A Tool for Easily Integrating Grammars as Language Models into the Kaldi Speech Recognition Toolkit

Bridges and Gaps between Formal and Computational Linguistics, ESSLLI 2022

Lucía Ormaechea Grijalba, Pierrette Bouillon
Université de Genève

Benjamin Lecouteux, Didier Schwab
Université Grenoble Alpes

UNIVERSITÉ DE GENÈVE
FACULTÉ DE TRADUCTION ET D'INTERPRÉTATION

UGA Université Grenoble Alpes

Plan

1. **Introduction**
    1. Language Modeling and ASR
    2. LM types: pros and cons
    3. Introducing a new tool
2. **Designing kaldi-grammar-compiler**
    1. Tool setup: Kaldi and Regulus Lite (RL)
    2. RL grammars into Kaldi-readable LMs
3. **Evaluation**
    1. Corpora
    2. ASR setup
    3. Results
4. **Conclusion**

1. Introduction

1. **Language Modeling and ASR**
2. LM types: pros and cons
3. Introducing a new tool

Language Models:

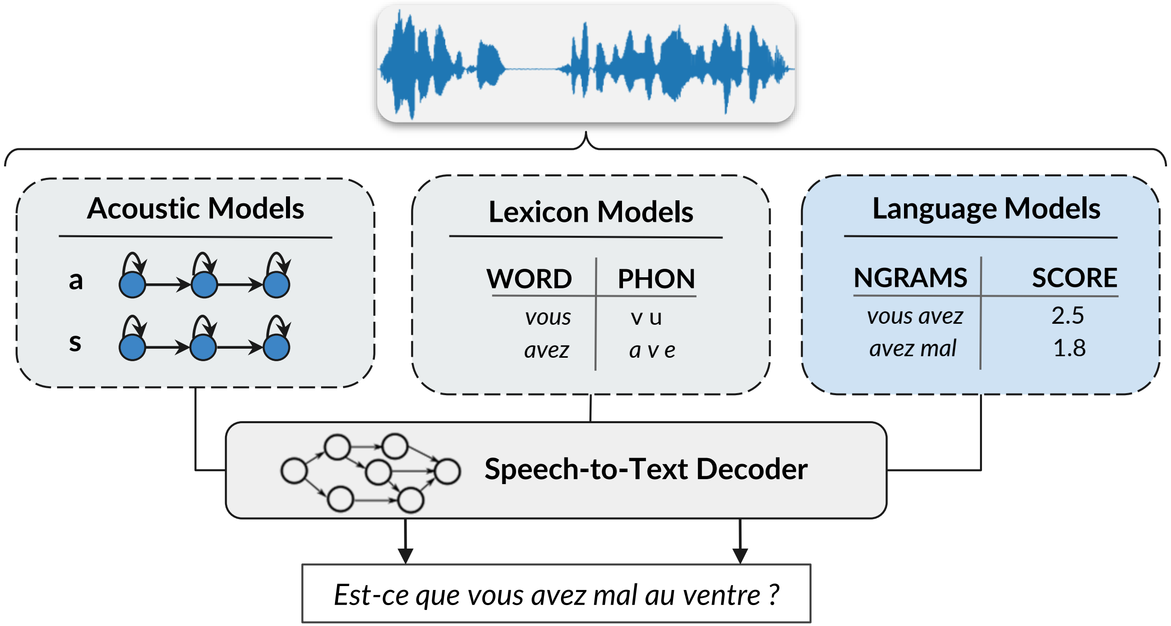
- Represent a **crucial component** in the design of **A**utomatic **S**peech **R**ecognition (**ASR**) systems.
- And more particularly, in the context of **traditional HMM-DNN ASR systems**.

1. Introduction

1. **Language Modeling and ASR**
2. LM types: pros and cons
3. Introducing a new tool

Language Models:

- Represent a **crucial component** in the design of **A**utomatic **S**peech **R**ecognition (**ASR**) systems.
- And more particularly, in the context of **traditional HMM-DNN ASR systems**.



1. Introduction

1. Language Modeling and ASR
2. **LM types: pros and cons**
3. Introducing a new tool

Two different types of Language Models (LMs) according to its way of reasoning:

1. Introduction

1. Language Modeling and ASR
2. **LM types: pros and cons**
3. Introducing a new tool

Two different types of Language Models (LMs) according to its way of reasoning:

CORPUS-BASED

LM is inferred from data

- ✓ Availability of resources and tools
- ✓ Generalizability
- ✗ Short memory span (n-gram-based)
- ✗ High-quality data not always available

1. Introduction

1. Language Modeling and ASR
2. **LM types: pros and cons**
3. Introducing a new tool

Two different types of Language Models (LMs) according to its way of reasoning:

CORPUS-BASED

LM is inferred from data

- ✓ Availability of resources and tools
- ✓ Generalizability
- ✗ Short memory span (n-gram-based)
- ✗ High-quality data not always available

KNOWLEDGE-BASED*

LM is inferred from rules

- ✓ Parsing efficiency
- ✓ Can alleviate the shortage of data
- ✓ Large scope of constraints between words
- ✗ Lack of software to easily feed them into ASR toolkits

*Also known as grammar-based.

