MALORCA

Machine Learning of Speech Recognition Models for Controller Assistance

MALORCA 2nd Stakeholder Meeting

Components of Machine Learning for Assistant Based Speech Recognition

Petr Motlicek

Vienna, 20th to 21st of February 2018

SESAR JOINT UNDERTAKING

Founding Members

EUROPEAN UNION EUROCONTROL

Abstract

The presentation presents the different components, which are needed to train an Assistant Based Speech Recognition System by Machine Learning. This includes e.g. acoustic, language and command prediction models, but also radar data and tool-supported controller utterance transcription.

Content of the presentation

Following modules will be presented (with challenges)

- Automatic Speech Recognition (ASR) module:
  - Acoustic Model (accents)
  - Language model (different phraseology)
  - Lexicon
- Command Prediction Model (CPM) and Hypotheses Generator
- Context extractor end corrector
- Command filtering and plausibility checker

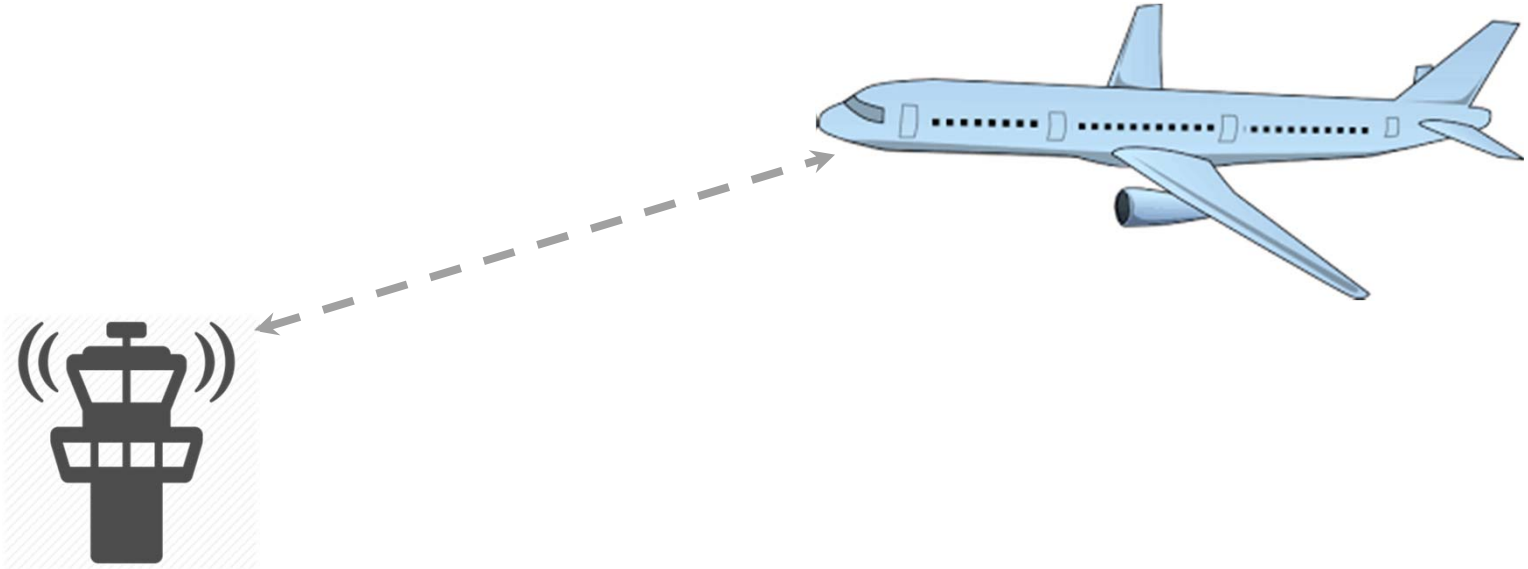
Automatic learning of model parameters

- Learning of AM/LM/CPM
- Audio and Radar data
- Context corrector

Objectives and challenges

Voice-text-command for Air Traffic Control

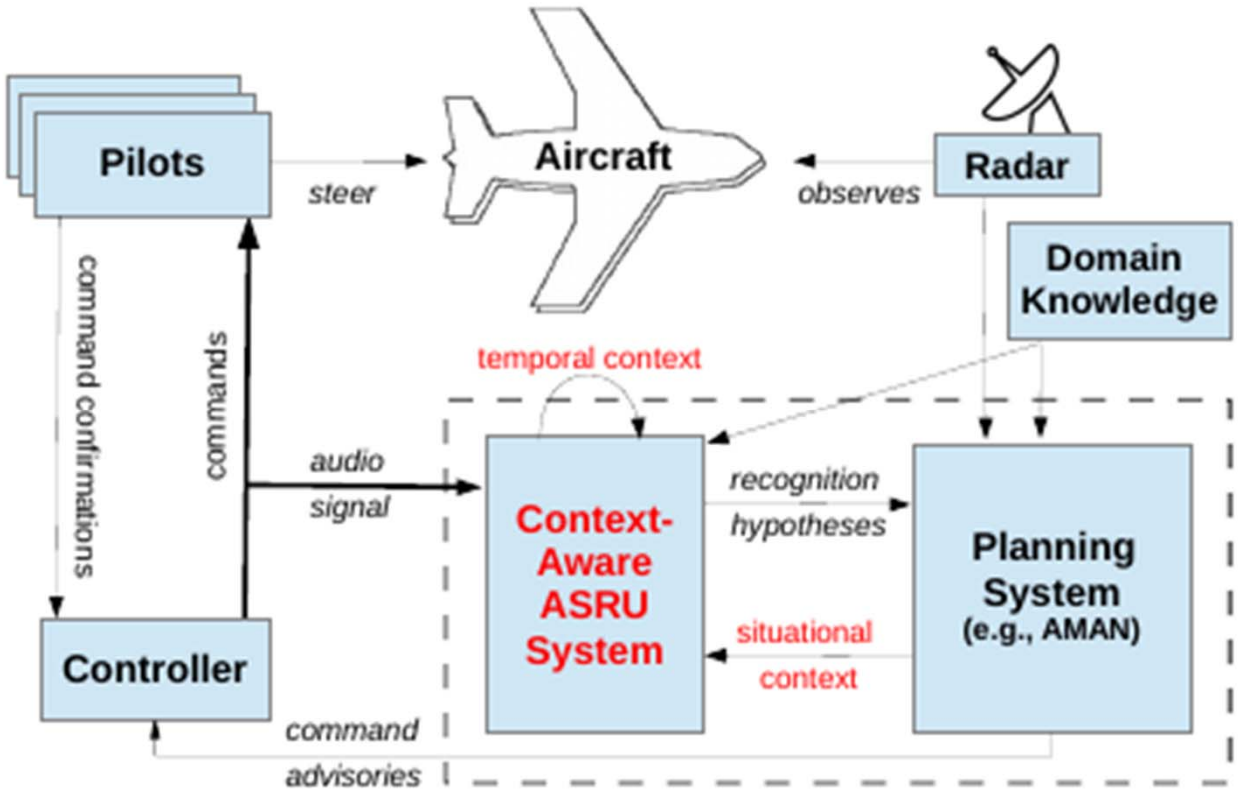
- Voice communication (1-channel) between controllers and pilots
- Limited vocabulary, constrained grammar, accented English, narrow-band speech
- Assistant Based Speech Recognition (ABSR) is applied



