-gram language model, adaptive language model) take a probabilistic viewpoint of language modeling (Huang et al. 2001). The main goal of stochastic language models is to assign higher probability to likely word sequences. Stochastic languages models have been widely used in modeling languages for domain-independent systems. Paeseler and Ney (1989) described the design of a stochastic language model and its integration into a continuous-speech recognition system. The authors claim that the WER was improved from 21.8% (without language model) to 9.1% (with the bigram model).

Among the above-mentioned approaches, n-gram models, especially trigram models have been used for modeling languages for state-of-the-art ASR systems (Xu and Jelinek 2007). However, n-gram language modeling typically suffered from the data sparseness problem, which is the exponential growth of the number of parameters in n-gram models as the order n increases. So, with the lack of ATC-related corpora, training a robust n-gram language model for ASR systems in the ATC domain is a very challenging task.

Fortunately, a few approaches have been proposed to deal with the problem of data sparseness. According to Xu and Jelinek (2007), smoothing is an approach which can be used to partially solve the data sparseness problem. The main idea of the smoothing approach is to assign nonzero probabilities to any word string. Studies have shown that Kneser-Ney is the best smoothing algorithm (Chen and Goodman 1996). Further improvement, class n-gram (Brown et al. 1992), which is also known as clusters of words, have recently been proved to be better than the Kneser-Ney smoothing under many test conditions in dealing with the data sparseness problem in language modeling (Xu and Jelinek 2007).

However, the class n-gram language model comes with a big challenge in searching the number of classes (clusters). Typically, experimentation is required to identify the number of classes (Xu and Jelinek 2007). In addition, using the class n-gram approach to model languages for ASR systems in the ATC domain comes with even bigger challenges. The lack of ATC-related corpora for training, and the problem of location-based data (e.g., callsigns, unit names and navigational aids/fixes) typically leads to the problem of ”unseen data” in language modeling.

In summary, we have described the general structure of an ASR system and its components. We have also presented in detail three main approaches for modeling languages for ASR systems in ATC: grammars, decision tree models and stochastic language models. Unfortunately, none of the presented approaches are well suited for use in ATC because of the unique challenges of the field which are the lack of ATC-related corpora for training and the problem of location-based data. Therefore, in this paper, we aim at addressing the challenges by combining a hybrid language model and a context-dependent model to propose a novel language modeling approach called ”context-dependent class n-gram language model”.

## 3 OUR APPROACH

With inspiration from the work of Rudnicky et al. (2000), we propose a hybrid class n-gram language model which is a combination of rule-based class n-gram language models proposed by Brown et al. (1992) and n-gram language models. In order to deal with the problem of lack of ATC-related corpora for training and location-based data in ASR systems, we improve the hybrid class n-gram language model by utilizing the ”context-dependent” language modeling approach. To facilitate this, we define three types of word classes:

- **Open class** - classes which are used to integrate external data into the language model (e.g., class “[CALLSIGN]” which includes airline telephony designators, and class “[FIX]” which includes navigational aids/fixes, class [UNIT] which includes air traffic control units name).
- **Fixed class** - classes which have fixed class members (e.g., [NUMBER], [LETTER])
- **Simple class** - classes which contain only one word.

While open classes and closed classes are manually defined in a class definition file based on the similarity in syntactic and semantic information of words, simple classes are identified from training corpora during the training process. A Simple class can be considered as a single word in word-based n-gram language model. (See Section 4.2 for more details about the language modeling process)

In our proposed model, we assume that a word $w_i$ can be uniquely mapped to only one class $c_i$. With that assumption, according to Huang et al. (2001), the class n-gram model can be computed based on the previous n-1 classes as follow:

$$P(w_i \mid c_{i-n+1}...c_{i-1}) = P(w_i \mid c_i)P(c_i|c_{i-n+1}...c_{i-1}). \tag{2}$$

Where $P(w_i \mid c_i)$ is the probability of word $w_i$ given class $c_i$ in the current position, and $P(c_i|c_{i-n+1}...c_{i-1})$ denotes the probability of class $c_i$ given n-1 previous classes.

In context-dependent class n-gram language model, the probability $P(w_i \mid c_{i-n+1}$ for Open classes and Fixed classes are computed by using equation (2). However, with simple classes, since the classes always contain only one word, $P(w_i \mid c_i)$ is always equal one. The equation (2) can be simplified as follow:

$$P(w_i \mid c_{i-n+1}...c_{i-1}) = P(c_i|c_{i-n+1}...c_{i-1}).$$

The two main goals of the proposed model are: Firstly, address the problem of lack of ATC-related corpora for training by adopting a hybrid class n-gram approach. Words are replaced by their corresponding class labels before training in order to reduce the amount of data required for training a high quality language model. Secondly, enable the integration of external data into the language model in order to address the problem of location-based data. In the training phase, location-based data (e.g., callsigns, unit names and navigational aids/fixes) are replaced by their corresponding class labels in order to increase the generality of the model. In the running phase, the class members are loaded into the trained model via a class definition file, which is the file that contains class labels and their corresponding class members. The use of class labels during training and a class definition file in the running phase has a great potential for solving the location-based data problem because it facilitates the update of location-based data (class members) at run time.

Currently, the process of switching among different class definition files are performed manually. However, with the contextual information from pragmatic analysis which is an approach that we are going to integrate into our system in the near future, the ASR system will be able to automatically choose the class definition file corresponding to the current context, state and configuration of the system (e.g., training scenario in ATC simulation and training).

In the following section, we describe how we implement the proposed approaches via three main steps: identifying classes, language modeling and evaluating with a baseline speech recognition system.

## 4 METHODOLOGY

In this section, we describe the implementation of our proposed context-dependent class n-gram language model. First of all, we identify 10 word classes based on our analysis of the ICAO standard phraseologies and ATCOSIM corpus (Konrad Hofbauer and Hering 2008). Secondly, we train our proposed language model with the identified classes using The CMU Statistical Language Modeling (SLM) Toolkit (Rosenfeld 1995) and SRILM - The SRI Language Modeling Toolkit (Stolcke et al. 2002). Finally, we build a baseline speech recognition system based on the Pocketsphinx recognizer from the CMU Sphinx framework (Sphinx 2011) and use the system for evaluating the proposed model in terms of Word Error Rate (WER).

### 4.1 Identifying Classes

Based on our analysis of the ICAO standard phraseologies (ICAO 2007), we found out that the general format of ATC clearances typically includes three main parts: Callsign (e.g., Speedbird, NordStar), Goal action (e.g., Climb, Descend, Turn left, Contact) and Goal value (e.g., radio frequency, time, flight level, name of unit). While the total number of goal actions are quite small and can be fully covered by the ICAO standard phraseologies, the number of callsigns, which is about 6000 (ICAO 2011), and goal values are very large. Thus, callsigns and goal values are the main contributors to the problem of location-based data and ”unseen data” in most of ATC-related corpora for modeling language for ASR systems. However, thanks to the ability to include syntactic and semantic knowledge to language models, the problem of location-based data and “unseen data” caused by callsigns and goal values can be resolved to some degree by manually obtaining the missing data and integrate into language models via predefined classes. Following are the 7 major classes and minor classes that we identified based on our analysis of the ICAO standard phraseologies and ATCOSIM corpus (Konrad Hofbauer and Hering 2008).

- **[CALLSIGN]** - ICAO airline designators/callsigns.
- **[UNIT-NAME]** - air traffic control units name.
- **[FIX]** - navigational aids/fixes.
- **[NUMBER]** - digits and keywords “hundred”, “thousand”.
- **[LETTER]** - ICAO phonetic spelling (e.g., alfa, bravo, charlie, delta, echo, foxtrot, golf).
- **[GREETING]** - greetings phrases (e.g., hello, good bye, good morning).
- **[NON-VERBAL-ARTICULATIONS]** - non-verbal articulations (e.g., ah, hm, ahm, yeah, aha, nah, ohh).
- **Minor classes: [DIRECTION]** (e.g., left right), **[POSITION]** (e.g., above, below), **[UNIT]** (e.g., feet, meters).

The class [CALLSIGN] can be used to handle the large diversity of callsigns. Most of the following goal values, flight level, time, speed, ratio frequency, unit names and navigational aids/fixes can be covered by classes: [UNIT-NAME], [FIX], [NUMBER] and [LETTER]. In addition, we also added the [GREETING] and [NON-VERBAL-ARTICULATIONS] classes in order to cover nonstandard phraseologies used by controllers.

However, not all identified classes are valuable for the context-dependent class n-gram model. The [NUMBER], [LETTER] and Minor Classes ([DIRECTION], POSITION] and [UNIT]) contain only words that can be found in the ICAO standard phraseologies, so ATC-related corpora (e.g., ATCOSIM) mostly covers the words. [GREETING] and [NON-VERBAL-ARTICULATIONS] classes contains words that occur in ATC clearances with a very low frequency.

In contrast, the [CALLSIGN], [UNIT-NAME] and [FIX] classes contain words that can not be found in the ATC standard phraseologies and occur with a very high frequency in ATC clearances. In addition, based on our observations of the outputs of our ASR system, most of the callsigns, units name and navigational aids/fixes are not English and hard to recognize, which can lead to a very high misrecognition rate. Following are examples of reference clearances and hypothesis clearances output by our baseline ASR system. In these clearances, Callsign (e.g., aero lloyd) and navigational aids/fixes (e.g., bilsa, gotil and fribourg) are the two parts that misrecognized by the ASR system at a very high frequency.

Reference: **aero lloyd** five six zero cleared direct **bilsa**
Hypothesis: **hello** five six zero cleared direct to **fusse**

Reference: sabena seven eight one six turn left to **gotil**
Hypothesis: sabena seven eight one six turn left to **go two**

Reference: **giant** one four four proceed to **danko**
Hypothesis: **seven** one four four proceed to **tango**

Therefore, in this paper, we focus only on training the context-dependent class n-gram models with three main classes: [CALLSIGN], [UNIT-NAME] and [FIX].

### 4.2 Language Modeling

We trained the context-dependent class n-gram language model using The CMU Statistical Language Modeling (SLM) Toolkit (Rosenfeld 1995) and SRILM - The SRI Language Modeling Toolkit (Stolcke et al. 2002). We used 85% of data from the ATCOSIM corpus for training the language model and the remaining 15% data of the corpus for evaluating the model. The training process includes three main steps: First, we created a class definition file for the above-mentioned classes. The class definition file is a file which contains class labels and lists of class members associated with the class labels. Secondly, we generated a class-based training corpus by replacing words in the ATCOSIM training corpus with their corresponding class labels by using the SRI Language Modeling Toolkit. Finally, we trained the context-dependent class n-gram language model as a normal n-gram model with the class-based training corpus using The CMU Statistical Language Modeling (SLM) Toolkit.

### 4.3 Evaluating with Baseline Speech Recognition System

We built a baseline speech recognition system based on the Pocketsphinx recognizer from the CMU Sphinx framework (Sphinx 2011) and used the system for evaluating the language models in term of Word Error Rate (WER). We also used K fold cross-validation to increase the reliability of the evaluations.

In addition, we recorded a dataset (called ATCSC - Air Traffic Control Speech Corpus) which contains 4800 ATC clearances generated from ICAO standard phraseology to evaluate the ability of the proposed model in recognizing general ATC clearances. The clearances in the corpus contain a different set of callsigns, unit names and navigational aids/fixes compared with the ATCOSIM corpus, which is the corpus used for training the language models. Thus, performance of a language model on this corpus can reflect its performance on recognizing general ATC clearances.

Based on a recommendation from the CMU Sphix framework authors, we used 20% of the data from the ATCSC corpus for acoustic adaptation and the remaining 80% of the data for evaluating the models. We also used K fold cross-validation to increase the reliability of the evaluations.

We performed the evaluations on seven different language models (US English Generic Language Model Version 5.0, ATC n-gram models (unigram, bigram and trigram) and context-dependent class-ngram models (unigram, bigram and trigram)).

In addition, in order to improve the accuracy of the baseline system, we also performed acoustic adaptation based on the training data from the ATCOSIM corpus and ATCSC corpus and pronunciation dictionary extension by adding Out-of-Vocabulary (OOV) words from the identified classes to the system's dictionary using the LOGIOS Lexicon Tool. We know that we can achieve even lower WER by using more data for acoustic adaptation, however, in order simulate a real-life setting and avoid overfitting, we performed acoustic adaptation with only 20% of the data which is also the recommended amount of data for acoustic adaptation according to the CMU Sphinx framework authors.

## 5 RESULTS

Below are the evaluation results of seven language models in terms of Word Error Rate (WER). We integrated the model into the Pocketsphinx recognizer and evaluated the model on the ATCOSIM and ATCSC corpora. Tables 1 and 2 show the average results from 3-fold cross-validation of the models on the ATCOSIM and ATCSC corpora respectively.

Table 1: The evaluation results of US English Generic Language Model 5.0 (US EGLM 5.0), ATC n-gram and context-dependent class n-gram language models on the ATCOSIM corpus in term of Word Error Rate (WER)

| Language Model | Word Error Rate (WER) |
|---|---|
| US EGLM 5.0 | 50.69% |
| Unigram | 21.59% |
| Bigram | 11.34% |
| **Trigram** | **9.69%** |
| Context-dependent class unigram | 23.57% |
| Context-dependent class bigram | 15.67% |
| **Context-dependent class trigram** | **14.20%** |

Table 2: The evaluation results of US English Generic Language Model 5.0 (US EGLM 5.0), ATC n-gram and context-dependent class n-gram language models on the ATCSC corpus in term of Word Error Rate (WER)

| Language Model | Word Error Rate (WER) |
|---|---|
| US EGLM 5.0 | 52.71% |
| Unigram | 36.49% |
| Bigram | 31.69% |
| **Trigram** | **31.58%** |
| Context-dependent class unigram | 21.52% |
| **Context-dependent class bigram** | **13.37%** |
| Context-dependent class trigram | 14.23% |

The evaluation results in Table 1 show that the generic n-gram language model is slightly better than our proposed model in recognizing ATC clearances from the ATCOSIM corpus with 9.69% WER. However, the evaluation results in Table 2 show that our proposed model outperforms the n-gram model in recognizing general ATC clearances from the ATCSC corpus with 13.37% WER.

### 5.1 Some Notes about Performance

Although it is not the main focus of the paper, we include some performance results, namely Real Time Factor (RTF) and the performance distribution of the baseline system among three main tasks: data pre-processing, feature extraction and search.

**Hardware configurations for the performance evaluations:**

- PC: Dell Optiplex 9020
- CPU: Intel Core i5-4690 Processor (Quad Core, 6MB, 3.50GHz w/HD4600 Graphics)
- RAM: 16 GB
- HDD: 256 GB SSD

Table 3 shows the RTF and the average time for recognizing an ATC clearance for the same seven models on the Pocketsphinx speech recognizer and their performance distribution among the three main tasks of ASR systems: data pre-processing, feature extraction and search.

Table 3: The evaluation results of performance distribution and Real Time Factor (RTF) of the baseline system with US English Generic Language Model 5.0 (US EGLM 5.0), ATC n-gram and context-dependent class n-gram language model

| **Language Model** | **Performance Distribution** | | | **Recognition Time (second)** | **RTF** |
|---|---|---|---|---|---|
| | **Data Preprocessing** | **Feature Extraction** | **Search** | | |
| US EGLM 5.0 | 28.59% | 0.17% | 71.24% | 6.3 | 1.76 |
| Unigram | 30.44% | 0.35% | 69.21% | 3.33 | 0.87 |
| Bigram | 22.56% | 0.42% | 77.40% | 2.71 | 0.74 |
| **Trigram** | 20.47% | 0.44% | 79.09% | **2.58** | **0.70** |
| Context-dependent class unigram | 28.12% | 0.33% | 71.55% | 3.37 | 0.92 |
| Context-dependent class bigram | 25.69% | 0.35% | 73.96% | 3.24 | 0.88 |
| Context-dependent class trigram | 25.01% | 0.36% | 74.63% | 3.16 | 0.86 |

The results show that, the trigram language model is the model that requires the least amount of time for recognizing ATC clearances. Search and and data pre-processing are the tasks that require the most processing time among the three evaluated tasks with average about 75% and 24.5% respectively.

## 6 DISCUSSION

In order to answer the research question "*which type of language model is well suited for use in automatic speech recognition system?*", we first evaluated different state-of-the-art language models. Then, we proposed a context-dependent class n-gram language model by combining a hybrid language model and a context-dependent language model. Finally, we compared our proposed model with state-of-the-art language models to identify the language model which is well suited for use in ASR in ATC.

The evaluation results in Table 1 show that n-gram language models are slightly better than our proposed models in recognizing ATC clearances from the ATCOSIM corpus. The best n-gram model is the trigram model with 9.69% WER, while the best of our proposed models is the context-dependent class trigram model with 14.20% WER.