1. Introduction

1. Language Modeling and ASR
2. **LM types: pros and cons**
3. Introducing a new tool

Two different types of Language Models (LMs) according to its way of reasoning:

CORPUS-BASED

LM is inferred from data

- ✓ Availability of resources and tools
- ✓ Generalizability
- ✗ Short memory span (n-gram-based)
- ✗ High-quality data not always available

KNOWLEDGE-BASED*

LM is inferred from rules

- ✓ Parsing efficiency
- ✓ Can alleviate the shortage of data
- ✓ Large scope of constraints between words
- ✗ Lack of software to easily feed them into ASR toolkits

OpenGrm

Kaldi Active Grammars

*Also known as grammar-based.

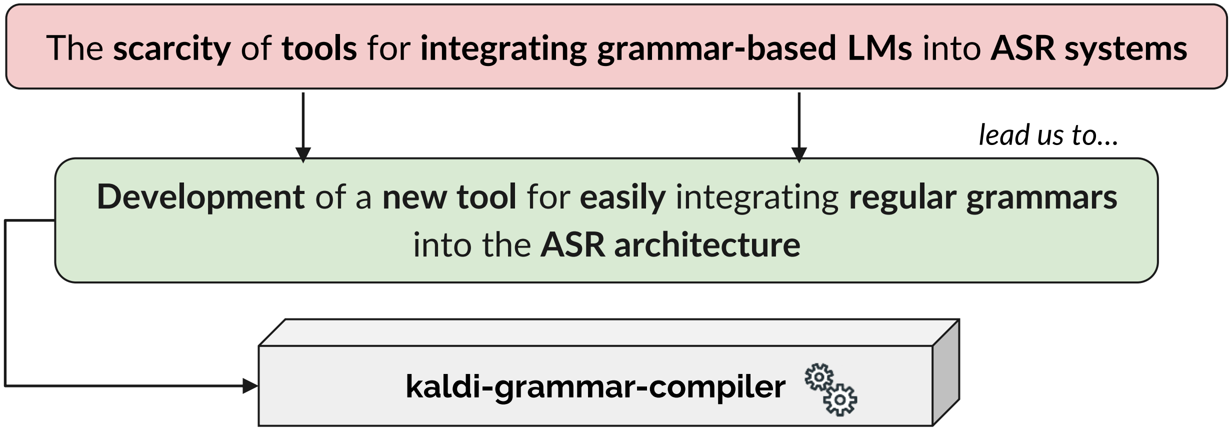
1. Introduction

1. Language Modeling and ASR
2. LM types: pros and cons
3. **Introducing a new tool**

The **scarcity** of **tools** for **integrating grammar-based LMs** into **ASR systems**

1. Introduction

1. Language Modeling and ASR
2. LM types: pros and cons
3. **Introducing a new tool**



1. Introduction

1. Language Modeling and ASR
2. LM types: pros and cons
3. **Introducing a new tool**

The **scarcity** of **tools** for **integrating grammar-based LMs** into **ASR systems**

lead us to...

Development of a **new tool** for **easily** integrating **regular grammars** into the **ASR architecture**

kaldi-grammar-compiler

Under two main principles:

1. **Prone to extensive use** → With its **implementation** in a **widely used ASR toolkit.**
2. **Easy-to-use** → Ensuring good usability for **linguists** and **translators**.

PRINCIPLE I: Prone to extensive use → **Kaldi – Speech Processing Toolkit**

- Introduced by (Povey *et al.*, 2011) as an **open source toolkit** for speech processing.
- **Widely used** within the **ASR community**.
- Highly **usable** and **modifiable**.
- Uses a **F**inite **S**tate **T**ransducer (**FST**) framework for training and decoding algorithms (Horndasch *et al.*, 2016).

PRINCIPLE I: Prone to extensive use → **Kaldi – Speech Processing Toolkit**

- Introduced by (Povey *et al.*, 2011) as an **open source toolkit** for speech processing.
- **Widely used** within the **ASR community**.
- Highly **usable** and **modifiable**.
- Uses a **F**inite **S**tate **T**ransducer (**FST**) framework for training and decoding algorithms (Horndasch *et al.*, 2016).