MALORCA: MAchine Learning Of speech Recognition models for Controller Assistance

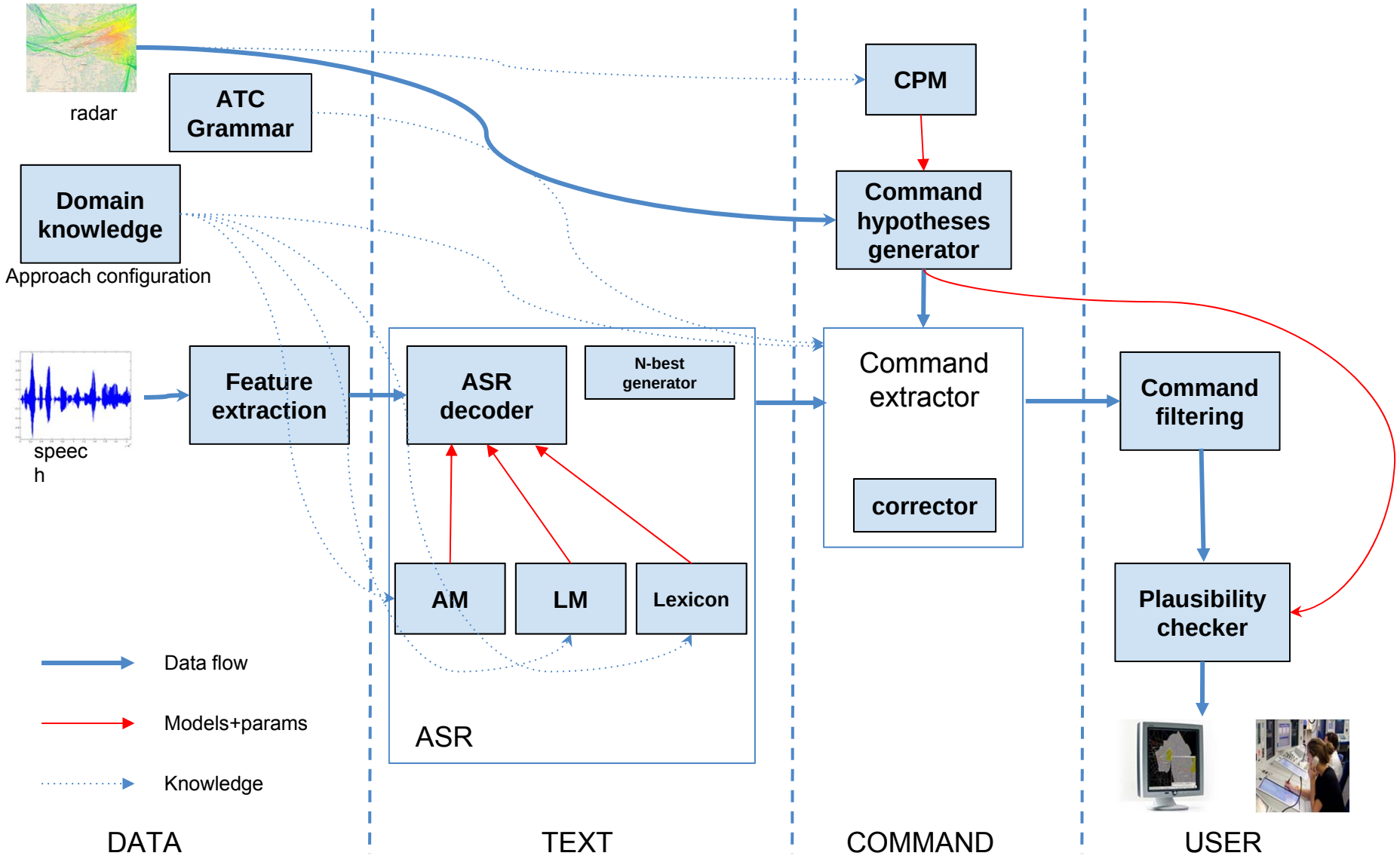
http://www.malorca-project.de/

Schematic view of MALORCA building components



ASRU - Automatic Speech Recognition and Understanding

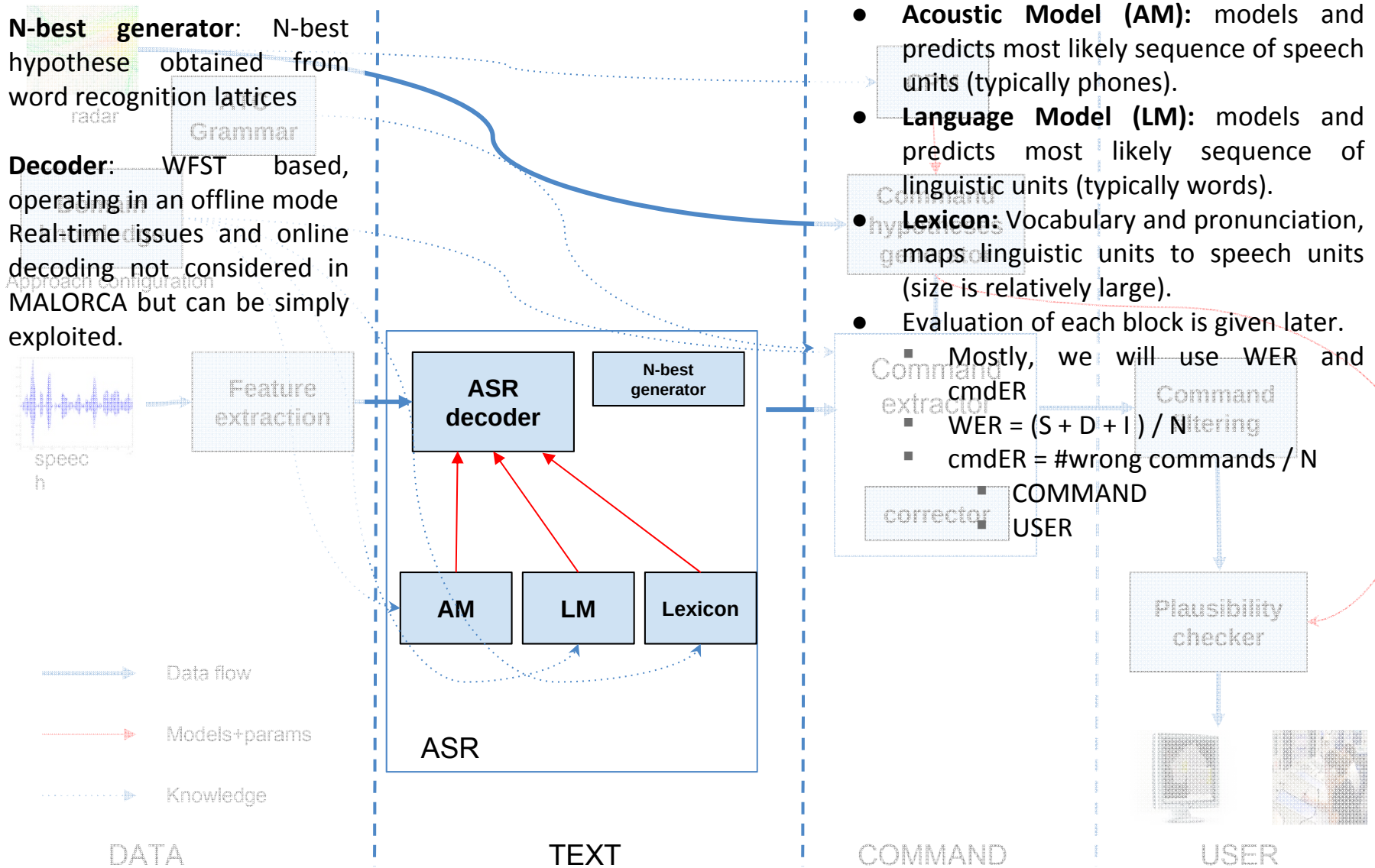
General Block Diagram (MALORCA system)



1. Automatic Speech Recognition (ASR)

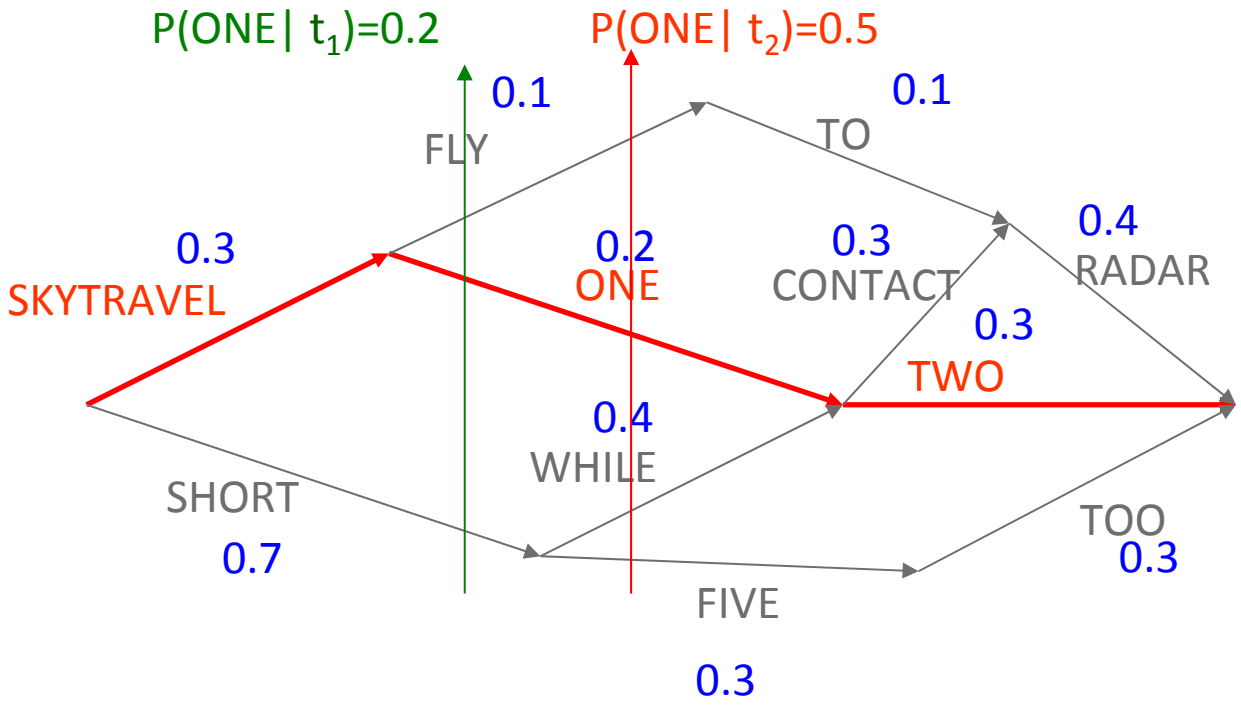
N-best generator: N-best hypothese obtained from word recognition lattices

Decoder: WFST based, operating in an offline mode Real-time issues and online decoding not considered in MALORCA but can be simply exploited.