The reason why n-gram models are very good at recognizing the ATC clearances from the ATCOSIM corpus is that the ATCOSIM corpus is the corpus used for training the n-gram models. Although, the corpus was split into two different parts for training and testing, and k-fold cross-validation was used to increase the reliability of the evaluations, both training and testing data still contain a lot of repetitions of location-based data (e.g., callsigns, units name, navigational aids/fixes, radio frequency) because all clearances of the corpus were recorded from the same control room of the EUROCONTROL Experimental Centre (EEC).

For example, in the ATCOSIM corpus, in 1500 times that the callsign “lufthansa” occurs, it is followed by the word sequence “three five” 243 times, “eight two” 151 times. The repetition of callsigns at high frequency leads to the high probability of corresponding sequence of words in the n-gram model, which can cause a problem called “overfitting”. When a n-gram model overfits, it tends to remember the training data, for example, “lufthansa” should be followed either by “three five” or “eight two”. So, when the testing data contains the same repetitions, the n-gram model can typically achieve a very low WER.

Therefore, in order to evaluate the ability of the proposed models in recognizing general ATC clearances, we recorded 4800 general ATC clearances (ATCSC corpus) which contain different location-based data (e.g., callsigns, units name, navigational aids/fixes) compared with the ATCOSIM corpus. The evaluation results in Table 2 show that our proposed model outperforms the n-gram models in recognizing general ATC clearances. The context-dependent class bigram (13.37% WER) shows 18.21% WER improvement compared with trigram model (31.58% WER), which is the best of the n-gram models.

The improvements in WER of the context-dependent class n-gram model on the new ATCSC corpus (not used for training the model) shows that our proposed model is better than the traditional n-gram models in recognizing general ATC clearances which contain different location-based data (e.g., callsigns, units name, navigational aids/fixes). The proposed model also demonstrates how linguistic knowledge can be used to improve automatic speech recognition accuracy in air traffic control via language modeling.

## 7 CONCLUSION AND FURTHER WORK

In this paper, we proposed a context-dependent class n-gram language model which enables the integration of external data into language model at run time and the use of class labels for training instead of words to address the problem of location-based data, and the lack of ATC-related corpora for training language models for ASR systems in ATC.

We evaluated the language models in terms of Word Error Rate (WER) on the ATCOSIM corpus. We used 85% of data from the corpus for training and the remaining 15% data for evaluations. We also used K fold cross-validation to increase the reliability of the evaluations.

In addition, we recorded a dataset called “ATCSC - Air Traffic Control Speech Corpus” which contains 4800 ATC clearances generated from ICAO standard phraseology to evaluate the ability of the proposed model in recognizing general ATC clearances. The clearances in the corpus contain a different set of callsigns, units name and navigational aids/fixes compared with clearances in the ATCOSIM corpus which is the corpus used for training the language models. Thus, performance of the model on this corpus can reflect its performance when recognizing general ATC clearances. We used 20% of the data from the ATCSC corpus for acoustic adaptation and the remaining 80% of the data for evaluations. In order to increase the reliability of the evaluations, we also used K fold cross-validation.

We compared the evaluation results of the proposed context-dependent class n-gram language models (unigram, bigram, trigram) with traditional n-gram language models (unigram, bigram, trigram) and the US English Generic Language Model 5.0 provided by CMU.

The evaluation results show that the context-dependent class n-gram models outperform the n-gram models in recognizing general ATC clearances from the ATCSC corpus. The context-dependent class bigram (13.37% WER) shows 18.21% WER improvement compared with trigram model (31.58% WER), which is the best of the n-gram models. It is possible to achieve lower WER by using more data for acoustic adaptation. For example, we can achieve 9.44% WER by performing acoustic adaptation with 85% of the data. However, in order to simulate a real-life setting and avoid overfitting, we only use 20% of the data, which is the recommended amount of data for acoustic adaptation according to the CMU Sphinx framework authors.

We now intend to take this further by integrating higher level linguistic knowledge, syntactic, semantic and pragmatic knowledge in post-processing to improve our systems accuracy. One possible approach is to use the linguistic knowledge to assist a widely used post-processing process called n-best list reranking.

## ACKNOWLEDGMENT

The authors would like to thank The Institute for Energy Technology (John E. Simensen and Christian Raspotnig), Edda Systems AS, and WP3 of "User driven innovation in smart homes for health and energy" at Østfold University College for support in the work with this paper.

## REFERENCES

Bahl, L. R., P. F. Brown, P. V. De Souza, and R. L. Mercer. 1989. "A tree-based statistical language model for natural language speech recognition". *Acoustics, Speech and Signal Processing, IEEE Transactions on* 37 (7): 1001–1008.

Brown, P. F., P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. 1992. "Class-based n-gram models of natural language". *Computational linguistics* 18 (4): 467–479.

Chen, S. F., and J. Goodman. 1996. "An empirical study of smoothing techniques for language modeling". In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, 310–318. Association for Computational Linguistics.

Chomsky, N. 1956. "Three models for the description of language". *Information Theory, IRE Transactions on* 2 (3): 113–124.

Cordero, J. M., M. Dorado, and J. M. de Pablo. 2012. "Automated speech recognition in ATC environment". In *Proceedings of the 2nd International Conference on Application and Theory of Automation in Command and Control Systems*, 46–53. IRIT Press.

Godfrey, J. 1994. "Air Traffic Control Complete LDC94S14A. Web Download.". *Philadelphia: Linguistic Data Consortium*.

Huang, X., A. Acero, H.-W. Hon, and R. Foreword By-Reddy. 2001. *Spoken language processing: A guide to theory, algorithm, and system development*. Prentice Hall PTR.

ICAO 2007. "Doc 4444/510: Procedures for Air Navigation Services Air Traffic Management". *International Civil Aviation Organization*.

ICAO 2011. "Doc 8585/155: Designators for Aircraft Operating Agencies Aeronautical Authorities and Services". *International Civil Aviation Organization*.

Jurafsky, D., C. Wooters, J. Segal, A. Stolcke, E. Fosler, G. Tajchaman, and N. Morgan. 1995. "Using a stochastic context-free grammar as a language model for speech recognition". In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, Volume 1, 189–192. IEEE.

Konrad Hofbauer, S. P., and H. Hering. 2008, may. "The ATCOSIM Corpus of Non-Prompted Clean Air Traffic Control Speech". In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, edited by B. M. J. M. J. O. S. P. D. T. Nicoletta Calzolari (Conference Chair), Khalid Choukri. Marrakech, Morocco: European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2008/.

Nguyen, V. N., and H. Holone. 2015. "Possibilities, Challenges and the State of the Art of Automatic Speech Recognition in Air Traffic Control". *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering* 9 (8): 1742–1751.

Organisation de l'aviation civile internationale 2007. *Outlook for Air Transport to the Year 2025*, Volume 313 of *ICAO circular*. International Civil Aviation Organization.

Paeseler, A., and H. Ney. 1989. "Continuous-speech recognition using a stochastic language model". In *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*, 719–722. IEEE.

Rosenfeld, R. 1995. “The CMU Statistical Language Modeling Toolkit and its use in the 1994 ARPA CSR Evaluation”. In *ARPA SLT*.

Rudnicky, A. I., C. Bennett, A. W. Black, A. Chotomongcol, K. Lenzo, A. Oh, and R. Singh. 2000. “Task and domain specific modelling in the Carnegie Mellon Communicator system”. *CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE*.

Segura, J., T. Ehrette, A. Potamianos, D. Fohr, I. Illina, P. Breton, V. Clot, R. Gemello, M. Matassoni, and P. Maragos. 2007. “The HIWIRE database, a noisy and non-native English speech corpus for cockpit communication”. *Online. http://www.hiwire.org*.

Šmídl, Luboš 2011. “Air Traffic Control Communication”. LINDAT/CLARIN digital library at Institute of Formal and Applied Linguistics, Charles University in Prague.

Sphinx, C. 2011. “CMU Sphinx. Open Source Toolkit For Speech Recognition”. *Online. http://cmusphinx.sourceforge.net*.

Stolcke, A. et al. 2002. “SRILM-an extensible language modeling toolkit.”. In *INTERSPEECH*.

Xu, P., and F. Jelinek. 2007. “Random forests and the data sparseness problem in language modeling”. *Computer Speech & Language* 21 (1): 105–152.

## AUTHOR BIOGRAPHIES

**VAN NHAN NGUYEN** is a Master Student at the Faculty of Computer Sciences at Østfold University College. His research interests include speech recognition, digital image processing and machine learning. His email address is nhan.v.nguyen@hiof.no.

**HARALD HOLONE** is Dean and Associate Professor at the the Faculty of Computer Sciences at Østfold University College. His research interests include novel interaction forms between humans and computers, the use of computer support in work environments, and corresponding design processes. His email address is h@hiof.no.

# Appendix B

# N-best List Re-ranking Using Syntactic Knowledge

2016 16th International Conference on Control, Automation and Systems (ICCAS 2016)
Oct. 16–19, 2016 in HICO, Gyeongju, Korea

# N-best List Re-ranking Using Syntactic Score: A Solution for Improving Speech Recognition Accuracy in Air Traffic Control

Van Nhan Nguyen and Harald Holone

Faculty of Computer Sciences
Østfold University College
PO Box 700, 1757 Halden, Norway
nhan.v.nguyen@hiof.no, h@hiof.no

**Abstract:** Recently, a lot of research has been conducted to bring Automatic Speech Recognition (ASR) into various areas of Air Traffic Control (ATC), such as ATC simulation and training, monitoring live operators for with the aim of safety improvements, ATC workload measurement and conducting analysis on large quantities of controller-pilot speech. Due to the high accuracy requirements of the ATC context and its unique challenges, ASR has not been widely adopted in this field. In this paper, in order take advantage of the opportunities offered by the ATC context such as standardized phraseology and small vocabulary size to reduce the Word Error Rate (WER) of ASR in ATC, we perform n-best list re-ranking using syntactic knowledge. We propose a novel feature called syntactic score which is computed using syntactic rules. We also propose a WER-Sensitive Pairwise Perceptron algorithm and use the perceptron to combine the proposed feature with the speech decoder's confidence score. We integrate the model into the Pocketsphinx speech recognizer and evaluate the model in terms of WER on the well known ATCOSIM and our own ATCSC corpora. The results shows that our proposed approach reduces 1.21% and 0.21% WER on the ATCSC and ATCOSIM corpora respectively.

**Keywords:** N-best List Re-ranking, Syntactic Knowledge, Automatic Speech Recognition, Air Traffic Control.

## 1. INTRODUCTION

The steady increase in levels of air traffic world wide creates an urgent need to investigate, review and improve Air Traffic Control (ATC) operations [1]. In ATC operations, communication between controllers and pilots is one of the key components. The quality of this communication significantly affects the performance as well as the safety of ATC operations. So, in the past few years, many attempts have been made to integrate Automatic Speech Recognition (ASR) technologies into ATC to improve the performance of controller-pilot communications and to increase the automation of ATC systems [2].

However, this technology has not been successfully adopted in this field because of its high accuracy requirements and unique challenges. For example, call sign detection, poor input signal quality, the problem of ambiguity, the use of non-standard phraseology and the problem of dialects, accents and multiple languages [3].

Fortunately, ATC domain offers many great opportunities to address the above-mentioned challenges such as small vocabulary size, standardized phraseology and the availability of linguistic knowledge such as syntactic, semantic and pragmatic knowledge.

The work presented in this paper is a part of an ongoing work involves taking advantage of the above-mentioned opportunities to integrate linguistic knowledge into ASR systems to improve recognition accuracy. In our previous work [4], we proposed a context-dependent class n-gram language model and built a baseline speech recognition system based on the Pocketsphinx recognizer from the CMU Sphinx framework [5]. Our proposed model outperformed the traditional n-gram model and showed 18.21% improvement in terms of Word Error Rate (WER) on our own Air Traffic Control Speech Corpus (ATCSC).

In this paper, we take this further by integrating syntactic knowledge into post-processing to assist ASR systems recognition process and improve recognition accuracy. Other linguistic knowledge such as semantic and pragmatic knowledge will be considered in our future work.

To facilitate the integration of syntactic knowledge into post-processing, we utilize the well known n-best list re-ranking approach. We propose a novel feature called syntactic score which is computed using syntactic rules. We also propose a WER-Sensitive Pairwise Perceptron algorithm and use the perceptron to combine the proposed feature with the speech decoder's confidence score to perform n-best list re-ranking.

We evaluate the proposed approach on the well known ATCOSIM Corpus of Non-prompted Clean Air Traffic Control Speech [6] and our own ATCSC corpus. The ATCSC corpus is a 4800 clearances corpus recorded using clearances generated from ICAO standardized phraseologies. We use K-fold cross-validation to increase the reliability of the evaluations.

The remainder of the paper is structured as follows: Section 2 presents background and related work covering different approaches for integrating syntactic knowledge into ASR systems, before we present the results of our preliminary tests in Section 3. In Section 4, we describe our proposed syntactic score feature and show how it is computed using syntactic rules. Section 5 describes our proposed WER-Sensitive Pairwise Perceptron algorithm. The evaluating settings and results are presented in Section 6. Finally, in Section 7 and Section 8 we discuss the properties of the proposed approach and conclude the paper with a summary and future work.

## 2. BACKGROUND AND RELATED WORKS

Speech recognition comes naturally to human being. We can easily listen to others and understand them even with people we never met before. In some cases, we can understand speech even when we mishear some words. We can also understand ungrammatical utterances or new expressions. These happens because we use not only acoustic information but also linguistic and contextual information to interpret speech.

On the other hand, speech recognition has been considered a difficult task for machines. Because unlike humans, machines typically use only acoustic information to perform speech recognition.

Over the past few years, many attempts has been made to integrate linguistic and contextual information into ASR systems to improve recognition accuracy. Typically, there are three main approaches that can be used to facilitate the integration of linguistic knowledge, particularly syntactic knowledge, into ASR systems: language modeling, N-best filtering and re-ranking, and word lattice filtering and re-ranking.

In our previous work, we covered the language modeling approach by proposing a context-dependent class n-gram language model [4]. We now take this further by utilizing the n-best list re-ranking approach.

### 2.1 Language Modeling

The main idea of this approach is to integrate syntactic knowledge into decoding to guide the search process. The main advantage of this approach is that it can reduce the search space in decoding which increases both accuracy and performance of the system.

Syntactic knowledge can be integrated into ASR systems via language modeling. Grammars is one of the most popular approach for integrating syntactic knowledge into language models. For example, L. Miller et al. used context-free grammars as the language model for a large vocabulary speech recognition [7].

In our previous work, we proposed a context-dependent class n-gram language model which enables the integration of external data into language models at run time and the use of class labels for training instead of words to solve the problem of lack of ATC-related corpora for training and location-based data in modeling language for ASR systems in ATC domain [4].

The language modeling process can be summarized as follows: Firstly, we proposed a hybrid class n-gram language model which is a combination of rule-based class n-gram language models and n-gram language models to address the problem of lack of ATC-related corpora for training. Secondly, in order to address the problem of location-based data, we improved the hybrid class n-gram language model by utilizing the context-dependent language modeling approach.

We implemented our proposed context-dependent class n-gram language model via three steps: First of all, we identified 10 word classes based on our analysis of the ICAO standard phraseologies and the ATCOSIM corpus. Secondly, we generated a class-based training corpus by replacing words in the ATCOSIM training corpus with their corresponding class labels. Finally, we trained the context-dependent class n-gram language model as a normal n-gram model with the class-based training corpus.

### 2.2 N-best Filtering and Reranking

N-best list re-ranking have been widely used for improving ASR systems accuracy. The main ideal of this approach is to re-score N-best hypotheses and then use the scores to perform re-ranking. The hypothesis that ranked highest will be the output of the system.

There are many different methods that can be used to perform N-best list re-ranking. For example, Z. Zhou et al. conducted a comparative study of discriminative methods: perceptron, boosting, ranking support vector machine (SVM) and minimum sample risk (MSR) for N-best list re-ranking in both domain adapting and generalizing task [8]. Another example is the work of T. Oba et al [9]. The authors compared three methods, Reranking Boosting (ReBst), Minimum Error Rate Training (MERT) and the Weighted Global Log-Linear Model (W-GCLM) for training discriminative n-gram language models for a large vocabulary speech recognition task.

With regard to N-best filtering, the main idea is to verify the list of N-best hypotheses which are already sorted by score with a verifier. The first hypothesis accepted by the verifier will be the output of the system. One approach that have been widely used to perform N-best filtering is using a natural language processing (NLP) module as a verifier [10].

### 2.3 Lattice Filtering and Re-raking

Lattices is a directed graph which represents a set of hypothesized words with different starting and ending positions in the input signal. Lattices are typically used to represent search results and served as intermediate format between recognition passes.