Four different levels of FSTs

	H → HMM	C → Context	L → Lexicon	G → Grammar
Input label	HMM state	Context phone	Phone	Word
Output label	Context phone	Phone	Word	Word

PRINCIPLE I: Prone to extensive use → **Kaldi – Speech Processing Toolkit**

- Introduced by (Povey *et al*., 2011) as an **open source toolkit** for speech processing.
- **Widely used** within the **ASR community**.
- Highly **usable** and **modifiable**.
- Uses a **F**inite **S**tate **T**ransducer (**FST**) framework for training and decoding algorithms (Horndasch *et al*., 2016).

Four different levels of FSTs

	H → HMM	C → Context	L → Lexicon	G → Grammar
Input label	HMM state	Context phone	Phone	Word
Output label	Context phone	Phone	Word	Word

PRINCIPLE II: Easy-to-use → **Regulus Lite (RL) Grammars**

- **Finite-state based** and **language independent.**
- Designed for the **rapid development** of small to medium vocabulary **speech translation applications** (Rayner *et al*., 2016).
- Featuring an **user-friendly syntax**, with **rules** describing individual **sentences**.

PRINCIPLE II: Easy-to-use → **Regulus Lite (RL) Grammars**

- **Finite-state based** and **language independent.**
- Designed for the **rapid development** of small to medium vocabulary **speech translation applications** (Rayner *et al.*, 2016).
- Featuring an **user-friendly syntax**, with **rules** describing individual **sentences**.

Source **pattern**

```
Utterance
Source $avez_vous ( mal | des
          douleurs ) quelque part
EndUtterance
```

PRINCIPLE II: Easy-to-use → **Regulus Lite (RL) Grammars**

- **Finite-state based** and **language independent.**
- Designed for the **rapid development** of small to medium vocabulary **speech translation applications** (Rayner *et al.*, 2016).
- Featuring an **user-friendly syntax**, with **rules** describing individual **sentences**.

Source **pattern**	PhraseId **pattern**
Utterance Source $avez_vous (mal | des douleurs) quelque part EndUtterance	PhraseId $avez_vous Source (avez-vous | vous avez) EndPhraseId

Kaldi supports any **LM** that is **representable** as an **FST**, given its **finite-state-based framework**

Kaldi supports any **LM** that is **representable** as an **FST**, given its **finite-state-based framework**

Chomsky Hierarchy		
Grammar	**Language**	**Automaton**
Type-0	Recursively enumerable	Turing machine
Type-1	Context-sensitive	Linear bounded automaton
Type-2	Context-free	Push-down automaton
Type-3	Regular	Finite automaton

Regulus Lite grammars fall into the category of **regular grammars**

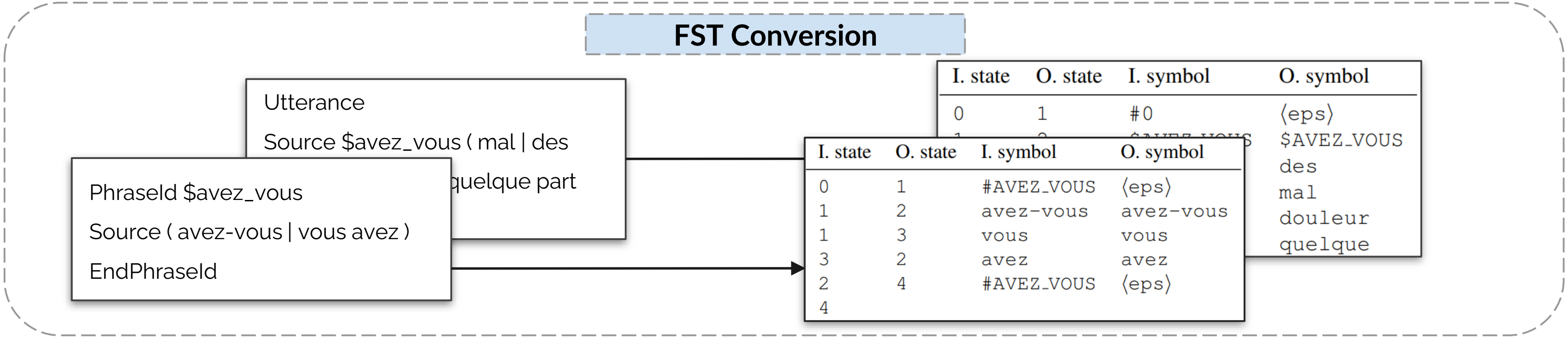
Kaldi supports any **LM** that is **representable** as an **FST**, given its **finite-state-based framework**

Chomsky Hierarchy		
Grammar	**Language**	**Automaton**
Type-0	Recursively enumerable	Turing machine
Type-1	Context-sensitive	Linear bounded automaton
Type-2	Context-free	Push-down automaton
Type-3	Regular	Finite automaton