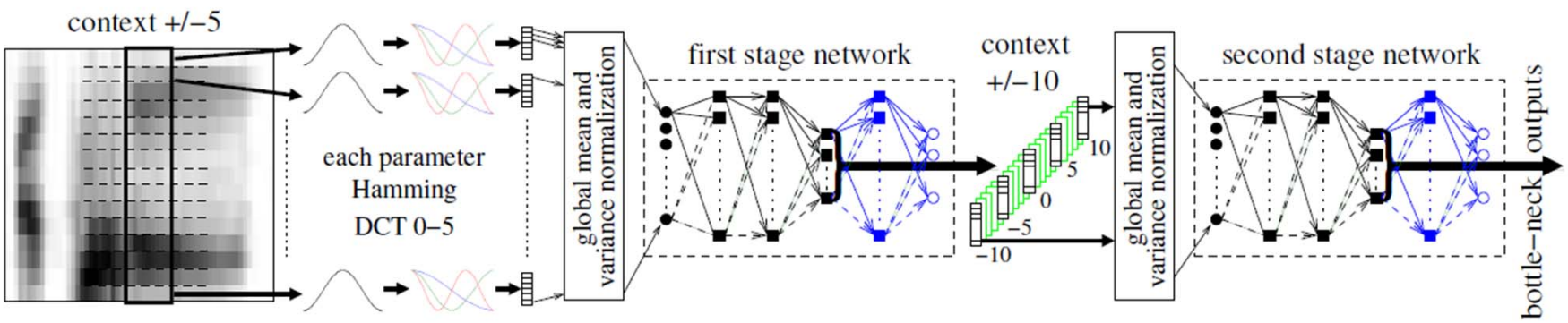
- **Acoustic Model (AM):** models and predicts most likely sequence of speech units (typically phones).
- **Language Model (LM):** models and predicts most likely sequence of linguistic units (typically words).
- **Lexicon:** Vocabulary and pronunciation, maps linguistic units to speech units (size is relatively large).
- Evaluation of each block is given later.
  - Mostly, we will use WER and cmdER
  - WER = (S + D + I) / N
  - cmdER = #wrong commands / N
    - COMMAND
    - USER

ASR decoding – acoustic N-best generation



ASR Components: Acoustic Modeling

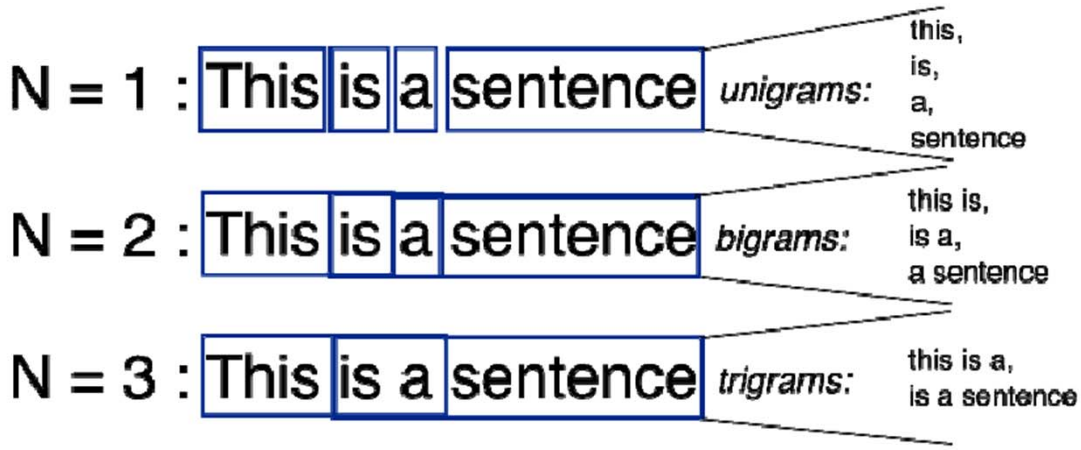
- Speech Signal --> Features --> Phones (context-dependent)
- Needs large amounts of training data (speech and transcripts) to learn.
- Baseline (easy to adapt): GMM+HMM.
- State-of-the-art (tricky adaptation): DNN + HMM.



ASR Components: Language Modeling

Language Models (LM):

- Disambiguate similar sounding words.
- Assigns probabilities to sequence of words (e.g., **"GO FOR A TEA BREAK, GOOD BYE"** is more likely than **"GO FOR A T-BRAKE, GOOD BUY"**
- Grammar language models: Rule-based LM (CFG):
  - a set of production rules that describe all possible strings in a given formal language. Production rules are simple replacements.
- Statistical language models (SLM): Learns "rules" from training data



ASR Components: Language modeling

Rule-based context-free grammar (deterministic way to model phraseology):

MALORCA ASR language modeling has evolved from CFG LM

- Limited vocabulary and standard phraseology - but deviations!
- Further experiments indicate SLM outperforms rule-based LM
- Performance in terms of cmdER (COMMAND, Prague):

  ~16% CGF

  ~8% SLM

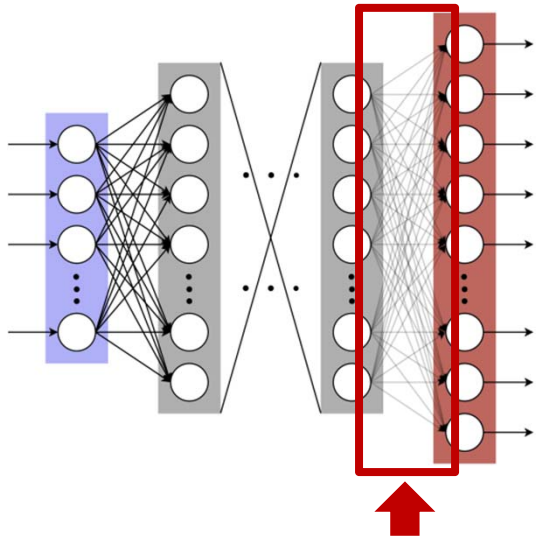
CFG is still applied for transducing text-to-commands.

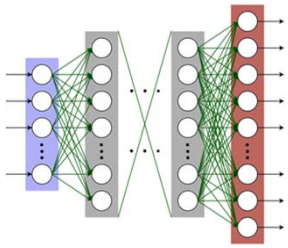
ASR Components (AM, LM adaptation)

Evaluation of AM, LM components:

- Especially w.r.t adapting to target domain

	WER
Base ASR system	13.3
+ AM adapted using in-domain data	10.0
+ LM adapted using in-domain data	8.5