The main idea of lattices filtering and re-ranking is to first generate lattices and then use post-processing parser to filter or re-rank the lattices [11]. One example is the work of Ariya Rastrow et al [12]. The authors proposed an approach for re-scoring speech lattices based on hill climbing via edit-distance based neighborhoods.

## 3. PRELIMINARY TESTS

In this section, we investigate whether our selected n-best list re-ranking approach has potential to facilitate the integration syntactic knowledge in to ASR systems in ATC by performing preliminary tests. The main goal of the tests are to test our hypothesis: “N-best list re-ranking can be used to improve ASR systems accuracy in ATC” and identify n-best hypotheses list size.

The tests are performed in three steps: Firstly, we use our baseline ASR system to generate N n-best hypotheses. Secondly, we filter the hypotheses by removing duplications, and then sort them by speech decoder’s confidence score. Finally, we choose M topmost ranked hypotheses, which is also called ”N-best hypotheses list size”, and calculate WER of each hypothesis with its corresponding reference hypothesis in order to get the oracle WER.

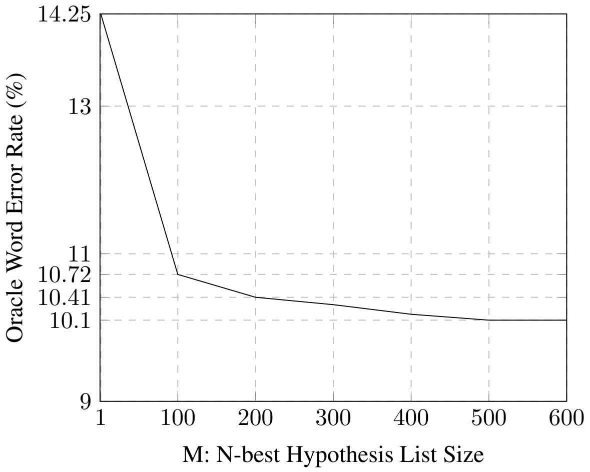

Fig. 1 Oracle Word Error Rate (WER) with different n-best hypothesis list size.

The testing results in Figure 1 shows that the oracle WER stops dropping at M = 500. The results also show that choosing the best hypothesis from the 500 topmost ranked hypotheses gives 4.24% WER improvement over the baseline ASR system. So our hypothesis ‘N-best list re-ranking can be used to improve ASR systems accuracy in ATC” is confirmed.

We have established that n-best list re-ranking is a valid approach for improving ASR systems accuracy in ATC. In the next two sections, we take this further by identifying features and proposing algorithms to implement the approach.

## 4. FEATURES FOR N-BEST LIST RE-RANKINNG

To implement the n-best list re-ranking approach, we propose a feature called syntactic score to take advantage of the availability of syntactic knowledge in ATC. The syntactic score is computed by using syntactic rules which are created by replacing expansions of word classes with their corresponding class labels.

We use 10 classes, which is identified based on our analysis of the ICAO standard phraseologies and the ATCOSIM corpus in our previous work [4], to create the syntactic rules.

- **[CALLSIGN]** - ICAO airline designators/callsigns.
- **[UNIT-NAME]** - air traffic control units name.
- **[FIX]** - navigational aids/fixes.
- **[NUMBER]** - digits and keywords “hundred”, “thousand”.
- **[LETTER]** - ICAO phonetic spelling (e.g., alfa, bravo).
- **[GREETING]** - greetings phrases (e.g., hello).
- **[NON-VERBAL-ARTICULATIONS]** - non-verbal articulations (e.g., ah, hm, ahm, yeah, aha, nah, ohh).
- **Minor classes: [DIRECTION]** (e.g., left, right), **[POSITION]** (e.g., above, below), **[UNIT]** (e.g., feet).

Below are some sample clearances and their corresponding syntactic rules:

```
speedbird one two nine turn left to london
[CALLSIGN] turn [DIRECTION] to [CITY]

lot three six one fly heading three two zero
[CALLSIGN] fly heading [NUMBER] [NUMBER] [NUMBER]
```

We use Word Error Rate (WER) as the metric for computing syntactic score for all hypotheses. The computing process can be briefly described as follows: First, We generate n-best hypotheses $h = (h_1, h_2, ...h_n)$ by using the speech decoder. After that, with each hypothesis $h_i$, we search among the syntactic rules $r = (r_1, r_2, ...r_n)$ to find a syntactic rule $r_j$ in which the pair $(h_i, r_j)$ has the lowest WER. The WER of the pair $(h_i, r_j)$ is used as the syntactic score $S(h_i)$ for the hypothesis $h_i$. Basically, the syntactic score computing process can be defined as follows:

$$S(h_i) = \underset{r' \in r}{\operatorname{argmin}}(WER(h_i, r')), \text{ where } h_i \in h.$$

In order to improve the accuracy of the n-best list re-ranking approach, we combine the syntactic score feature with the well known speech decoder’s confidence score feature.

### 4.1 Context-dependent Syntactic Rules

We improve the syntactic score computation process by introducing context-dependent syntactic rules. We first generate different rule sets for different contexts and then during run time, the corresponding rules set is selected based on current contextual information.

By using different context-dependent syntactic rules sets for different contexts, we can reduce the number of rules that are needed to compute the syntactic score, which can improve both accuracy and performance of the syntactic score computation process.

The main benefit of using context-dependent syntactic rules is that the system can be easily adapted to a new context without re-training, which makes our proposed approach a practical approach.

Currently, the selection of rule sets is performed manually, however, with the availability of pragmatic knowledge and contextual information from pragmatic analysis the rule sets selection process can be performed automatically. We are planing to implement this in our future work.

# 5. N-BEST LIST RE-RANKING WITH PERCEPTRON

In machine learning, the perceptron is a linear classifier. The main goal of the peceptron algorithm is to learn a weight vector that minimizes the number of misclassifications [13]. In the field of speech recognition, the variants of the perceptron algorithm have been proved to be very successful in re-ranking n-best list [8][14].

In this paper, we improve the Average Perceptron algorithm presented in [15] to combine the two following features for re-ranking the n-best list:

- $D_1$: Syntactic score
- $D_2$: Speech decoder's confidence score

We use the following definitions and notions adapted from [8][14] to describe the perceptron algorithm:

- With each utterance $x_i$ in a training set which includes n utterances, define $x_{i,j}$ as the $j$-th hypothesis and $y_i$ as the oracle hypothesis of the utterance $x_i$.
- Define D+1 features $f_d(h), d = 0...D$, h is a hypothesis.
- Define a function $f(h) = (f_0(h), f_1(h), ..., f_D(h))$ which can map each hypothesis $h_i$ to a feature vector $f(h_i) = (f_0(h_i), f_1(h_i), ..., f_D(h_i))$.
- Define $\Delta(x_{ij}, y_i)$ as the difference in WER of $x_{ij}$ (the $j$-th hypothesis of the utterance $x_i$) and $y_i$ (the oracle hypothesis of utterance $x_i$) with the reference transcription of utterance $x_i$.
- Define $0 < \alpha < 1$ as the momentum constant.
- Define $adapt\_lr(\eta, w, \bar{w})$ as a version of the Bold Driver learning rate adaptation function. The function is simple: after each utterance $x_i$, compare perceptron's loss $L(w_t(i))$ to its previous value, $L(w_t(i-1))$. If the error has increased by more than a tiny proportion (say, $10^{-10}$), undo the last weight change, and decrease the learning rate $\eta$ sharply - typically by 50%. If the error has decreased, increase the learning rate $\eta$ by a small proportion (typically 1%-5%). In order to improve the training performance, we make two minor modifications to the original Bold Driver learning rate adaptation algorithm. We increase the learning rate $\eta$ even when the error remained unchanged and reset the learning rate $\eta$ to its initial value after each iteration.

## 5.1 The WER-Sensitive Pairwise Perceptron Algorithm

We improve the Average Perceptron algorithm (Algorithm 1) in three ways:

**Algorithm 1** The average perceptron algorithm

```
input set of training examples $(x_i, y_i) : 1 \leq i \leq n$
Input number of iterations $T$
$w = 0, \bar{w} = 0$
for $t = 1...T, i = 1...n$ do
    Choose the $x_{ij}$ with the largest $f(x_{ij}) \cdot w$
    if $x_{ij} \neq y_i$ then
        $w = w + \eta(f(y_i) - f(x_{ij}))$
    end if
    $\bar{w} = \bar{w} + w$
end for
return $\bar{w}/(nT)$
```

First, we adopt the idea of the WER-Sensitive Perceptron algorithm presented in [14] to incorporate WER metric into the training of the perceptron. Secondly, we improve the algorithm by utilizing the pairwise ranking approach. We define a better word error rate sensitive pairwise loss function as follows:

$$L(w) = \sum_{i=1}^{n} \sum_{j=1}^{m} \Delta(x_{ij}, y_i)[\![(w \cdot f(x_{ij}) - w \cdot f(y_i)]\!]$$

Where $[\![x]\!] = 0$ if $x < 0$ and 1 otherwise. Finally, we adopt the mini-batch gradient descent, momentum and the Bold Driver learning rate adaptation [16] approaches to optimize the perceptron training process. In addition, in order to improve the quality of the perceptron training process, we also adopt the practical tricks presented in [17] for shuffling the examples, normalizing the inputs and initializing the weights. The full version of our perceptron algorithm is described in Algorithm 2.

**Algorithm 2** The WER-sensitive pairwise perceptron

```
input set of training examples $(x_i, y_i) : 1 \leq i \leq n$
Input n-best hypotheses list size $m$
Input number of iterations $T$
$w = 0, \bar{w} = 0$
for $t = 1...T, i = 1...n$ do
    $\Delta w_t(i) = 0$
    for $j = 1...m$ do
        if $f(x_{ij}) \cdot w > f(y_i) \cdot w$ then
            $\Delta w_t(i) = \Delta w_t(i) + \Delta(x_{ij}, y_i)(f(y_i) - f(x_{ij}))$
        end if
    end for
    $\Delta w_t(i) = \Delta w_t(i)/m$
    $w = w + \eta \Delta w_t(i) + \alpha \Delta w_t(i-1)$
    $\bar{w} = \bar{w} + w$
    $adapt\_lr(\eta, w, \bar{w})$
end for
return $\bar{w}/(nT)$
```

Given a training set size $n$, define $m$ as the n-best hypotheses list size and $T$ as the number of iterations. The complexity of the WER-Sensitive Pairwise Perceptron algorithm without learning rate adaptation is the same as the Average Perceptron algorithm, $O(nmT)$. With learning rate adaptation, the complexity of our proposed approach is $O(n^2mT)$. This is not a big problem because the learning rate adaptation can speed up the training process by reducing the number of iterations $T$ that is needed for the perceptron to converge.

## 6. EVALUATING SETTINGS AND RESULTS

### 6.1 Evaluating Settings

First, we use the Pocketsphinx recognizer, the CMUSphinx US English generic acoustic model, the generic cmudict_SPHINX_40 pronunciation dictionary and the context-dependent class n-gram language model proposed in our previous work [4] to build a baseline speech recognition system.

Then, we use the baseline system to evaluate our proposed approach on the ATCOSIM and ATCSC corpora. We use 85% of the data from the corpora for training language models and adapting acoustic models, we use the remaining 15% of the data for evaluations. We also use k-fold cross-validation to increase to reliability of the evaluations.

### 6.2 Results

Table 1 The evaluation results of traditional n-gram, context-dependent class n-gram (C-DC n-gram) models, and the WER-Sensitive Pairwise Perceptron (WER-SPP) algorithm on the ATCOSIM and ATCSC corpora.

| Models -Algorithms | Speech Corpora | |
|---|---|---|
| | ATCOSIM | ATCSC |
| N-gram | 9.69% | 31.58% |
| C-DC n-gram 500-best oracle | 8.51% | 8.45% |
| C-DC n-gram 1-best | 12.62% | 14.39% |
| WER-SPP Algorithm<br>+ Syntactic score<br>+ Decoder's confidence score | 12.41% | 13.18% |

The evaluations results in Table 1 show that our proposed approach reduces the WER by 0.21% and 1.21% compared with the context-dependent class n-gram model on the ATCOSIM and ATCSC corpora respectively. Our proposed approach also shows 18.40% improvement in terms of WER compared with traditional n-gram model on the ATCSC corpus.

As explained in our previous work [4], the evaluation result of the n-gram model on the ATCOSIM corpus (9.69% WER) is not relevant for comparison because the model was trained and evaluated using data from the same corpus which contains a lot of repetitions of location-based data.

## 7. DISCUSSION

The evaluations results show that our proposed approach reduces the WER by 0.21% and 1.21% compared with the context-dependent class n-gram model on the ATCOSIM and ATCSC corpora respectively.

The main reason that the proposed approach shows a very small improvement in term of WER on the ATCOSIM corpus is that it contains a large number of non-standard phraseologies and clearances. Since the syntactic score of a hypothesis reflects how well the hypothesis matches its closest standardized clearance, the performance of the n-best list re-ranking process using syntactic score on a corpus depends heavily on the proportion of the standardized clearances in the corpus.

The ATCSC corpus, on the other hand, is recorded using ICAO standardized clearances, so the proportion of the standardized clearances in the corpus is relatively high. Thus, the performance of the n-best hypotheses re-ranking process using the syntactic score feature on the corpus is higher. This resulted in 1.21% improvement in terms of WER.

The evaluation results demonstrates that if we use more syntactic knowledge in the n-best list re-ranking process, the recognition accuracy can be improved significantly.

The work described in this paper is first aimed at integrating speech technologies into air traffic control simulation and training environment in which air traffic controller students are usually required to use ICAO standard phraseology. This means that the proportion of standardized clearances is a good fit for our proposed approach.

## 8. CONCLUSION AND FURTHER WORK

In this paper, in order take advantage of the opportunities offered by the ATC context such as standardized phraseology and small vocabulary size to improve the accuracy of ASR in ATC, we perform n-best list re-ranking using syntactic knowledge.

To facilitate the re-ranking process, we propose a novel feature called syntactic score. We compute the syntactic score using syntactic rules which are generated by a syntactic rules generator by replacing expansions of word classes with their corresponding class labels. We also propose a WER-sensitive pairwise perceptron algorithm and use the perceptron to combine the proposed feature with the speech decoder's confidence score.

We use the baseline system proposed in our previous work to evaluate our proposed approach in terms of Word Error Rate (WER) on the well known ATCOSIM and our own ATCSC corpora. We compare the evaluation results of the proposed approach with traditional n-gram and context-dependent class n-gram models.

The evaluations results show that our proposed approach reduces the WER by 0.21% and 1.21% compared with the context-dependent class n-gram model on the ATCOSIM and ATCSC corpora respectively. Our proposed approach also shows 18.40% WER improvement compared with traditional n-gram model on the ATCSC corpus.

We now intend to take this further by integrating higher level linguistic knowledge such as semantic and pragmatic knowledge into post-processing to assist the recognition process.

## ACKNOWLEDGMENT

The authors would like to thank The Institute for Energy Technology (John E. Simensen and Christian Raspotnig), Edda Systems AS, and WP3 of "Smart Buildings for Welfare (SBW)" at Østfold University College for support in the work with this paper.

## REFERENCES

[1] Organisation de l'aviation civile internationale, *Outlook for Air Transport to the Year 2025*, vol. 313 of *ICAO circular*. International Civil Aviation Organization, 2007.

[2] J. M. Cordero, M. Dorado, and J. M. de Pablo, "Automated speech recognition in atc environment," in *Proceedings of the 2nd International Conference on Application and Theory of Automation in Command and Control Systems*, pp. 46–53, IRIT Press, 2012.

[3] V. N. Nguyen and H. Holone, "Possibilities, challenges and the state of the art of automatic speech recognition in air traffic control," *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, vol. 9, no. 8, pp. 1742–1751, 2015.

[4] V. N. Nguyen and H. Holone, "Using context-dependent class n-gram language model for improving speech recognition accuracy in air traffic control," in *Manuscript submitted for publication*, 2016.

[5] C. Sphinx, "Cmu sphinx. open source toolkit for speech recognition," *Online. http://cmusphinx.sourceforge.net*, 2011.

[6] S. P. Konrad Hofbauer and H. Hering, "The atcosim corpus of non-prompted clean air traffic control speech," in *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)* (B. M. J. M. J. O. S. P. D. T. Nicoletta Calzolari (Conference Chair), Khalid Choukri, ed.), (Marrakech, Morocco), European Language Resources Association (ELRA), may 2008. http://www.lrec-conf.org/proceedings/lrec2008/.

[7] L. Miller and S. Levinson, "Syntactic analysis for large vocabulary speech recognition using a context-free covering grammar," in *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*, pp. 271–274, IEEE, 1988.