Regulus Lite grammars fall into the category of **regular grammars**

Meaning that…

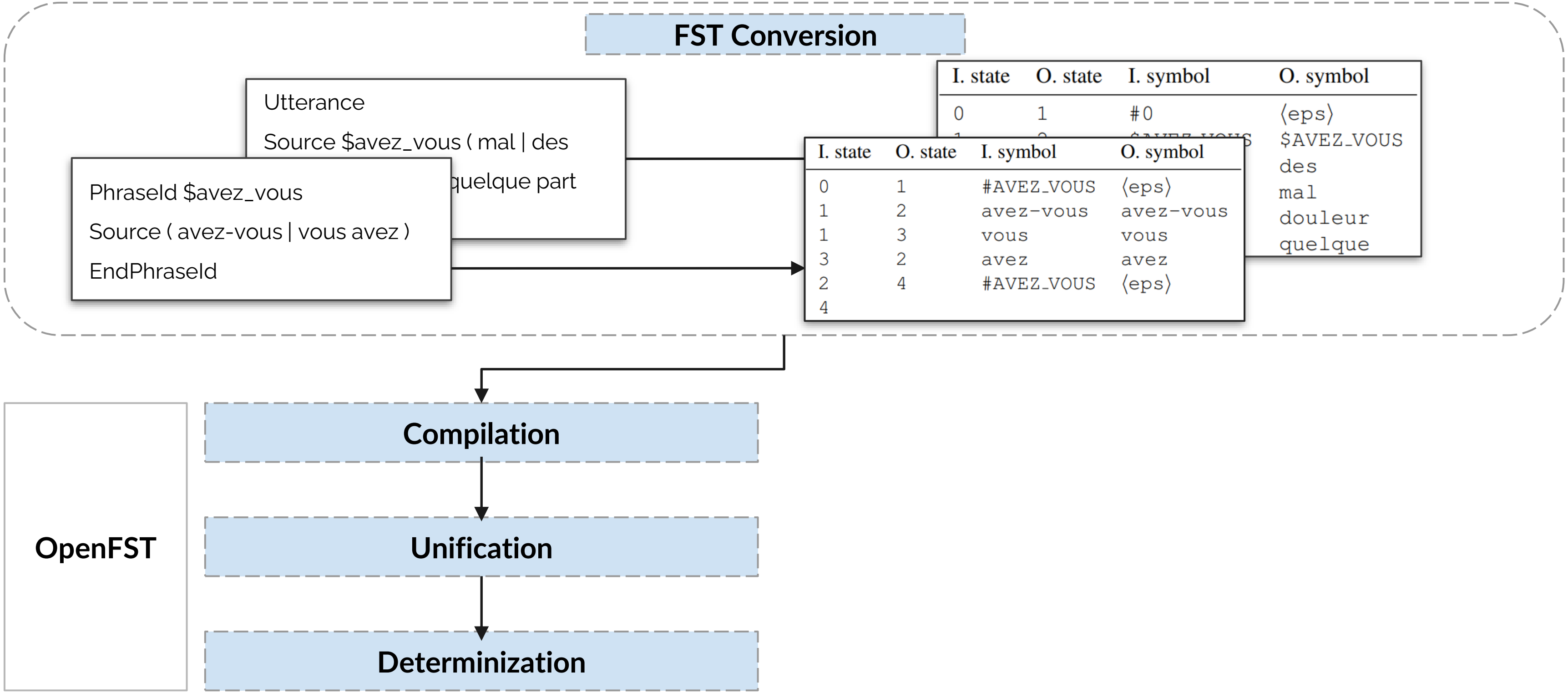
The **language produced** by this **type of grammars** is **recognized** or **accepted** by a **FST**



FST Conversion
Utterance
Source $avez_vous (mal | des
quelque part
PhraseId $avez_vous
Source (avez-vous | vous avez)
EndPhraseId
I. state O. state I. symbol O. symbol
0 1 #0 ⟨eps⟩
$AVEZ_VOUS
des
mal
douleur
quelque
I. state O. state I. symbol O. symbol
0 1 #AVEZ_VOUS ⟨eps⟩
1 2 avez-vous avez-vous
1 3 vous vous
3 2 avez avez
2 4 #AVEZ_VOUS ⟨eps⟩
4

2. Designing *kaldi-grammar-compiler*

1. Tool setup: Kaldi and Regulus Lite (RL)
2. **RL grammars into Kaldi readable LMs**



2. Designing *kaldi-grammar-compiler*

1. Tool setup: Kaldi and Regulus Lite (RL)
2. **RL grammars into Kaldi readable LMs**

FST Conversion

Utterance

Source $avez_vous (mal | des

quelque part

PhraseId $avez_vous

Source (avez-vous | vous avez)