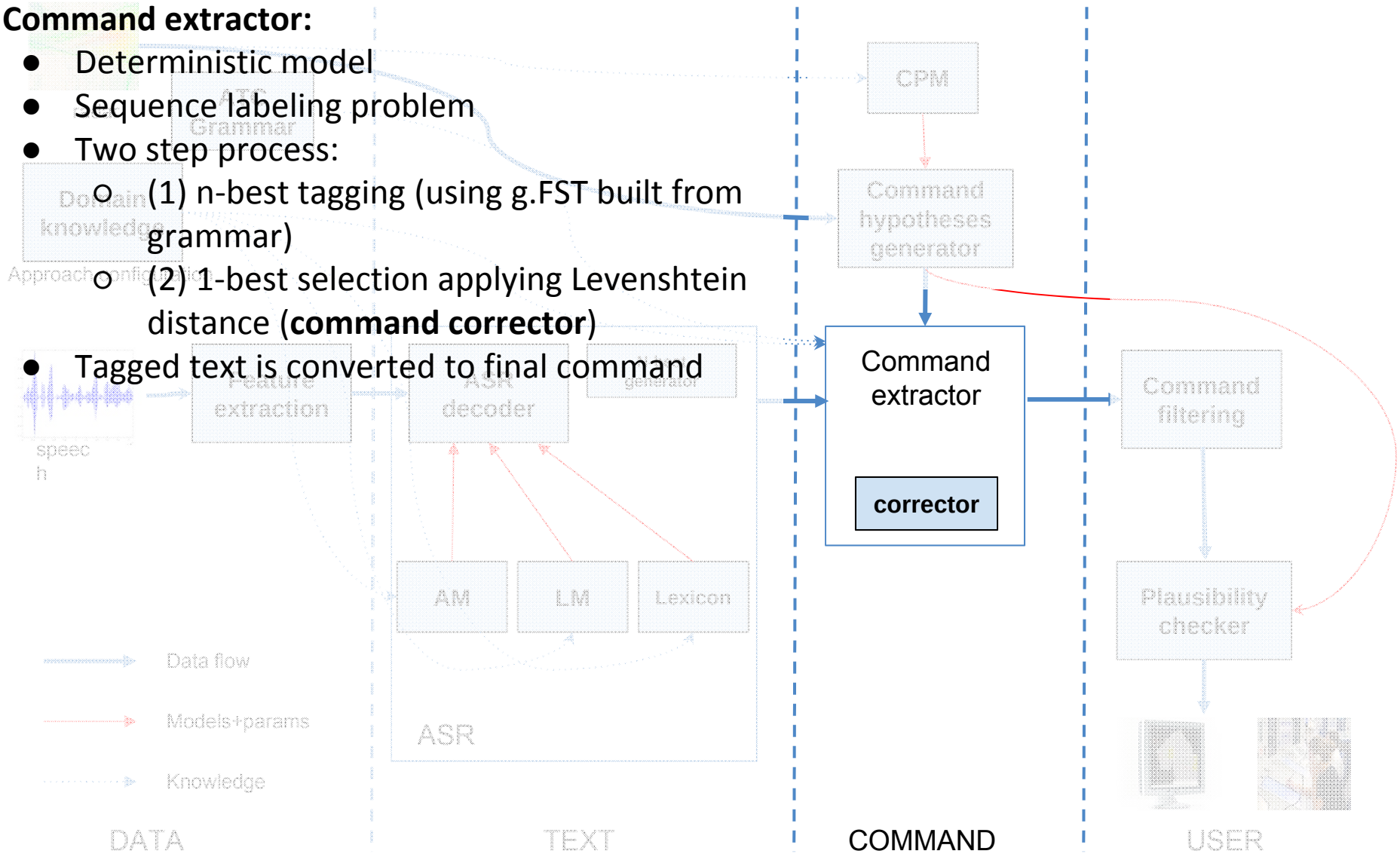


N-gram LM

2. Command Extractor and corrector

Command extractor:

- Deterministic model
- Sequence labeling problem
- Two step process:
  - (1) n-best tagging (using g.FST built from grammar)
  - (2) 1-best selection applying Levenshtein distance (**command corrector**)
- Tagged text is converted to final command



Command extractor and corrector

hello lufthansa eight echo kilo start reduce

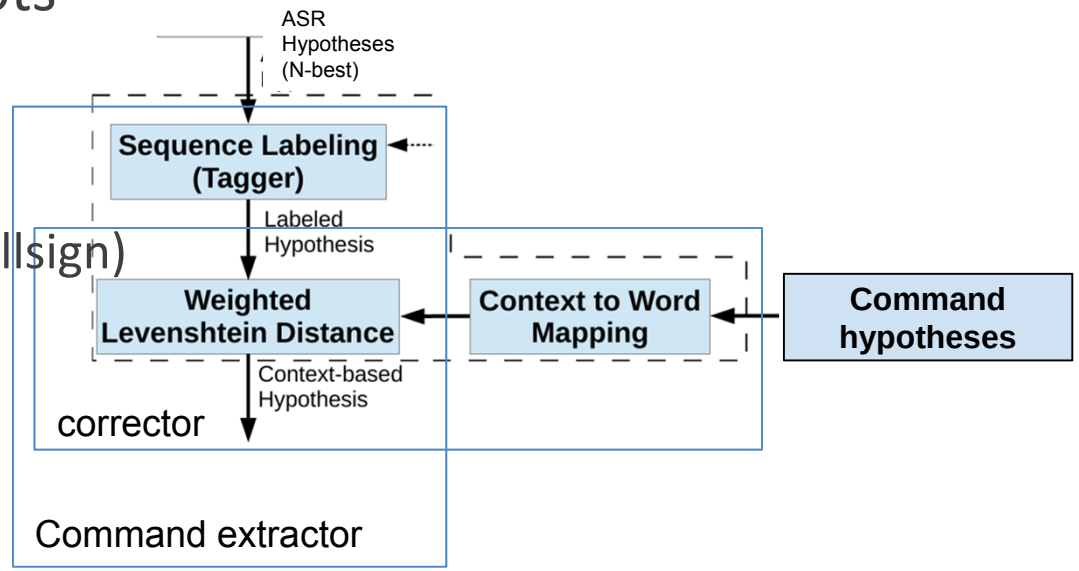
your speed to two two zero knots

Concepts:

DLH8EK (lufthansa eight echo kilo - callsign)

REDUCE (reduce - command word)

220 (two two zero - speed attribute)

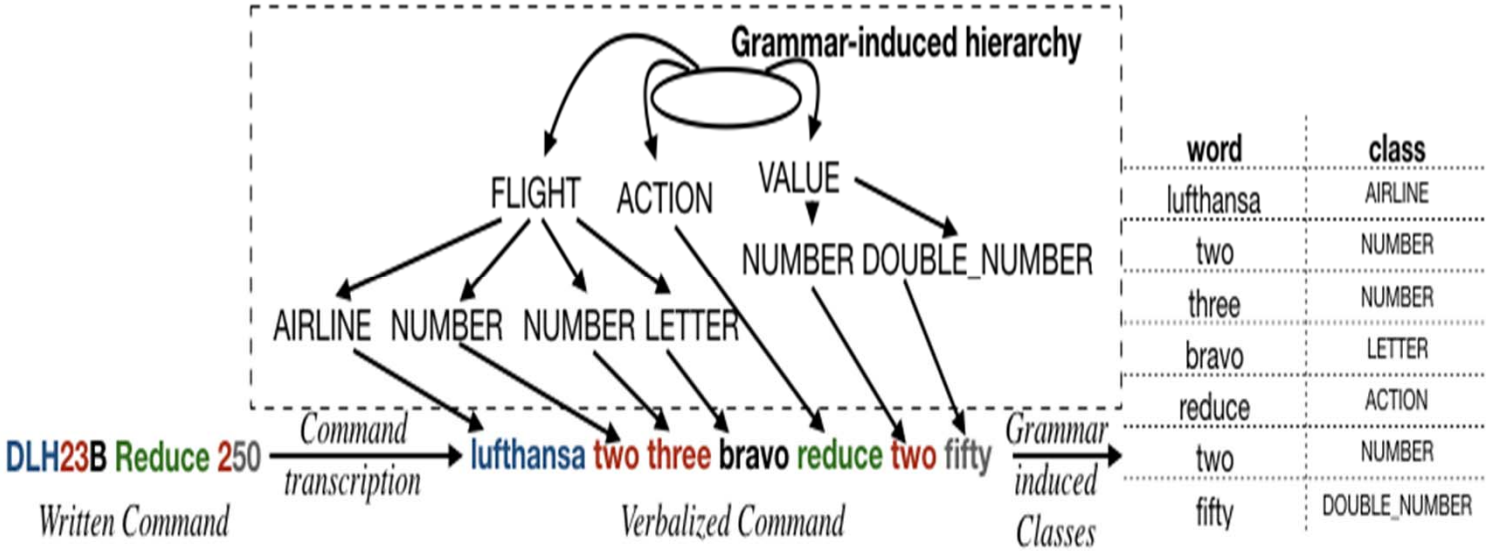


hello <callsign> <airline> lufthansa </airline> <flightnumber> eight echo kilo </flightnumber> </callsign> start <commands> <command="reduce"> reduce your speed to <speed> two two zero </speed> knots </command> </commands>

Command: DLH8EK REDUCE 220

Command corrector

- Block to correct the commands hypothesized by command extractor (N-best)
- Block of verbalisation applied on commands issued by command hypotheses generator
- Levenshtein distance applied to selected the final 1-best hypothesis



Command corrector - evaluation (SLM vs. CFG model)

	WER [%]	CmdER [%] (Command)	CmdER %] (User)
SLM (no context)	10.9	30.9	21.9
SLM+Rescoring	8.9	16.5	12.7
CFG (no context)	18.0	50.5	37.5
CFG_Rescoring	19.1	31.8	24.2

3. CPM and command hypotheses generator

CPM (command prediction model):

