Application for the Post Doc Position in Speech Recognition (Malorca project)

Luiza Orosanu

INRIA-Loria, Nancy, France

5th February 2016

Short introduction

2006 -2010	-2011	-2012	-2015
BSc in computer science	**MSc** in computer science	**Engineer:** ALLEGRO project	**PhD student:** RAPSODIE project
head tracking Wii remote infrared sensors	speech recognition remote sound adaptation	incorrect entries non-native speech speech-text alignments	hybrid language models add new words question detection
Romania	France		

Short introduction

2006 -2010	-2011	-2012	-2015
BSc in computer science	**MSc** in computer science	**Engineer:** ALLEGRO project	**PhD student:** RAPSODIE project
head tracking Wii remote infrared sensors	speech recognition remote sound adaptation	incorrect entries non-native speech speech-text alignments	hybrid language models add new words question detection
Romania	France		

Sommaire

Hybrid language models

Context

* **OOV words** (regardless the size of vocabulary)

Reference:	dans	un		village	du	nord
Hypothesis:	dans	++parole++	l'	âge	du	nord

* maximize the **understanding of the resulting transcription** for the deaf community

Hybrid language models

* combining words with word-fragments

Approach

- Choice of a **hybrid language model of words & syllables**
  - ensure the **proper recognition of the most frequent words**
  - provide a sequence of syllables for out-of-vocabulary words

⚠ syllables trained on **real pronunciations**

(1 syllable = 1 single sequence of phonemes = 1 pronunciation)

Approach

- Choice of a **hybrid language model of words & syllables**
  - ensure the **proper recognition of the most frequent words**
  - provide a sequence of syllables for out-of-vocabulary words

> ⚠ syllables trained on **real pronunciations**
> (1 syllable = 1 single sequence of phonemes = 1 pronunciation)

- Motivations
  - **syllables** $\longleftarrow$ study on optimising the phonetic decoding
  - **words** $\longleftarrow$ interviews conducted with deaf people

Approach

- Choice of a **hybrid language model of words & syllables**
  * ensure the **proper recognition of the most frequent words**
  * provide a sequence of syllables for out-of-vocabulary words

⚠ syllables trained on **real pronunciations**

(1 syllable = 1 single sequence of phonemes = 1 pronunciation)

- Motivations
  * **syllables** ⟵ study on optimising the phonetic decoding
  * **words** ⟵ interviews conducted with deaf people

Example of a hybrid transcription

Decoding:	une femme a été _b_l_e _s_e
Display:	une femme a été b l é s é

Syllabification

- Training corpus for hybrid language models
  - keep only the **most frequent words** ($\#occ \geq N$)
  - **syllabify** the other words (less frequent)

Syllabification

- Training corpus for hybrid language models
  - keep only the **most frequent words** ($\#occ \geq N$)
  - **syllabify** the other words (less frequent)

- Syllabification
  - forced alignement **words** $\rightarrow$ **phonemes**
  - syllabification rules **phonemes** $\rightarrow$ **syllables** [Bigi et al. 2010]

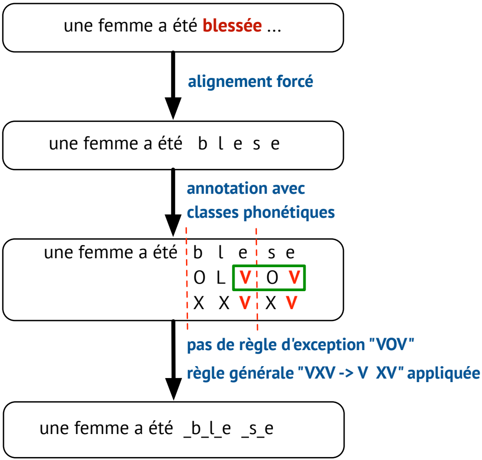
Syllabification

- Training corpus for hybrid language models
  - keep only the **most frequent words** ($\#occ \geq N$)
  - **syllabify** the other words (less frequent)