EndPhraseId

I. state	O. state	I. symbol	O. symbol
0	1	#0	⟨eps⟩
1			$AVEZ_VOUS
			des
			mal
			douleur
			quelque

I. state	O. state	I. symbol	O. symbol
0	1	#AVEZ_VOUS	⟨eps⟩
1	2	avez-vous	avez-vous
1	3	vous	vous
3	2	avez	avez
2	4	#AVEZ_VOUS	⟨eps⟩
4			

OpenFST

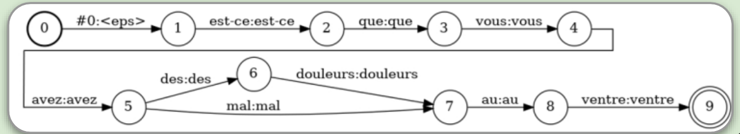
Compilation

Unification

Determinization

G.fst Graph

- **Representing** the **input Regulus Lite grammar**.
- **Readable** as a **LM** in **Kaldi**.



3. Evaluation

1. **Corpora**
2. ASR setup
3. Results

For evaluation purposes:

- **Dedicated corpora** → Gathered via **data collection campaigns**.
- **Highly domain-specific** → Derived from **ASR systems** being used under very **particular scenarios**.

3. Evaluation

1. **Corpora**
2. ASR setup
3. Results

For evaluation purposes:

- **Dedicated corpora** → Gathered via **data collection campaigns**.
- **Highly domain-specific** → Derived from **ASR systems** being used under very **particular scenarios**.

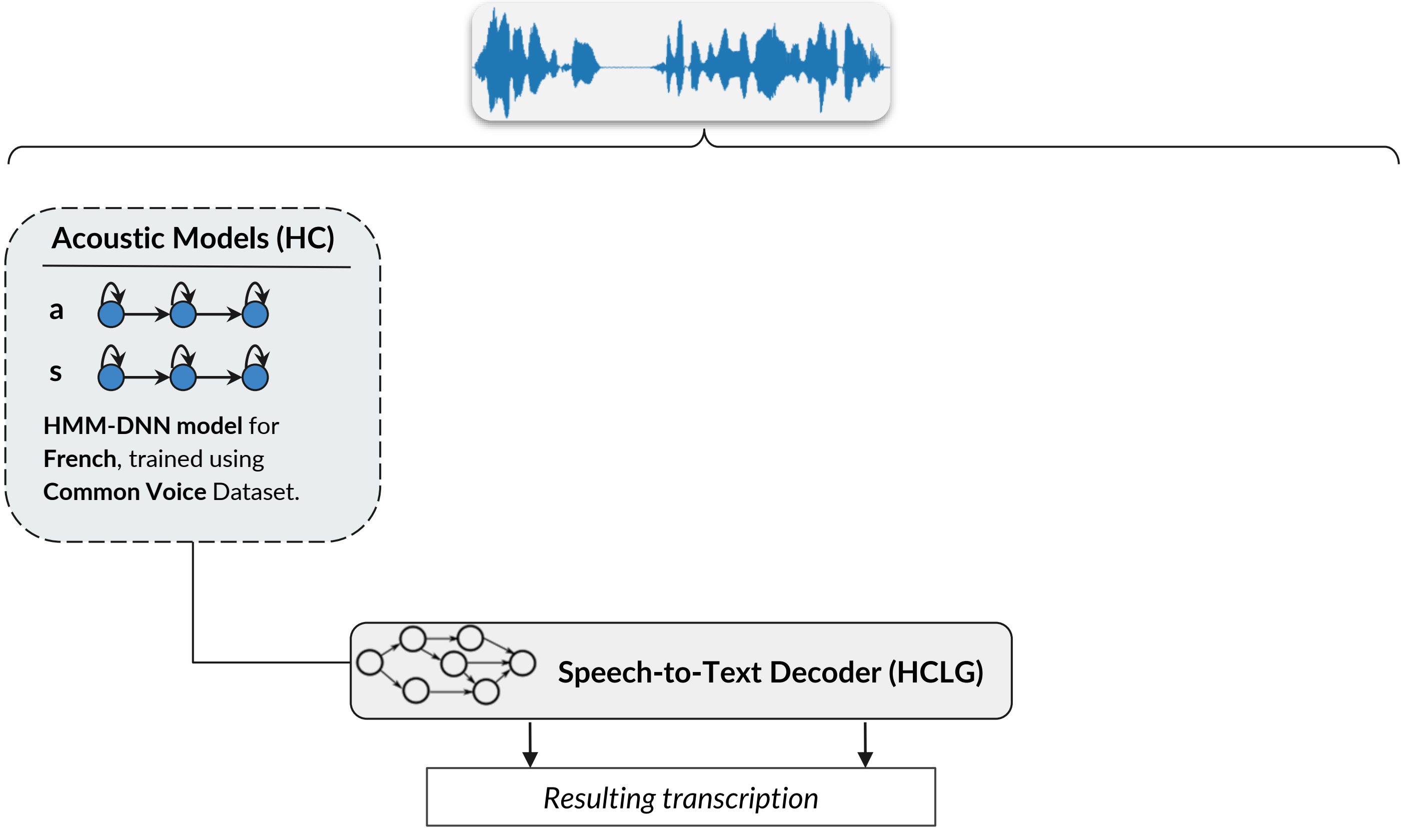
	Language	**Speakers**	**Gender**	**Length**	**Utterances**	**Words**
MeDiCo (*Medical Discourse Corpus*)	French	14	9F, 5M	0h 41mn	713	≲6k
HomeAutomation (Vacher *et al.*, 2014)	French	23	9F, 14M	1h 38mn	3114	≲10k