[8] Z. Zhou, J. Gao, F. Soong, and H. Meng, "A comparative study of discriminative methods for reranking lvcsr n-best hypotheses in domain adaptation and generalization," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 1, pp. I–I, May 2006.

[9] T. Oba, T. Hori, and A. Nakamura, "A comparative study on methods of weighted language model training for reranking lvcsr n-best hypotheses," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pp. 5126–5129, IEEE, 2010.

[10] V. Zue, J. Glass, D. Goodine, H. Leung, M. Phillips, J. Polifroni, and S. Seneff, "Integration of speech recognition and natural language processing in the mit voyager system," in *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*, pp. 713–716 vol.1, Apr 1991.

[11] R. Beutler, *Improving speech recognition through linguistic knowledge*. PhD thesis, SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZURICH, 2007.

[12] A. Rastrow, M. Dreyer, A. Sethy, S. Khudanpur, B. Ramabhadran, and M. Dredze, "Hill climbing on speech lattices: A new rescoring framework," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 5032–5035, IEEE, 2011.

[13] R. Frank, "The perceptron a perceiving and recognizing automaton," tech. rep., Technical Report 85-460-1, Cornell Aeronautical Laboratory, 1957.

[14] H. Sak, M. Saraclar, and T. Gungor, "Discriminative reranking of asr hypotheses with morpholexical and n-best-list features," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*, pp. 202–207, Dec 2011.

[15] M. Collins, "Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms," in *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, (Stroudsburg, PA, USA), pp. 1–8, Association for Computational Linguistics, 2002.

[16] R. Battiti, "Accelerated backpropagation learning: Two optimization methods," *Complex systems*, vol. 3, no. 4, pp. 331–342, 1989.

[17] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*, pp. 9–48, Springer, 2012.

# Appendix C

# N-best List Re-ranking Using Semantic Knowledge

2016 16th International Conference on Control, Automation and Systems (ICCAS 2016)
Oct. 16–19, 2016 in HICO, Gyeongju, Korea

# N-best List Re-ranking Using Semantic Relatedness: An Approach for Improving Speech Recognition Accuracy in Air Traffic Control

Van Nhan Nguyen and Harald Holone

Faculty of Computer Sciences
Østfold University College
PO Box 700, 1757 Halden, Norway
nhan.v.nguyen@hiof.no, h@hiof.no

**Abstract:** In this paper, with the aim of bringing Automatic Speech Recognition (ASR) technologies into Air Traffic Control (ATC), we investigate how we can take advantage of the availability of linguistic knowledge in the ATC context to reduce the Word Error Rate (WER) of ASR systems by performing n-best list re-ranking. We first propose a feature called semantic relatedness. We then use a WER-Sensitive Pairwise Perceptron algorithm which is proposed in our previous work to combine the semantic relatedness, syntactic score and speech decoder's confidence score features to perform n-best list re-ranking. We evaluate the proposed approach in terms of WER on the well known ATCOSIM Corpus of Non-prompted Clean Air Traffic Control Speech (ATCOSIM) and our own Air Traffic Control Speech Corpus (ATCSC). The evaluations results show that our proposed approach reduces the WER by 0.31% and 1.53% on the ATCOSIM and ATCSC corpora respectively. Our proposed approach also shows 19.93% WER improvement compared with traditional n-gram model on the ATCSC corpus.

**Keywords:** N-best List Re-ranking, Semantic Knowledge, Automatic Speech Recognition, Air Traffic Control.

## 1. INTRODUCTION

In the past few years, many attempts have been made to integrate Automatic Speech Recognition (ASR) into Air Traffic Control (ATC) to increase the automation of ATC systems. However, this technology has not been successfully adopted in this field because of its high accuracy requirements and unique challenges. For example, call sign detection, poor input signal quality, the problem of ambiguity, the use of non-standard phraseology and the problem of dialects, accents and multiple languages [1].

With the aim of bringing ASR technologies into ATC, we have been investigating how we can take advantage of the availability of linguistic knowledge in the ATC context to address the above-mentioned challenges.

The work presented in this paper is a part of an ongoing work involves using linguistic knowledge to improve the accuracy ASR systems in ATC. In our previous work [2], we proposed a context-dependent class n-gram language model and built a baseline speech recognition system based on the Pocketsphinx recognizer from the CMU Sphinx framework [3]. We also integrated syntactic knowledge into post-processing to assist ASR systems recognition process by performing n-best list re-ranking with syntactic knowledge [5]. We proposed a novel feature call syntactic score and a WER-Sensitive Pairwise Peceptron algorithm to combine the proposed feature with the speech decoder's confidence score. Our proposed approach outperformed tradition n-gram model and showed 18.4% improvement in terms of Word Error Rate (WER) on our own Air Traffic Control Speech Corpus (ATCSC).

In this paper, we take this further by looking into combining syntactic and semantic knowledge in re-ranking the n-best list to improve the accuracy of ASR systems in ATC.

In order to take advantage of the availability of syntactic and semantic knowledge in the ATC context, we first propose a feature called semantic relatedness. We then use the WER-Sensitive Pairwise Peceptron algorithm to combine the proposed feature with the syntactic score and speech decoder's confidence score features to perform n-best list re-ranking.

We evaluate the proposed approach on the well known ATCOSIM Corpus of Non-prompted Clean Air Traffic Control Speech [4] and our own ATCSC corpus. The ATCSC corpus is a 4800 clearances corpus recorded using clearances generated from ICAO standardized phraseologies. We use K-fold cross-validation to increase the reliability of the evaluations.

The remainder of the paper is structured as follows: Section 2 presents background and related work covering the n-best list re-ranking approach, and semantic relatedness and semantic similarity, before we present our proposed feature, semantic relatedness in Section 3. In section 4, we describe the perceptron algorithm proposed in our previous work for combining features for n-best list re-ranking. The evaluating settings and results are presented in Section 5. Finally, in Section 6 and Section 7, we discuss the properties of the proposed approach and conclude the paper with a summary and further work.

## 2. BACKGROUND AND RELATED WORK

In our previous work [5], we investigated how we can use the first level of linguistic knowledge, syntactic knowledge to improve the accuracy of ASR systems in ATC by performing n-best list re-ranking. We now take this further by looking into how can the syntactic knowledge be used together with the next level of linguistic knowledge, semantic knowledge in re-ranking the n-best list.

### 2.1 N-best Re-ranking

N-best list re-ranking have been widely used for improving ASR systems accuracy. The main ideal of this approach is to re-score N-best hypotheses and then use the scores to perform re-ranking. The hypothesis that ranked highest will be the output of the system.

There are many different methods that can be used to perform N-best list re-ranking. For example, Z. Zhou et al. conducted a comparative study of discriminative methods: perceptron, boosting, ranking support vector machine (SVM) and minimum sample risk (MSR) for N-best list re-ranking in both domain adapting and generalizing task [6]. Another example is the work of T. Oba et al [7]. The authors compared three methods, Reranking Boosting (ReBst), Minimum Error Rate Training (MERT) and the Weighted Global Log-Linear Model (W-GCLM) for training discriminative n-gram language models for a large vocabulary speech recognition task.

### 2.2 Semantic Relatedness and Semantic Similarity

Semantic relatedness is the degree of any semantic relation (e.g., antonym, meronymy) between two words, terms or documents whereas semantic similarity is a special case of semantic relatedness which only includes "is a" relations. For example "apple" is similar to "orange", but is only related to "juice" and "pie". Both semantic relatedness and semantic similarity are fundamental and widely used concepts for measuring semantic relations.

In this paper, in order to capture all semantic relations between ATC standardized phraseologies, we use semantic relatedness as a feature for performing n-best list re-ranking instead of semantic similarity.

There are many semantic relatedness measures have been proposed, and they can be generally categorized into two categories: knowledge-based measures and corpus-based measures. While knowledge-based measures employ information extracted from lexical resources such as dictionaries [8] (e.g., Longman Dictionary of Contemporary English), thesaurus [9] (e.g., Rogets Thesaurus), WordNet and other semantic networks [10], corpus-based measures utilize probabilistic approaches to extract semantics relations between words, terms or documents from text corpora, for examples, Latent Semantic Analysis (LSA) [11], Pointwise Mutual Information (PMI) [12], Second Order Co-occurrence PMI (SOC-PMI) [13], Distributional Similarity [14] and Salient Semantic Analysis [15].

In the following section, we show how we turn the semantic relatedness measure into a feature for re-ranking the n-best list.

## 3. SEMANTIC RELATEDNESS FEATURE

In order to take advantage of the availability of linguistic knowledge in the ATC context, we utilize the well known n-best list re-ranking approach to integrate linguistic knowledge into post-processing to improve the accuracy of ASR systems. In our previous work [5], we started with the first level of linguistic knowledge, syntactic knowledge by proposing a featured called syntactic score to perform n-best list re-ranking. We now take this further by looking in to combining the syntactic knowledge with the next level of linguistic knowledge, semantic knowledge. The last level of linguistic knowledge, pragmatic knowledge will be considered in our future work.

To combine the syntactic and semantic knowledge in re-ranking the n-best list, we propose a feature called semantic relatedness. We adopt the Pointwise Mutual Information (PMI) approach proposed in [12] to measure the semantic relatedness. The main reason that we choose the PMI approach is that it can capture long-span semantic relationships between words in ATC clearances, which typically overlooked by n-gram language models.

To address the problem of lack of training corpora and "location-based" data, which occur in most of exiting ATC-related corpora, we improve the PMI approach by estimating the association ratio on syntactic rules instead of original ATC-related speech corpora. In addition, we also make three assumptions about semantic relatedness that we believe to be reasonable:

- **Assumption 1:** If x is followed by y in the syntactic rules in R, x and y is semantic related.
- **Assumption 2:** If either x or y does not occur in the R or x is not followed by y in any syntactic rules in R, x and y is not semantic related.
- **Assumption 3:** Let d(x,y) be the distance from x to y. If x is not followed by y with exact distance d(x,y) in any syntactic rules in R, x and y is not semantic related.

Based on the above-mentioned assumptions, the process of calculating semantic relatedness using pointwise mutual information is described as follows: With $C = c_1, c_2, ..., c_m$ denotes an ATC-related speech corpus. Let $R = r_1, r_2, ..., r_m$ be a set of syntactic rules generated by replacing expansions of word classes with their corresponding class labels. We use 10 classes, which is identified based on our analysis of the ICAO standard phraseologies and the ATCOSIM corpus in our previous work [2], to create the syntactic rules.

- **[CALLSIGN]** - ICAO airline designators/callsigns.

- **[UNIT-NAME]** - air traffic control units name.
- **[FIX]** - navigational aids/fixes.
- **[NUMBER]** - digits and keywords “hundred”, “thousand”.
- **[LETTER]** - ICAO phonetic spelling (e.g., alfa, golf).
- **[GREETING]** - greetings phrases (e.g., hello).
- **[NON-VERBAL-ARTICULATIONS]** - non-verbal articulations (e.g., ah, hm, ahm, yeah, aha, nah, ohh).
- **Minor classes: [DIRECTION]** (e.g., left, right), **[POSITION]** (e.g., above, below), **[UNIT]** (e.g., feet).

Let $P(x), P(y)$ be the probabilities of word x and y, $P(x, y)$ be the probability of x is followed by y in the syntactic rules in R, d(x,y) is the distance from x to y and $P(d(x, y))$ is the probability of x is followed by y with exact distance d(x,y) in R. The mutual information I(x,y) [12], which is also the semantic relatedness between x and y, is defined as follow:

$$I(x,y) = \begin{cases} \log_2\left(\frac{P(x,y)P(d(x,y))}{P(x)P(y)} + \alpha\right), & \text{if } P(d(x,y)) > 0, \\ 0, & \text{otherwise} \end{cases}$$

Where $\alpha$ is a smoothing constant used to guarantee that the mutual information I(x,y) between x and y is always greater than zero if x and y are semantic related. The word probabilities P(x) and P(y) are estimated by counting the number of observations of x and y in the syntactic rules set $R$ , and normalizing by N, the size of the syntactic rules set. The joint probabilities $P(x, y)$, are estimated by counting the number of times that x is followed by y in the syntactic rules in $R$, and normalizing by N. d(x,y) is calculated by counting the number of words between x and y plus one. $P(d(x, y))$ is calculated by dividing the distance $d(x, y)$ by the number of x is followed by y in R.

With the above-mentioned assumptions, the accumulated semantic relatedness of a clearance $X = x_1, x_2, ..., x_n$ is defined as follow:

$$I(X) = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i}^{n} I(x_i, x_j)$$

Where n is the size of the clearance.

We have demonstrated how we turn the semantic relatedness measure into a feature for re-ranking the n-best list. In the next section, we describe how we use a perceptron algorithm to combine the semantic relatedness, syntactic score and speech decoder’s confidence score features to perform n-best list re-ranking.

## 4. N-BEST LIST RE-RANKING USING PERCEPTRON

To perform n-best list re-reanking, we use the WER-Sensitive Pairwise Perceptron, which proposed in our previous work [5], to combine the three following features:

- $D_1$: Syntactic score
- $D_2$: Speech decoder’s confidence score
- $D_3$: Semantic Relatedness

The algorithm is described using definitions and notions adapted from [6][16] as follows:

- With each utterance $x_i$ in a training set which includes n utterances, define $x_{i,j}$ as the $j$-th hypothesis and $y_i$ as the oracle hypothesis of the utterance $x_i$.
- Define D+1 features $f_d(h), d = 0...D$, h is a hypothesis.
- Define a function $f(h) = (f_0(h), f_1(h), ..., f_D(h))$ which can map each hypothesis $h_i$ to a feature vector $f(h_i) = (f_0(h_i), f_1(h_i), ..., f_D(h_i))$.
- Define $\Delta(x_{ij}, y_i)$ as the difference in WER of $x_{ij}$ (the $j$-th hypothesis of the utterance $x_i$) and $y_i$ (the oracle hypothesis of utterance $x_i$) with the reference transcription of utterance $x_i$.
- Define $0 < \alpha < 1$ as the momentum constant.
- Define $adapt\_lr(\eta, w, \bar{w})$ as a version of the Bold Driver learning rate adaptation function [17]. The function is simple: after each utterance $x_i$, compare perceptron’s loss $L(w_t(i))$ to its previous value, $L(w_t(i-1))$. If the error has increased by more than a tiny proportion (say, $10^{-10}$), undo the last weight change, and decrease the learning rate $\eta$ sharply - typically by 50%. If the error has decreased, increase the learning rate $\eta$ by a small proportion (typically 1%-5%). In order to improve the training performance, we make two minor modifications to the original Bold Driver learning rate adaptation algorithm. We increase the learning rate $\eta$ even when the error remained unchanged and reset the learning rate $\eta$ to its initial value after each iteration.

**Algorithm 1** The WER-Sensitive Pairwise Perceptron

```
input set of training examples $(x_i, y_i) : 1 \leq i \leq n$
Input n-best hypotheses list size $m$
Input number of iterations $T$
$w = 0, \bar{w} = 0$
for $t = 1...T$, $i = 1...n$ do
    $\Delta w_t(i) = 0$
    for $j = 1...m$ do
        if $f(x_{ij}) \cdot w > f(y_i) \cdot w$ then
            $\Delta w_t(i) = \Delta w_t(i) + \Delta(x_{ij}, y_i)(f(y_i) - f(x_{ij}))$
        end if
    end for
    $\Delta w_t(i) = \Delta w_t(i)/m$
    $w = w + \eta\Delta w_t(i) + \alpha\Delta w_t(i-1)$
    $\bar{w} = \bar{w} + w$
    $adapt\_lr(\eta, w, \bar{w})$
end for
return $\bar{w}/(nT)$
```

We have demonstrated how we combine the semantic relatedness, syntactic score and speech decoder’s confidence score features using the WER-Sensitive Pairwise Perceptron algorithm. In the following section, we show how the approach is evaluated with the ATCOSIM and ATCSC corpora.

## 5. EVALUATING SETTINGS AND RESULTS

### 5.1 Evaluating Settings

First, we use the Pocketsphinx recognizer, the CMUSphinx US English generic acoustic model, the generic cmudict_SPHINX_40 pronunciation dictionary and the context-dependent class n-gram language model proposed in our previous work [2] to build a baseline speech recognition system.

Then, we use the baseline system to evaluate our proposed approach on the ATCOSIM and ATCSC corpora. We compare our proposed approach with traditional n-gram and context-dependent class n-gram models, and the WER-Sensitive Pairwise Perceptron algorithm with two features, syntactic score and decoder's confidence score. We use 85% of the data from the corpora for training language models and adapting acoustic models, we use the remaining 15% of the data for evaluations. We use k-fold cross-validation to increase to reliability of the evaluations.