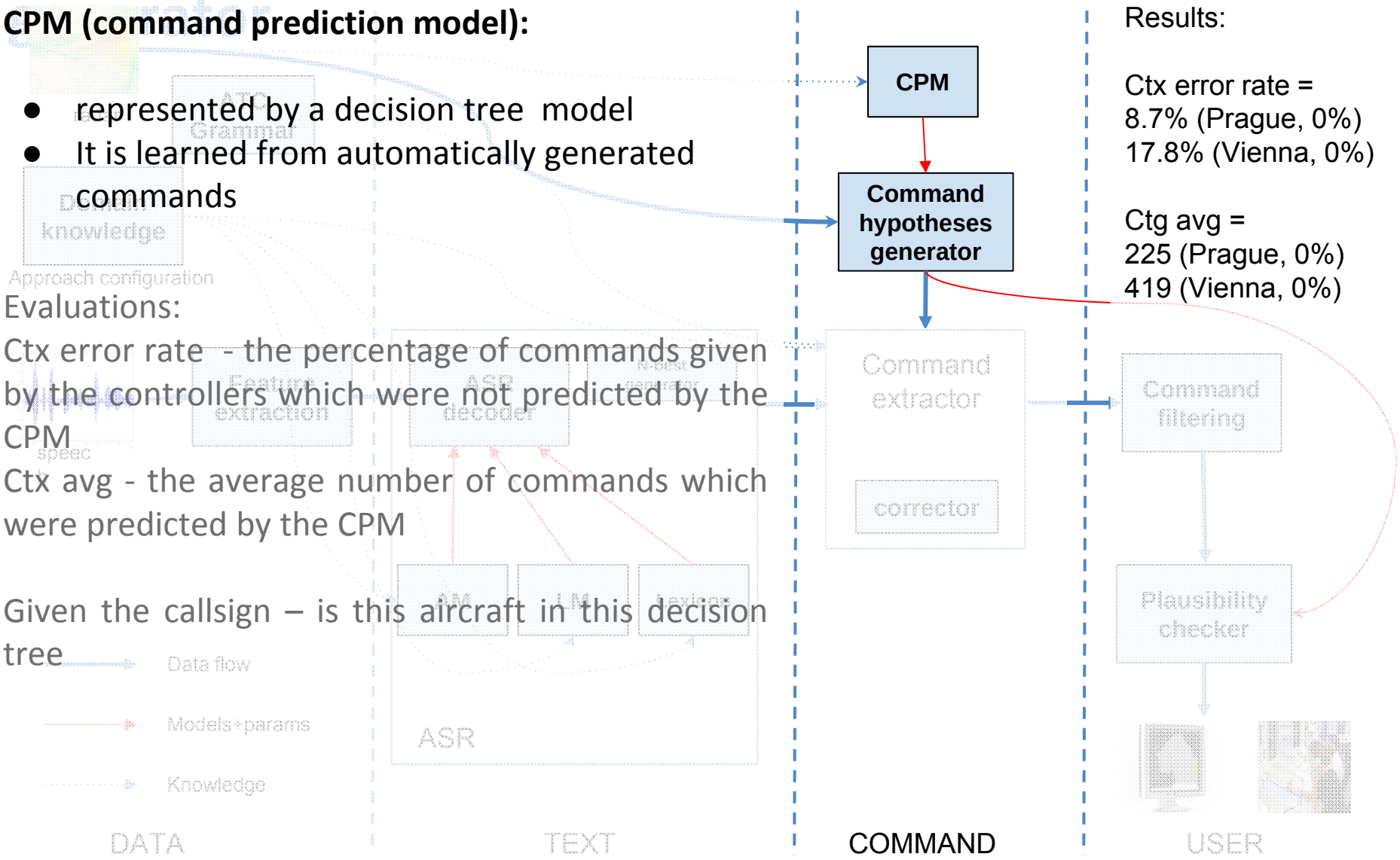
- represented by a decision tree model
- It is learned from automatically generated commands

Evaluations:

Ctx error rate - the percentage of commands given by the controllers which were not predicted by the CPM

Ctx avg - the average number of commands which were predicted by the CPM

Given the callsign – is this aircraft in this decision tree

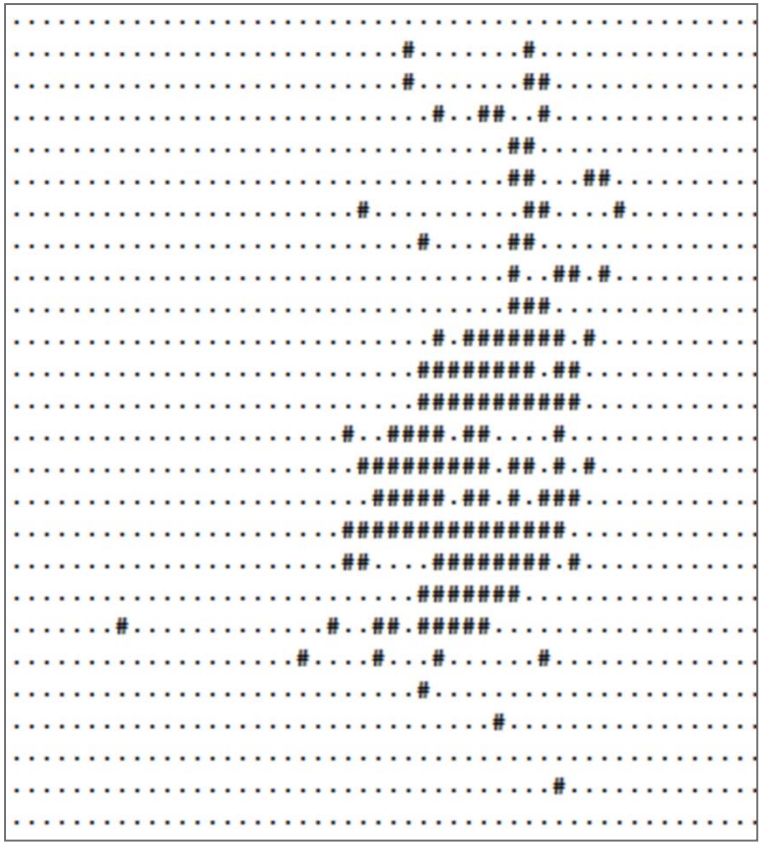


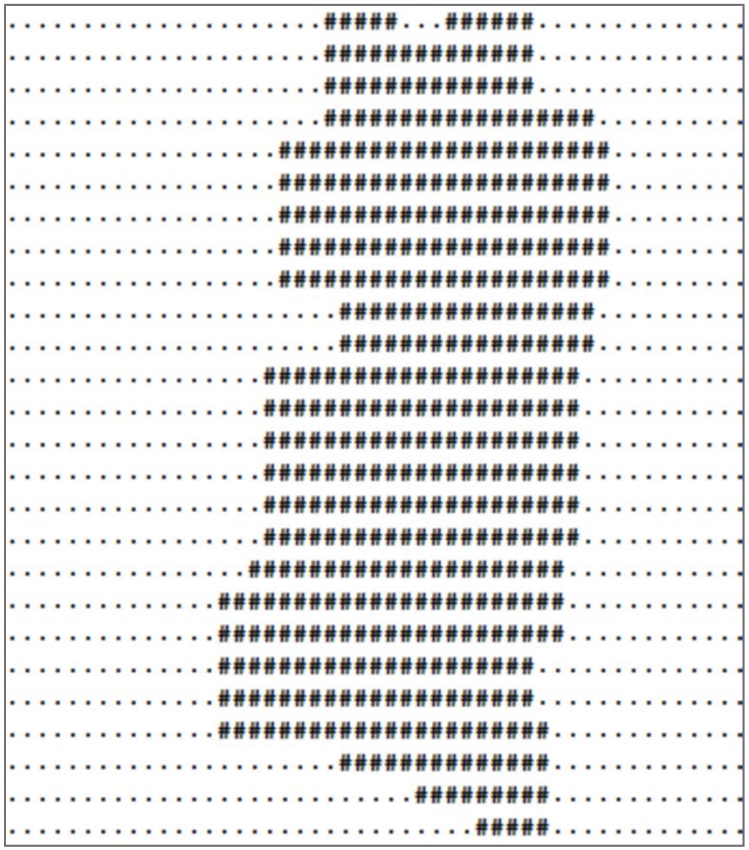
Results:

Ctx error rate =
8.7% (Prague, 0%)
17.8% (Vienna, 0%)