- **Two different Kaldi ASR engines** were built.
- Both integrated a **regular grammar** as **LM** in their **decoding graph (HCLG)**.

3. Evaluation

1. Corpora
2. **ASR setup**
3. Results

- **Two different Kaldi ASR engines** were built.
- Both integrated a **regular grammar** as **LM** in their **decoding graph (HCLG)**.



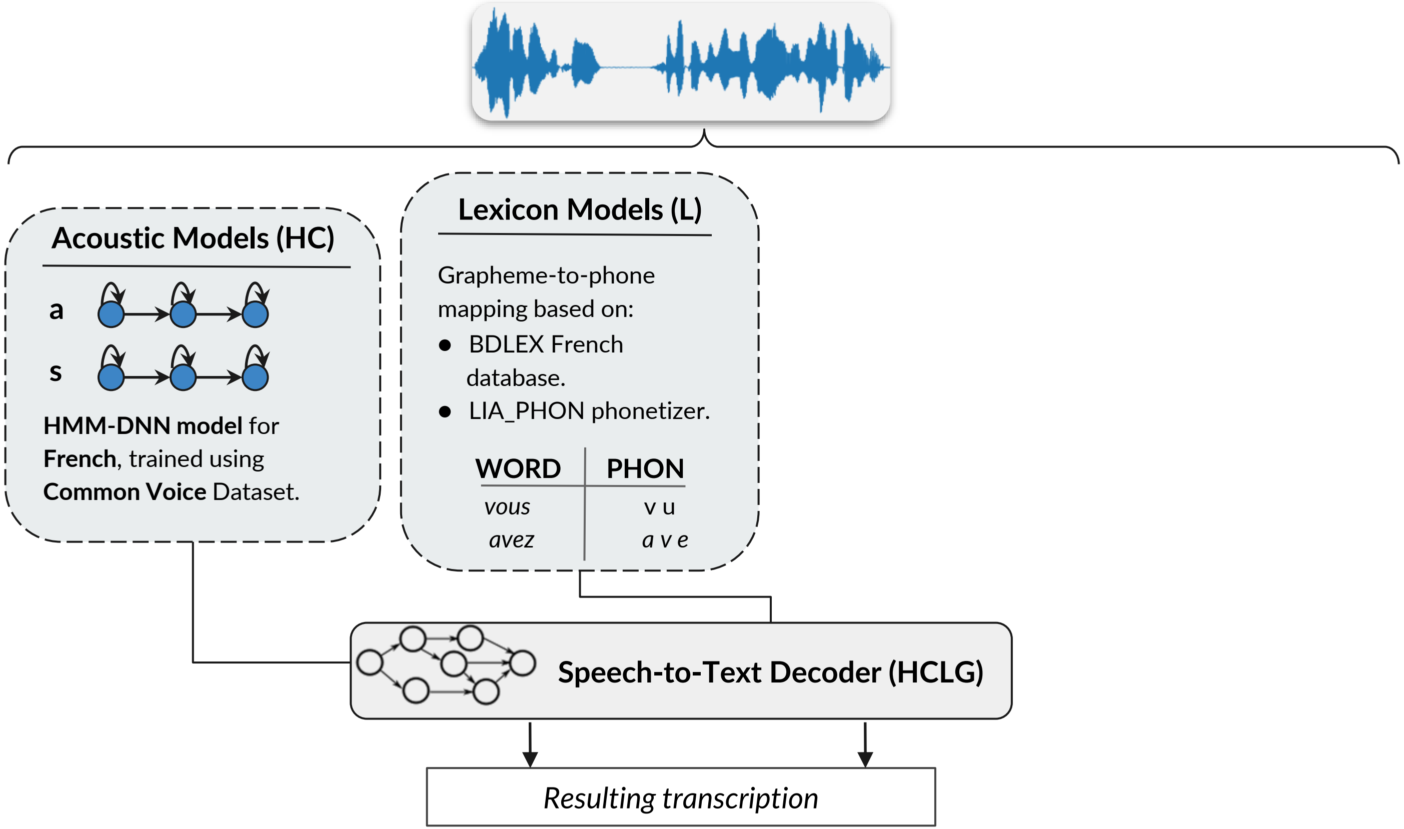
3. Evaluation

1. Corpora