### 5.2 Results

Table 1 The evaluation results of traditional n-gram and context-dependent class n-gram (C-DC n-gram) models, and the WER-Sensitive Pairwise Perceptron (WER-SPP) algorithm with three features, syntactic score, decoder's confidence score and semantic relatedness on the ATCOSIM and ATCSC corpora.

| Models -Algorithms | Speech Corpora | |
|---|---|---|
| | ATCOSIM | ATCSC |
| N-gram | 9.69% | 31.58% |
| C-DC n-gram 500-best oracle | 8.51% | 8.45% |
| C-DC n-gram 1-best | 12.62% | 14.39% |
| WER-SPP algorithm<br>+ Syntactic score<br>+ Decoder's confidence score | 12.41% | 13.18% |
| WER-SPP algorithm<br>+ Syntactic score<br>+ Decoder's confidence score<br>+ Semantic Relatedness | 12.10% | 11.65% |

The results show that using our proposed semantic relatedness feature in n-best list re-ranking reduces the WER by 0.31% and 1.53% on the ATCOSIM and ATCSC corpora respectively.

As explained in our previous work [2], the evaluation result of the n-gram model on the ATCOSIM corpus (9.69% WER) is not relevant for comparison because the model was trained and evaluated using data from the same corpus which contains a lot of repetitions of location-based data. The repetitions of location-based data in both training and testing data led to a problem called overfitting, which resulted in a very high WER (31.58%) when the models were used for recognizing general ATC clearances from the ATCSC corpus. Fortunately, our approach demonstrates that the combination of a context-dependent class n-gram language model and n-best list re-ranking using semantic relatedness and syntactic score can overcome the overfitting problem and improve the WER by 19.93% on the ATCSC corpus.

## 6. DISCUSSION

The evaluations results show that our proposed approach reduces the WER by 0.31% and 1.53% compared with the context-dependent class n-gram model on the ATCOSIM and ATCSC corpora respectively.

The significant difference between the evaluation results of the proposed approach on the ATCOSIM and ATCSC corpora indicates that the performance of the n-best list re-ranking process using linguistic knowledge on a corpus depends heavily on the amount of linguistic knowledge available in the corpus. This demonstrates that linguistic knowledge has great potential in solving the existing challenges of ASR in contexts which have significant amount of linguistic knowledge.

The work described in this paper is first aimed at integrating ASR into ATC simulation and training environment in which air traffic controller students are usually required to use standardized phraseologies. This means that the amount of linguistic knowledge available is relatively high, which is a good fit for our proposed approach.

The 19.93% improvement in terms of WER on the ATCSC corpus, which is not the corpus used for training, reveals that the combination of a context-dependent class n-gram language models, and n-best list re-ranking using linguistic knowledge (semantic relatedness and syntactic score), can be easily adapted to recognize general ATC clearances. This makes our proposed approach a practical approach because the models can be easily adapted to new contexts without re-training.

## 7. CONCLUSION AND FURTHER WORK

In this paper, in order take advantage of the availability of linguistic knowledge in the ATC context to improve the accuracy of ASR, we perform n-best list re-ranking using semantic knowledge.

To facilitate the re-ranking process, we first propose a feature called semantic relatedness which is measured using the Pointwise Mutual Information approach. We then use the WER-Sensitive Pairwise Perceptron algorithm proposed in our previous work to combine the semantic relatedness feature with the syntactic score and speech decoder's confidence score features.

We use our baseline ASR system to evaluate our proposed approach in terms of Word Error Rate (WER) on the well known ATCOSIM and our own ATCSC corpora. We compare our proposed approach with traditional n-gram and context-dependent class n-gram language models, and the WER-Sensitive Pairwise Perceptron algorithm with two features, syntactic score and decoder's confidence score

The evaluations results show that our proposed approach reduces the WER by 0.31% and 1.53% on the ATCOSIM and ATCSC corpora respectively. Our proposed approach also shows 19.93% WER improvement compared with traditional n-gram model on the ATCSC corpus.

We now intend to take this further by integrating the last level of linguistic knowledge, pragmatic knowledge into post-processing to improve the accuracy of ASR systems in ATC.

## ACKNOWLEDGMENT

The authors would like to thank The Institute for Energy Technology (John E. Simensen and Christian Raspotnig), Edda Systems AS, and WP3 of "Smart Buildings for Welfare (SBW)" at Østfold University College for support in the work with this paper.

## REFERENCES

[1] V. N. Nguyen and H. Holone, "Possibilities, challenges and the state of the art of automatic speech recognition in air traffic control," *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, vol. 9, no. 8, pp. 1742–1751, 2015.

[2] V. N. Nguyen and H. Holone, "Using context-dependent class n-gram language model for improving speech recognition accuracy in air traffic control," in *Manuscript submitted for publication*, 2016.

[3] C. Sphinx, "Cmu sphinx. open source toolkit for speech recognition," *Online. http://cmusphinx.sourceforge.net*, 2011.

[4] S. P. Konrad Hofbauer and H. Hering, "The atcosim corpus of non-prompted clean air traffic control speech," in *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)* (B. M. J. M. J. O. S. P. D. T. Nicoletta Calzolari (Conference Chair), Khalid Choukri, ed.), (Marrakech, Morocco), European Language Resources Association (ELRA), may 2008. http://www.lrec-conf.org/proceedings/lrec2008/.

[5] V. N. Nguyen and H. Holone, "N-best list re-ranking using syntactic score: A solution for improving speech recognition accuracy in air traffic control," in *Manuscript submitted for publication*, 2016.

[6] Z. Zhou, J. Gao, F. Soong, and H. Meng, "A comparative study of discriminative methods for reranking lvcsr n-best hypotheses in domain adaptation and generalization," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 1, pp. I–I, May 2006.

[7] T. Oba, T. Hori, and A. Nakamura, "A comparative study on methods of weighted language model training for reranking lvcsr n-best hypotheses," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pp. 5126–5129, IEEE, 2010.

[8] H. Kozima and T. Furugori, "Similarity between words computed by spreading activation on an english dictionary," in *Proceedings of the sixth conference on European chapter of the Association for Computational Linguistics*, pp. 232–239, Association for Computational Linguistics, 1993.

[9] J. Morris and G. Hirst, "Lexical cohesion computed by thesaural relations as an indicator of the structure of text," *Computational linguistics*, vol. 17, no. 1, pp. 21–48, 1991.

[10] S. Patwardhan and T. Pedersen, "Using wordnet-based context vectors to estimate the semantic relatedness of concepts," in *Proceedings of the EACL 2006 Workshop Making Sense of Sense-Bringing Computational Linguistics and Psycholinguistics Together*, vol. 1501, pp. 1–8, Citeseer, 2006.

[11] M. W. Berry, S. T. Dumais, and G. W. O'Brien, "Using linear algebra for intelligent information retrieval," *SIAM review*, vol. 37, no. 4, pp. 573–595, 1995.

[12] K. W. Church and P. Hanks, "Word association norms, mutual information, and lexicography," *Computational linguistics*, vol. 16, no. 1, pp. 22–29, 1990.

[13] A. Islam and D. Inkpen, "Second order co-occurrence pmi for determining the semantic similarity of words," in *Proceedings of the International Conference on Language Resources and Evaluation, Genoa, Italy*, pp. 1033–1038, 2006.

[14] D. Lin, "An information-theoretic definition of similarity," in *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, (San Francisco, CA, USA), pp. 296–304, Morgan Kaufmann Publishers Inc., 1998.

[15] S. Hassan and R. Mihalcea, "Semantic relatedness using salient semantic analysis.," in *AAAI*, 2011.

[16] H. Sak, M. Saraclar, and T. Gungor, "Discriminative reranking of asr hypotheses with morpholexical and n-best-list features," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*, pp. 202–207, Dec 2011.

[17] R. Battiti, “Accelerated backpropagation learning: Two optimization methods,” *Complex systems*, vol. 3, no. 4, pp. 331–342, 1989.

# Appendix D

# Possibilities, Challenges and the State of the Art of ASR in ATC

# Possibilities, Challenges and the State of the Art of Automatic Speech Recognition in Air Traffic Control

Van Nhan Nguyen, Harald Holone

*Abstract*—Over the past few years, a lot of research has been conducted to bring Automatic Speech Recognition (ASR) into various areas of Air Traffic Control (ATC), such as air traffic control simulation and training, monitoring live operators for with the aim of safety improvements, air traffic controller workload measurement and conducting analysis on large quantities controller-pilot speech. Due to the high accuracy requirements of the ATC context and its unique challenges, automatic speech recognition has not been widely adopted in this field. With the aim of providing a good starting point for researchers who are interested bringing automatic speech recognition into ATC, this paper gives an overview of possibilities and challenges of applying automatic speech recognition in air traffic control. To provide this overview, we present an updated literature review of speech recognition technologies in general, as well as specific approaches relevant to the ATC context. Based on this literature review, criteria for selecting speech recognition approaches for the ATC domain are presented, and remaining challenges and possible solutions are discussed.

*Keywords*—Automatic Speech Recognition, ASR, Air Traffic Control, ATC.

## I. Introduction

STEADILY increasing levels of air traffic world wide poses corresponding capacity challenges for air traffic control services. According to the “Outlook for Air Transport to the Year 2025” report of International Civil Aviation Organization (ICAO) [55], passenger traffic on the major international routes is expected to grow about 3 to 6 percent each year through to the year 2025. Thus, ATC operations has to investigate, review and improve in order to be able to meet with the increasing demands [9]. In ATC operations, communication between controllers and pilots is one of the key components. The quality of this communication significantly affects the performance as well as the safety of ATC operations.

Integration of automatic speech recognition (ASR) technologies in the ATC domain has been investigated in order to improve the performance of controller-pilot communications and to increase the automation of ATC systems. The introduction of automatic speech recognition to ATC and the steadily improvement in accuracy and performance of ASR technologies have opened many potential opportunities to investigate, review and improve ATC operations. For example, facilitating applications such as simulating the work environment of controllers for testing and training, controller workload measurement and balancing, assistant systems that support controllers in operational environment by catching potential dangerous situations that might be missed by the controllers, and providing suggestions as well as safety information to the operators.

Automatic speech recognition (ASR) technology, which is capable of translating human speech into sequences of words, has advanced significantly over the past decades. By 2015, ASR technologies has been successfully used in many applications like dictation, command and control, voice user interfaces such as voice dialing or call routing, medical applications, personal assistants on mobile phones, home automation, and automatic voice translation into foreign languages [52].

However, integrating ASR technologies into the ATC domain comes with many challenges such as call sign detection, poor input signal quality, the problem of ambiguity and the use of non-standard phraseology which dramatically reduce the recognition rate and the performance of speech recognition systems. Although the integration of ASR technologies into the ATC domain was introduced in the early 90s (or earlier) [30], it still has not been able to provide acceptable results in terms of recognition rate and overall performance.

With the aim of providing a comprehensive overview of current state-of-the-art speech recognition technologies, challenges as well as possibilities for applying ASR in the ATC domain, we have conducted a thorough literature review.

Based on the literature we identify five major existing challenges which make the integration of ASR technologies to the ATC domain difficult, and suggest possible approaches to address the challenges and improve the recognition rate of ASR systems in the ATC domain. Criteria for selecting ASR systems which well suited for use in ATC domain were also identified. The main contribution of this paper is to provide a fundamental starting point for researchers who are interested in integrating ASR systems in the ATC domain for both operational and simulation environments.

The remainder of the paper is structured as follows: Section II describes the methodology for conducting the literature review, before we present general introduction to automatic speech recognition, classification of ASR approaches as well as history of the field in section III. In Section IV we presents a brief introduction to air traffic control, possible applications of ASR in the ATC domain, criteria for selecting ASR approaches for the ATC domain, and an extended literature review of ASR research relevant to ATC. Finally, in Section V and Section

Authors are with Faculty of Computer Science, Østfold University College, PO Box 700, 1757 Halden, Norway, emails: nhan.v.nguyen@hiof.no, h@hiof.no.

VI we identify remaining challenges for ASR in ATC, discuss possible solutions to these challenges, and conclude the paper with a summary and outlook for this field.

## II. Methodology

The literature review was conducted using the following keyword phrases: "Speech Recognition in Air Traffic Control OR Voice Recognition in Air Traffic Control", "Speech Command Recognition OR Voice Command Recognition", and "Medium Vocabulary AND Continuous Speech Recognition AND Speaker Independent". Searches were performed in ACM Digital Library, IEEEXplore Digital Library, Google Scholar and Google Search. From the search results we identified and reviewed 60 papers that focus on speech command recognition systems, the use of medium sized vocabularies, continuous speech, and speaker independent recognition, as well as speech recognition specifically in the context of air traffic control.

The purpose of including the last keyword phrase "Medium Vocabulary AND Continuous Speech Recognition AND Speaker Independent" is to capture articles about speech recognition techniques well suited for use in air traffic control (See Section IV for more details).

TABLE I
Search results summary. Keyword Phrase 1: "Speech Recognition in Air Traffic Control OR Voice Recognition in Air Traffic Control", Keyword Phrase 2: "Speech Command Recognition OR Voice Command Recognition", Keyword Phrase 3: "Medium Vocabulary AND Continuous Speech Recognition AND Speaker Independent"

| Search Engine<br>Keyword Phrase | ACM | IEEE | Google Scholar | Google |
|---|---|---|---|---|
| Keyword Phrase 1 | 2 | 4 | 12 | 9 |
| Keyword Phrase 2 | 1 | 7 | 10 | 9 |
| Keyword Phrase 3 | 4 | 6 | 13 | 10 |

The literature review provides background for identification of suitable speech recognition systems for air traffic control, as well as a discussion of remaining challenges and possible solutions for these types of applications.

## III. Automatic Speech Recognition (ASR)

Speech recognition is the process of converting a speech signal into a sequence of words. It also called Automatic Speech Recognition (ASR) or Speech-to-Text (STT). In recent years, the technology and performance of speech recognition systems have been improving steadily. This has resulted in their successful use in many application areas such as in-car systems or environment in which users are busy with their hands (e.g., "voice user interfaces") [34], hospital-based healthcare applications (e.g., systems for dictation into patient records, speech-based interactive voice response systems, systems to control medical equipment and language interpretation systems) [15], home automation (e.g., voice command recognition systems) [1], speech-to-text processing (e.g., word processors or emails), and personal assistants on mobile phones (e.g., Apple's Siri on iOS, Microsoft's Cortana on Window Phone, Google Now on Android). Speech recognition has also been widely used in air traffic control for many applications such as air traffic controllers' work load measurement [10], speech interface for air traffic control terminals [20], automated analysis and transcription of ATC voice communications [9], replacing the "pseudo-pilot" in air traffic control simulation and training by "automated pilot" which can recognize and understand the controller's speech using speech recognition modules [45].

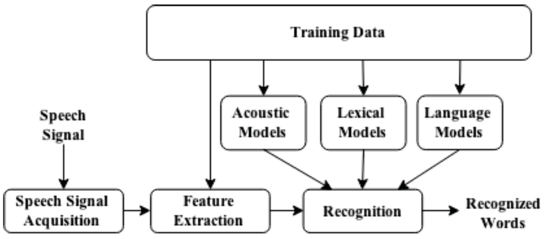

Fig. 1. General structure of speech recognition system

### A. Modules of Speech Recognition Systems

The general speech recognition approach can be described in two steps. 1) Given an acoustic observation, identify a feature vector sequence $X = X_1, X_2, ..., X_n$ using a feature extraction module. 2) Given this vector, find the corresponding word sequence $W = W_1, W_2, ..., W_n$ that has the maximum posterior probability $P(W \mid X)$ [35], expressed using Bayes theorem in (1).

$$W = \underset{w}{\operatorname{argmax}}\, P(W \mid X) = \underset{w}{\operatorname{argmax}} \frac{P(W)P(X \mid W)}{P(X)} \quad (1)$$

Fig. 1 shows the general structure of a speech recognition system. The system consist of six main modules: Speech Signal Acquisition, Feature Extraction, Acoustic Modeling, Language Modeling, Lexical Modeling, and Recognition.

1. Signal Acquisition: The signal acquisition module is responsible for obtaining the speech signal to be analyzed, for example by using microphones.

2. Feature Extraction: The feature extraction module is responsible for converting the speech signal into a feature vector. The performance of the ASR system depends heavily on this process. There are many feature extraction techniques such as Principal Component Analysis (PCA), Mel Frequency Cepstral Coefficients (MFCC), Independent Component Analysis (ICA), Linear Predictive Coding (LPC), Autocorrelation Mel Frequency Cepstral Coefficients (AMFCCs), Relative Autocorrelation Sequence (RAS), and Perceptual Linear Predictive Analysis (PLP). [23], [28], [57]. Studies have shown that Mel Frequency Cepstral Coefficients (MFCC) and Linear Predictive Coding (LPC) are techniques extensively used in speech recognition [52].