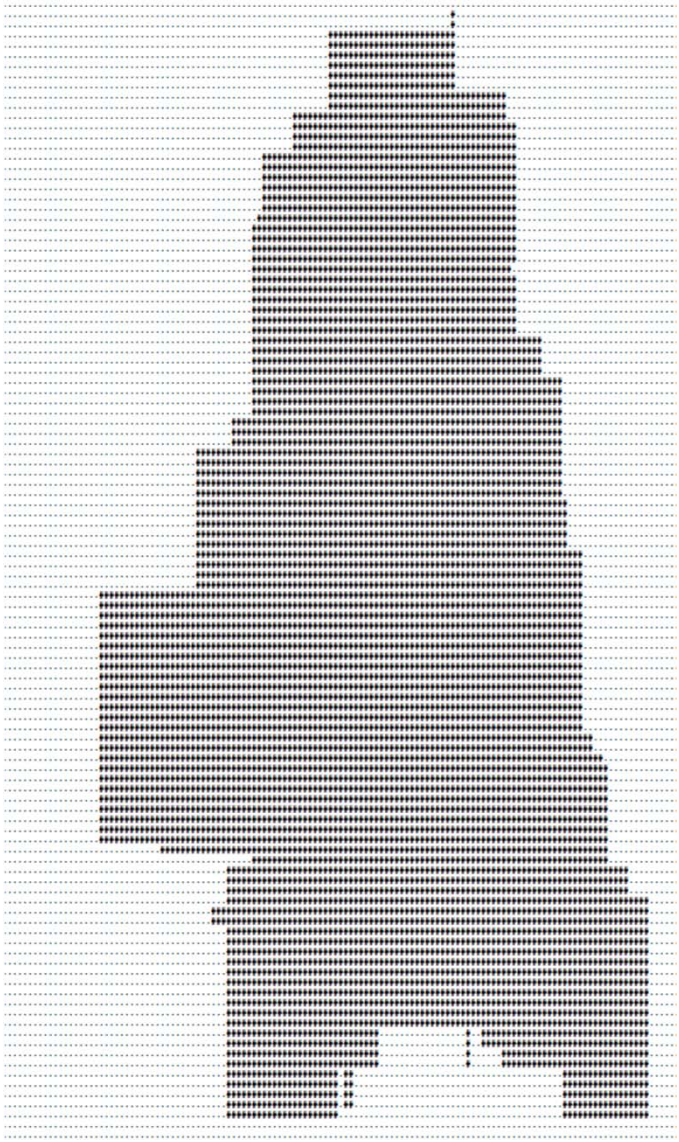
Ctg avg =
225 (Prague, 0%)
419 (Vienna, 0%)

CPM: Filtering gaps and finding outliers





CPM: Real examples



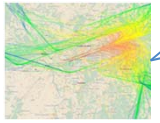
TURN_RIGHT
(arrivals)

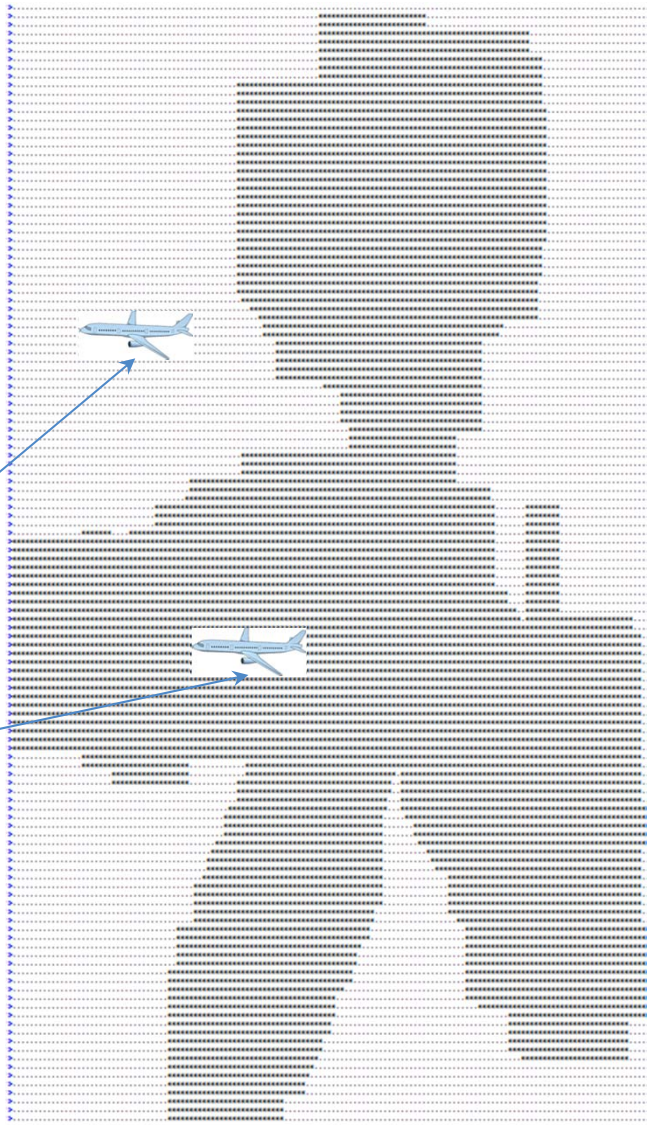
DIRECT_TO
(arrivals)

Command hypotheses generator

Command hypotheses generator:
Decision tree algorithm checks the radar information w.r.t CPM and generates list of plausible commands

- Example for “DIRECT_TO” (for flight arrivals)
- For teach wav-file (at the timestamp of the wavefile), both static and dynamic contexts are generated





Command hypotheses generator

Evaluations:

Given the radar information – we generate list of commands that are plausible (this we call context)

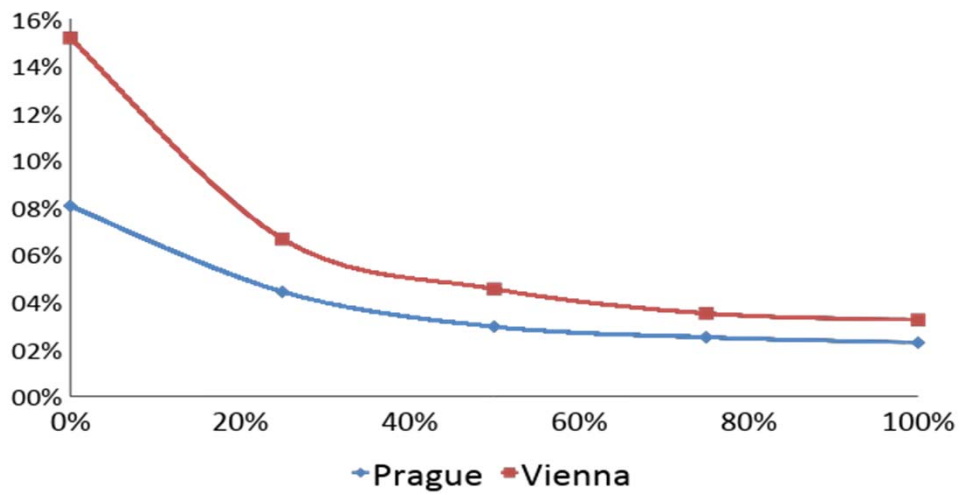
•Static context (snapshot of radar)

1470647283 STATIC RESET
    ADD AUI807
    ADD BER161K

Dynamic context (all possible commands)

1470647283 DYNAMIC RESET
    ADD AUI807 CLEARED_ILS 24 1
    ADD AUI807 CLEARED_ILS 30 1
    ADD AUI807 DESCEND 3000 Alt 1
    ADD AUI807 DESCEND 3500 Alt 1
    ADD AUI807 DESCEND 4000 Alt 1
    ADD AUI807 DESCEND 5000 Alt 1

Vienna:
Decrease from 15.2% to **3.2%**
Prague:
Decrease from 8.1% to **2.3%**



4. Command filter and plausibility checker

Both represent acceptance/rejection modules ⇒

Evaluation as an information retrieval task (similar to e.g. AIRBUS challenge)

- **Input** = command; **Output** = acceptance/rejection decision
- TP/FP = recognised command is correct/wrong (FALSE ALARM) and accepted
- TN/FN = recognised command is wrong/correct (MISS) and rejected

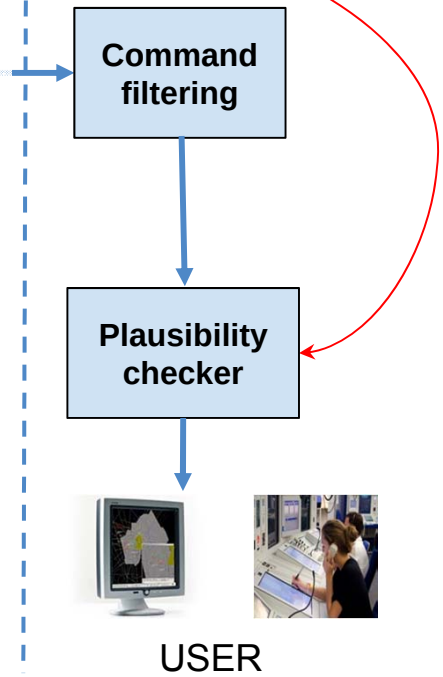
⇒ optimising Precision/Recall

Precision → seems to be more important (related to FP) in ATC domain

MALORCA evaluations: (0% untranscribed data, Prague)