2. **ASR setup**
3. Results

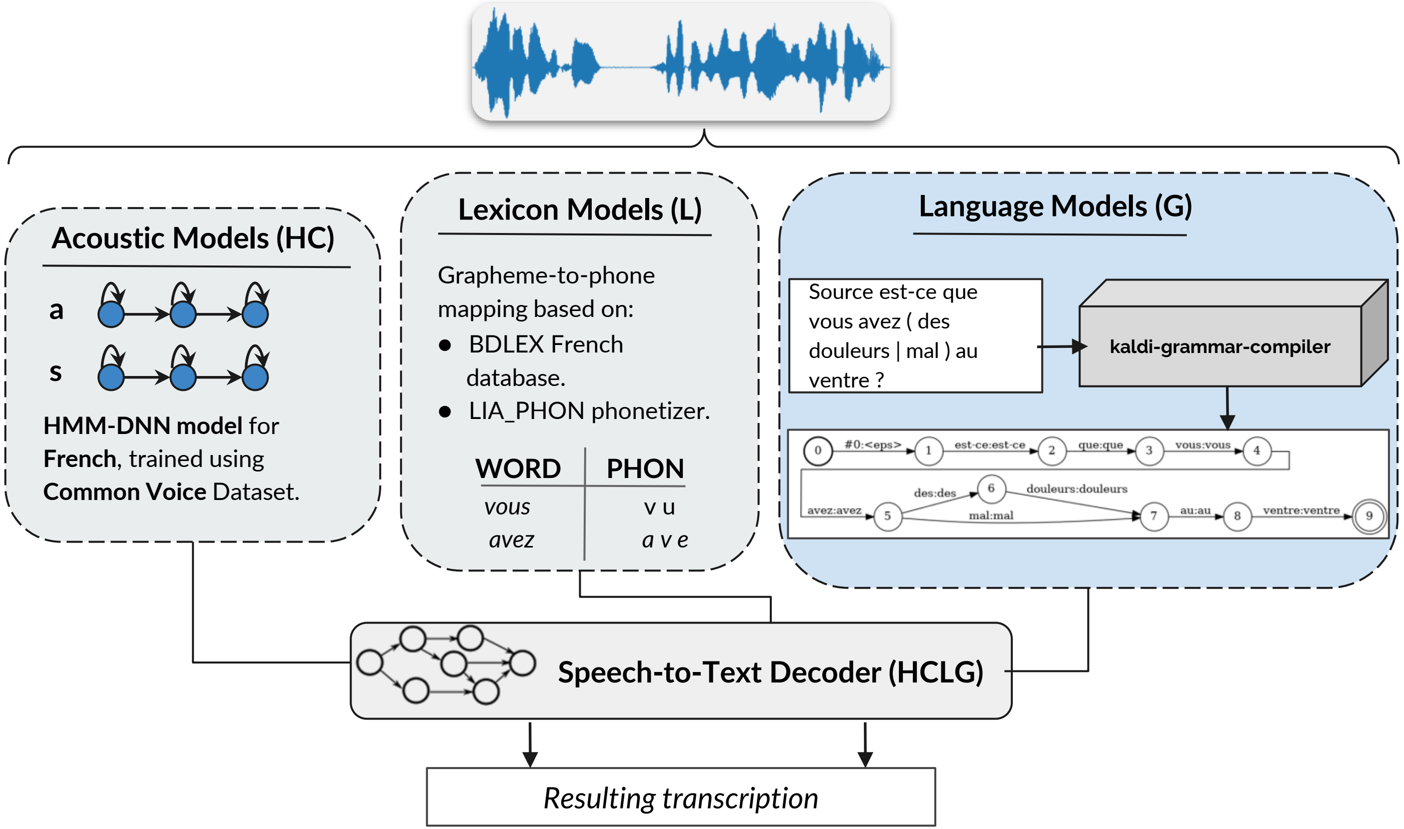
3. Evaluation

1. Corpora
2. **ASR setup**
3. Results

- **Two different Kaldi ASR engines** were built.
- Both integrated a **regular grammar** as **LM** in their **decoding graph (HCLG)**.



- **Two different Kaldi ASR engines** were built.
- Both integrated a **regular grammar** as **LM** in their **decoding graph (HCLG)**.