3. Acoustic Models: The acoustic model plays a critical role in improving accuracy of the ASR system by linking the input features with the expected phonetics of the hypothesis sentence [28] [35]. In (1), P$(X \mid W)$ represents the acoustic model, which is is the probability of acoustic observation of X when the word W is uttered.

4. Language Models: The main task of a language model is detecting connections between the words in a sentences with the help of lexical models. ASR systems usually use an n−gram language model to provide context for distinguishing words and phrases that sound similar. The use of a language model not only makes speech recognition more accurate but also helps to reduce the search space for recognition [35]. In (1), $P(W)$ represents the language model, which is the probability of word W uttered.

5. Lexical Models: A lexical model is also known as a pronunciation dictionary. It is developed to provide pronunciations of words in a given language. The lexical model links the acoustic-level representation with the word sequence which is output by the speech recognizer [7].

6. Recognition: The recognition module takes input from the feature extraction module and then uses acoustic models, language models and lexical models to recognize which words were spoken.

### B. *Classification of Speech Recognition Systems*

Speech recognition systems can be classified by type of speech utterance, type of speaker model and type of vocabulary that the systems can recognize [52].

1. Types of Speech Utterance: In ASR, an utterance is the smallest unit of speech and it is the sound of a word or set of words. Types of utterance can be classified into four classes as follows:

- Isolated Words - according to Radha et al., “isolated word recognizers usually require each utterance to have quiet on both sides of the sample window. It doesn’t mean that it accepts single words, but does require a single utterance at a time” [52]. It is also known as “Isolated Utterance”. This type of speech recognizer is comparatively simple and easy to develop because word boundaries are obvious.
- Connected Words - connected word recognizers are quite similar to isolated word recognizers, but require smaller pauses between utterances. It also known as “connected utterances”.
- Continuous Speech - continuous speech recognizers require special techniques for determining utterance boundaries, and allow speakers to speak almost naturally [52]. Although this kind of system is very difficult to develop, it has been widely used in many applications because of its flexibility.
- Spontaneous Speech - spontaneous speech recognizers are capable of recognizing unrehearsed speech, words being run together, “ums” and “ahs”, and even slight stutters [52]. Because of the large linguistic variation of spontaneous speech, recognition is extremely difficult. However, it has been shown that acoustic and language models with very large training data sets are able to overcome the problem of variation to some degree. This has resulted in increased recognition rates in spontaneous speech recognition systems [22].

Speech recognition systems for isolated words and connected words are considered relatively easy to develop because word boundaries are easy to find and the pronunciation of a word tends not to affect others. In contrast, continuous speech and spontaneous speech is more difficult to handle for a number of reasons. Challenging aspects of this type of ASR includes word boundary detection, the problem of coarticulation, and varying speech rates.

2. Types of Speaker Models: Because of the uniqueness physical bodies and personalities among people, speakers usually have distinct voice characteristics. ASR speaker models can be divided into two classes depending on how they handle these differences; speaker dependent and speaker independent models. [52].

- Speaker Dependent Models - speaker dependent systems depends on knowledge of a specific speaker’s voice characteristics. This kind of system must usually be trained for a specific user before it can recognize the speech of the user. Although these systems are easy to develop and achieve high accuracy, they are not used widely because they are usually not as flexible as speaker adaptive or speaker independent systems.
- Speaker Independent Models - speaker independent systems does not require knowledge of specific speakers, and can recognize speech from practically any people speaking a given language. Apple’s Siri assistant is an example of a system using a speaker independent model. Compared with speaker dependent systems, these systems are more flexible, however they offer less accuracy and are more difficult to develop.

Speaker dependent systems are commonly used for speech-to-text software (e.g., word processors, emails and dictation applications), while speaker independent systems are more commonly found in telephone applications (e.g., call centers). There is a third type of speaker model called a speaker adaptive model. These systems are developed to adapt its operation to the characteristics of new speakers. Implementing speaker adaptive systems is more complex than speaker dependent systems, but easier than the use of speaker independent models.

3. Types of Vocabulary: Another distinguishing factor of ASR systems is the size of the vocabulary they are able to recognize. The size of vocabulary affects the complexity, performance and the accuracy of the system [52]. In the literature, these vocabularies are usually classified into five classes as follows:

- Small vocabulary - tens of words
- Medium vocabulary - hundreds of words
- Large vocabulary - thousands of words
- Very-large vocabulary - tens of thousands of words
- Unlimited vocabulary - the system is able to suggest recognized words based on the phonems even when the word is not found in the (very large) vocabulary.

Generally, the smaller the vocabulary the easier it is to implement the ASR system.

### C. *Performance of Speech Recognition Systems*

Accuracy and speed are the two most common metrics for measuring speech recognition system performance. Word Error Rate (WER) is usually used for measuring accuracy, whereas speed is usually rated with Real Time Factor (RTF) [52]. WER can be computed by using (2):

$$WER = \frac{S + D + I}{N} \tag{2}$$

Where S is the number of substitutions, D is the number of deletions, I is the number of insertions and N is the number of words in the reference.

If the input of duration I requires time P to process, RTF can be computed by using (3):

$$RTF = \frac{P}{I} \tag{3}$$

Other measures of performance include Concept Error Rate (CER), Single Word Error Rate (SWER) and Command Success Rate (CSR).

### D. History of Automatic Speech Recognition

The history of ASR started in 1952 with an isolated digit recognition system for a single speaker. It was built by Davis, Biddulph, and Balashek of Bell Laboratories [11]. Over the last 60 years, technology development has led to a dramatic improvement of speech recognition systems. Juang and Rabiner [39] describes the development during the first four decades:

- 1960's - speech recognition systems were able to recognize small vocabularies (10 - 100 words) of isolated words with the help of filter-bank analyses and simple time normalization methods.
- 1970's - by using simple template-based, pattern recognition methods, researchers were able to build connected words, speaker independent speech recognition systems which can recognize medium vocabularies (100 - 1000 words).
- 1980's - large vocabulary (1000 - unlimited number of words) further advances in speech recognition problem was addressed using Hidden Markow Models (HMM) and stochastic language models.
- 1990's - with the helps of stochastic language understanding, statistical learning of acoustic and language models, and finite state transducer framework (and the FSM Library), researchers were able to build large vocabulary systems for continuous speech recognition and understanding.

In beginning of the new millennium, speech recognition systems were expanded to recognize very large vocabularies [52] [51]. Spontaneous speech recognition has started to receive attention from many researchers. In addition, researchers have started to use multimodal speech recognition, in which visual face information, particularly lip information is utilized. Results from multimodal speech recognition research show that performance can be improved compared with using audio only [21].

Currently (2015), we are able to build unlimited vocabulary speech recognition systems which can solve a large number of tasks, including the multiple languages problem [36], [51], [52]. Although artificial neural networks has been explored since the 1980's, they have so far not been able to compete with the Gaussian Mixture Model/Hidden Markov Model (GMM-HMM) approaches, which continues to be the dominating approach [13]. Nowadays, the introduction of deep learning [14], [32] and hybrid approaches [29], [67], [68] has overcome most of these difficulties and significantly increased the recognition rate of ASR systems.

## IV. Air Traffic Control (ATC)

### A. Introduction to Air Traffic Control

According to the Oxford English Dictionary, Air Traffic Control (ATC) is "the ground-based personnel and equipment concerned with controlling and monitoring air traffic within a particular area" [60]. The main purpose of ATC systems is to prevent collisions, provide safety, organize aircraft operating in the system and expedite air traffic [18]. With the steady increase in air traffic, ATC has become more and more important. This increase has also resulted in more complex procedures, regulations and technical systems [54].Thus, air traffic control systems have to be continuously improved to meet the evolving demands in air traffic.

In ATC, air traffic controller (ATCO) have an incredibly large responsibility for maintaining the safe, orderly and expeditious conduct of air traffic. Given the important roles of air traffic control and air traffic controllers, there is an ongoing need to strengthen training and testing of the operators. Further, being able to simulate the working environment of controllers enables increased safety through the use of support systems that can assist controllers and improve procedures, and by analyzing controller-pilot communications. In the past few years, the advances in technology and performance of ASR systems has offered many promising ways to deal with these needs.

### B. Applications of ASR in ATC

Because voice communication plays a critical role in ATC, many researchers have been interested in using automatic speech recognition technology for various applications in ATC operations as well as for simulation environments [41].

1. Air Traffic Control Simulation and Training: Air traffic control simulation provides facilities for testing and evaluation of new systems and concepts, and training of traffic controller students to handle realistic scenarios. Current air traffic control simulation typically requires "pseudo-pilots" who will act as real pilots in the simulation of controller-pilot communications with air traffic controller students. The use of "pseudo-pilots" make air traffic control simulators less flexible and comes at a relatively high cost.

By introducing speech technologies in ATC simulation and training the "pseudo pilots" can be replaced with so-called "automated pilots". The "automated pilot" will understand and process air traffic controllers' speech using a speech recognition module and generate responses that is sent back to the controllers using a speech synthesis module. The use of "automated pilot" instead of "pseudo-pilot" can dramatically reduce the cost of ATC systems and make the systems more flexible [61].

International Science Index, Computer and Information Engineering Vol:9, No:8, 2015 waset.org/Publication/10002245

2. Air Traffic Controllers Workload Measurement and Balancing: In ATC systems, air traffic controller workload is the key factor that limit the capacity of the whole system. With the increase in air traffic, measuring and balancing air traffic controller workload becomes important.

However, measuring controller workload is currently not an easy task because workload is difficult to measure directly. It is a costly process that requires manual observation and analysis of spoken communication. With the help of ASR systems, detecting spoken control events that the controller has to perform becomes easier, thus facilitating more direct measurements of controller workload. The detected events can be used for automated controller workload balancing [9], [10].

3. Controller-pilot Speech Analysis and Transcription: With the help of ASR systems in transcribing controller-pilot communications, it is possible to analyze large quantities of voice data for ATC research and analysis [41]. This analysis can be used for investigating and improving procedures and regulations, detecting air traffic controllers' events for workload measurement and balancing of controller workloads.

4. Backup Controller: An ASR system combined with other information sources in the ATC context (e.g., radar information, minimum safe altitudes, restricted zones, and weather information) could be used as input for a system called a "backup controller" to catch potentially dangerous situations that might be missed by the controller. It can also provide suggestions and safety information to the controllers in real time [41], [65].

### C. Criteria for Selecting ASR Systems for ATC

Applying automatic speech recognition in the ATC domain comes with many challenges and opportunities because of the unique characteristics of communication between controllers and pilots, such as small vocabulary sizes, high accuracy requirements, close to real time demands, and standardized formats for communication [41]. Based on these characteristics, studies has suggested that an ASR system that is suitable for ATC should be a speaker independent system which can recognize medium sized vocabularies and continuous speech [54], [65].

1. Speaker Dependence: Although Air Traffic Control Command Recognition (ATCCR) applications require only one controller at the same time, there are situations where multiple controllers are required in the operational environment.

Additionally, in the context of simulation and training, the system has to be able to recognize many air traffic controller students without the requirement to retrain or reconfigure the system. Thus, speaker independent systems are best suited for these applications, despite the reduced recognition accuracy of such systems [37], [65].

2. Continuous Speech Recognition: Although isolated words and connected words recognition systems usually have higher accuracy than continuous speech recognition systems, they are not well suited in the context of ATC. This is because they require the controllers to pause between each word when giving commands. Isolated words and connected words recognition systems will therefore cause delay in pilot-controller communication. A continuous speech recognition system, which permits the controller to speak in a natural way without pauses [54], is the system of choice when applying ASR in ATC [37].

3. Vocabulary Size: In the ATC domain, vocabularies used in communication between controllers and pilots follows International Civil Aviation Organization (ICAO) Standard Phraseology. The entire vocabulary of words (excluding names of specific places and call signs) is only about a few hundred words [65] [37] [17]. Thus, a medium sized vocabulary speech recognition system is adequate in the context of air traffic control.

4. Performance: In ATC, it is not important that ASR systems can recognize every single word, but it is important that the conveyed concepts are correctly detected. For example, the ASR system is not required to recognize all of the words in the following sentence: "Good morning Lufthansa one zero one descend level one two three", however it has to be able to extract the concept "DLH101 DESCEND FL 123".

The Concept Error Rate (CER) metric is used to measure the systems ability to extract the concepts from speech [30]. The CER of an ASR system which can be applied in ATC should not exceed those of pilots or pseudo pilots, which is 0.73% [54]. In addition, the system should be able to recognize and understand the concepts in real time without causing delays in communication between controllers and pilots or pseudo pilots.

### D. State-of-the-art of ASR suitable for use in ATC

Based on the previously mentioned criteria for selecting ASR system for the ATC domain, the number of suitable systems is limited. In this section, we highlight progress made so far for ASR systems that match these criteria. Although some of the systems were not developed for ATC or the English language, the approaches and technologies of the systems are still applicable to the ATC domain. The research presented in this section are grouped into three: The Hidden Markov Model approach, hybrid approaches and other approaches.

1. The Hidden Markov Model (HMM): In ASR, HMM has been the dominant approach over the last two decades.

Although the method has it's own weaknesses, it is still popular because it can be trained automatically, it is simple and computationally feasible.

In 1994, Daniel Jurafsky et al. used HMM combined with a Viterbi decoder, a bigram language model and a phonetic likelihood estimator to develop the Berkeley Restaurant Project (BeRP), which is a medium-vocabulary, speaker-independent, spontaneous continuous speech recognition system which functions as a knowledge consultant [40]. The recognition error rate and understanding error rate were quite high at 32.1% and 34% respectively.

Three years later, Jones et al. developed a continuous speech recognition system using syllable-based HMMs [38]. The authors concluded that the introduction of syllable-level bigram probabilities, word- and syllable-level insertion penalties, and the investigation of different model topologies can improve the recognizer performance. Compared with 35% of the baseline accuracy for monophone recognition, the proposed system achieved over 60% recognition accuracy.

Recognition of non-English languages have also been investigated by many ASR researchers, including Arabic, Tamil, Estonian, Amharic and Malayalam.

An acoustic training system for building acoustic models for a medium vocabulary speaker independent continuous speech recognition system for the Arabic language was developed Nofal et al. [47]. Cross-word triphones HMMs were used for acoustic modeling, and the models were trained using maximum likelihood estimation. The best word error rate was 0.19%.

A continuous speech recognition system for the Tamil language using a monophone-based HMM was developed by Radha et al. in 2012 [53]. The system used Mel Frequency Cepstral Coefficients (MFCC) for feature extraction. The results were relatively good, with the system yielding 92% word recognition accuracy and 81% sentence accuracy.

Thangarajan et al. built a small vocabulary word based and a medium vocabulary triphone based continuous speech recognizers for the Tamil language using HMM based word and triphone acoustic models [62]. 92.06% and 70.08% accuracy were achieved with new speakers on test sentences for the word-model and triphone-model respectively.

Thangarajan et al. used syllable modeling for developing a continuous speech recognition system for the Tamil language [63]. A small vocabulary context independent word model and medium vocabulary context dependent phone model were developed. The models were trained using SphinxTrain, a HMM-based acoustic model trainer from Carnegie Mellon University (CMU) [58]. The Word Error Rate of the proposed system was 10.63%.

A limited-vocabulary Estonian continuous speech recognition system using HMM was proposed by Alumäe et al [2]. Clustered triphones with multiple Gaussian mixture components were used to model words. The recognizer yielded 82.9% accuracy with a medium-sized vocabulary. If the real-time requirement was discarded, the correctness increased to 90.6%.

Although HMM has been the dominant technique for acoustic modeling in speech recognition for over two decades, it has two main weaknesses: it discards information about time dependencies, which creates problems for recognizing speech with varying speeds, and is prone to overgeneralization. De Wachter et. al (2007) [12] attempted to overcome these problems by relying on straightforward template matching. The authors extended the Dynamic Time Warping (DTW) framework with a flexible subword unit mechanism and a class sensitive distance measure. This resulted in an error rate reduction of 17% compared to the HMM results.

Gebremedhin et al (2013) built a syllable based, medium vocabulary size, continuous Amharic speech recognition for weather forecast and business report applications based on HMM [27]. To do this, they introduced a new approach for reducing the number of acoustic models that are required to build a syllable based Amharic ASR by combining similarly pronounced syllables. Finite state transducers were also explored to specify the grammar rules. The recognition accuracy of 93.6% was achieved on a 4000 words test set.

Kurian and Balakriahnan developed a continuous speech recognition system for the Malayalam language using PLP (Perceptual Linear Predictive) Cepstral Coefficient [42]. The developed system was evaluated with different number of states of HMM, Gaussian mixtures, and tied states. The word recognition accuracy and sentence recognition accuracy were 89% and 83% respectively.