Error-rates [%] if the command is presented to user (FP)

- 7.64% after command filtering (are falsely accepted)) (CmdER = 15.4%)
- 0.44% after plausibility checker (are falsely accepted) (CmdER = 22.3%)



Command filter (example)

Unrecognized commands are filtered by command filter

AUA790Z TURN_LEFT_HEADING 380

AUA806Q REDUCE 160 KT

NO_CALLSIGN NO_CONCEPT

EZY6936 CLIMB 220 FL

EZY75EY DIRECT_TO LEDVA

NLY930N CLIMB 240 FL

NO_CALLSIGN NO_CONCEPT

IBE31ST HANDOVER_FREQUENCY 133.2

AUA806Q TURN_RIGHT_HEADING 80

Command filter

AUA790Z TURN_LEFT_HEADING 380

AUA806Q REDUCE 160 KT

EZY6936 CLIMB 220 FL

EZY75EY DIRECT_TO LEDVA

NLY930N CLIMB 240 FL

IBE31ST HANDOVER_FREQUENCY 133.2

AUA806Q TURN_RIGHT_HEADING 80

Plausibility checker (example)

Commands with semantic or context error are rejected

AUA790Z TURN_LEFT_HEADING 380

AUA806Q REDUCE 160 KT

EZY6936 CLIMB 220 FL

EZY75EY DIRECT_TO LEDVA

NLY930N CLIMB 240 FL

IBE31ST HANDOVER_FREQUENCY 133.2

AUA806Q TURN_RIGHT_HEADING 80

Plausibility Checker

Command not meaningful (semantic error)l

AUA806Q REDUCE 160 KT

EZY6936 CLIMB 220 FL

EZY75EY DIRECT_TO LEDVA

Command not in context (context error)

IBE31ST HANDOVER_FREQUENCY 133.2

AUA806Q TURN_RIGHT_HEADING 80

Domain knowledge(approach configuration) ATC grammar

Domain knowledge: definition of

- Airlines
- waypoints
- Frequencies
- non-english words
- Runways

➔ ATC grammar is built around CFG:

- set of production rules that describe all possible strings in an ATC phraseology.
- Each command is specified using CFG (allowing all possible replacements)

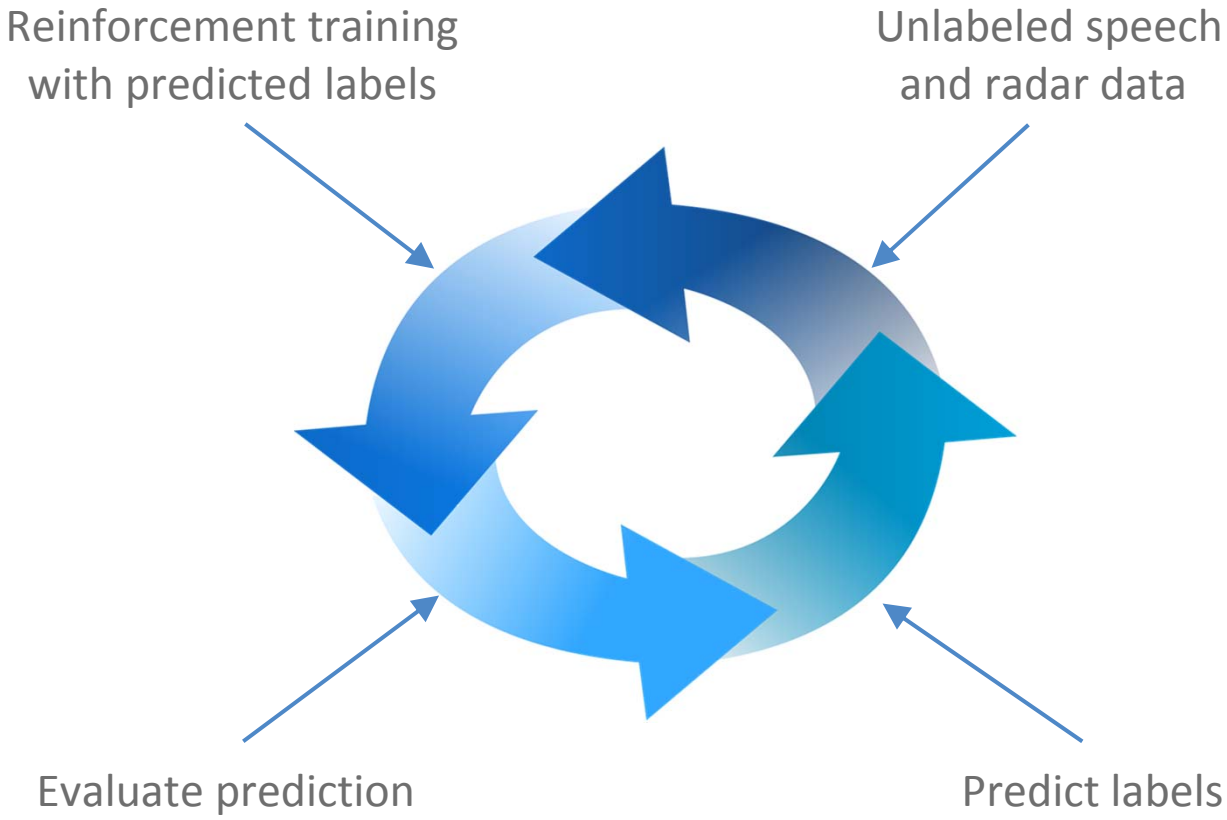
Final system results – without semi-supervised learning

	WER	CmdER	Error Rate	Rejection rate
COMMAND (pure) rates	13.3	24	7.6	6.3
USER rates	10	20	0.3	20

5. Automatic Learning

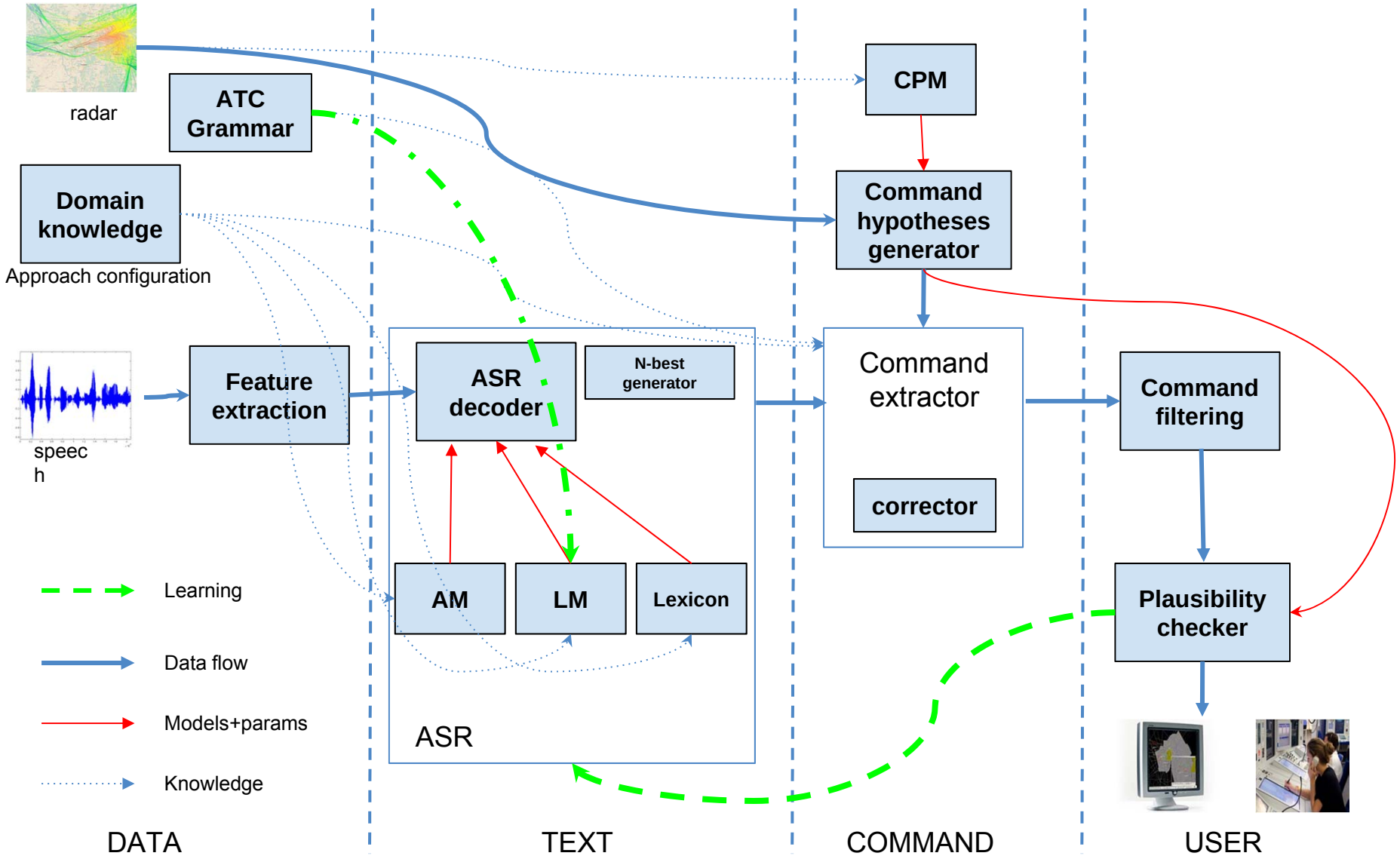
- Adapt ABSR systems to new ATC environments
  - Continuously in semi/un-supervised manner
  - Utilize increasing amounts of (untranscribed) data
  - Exploit local constraints (accents, acoustic conditions)
  - Combine data from two modalities