3. Evaluation

1. Corpora
2. **ASR setup**
3. Results

3. Evaluation

1. Corpora
2. ASR setup
3. **Results**

Key points:

- **Evaluation** measured in terms of **W**ord **E**rror **R**ate (**WER**), to calculate the **transcription accuracy.**

$$WER = \frac{S + D + I}{N} \times 100$$

where:
S = number of **substitutions**
D = number of **deletions**
I = number of **insertions**
N = number of **words** in the **reference**

3. Evaluation

1. Corpora
2. ASR setup
3. **Results**

Key points:

- **Evaluation** measured in terms of **W**ord **E**rror **R**ate (**WER**), to calculate the **transcription accuracy.**

$$\boxed{WER = \frac{S + D + I}{N} \times 100}$$

where:
S = number of **substitutions**
D = number of **deletions**
I = number of **insertions**
N = number of **words** in the **reference**

- **Compared** the **grammar-based ASR systems** against a **baseline 3-gram LM**, inferred from **data generated** by the **Regulus Lite grammars**.

3. Evaluation

1. Corpora
2. ASR setup
3. **Results**

Model	*Corpus*	Recognized words	I	D	S	WER (%)
Grammar-based LM	MeDiCo	5208 / 5598	58	76	256	**6.97**
	Home Automation	8975 / 9639	86	338	240	**6.89**
Baseline n-gram LM	MeDiCo	4690 / 5598	298	85	525	**16.22**
	Home Automation	8850 / 9639	156	161	472	**8.19**

- Both **MeDiCo** and **HomeAutomation** return a significantly **low WER**.
- The **ability of the grammars** to **extend** the **span** of **linguistic constraints** between words has a **positive effect** in the context of **highly domain-specific ASR applications**.

3. Evaluation

1. Corpora
2. ASR setup
3. **Results**

Model	*Corpus*	Recognized words	I	D	S	WER (%)
Grammar-based LM	MeDiCo	5208 / 5598	58	76	256	**6.97**
	Home Automation	8975 / 9639	86	338	240	**6.89**
Baseline n-gram LM	MeDiCo	4690 / 5598	298	85	525	**16.22**
	Home Automation	8850 / 9639	156	161	472	**8.19**

- Both **MeDiCo** and **HomeAutomation** return a significantly **low WER**.
- The **ability of the grammars** to **extend** the **span** of **linguistic constraints** between words has a **positive effect** in the context of **highly domain-specific ASR applications**.

4. Conclusion

Key points:

- We introduced an **extension** for **easily integrating regular grammars** as **LMs** into **Kaldi**.

kaldi-grammar-compiler

- Achieved **satisfactory results** in the **experiments** performed.

This shows that **grammar-based ASR systems** obtain a **competitive performance** when applied in **constrained domain-specific applications**.

4. Conclusion

Key points:

- We introduced an **extension** for **easily integrating regular grammars** as **LMs** into **Kaldi**.

kaldi-grammar-compiler

- Achieved **satisfactory results** in the **experiments** performed.

This shows that **grammar-based ASR systems** obtain a **competitive performance** when applied in **constrained domain-specific applications**.

Further work:

- Explore how to **leverage grammar knowledge**, so as to **specialize** a **neural-based LM** (Lee, 2020).
- **Generalize** the **input grammar format**, so as to **extend** the **applicability** of our designed tool beyond the **Regulus Lite syntax**.

Thanks for your attention!

Any questions?

https://luciaormaechea.com

Lucia.OrmaecheaGrijalba@unige.ch

4. References

[In order of appearance]

- **Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlıcek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., and Vesel, K**. The Kaldi Speech Recognition Toolkit. *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, **2011**.

- **Rayner, M., Armando, A., Bouillon, P., Ebling, S., Gerlach, J., Halimi, S., Strasly, I., and Tsourakis, N.** Helping Domain Experts build Phrasal Speech Translation Systems. In Jose F. Quesada, *et al*., editors, *Future and Emergent Trends in Language Technology*, pages 41–52. Springer International Publishing, **2016**.

- **Horndasch, A., Kaufhold, C., and Noth, E.** How to add Word Classes to the Kaldi Speech Recognition Toolkit. In Petr Sojka, *et al*., editors, *Text, Speech, and Dialogue*, pages 486–494. Springer International Publishing, **2016**.

- **Vacher, M., Lecouteux, B., Chahuara, P., Portet, F., Meillon B., and Bonnefond, N.** The Sweet-Home Speech and Multimodal Corpus for Home Automation Interaction. In *The 9th edition of the Language Resources and Evaluation Conference* (LREC), pages 4499–4506, **2014**. URL http://hal.archives-ouvertes.fr/hal-00953006

- **Lee, Jay Yoon**. Injecting Output Constraints into Neural NLP Models, *Ph.D. Thesis*. Carnegie Mellon University, **2020**.