Edward C. Lin implemented a 1000-word vocabulary, speaker independent, continuous live-mode speech recognizer in a single FPGA (A field-programmable gate array) [44]. A 4-state HMM is used to represent triphones in the implemented system. Although the implementation is extraordinarily small, it can still achieve almost the same accuracy as the state-of-the-art software recognizer at 10.9% Word Error Rate.

In order to address the problem of automatic speech recognition in the presence of interfering noise, Gales et al. developed a robust continuous speech recognition system using parallel model combination [24]. The model used in the system is a standard HMM with Gaussian output probability distributions.

Novotnỳ et al. developed a speech command recognition system using hidden Markov models of context dependent phones (triphones) and mel-frequency cepstral coefficients analysis of speech (MFCC) [48].

Although HMM-based ASR systems have not achieved the required accuracy in the ATC domain (0.73% CER), the steady improvement in term of accuracy and performance makes HMM-based ASR systems potential candidates for use in ATC. Approaches facilitated by the characteristics of the ATC domain can be applied to improve the accuracy of the systems in order to achieve the required results.

2. Hybrid Approaches: Although HMM is the dominant method for speech recognition over the last two decades, it still has it’s weaknesses. Many research initiatives have been conducted to overcome those weaknesses, for instance by proposing hybrid approaches. Combining HMM and Artificial Neural Networks (ANN) is a new research area that has received focus from many researchers. A survey of hybrid ANN/HMM models for automatic speech recognition was conducted by Edmondo Trentin et al. [64].

Hussien Seid et al. developed an Amharic speaker independent continuous speech recognizer based on an HMM/ANN hybrid approach [56]. With the help of the CSLU Toolkit [33], the model was constructed at a sub-word level using context dependent phonemes. This resulted in the achievement of 74.28% word and 39.70% sentence recognition.

Shantanu Chakrabartty et al (2000) proposed a hybrid Support Vector Machine (SVM), Hidden Markov Model approach for continuous speech recognition [6]. The architecture of the proposed system is based on the MAP (maximum a posteriori) framework [25].

Wroniszewska et al developed a voice command recognition system based on the combination of genetic algorithms (GAs) and K-nearest neighbor classifier (KNN). 94.2% recognition rate was achieved [67].

The ability to overcome the existing weaknesses of HMM and the improvement in terms of accuracy and performance with hybrid speech recognition systems makes this a good candidate for applications in environments like ATC.

3. Other Approaches: Although it has been proven that Support Vector Machines (SVM) have problems which make them difficult to apply to speech recognition, Padrell-Sendra et al. proposed a pure SVM-based continuous speech recognizer, using the SVM to make decisions at frame level, and a Token Passing algorithm to obtain the chain of recognized words [49]. The proposed system achieved a better recognition rate than traditional HMM-based systems (96.96% vs 96.47%).

Pellom et al. proposed fast likelihood computation techniques in nearest-neighbor based search for continuous speech recognition systems [50]. The authors concluded that the combination of the two techniques with partial distance elimination (PDE) reduced the computational complexity for likelihood computation by 29.8% over straightforward likelihood computation.

Leung et al. proposed a neural fuzzy network and genetic algorithm approach for Cantonese speech command recognition [43].

Beritelli et al. (2006) proposed a noise robust, low-complexity algorithm for voice command recognition using Vector Quantization-Weighted Hit Rate (VQWHR) and Dynamic Time Warping (DTW). The authors concluded that the proposed algorithm was robust to various types of background noise [4].

Although HMM and hybrid approaches have been used very widely for speech recognition, they are still facing challenges like computational complexity and background noise. Approaches such as Support Vector Machines or combinations of Vector Quantization-Weighted Hit Rate (VQWHR) and Dynamic Time Warping can deal with those challenges to some degree, and is still being explored by many researchers.

The following discussion is based on the state-of-the-art of ASR presented in section III and the specifics of the ATC context presented in section IV.

## V. Discussion

The discussion is divided in two. First, we identify challenges of applying automatic speech recognition (ASR) in the ATC domain, and second we suggests possible approaches which can be used to address the challenges and improve the recognition rate of ASR systems in general.

### A. *Challenges of ASR in ATC*

There are five major challenges to overcome in order to successfully apply ASR in ATC. While some challenges are unique to the ATC domain, such as call sign detection and the use of non-standard phraseology, others are general challenges of ASR systems such as poor input signal quality, the problem of ambiguity, and the use of dialects, accents and multiple languages. The latter challenges becomes even more pronounced when ASR is introduced to a high-risk domain such as ATC.

1. Call Sign Detection: Because of the variety of ways to refer to the same flight call sign and the use of airline aliases (e.g., “Speedbird” for British Airways Plc (United Kingdom), “Norstar” for Norwegian Long Haul (Norway), “Pacific” for Jetstar Pacific Airlines (Vietnam)), call sign detection is an extremely challenging task of ASR in ATC domain. It especially increases the CER (Concept Error Rate), but also affects WER (Word Error Rate) because of the requirement to identify all airline aliases, and then train the system for these alternative names.

2. Poor Input Signal Quality: The input signal quality can be affected by both technological and human factors. While technological problems such as background noise in cockpits and communication via radio links physically reduce the quality of input signal, human related problems such as spontaneous speed, high speed and/or slurred speech increase ambiguity in the ASR process. Both factors lead to increased misrecognition rates. The above-mentioned technological problems can to a certain degree be resolved by using noise canceling microphones and high quality radio links. However, solving human related problems would be very challenging because it is not likely that we can force controllers and pilots to significantly change the way they speak in order to adapt to ASR systems.

3. The Problem of Ambiguity: In the ATC domain, the problem of ambiguity (e.g., the number two-four-five can refer to a speed, heading or flight level) and references to confusable entities such as call signs or flight levels is one of the main factors which contribute to the reduction in the speech recognition rate of ASR systems [41], especially with regards to CER.

4. The Use of Non-Standard Phraseology: The use of non-standard phraseology leads to errors in controller-pilot radio messages. Studies have shown that about 80% of all pilot radio messages contain at least one error [26]. In addition, only a small number (less than 30%) of the examined utterances fully conform to the ICAO recommended phraseology [31]. This starting point adds to the difficulty of introduction of ASR in ATC.

5. Dialects, Accents and Multiple Languages: Because Air Traffic Control services are global services, ASR systems must be able to recognize foreign accents, different dialects and commands with a combination of multiple languages. For example, German controllers may say “Guten morgen Lufthansa one two three descend level one two zero”, where “Guten morgen” is good morning in German.

### B. *Approaches that can be used to improve the accuracy of ASR in ATC*

Although the ATC context poses many challenges to ASR systems, it also offers many distinct opportunities such as the use of context knowledge, the structured format of controller-pilot communications, and small vocabulary sizes. The use of post-processing approaches to reduce the uncertainties and ambiguities which resulted from the speech recognition process in order to improve recognition accuracy is a very well-known approach in ASR. There are three main post-processing approaches that are well suited for the ATC domain; syntactic analysis, semantic analysis and pragmatic analysis.

1. Syntactic Analysis: Syntactic analysis is the process of representing the language domain of the speech recognition system by a grammar, and then parsing inputs to eliminate invalid words or sentences [46]. Finite State Networks, Augmented Transition Networks (ATNs) and heuristics are the three methods that can be used to implement syntax in ASR [37].

In the ATC domain, syntactic analysis can be performed with the help of grammar files, which is made easier because of the structured format of controller-pilot communications and the predefined vocabularies. These grammar files define structure of sentences used in the operation. By using these grammar files, improved recognition can be achieved by focusing on the words likely to be spoken next in a sentence.

The ASR system use the grammar files to compile lexical trees which will be used recognize a statement by parsing the tree. For example, the simplest form of an ATC command consists of a call sign (e.g., SpeedBird, Norstar) followed by a goal action (e.g., descent, heading, fly direct) and a goal value (e.g., FL 90, 260 (degrees)) [17]. After a call sign is detected, the speech recognition system should expect to find a goal action. Thus, words which are not goal actions (e.g.,"Ahs", 'Ums") can be eliminated through the syntactic analysis.

The ability to eliminate invalid words and sentences of syntax analysis offers great potential to address the poor quality input signal challenge. In addition, syntax analysis can be used to deal with the problem of ambiguity.

With the help of the list of known ATC vocabularies, syntax analysis is able to correct misrecognized words, for example due to the problem of ambiguity, by replacing them with valid words with similar pronunciation.

2. Semantic Analysis: Semantic analysis is the process of testing the meaningfulness of sentences recognized by a speech recognition system. The method has been used to improve speech recognition performance by many researchers [8], [16], [69]. In the ATC domain, semantic analysis can be performed with the help of grammar files. Semantic knowledge is static, so it can be obtained and implemented into the syntax.

One possible method of using context knowledge is N-best list. The speech recognizer first analyzes the input signal and transforms it into a N-best list, and the list is then reduced by eliminating word sequences that parse syntactically, but are not actually meaningful [30] [37].

Because semantic analysis have the ability to eliminate words and sentences which are not meaningful even when they are parsed syntactically, it can be used to assist syntax analysis in dealing with the problem of ambiguity, poor input signal quality, and even the use of non-standard phraseology.

3. Pragmatic Analysis: Pragmatic analysis is the process of predicting likely future words based on the previously recognized words and the state of the system [37] [59].

A few methods exist that can be used to perform pragmatic analysis in the ATC domain with the help of context knowledge. One example is work by Schaefer, who developed a context-sensitive speech recognition system for air traffic control simulation using a cognitive model of the ATC controller. The model can continuously observe the present situation and generate a prediction of sentences the controller is most likely to say next [54].

Further, by using a so-called "Dialog Model" combined with context knowledge, the ASR system is able to predict the form and content of the next utterance from the previously recognized utterances [66]. The dialog models allow the system to consider only a subset of the application's full grammar and vocabulary, so both performance and accuracy of the ASR system can be improved.

In addition, radar information and flight plans could be used to reduce the list of likely aircraft call signs that a controller may refer to in a sector to only those in the sector or about to enter the sector [41]. With the ability to reduce the list of likely aircraft call signs, pragmatic analysis can be used to mitigate the challenge in call sign detection.

Finally, knowledge based rules, Finite State Networks, and knowledge state databases can also be used to implement pragmatic analysis in the ATC domain.

4. Other Approaches: Although, the three suggested post-processing approaches cannot address all the ATC challenges completely, they offer great potential to improve the recognition rate of ASR systems in the ATC domain.

Issues related to the use of dialects, accents, and multiple languages remain difficult to address. One possible way forward is to use detector modules for identifying which dialects, accents and languages which are spoken. This approach has been demonstrated by Fernandez et al. [19], who devised an ATC speech understanding system which can understand both English and Spanish. They achieved this by using a language detection module, which is capable of detecting the languages spoken by air traffic controllers. Detecting dialects and accents for tuning of a speech recognition system has been investigated by other researchers (see for example [3] and [5]).

## VI. Conclusion

In this paper, we have presented a thorough review of the Automatic Speech Recognition literature, including a look at the research history, and a presentation of the state-of-the-art of ASR approaches.

Further, we have presented possible applications of ASR in air traffic control, and identified central criteria for ASR approaches applicable to the ATC domain.

Following a detailed review of current ASR research approaches, we identified existing challenges applications of ASR in ATC, and discussed possible solutions to these challenges.

Because of the operation critical nature of systems in the ATC domain, there are still challenges that remain before ASR systems can be applied fully both in training, testing and ATC operations. However, as we have pointed out in this paper, research is steadily providing better results, both in terms of accuracy and speed.

Combining state-of-the art ASR approaches with contextual information to include syntactic, semantic and pragmatic analysis in the recognition process, and the identification of dialects, accents and languages holds great promise for the application of automatic speech recognition in the air traffic control domain.

## Acknowledgment

The authors would like to thank The Institute for Energy Technology (John E. Simensen and Christian Raspotnig), Edda Systems AS, and WP3 of "User driven innovation in smart homes for health and energy" at Østfold University College for support in the work with this paper.

## References

[1] H. AlShu'eili, G. Sen Gupta, and S. Mukhopadhyay. Voice recognition based wireless home automation system. In *Mechatronics (ICOM), 2011 4th International Conference On*, pages 1–6, May 2011.

[2] Tanel Alumäe and Leo Võhandu. Limited-vocabulary estonian continuous speech recognition system using hidden markov models. *Informatica*, 15(3):303–314, 2004.

[3] Hamid Behravan. *Dialect and accent recognition*. PhD thesis, 2012.

[4] Francesco Beritelli and Salvatore Serrano. A robust low-complexity algorithm for voice command recognition in adverse acoustic environments. In *2006 8th International Conference on Signal Processing*, volume 3. IEEE, 2006.

[5] Fadi Biadsy. *Automatic dialect and accent recognition and its application to speech recognition*. PhD thesis, Columbia University, 2011.

[6] Shantanu Chakrabartty, Guneet Singh, and Gert Cauwenberghs. Hybrid support vector machine/hidden markov model approach for continuous speech recognition. In *Circuits and Systems, 2000. Proceedings of the 43rd IEEE Midwest Symposium on*, volume 2, pages 828–831. IEEE, 2000.

[7] Rahul Chitturi, Venkatesh Keri, Gopalakrishna Anumanchipalli, and Sachin Joshi. Lexical modeling for non native speech recognition using neural networks. In *Proceedings of the International Conference on Natural Language Processing (ICON–2005)*, page 79. Allied Publishers, 2005.

[8] Noah B. Coccaro. *Latent Semantic Analysis As a Tool to Improve Automatic Speech Recognition Performance*. PhD thesis, Boulder, CO, USA, 2005. AAI3190360.

[9] José Manuel Cordero, Manuel Dorado, and José Miguel de Pablo. Automated speech recognition in atc environment. In *Proceedings of the 2nd International Conference on Application and Theory of Automation in Command and Control Systems*, pages 46–53. IRIT Press, 2012.

[10] José Manuel Cordero, Natalia Rodríguez, José Miguel, and Manuel Dorado. Automated speech recognition in controller communications applied to workload measurement. *Third SESAR Innovation Days*, 2013.

[11] KH Davis, R Biddulph, and Stephen Balashek. Automatic recognition of spoken digits. *The Journal of the Acoustical Society of America*, 24(6):637–642, 1952.

[12] M. De Wachter, M. Matton, K. Demuynck, P. Wambacq, R. Cools, and D. Van Compernolle. Template-based continuous speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(4):1377–1390, May 2007.

[13] Li Deng, Khaled Hassanein, and M Elmasry. Analysis of the correlation structure for a neural predictive model with application to speech recognition. *Neural Networks*, 7(2):331–339, 1994.

[14] Li Deng, Geoffrey Hinton, and Brian Kingsbury. New types of deep neural network learning for speech recognition and related applications: An overview. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8599–8603. IEEE, 2013.

[15] Scott Durling and Jo Lumsden. Speech recognition use in healthcare applications. In *Proceedings of the 6th international conference on advances in mobile computing and multimedia*, pages 473–478. ACM, 2008.

[16] Hakan Erdogan, Ruhi Sarikaya, Stanley F Chen, Yuqing Gao, and Michael Picheny. Using semantic analysis to improve speech recognition performance. *Computer Speech & Language*, 19(3):321–343, 2005.

[17] Eurocontrol. All clear? the path to clear communication. icao standard phraseology a quick reference guide for commercial air transport pilots. http://www.skybrary.aero/bookshelf/books/115.pdf, 2011.

[18] AJV-0 VP Mission Support Federal Aviation Administration. Air traffic control - chapter 2. general control, faa 7110.65 2-1-1. Technical report, February 19, 2014.

[19] F Fernández, J Ferreiros, JM Pardo, V Sama, R de Córdoba, J Marias-Guarasa, JM Montero, R San Segundo, LF d'Haro, M Santamaría, et al. Automatic understanding of atc speech. *Aerospace and Electronic Systems Magazine, IEEE*, 21(10):12–17, 2006.

[20] J. Ferreiros, J.M. Pardo, R. de Crdoba, J. Macias-Guarasa, J.M. Montero, F. Fernndez, V. Sama, L.F. d'Haro, and G. Gonzlez. A speech interface for air traffic control terminals. *Aerospace Science and Technology*, 21(1):7 – 15, 2012.

[21] Sadaoki Furui. 50 years of progress in speech and speaker recognition. *SPECOM 2005, Patras*, pages 1–9, 2005.

[22] Sadaoki Furui, Masanobu Nakamura, Tomohisa Ichiba, and Koji Iwano. Why is the recognition of spontaneous speech so hard? In *Text, Speech and Dialogue*, pages 9–22. Springer, 2005.

[23] Santosh K Gaikwad, Bharti W Gawali, and Pravin Yannawar. A review on speech recognition technique. *International Journal of Computer Applications*, 10(3):16–24, 2010.