Automatic Learning Cycle



1) The initial (starting) system is trained with a minimal effort.
2) The automatic learning is done offline (e.g., at the end of each session or once a week).
3) The system is expected to improve over time and can be trained for different users.

MALORCA system - automatic learning



Semi-supervised learning (AM, LM, CPM)

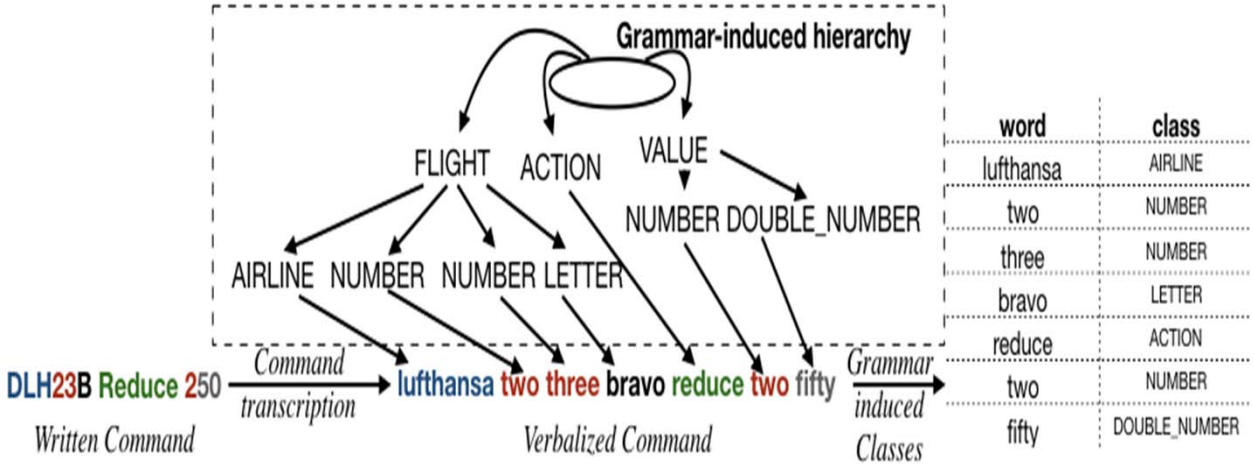
- Exploit untranscribed data (VDev2) to improve ASR
- Transcript generation using ASR-SA
  - Generate text and command hypotheses for VDev2
- Data selection
  - Select “good” or “bad” data ?
  - Assign confidence scores to ASR outputs
- Semi-supervised training

Automatic learning of LM by sampling from a grammar

Sampling from CFG (grammar) – building adapted LM

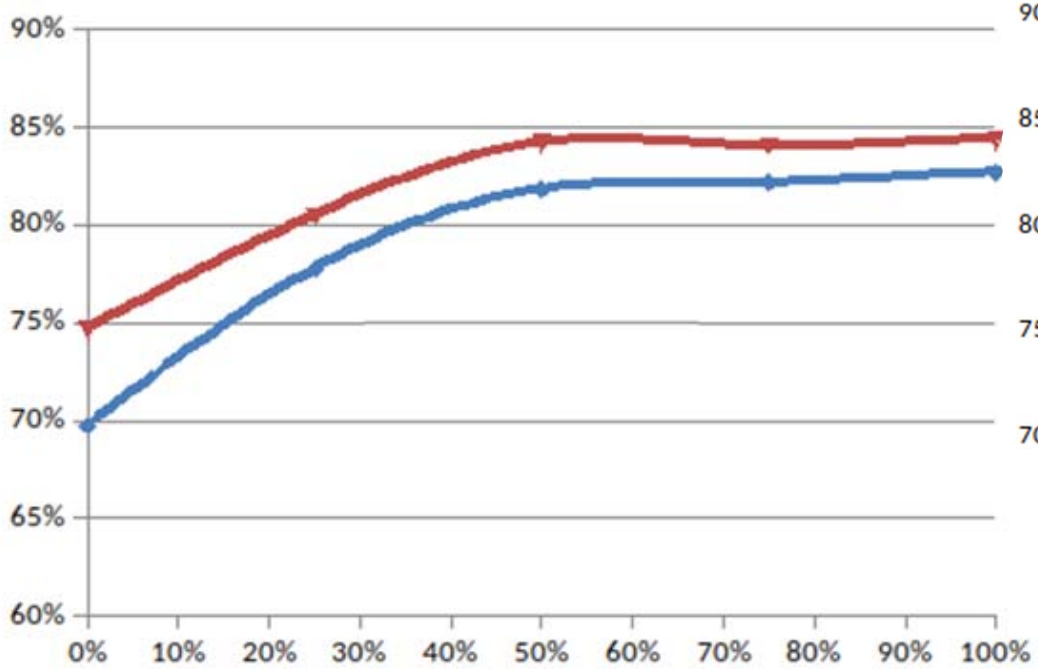
- Commands first generated from grammar
- Then verbalisation is applied
- New n-gram is built with these samples + all transcribed data

(CFG used to generate grammatical sentences and then combined with operational data (transcriptions) to build final SLM)

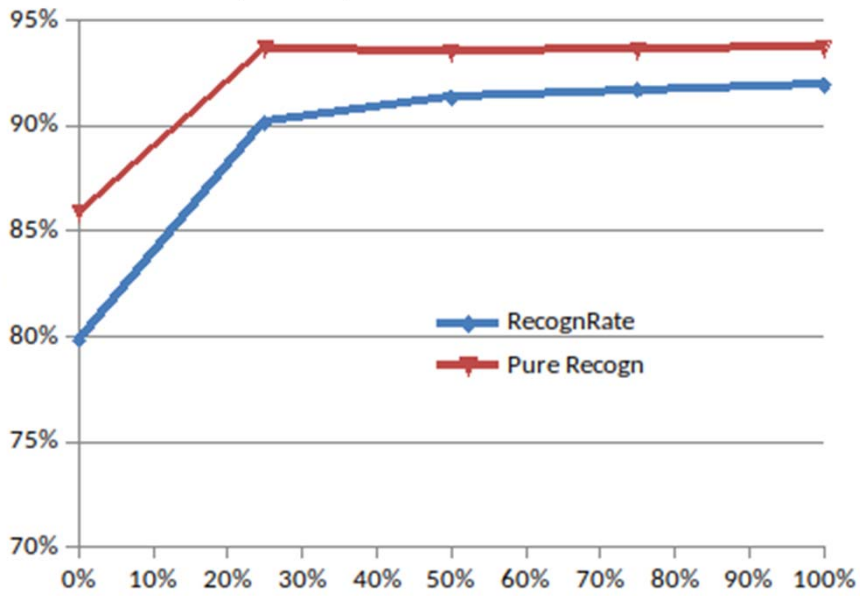


Final performance - with/without learning
Cmd recognition accuracy

WITHOUT correction
- Small LM only (same AM)
- Learning from audio+radar
- Output commands NOT corrected using CPM



WITH correction (T2):
- Large LM (same AM)
- Learning only from audio+radar
- Output commands corrected using CPM (>25%)



MALORCA

Machine Learning of Speech Recognition Models for Controller Assistance

DLR

UNIVERSITÄT DES SAARLANDES

idiap RESEARCH INSTITUTE

austro CONTROL

Air Navigation Services of the Czech Republic

Covering the sky...

Thank you very much for your attention!

This project has received funding from the SESAR Joint Undertaking under the European Union's Horizon 2020 research and innovation programme under grant agreement No 698824.

SESAR JOINT UNDERTAKING

Founding Members

EUROPEAN UNION EUROCONTROL