[24] M.J.F. Gales and S.J. Young. Robust continuous speech recognition using parallel model combination. *Speech and Audio Processing, IEEE Transactions on*, 4(5):352–359, Sep 1996.

[25] J. Gauvain and Chin-Hui Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *Speech and Audio Processing, IEEE Transactions on*, 2(2):291–298, Apr 1994.

[26] Claudiu-Mihai Geacăr. Reducing pilot/atc communication errors using voice recognition. In *Proceedings of ICAS*, volume 2010, 2010.

[27] Yitagessu Birhanu Gebremedhin, Frank Duckhorn, Rüdiger Hoffmann, and Ivan Kraljevski. A new approach to develop a syllable based, continuous amharic speech recognizer. In *EUROCON, 2013 IEEE*, pages 1684–1689. IEEE, 2013.

[28] Wiqas Ghai and Navdeep Singh. Literature review on automatic speech recognition. *International Journal of Computer Applications*, 41(8):42–50, 2012.

[29] A. Graves, N. Jaitly, and A.-R. Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 273–278, Dec 2013.

[30] Hartmut Helmke, Heiko Ehr, and Matthias Kleinert. Increased acceptance of controller assistance by automatic speech recognition. *Tenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2013)*, 2013.

[31] Horst Hering. Technical analysis of atc controller to pilot voice communication with regard to automatic speech recognition systems. *EEC note*, 1, 2001.

[32] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97, 2012.

[33] John-Paul Hosom. The cslu toolkit: A platform for research and development of spoken-language systems. *Center for Spoken Language Understanding (CSLU), OGI Campus, Oregon Health & Science University (OGI/OHSU), visitado em Janeiro de*, 2002.

[34] Zhang Hua and Wei Lieh Ng. Speech recognition interface design for in-vehicle system. In *Proceedings of the 2nd International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pages 29–33. ACM, 2010.

[35] Xuedong Huang, Alex Acero, Hsiao-Wuen Hon, and Raj Foreword By-Reddy. *Spoken language processing: A guide to theory, algorithm, and system development*. Prentice Hall PTR, 2001.

[36] Xuedong Huang, James Baker, and Raj Reddy. A historical perspective of speech recognition. *Commun. ACM*, 57(1):94–103, January 2014.

[37] Karlsson Joakim. The integration of automatic speech recognition into the air traffic control system. Technical report, Cambridge, Mass.: Flight Transportation Laboratory, Dept. of Aeronautics and Astronautics, Massachusetts Institute of Technology,[1990], 1990.

[38] Rhys James Jones, Simon Downey, and John S. Mason. Continuous speech recognition using syllables. In *In Proc. Eurospeech '97*, pages 1171–1174, 1997.

[39] Biing-Hwang Juang and Lawrence R Rabiner. Automatic speech recognition–a brief history of the technology development. *Georgia Institute of Technology. Atlanta Rutgers University and the University of California. Santa Barbara*, 1, 2005.

[40] Daniel Jurafsky, Chuck Wooters, Gary Tajchman, Jonathan Segal, Andreas Stolcke, Eric Foster, and Nelson Morgan. The berkeley restaurant project. In *ICSLP*, volume 94, pages 2139–2142, 1994.

[41] H.D. Kopald, A. Chanen, Shuo Chen, E.C. Smith, and R.M. Tarakan. Applying automatic speech recognition technology to air traffic management. In *Digital Avionics Systems Conference (DASC), 2013 IEEE/AIAA 32nd*, pages 6C3–1–6C3–15, Oct 2013.

[42] Cini Kurian and Kannan Balakriahnan. Continuous speech recognition system for malayalam language using plp cepstral coefficient. *Journal of Computing and Business Research*, 3(1), 2012.

[43] KF Leung, FH Frank Leung, HK Lam, and Peter Kwong-Shun Tam. Neural fuzzy network and genetic algorithm approach for cantonese speech command recognition. In *2003. FUZZ'03. The 12th IEEE International Conference on Fuzzy Systems*, volume 1, pages 208–213. IEEE, 2003.

[44] Edward C Lin, Kai Yu, Rob A Rutenbar, and Tsuhan Chen. A 1000-word vocabulary, speaker-independent, continuous live-mode speech recognizer implemented in a single fpga. In *Proceedings of the 2007 ACM/SIGDA 15th international symposium on Field programmable gate arrays*, pages 60–68. ACM, 2007.

[45] F Marque, SK Bennacef, F Neel, and S Trinh. Parole: a vocal dialogue system for air traffic control training. In *Applications of Speech Technology*, 1993.

[46] LG Miller and S Levinson. Syntactic analysis for large vocabulary speech recognition using a context-free covering grammar. In *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*, pages 271–274. IEEE, 1988.

[47] M. Nofal, E. Abdel-Raheem, H. El Henawy, and N.A. Kader. Acoustic training system for speaker independent continuous arabic speech recognition system. In *Proceedings of the Fourth IEEE International Symposium on Signal Processing and Information Technology, 2004.*, pages 200–203, Dec 2004.

[48] Jan Novotnỳ, Pavel Sovka, and Jan Uhlı̌r̆. Analysis and optimization of telephone speech command recognition system performance in noisy environment. *Radioengineering*, 13(1):1, 2004.

[49] JM Pardo, J Ferreiros, F Fernandez, Valentin Sama, R De Cordoba, Javier Macias-Guarasa, JM Montero, R San-Segundo, LF D'Haro, and Germán González. 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, 2011.

[50] B.L. Pellom, R. Sarikaya, and J.H.L. Hansen. Fast likelihood computation techniques in nearest-neighbor based search for continuous speech recognition. *Signal Processing Letters, IEEE*, 8(8):221–224, Aug 2001.

[51] Omprakash Prabhakar and Navneet Kumar Sahu. A survey on: Voice command recognition technique. *International Journal of Advanced Research in Computer Science And Software Engineering*, 3(5), 2013.

[52] V Radha and C Vimala. A review on speech recognition challenges and approaches. *doaj. org*, 2(1):1–7, 2012.

[53] V. Radha, C. Vimala, and M. Krishnaveni. Continuous speech recognition system for tamil language using monophone-based hidden markov model. In *Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology*, CCSEIT '12, pages 227–231, New York, NY, USA, 2012. ACM.

[54] D. Schaefer. Context-sensitive speech recognition in the air traffic control simulation. *EEC Technical/Scientific Report No. 2001-004*, 2001.

[55] ICAO Secretariat. Outlook for air transport to the year 2025. *Report No. Cir*, 313, 2007.

[56] Hussien Seid and Björn Gambäck. A speaker independent continuous speech recognizer for amharic. *INTERSPEECH 2005*, 2005.

[57] Benjamin J Shannon and Kuldip K Paliwal. Feature extraction from higher-lag autocorrelation coefficients for robust speech recognition. *Speech Communication*, 48(11):1458–1485, 2006.

[58] CMU Sphinx. Cmu sphinx: Open source toolkit for speech recognition. *Retrieved*, 8(13):2010, 2010.

[59] Georg Stemmer, Elmar Nöth, and Heinrich Niemann. The utility of semantic-pragmatic information and dialogue-state for speech recognition in spoken dialogue systems. In *Text, Speech and Dialogue*, pages 439–444. Springer, 2000.

[60] Stevenson. Oxford dictionary of english.

[61] Glenn Taylor, J Miller, and Jeff Maddox. Automating simulation-based air traffic control. In *Interservice/Industry Training, Simulation, and Education Conference*, volume 2193, 2005.

[62] R. Thangarajan, A. M. Natarajan, and M. Selvam. Word and triphone based approaches in continuous speech recognition for tamil language. *WSEAS Trans. Sig. Proc.*, 4(3):76–85, March 2008.

[63] R Thangarajan, AM Natarajan, and M Selvam. Syllable modeling in continuous speech recognition for tamil language. *International Journal of Speech Technology*, 12(1):47–57, 2009.

[64] Edmondo Trentin and Marco Gori. A survey of hybrid ann/hmm models for automatic speech recognition. *Neurocomputing*, 37(1):91–126, 2001.

[65] Thanassis Trikas. Automated speech recognition in air traffic control. Technical report, Cambridge, Mass.: Massachusetts Institute of Technology, Dept. of Aeronautics and Astronautics, Flight Transportation Laboratory, 1987, 1987.

[66] Karen Ward. A speech act model of air traffic control dialogue. 1992.

[67] MARTA WRONISZEWSKA and JACEK DZIEDZIC. Voice command recognition using hybrid genetic algorithm. *TASK QUARTERLY*, 14(4):377–396, 2010.

[68] Dong Yu and Li Deng. Deep neural network-hidden markov model hybrid systems. In *Automatic Speech Recognition*, pages 99–116. Springer, 2015.

[69] Bartosz Ziółko, Suresh Manandhar, Richard C Wilson, and Mariusz Ziółko. Semantic modelling for speech recognition. *Proceedings of Speech Analysis, Synthesis and Recognition. Applications in Systems for Homeland Security, Piechowice, Poland*, 2008.

# Appendix E

# ATC Phraseology

A
ABEAM
ABLE
ABOVE
ACCELERATION
ACCEPT
ACKNOWLEDGE
ACROSS
ACTION
ADDITIONAL
ADJACENT
ADJUST
ADS-B
ADS-C
ADS-CONTRACT
ADVISE
ADVISED
AERODROME
AFFIRM
AFTER
AGAIN
AGREED
AHEAD
AILERONS
AIR
AIRBORNE
AIRCRAFT
AIRSPACE
AIR-TAXI
AIR-TAXIING
ALERT
ALL
ALTERNATIVE
ALTIMETER
ALTITUDE
AND
ANOTHER
APPEAR
APPEARS
APPROACH
APPROACHING
APPROVAL
APPROVED
ARC
ARE
AREA
AROUND
ARRIVAL
ARRIVING
AS
AT
ATC
ATIS
ATTENTION
AVAILABLE
AVIATION
AVOID
BACK
BACKTRACK
BALLOON(S)
BASE
BASIC
BAY
BE
BEFORE
BELOW
BETWEEN
BLAST
BLOCK
BOTH
BOUND
BRAKES
BRAKING
BREAK
BY
CALL
CANCEL
CANCELLED
CAPABILITY
CASE
CATEGORY
CAUTION
CAVOK
CENTRE
CHANGE
CHARLIE
CHECK
CIRCLE
CIRCLING
CIRCUIT
CLEAN
CLEAR
CLEARANCE
CLEARED
CLEARS
CLIMB
CLIMBING
CLOSING
CLOUD
CODE
COEFFICIENT
COMING
COMMAND
COMMENCE
COMMENCING
COMPACTED
COMPLETED
CONDITION
CONDITIONS
CONFIRM
CONFLICT
CONSTRUCTION
CONTACT
CONTINUE
CONTROL
CONTROLLED
CONVENIENT
CORRECT
COURSE
COVERED
CPDLC
CROSS
CROSSED
CROSSING
CRUISE
CURRENT
DAMP
DECELERATION

DECISION
DEGREE
DEGREES
DELAY
DEPARTING
DEPARTURE
DESCEND
DESCENDING
DESCENT
DETAILED
DETAILS
DETERMINED
DEVIATING
DEWPOINT
DIRECT
DIRECTION
DISCONNECT
DISCONNECTING
DISCRETION
DISREGARD
DISTANCE
DME
DO
DOES
DOWN
DOWNWIND
DRY
DUE
EIGHT
ELEMENT
ELEVATION
EMERGENCY
ENTER
EQUIPMENT
ESTABLISH
ESTABLISHED
ESTIMATE
ESTIMATED
ESTIMATING
EXCEED
EXEMPTED
EXPECT
EXPECTED
EXPEDITE
EXPEDITING
EXTEND
EXTENDED
FAILURE
FAMILIAR
FAST
FEET
FIELD
FINAL
FIR
FIRST
FIX
FLASHING
FLIGHT
FLOODED
FLY
FOLLOW
FOLLOWED
FOR
FREE
FREQUENCY
FROM
FROZEN
FULL
FURTHER
GATE
GBAS
GEAR
GENERAL
GIVE
GIVING
GLIDE
GNSS
GO
GOING
GOOD
GREATER
GROUND
HALF
HAND
HANDOVER
HAVE
HEADING
HEIGHT
HELICOPTER
HIGH
HOLD
HOLDING
HOUR
I
ICE
ICING
IDENT
IDENTIFICATION
IDENTIFIED
IF
ILS
IMMEDIATE
IMMEDIATELY
IN
INBOUND
INCREASE
INDICATION
INFORMATION
INSTRUCTIONS
INTENTIONS
INTERCEPT
INTERCEPTION
INTERFERENCE
INTO
IS
ISSUE
JET
JOIN
KILOMETRES
KNOTS
LAND
LANDING
LATER
LEAST
LEAVE
LEAVING
LEFT
LESS
LEVEL(S)
LIGHTING
LIGHTS
LINE
LINING
LOCALIZER
LOCKED
LONG
LOOKING
LOSE
LOSS
LOST
LOW
MACH
MAGNETIC
MAINTAIN
MAKE
MAY
MAYDAY
MEDIUM
MESSAGE
METRES
MILES
MILLIMETRES
MINIMA
MINIMUM
MINUS
MINUTE
MINUTES
MLS
MODE
MONITOR
MONITORING
MOVING
NAVIGATION
NEGATIVE
NEXT
NO
NORMAL
NOSE
NOT
NOTICE
NOW
NUMBER
OBSERVED
OBSERVES
OBSTACLE
OBSTRUCTION
OCLOCK
OF
OFF
OFFSET
OMIT
ON
ONE
ONLY
OPERATIONS
OPPOSITE
OR
ORBIT

OUT
OUTBOUND
OVER
OVERTAKING
OWN
PARALLEL
PARKING
PASS
PASSING
PATCHES
PATH
PATTERN
PER
PILOT
PLAN
PLANNED
POINT
POOR
POSITION
POSSIBLE
PRECEDING
PRECISION
PREPARE
PRESENT
PRIMARY
PROCEDURE
PROCEED
PROCEEDING
PROGRESS
PUBLISHED
PUSHBACK
QFE
QNH
QUICKLY
RA
RADAR
RADIAL
RADIO
RAIM
RANGE
RATE
REACH
REACHING
READ
READY
RECEIVED
RECEIVER
RECLEARED
REDUCE
RE-ENTER
RELEASE
RELEASED
REMAIN
REMARKS
REMOVED
REPLY
REPORT
REPORTED
REPORTING
REPORTS
REQUEST
REQUESTS
REQUIRED
REQUIREMENT
RESET
RESETTING
REST
RESTRICTION(S)
RESUME
RESUMED
RETURN
RETURNING
REVERT
REVISED
REVISION
RIDGES
RIGHT
RNAV
RNP
ROCKING
ROGER
ROUTE
RUDDER
RUNWAY
RUTS
RVR
RVSM
S
SAME
SAY
SBAS
SECOND
SECONDARY
SECONDS
SENDING
SEPARATION
SERVICE
SET
SETTING
SEVEN-SEVEN-ZERO-ZERO
SHORT
SHORTLY
SHOULD
SHOW
SID
SIDES
SIGHT
SIGN
SIXTY
SKY
SLIGHTLY
SLIPSTREAM
SLOW
SLOWER
SLOWING
SLOWLY
SLUSH
SNOW
SNOWDRIFTS
SPEED
SQUAWK
SQUAWKING
SSR
STAND
STANDBY
STAR
START
STARTING
STATIONS
STILL
STOP
STOPPING
STRAIGHT
STRAIGHT-IN
SURFACE
SURVEILLANCE
TAKE
TAKE-OFF
TAKING
TAS
TAXI
TAXIWAY
TCAS
TEMPERATURE
TERMINAL
TERMINATED
TERMINATING
TERRAIN
THAT
THE
THIS
THREE
THRESHOLD
THROUGH
TIME
TO
TOO
TOUCH
TOUCHDOWN
TOW
TOWER
TRACK
TRAFFIC
TRANSMISSION
TRANSMISSIONS
TRANSMIT
TRANSMITTER
TRANSPONDER
TREATED
TRUE
TURBULENCE
TURN
TURNS
UHF
UNABLE
UNAVAILABLE
UNCHANGED
UNDERNEATH
UNIDENTIFIED
UNKNOWN
UNMANNED
UNRELIABLE
UNSERVICEABLE
UNTIL
UP
VACATE
VACATED
VACATING

VECTOR
VECTORING
VECTORS
VIA
VICINITY
VISIBILITY
VISUAL
VISUALLY
VMC
VOR
WAIT
WAKE
WANT
WARNING
WAS
WATER
WAY
WE
WEATHER
WELL
WET
WHEEL
WHEELS
WHEN
WHILE
WILL
WIND
WINGS
WITH
WORK
WRONG
YOU
YOUR
ZONE