- Syllabification
  - forced alignement **words** $\rightarrow$ **phonemes**
  - syllabification rules **phonemes** $\rightarrow$ **syllables** [Bigi et al. 2010]
    - a syllable contains a single vowel (V)
    - a pause designates a syllable's boundary

Rule type	**Sequence of phonetic classes**	**Split position**	**Resulting syllables**	
GEN	VV	0	V	V
GEN	VxV	0	V	xV
GEN	VxxV	1	Vx	xV
EXC	VOLV	0	V	OLV

Syllabification



Evaluation of hybrid language models

Word-based model:	une femme a été blessée
Hybrid model:	une femme a été _b_l_e _s_e

- **Performance of hybrid models**
  - phoneme error rate
  - percentage of words in the automatic transcription
  - percentage of correctly recognized words and syllables
  - percentage of out-of-vocabulary words recognized as syllables

- Apply a filter on the **confidence measure of words**

  → phonetize words with low confidence measures

Conclusions

- our hybrid modeling solution takes into acount **real pronunciations**
- the speech recognition outputs contain mainly words
- over 70% of words are correctly recognized
- the confidence measures can effectively select the correctly recognized words
- an increased amount of syllables in the training corpus
  - improves the percentage of correctly recognized syllables
  - improves the percentage of OOV words decoded as syllables

Sommaire

Adding new words to a language model

- **Context**
  - OOV words that are frequently pronounced
    - ex: words specific to a certain area
- **Adding new words to an ASR system involves**
  - generating the pronunciation variants
  - modifying the language model

Approach

- without retraining or adapting the model
  (which requires a lot of new data relative to the new words)

- approach based on the similarity between words

on ignorait encore lundi	**soir**	les conditions de sa survie
on ignorait encore lundi	**matin**	les conditions de sa survie

Approach

- without retraining or adapting the model
  (which requires a lot of new data relative to the new words)

- approach based on the similarity between words

on ignorait encore lundi	**soir**	les conditions de sa survie
on ignorait encore lundi	**matin**	les conditions de sa survie

→ use **a few examples of sentences** for each new word

→ find similar known words (having similar neighbor distributions)

→ **transpose the LM probabilities** of known words onto the new words

Neighbors of new words

1. Use a few **examples of sentences** with the new word

$\rightarrow$ compute the neighbor distributions of the new word **nW**

$$P_k(w|\textbf{nW}) \text{ for } k \in \{..., -2, -1, +1, +2, ...\}$$

Neighbors of new words

1. Use a few **examples of sentences** with the new word

   → compute the neighbor distributions of the new word **nW**

$$P_k(w|\mathbf{nW}) \text{ for } k \in \{..., -2, -1, +1, +2, ...\}$$

- example of a new word: **soir**
- examples of sentences

	−2	−1		+1	+2	
on ignorait	encore	lundi	**soir**	les	conditions	de sa survie
devine qui vient	dîner	ce	**soir**			
pas de consigne de	vote	au	**soir**	du	premier	tour

- preceding and succeeding neighbors

$k = -2$	encore, dîner, vote, ...
$k = -1$	lundi, ce, au, ...
$k = +1$	les, du, ...
$k = +2$	conditions, premier, ...

Neighbors of known words

2. Search for similar words in a **reference corpus**

$\rightarrow$ compute the neighbor distributions of each known word **kW**

$$P_k(w'|\mathbf{kW}) \text{ for } k \in \{..., -2, -1, +1, +2, ...\}$$

Neighbors of known words

2. Search for similar words in a **reference corpus**

$\rightarrow$ compute the neighbor distributions of each known word **kW**

$$P_k(w'|\mathbf{kW}) \text{ for } k \in \{..., -2, -1, +1, +2, ...\}$$

- use directly the n-gram counts file
  - * 3-gram $\Rightarrow$ maximum 4 neighbors $k \in \{-2, -1, +1, +2\}$

- examples of 3-gram entries with the known word '**matin**'

"**matin**	a	été	*10*"	$\rightarrow$ voisin $k = +1$ 'a'; voisin $k = +2$ 'été'
"beau	**matin**	de	*9*"	$\rightarrow$ voisin $k = -1$ 'beau'; voisin $k = +1$ 'de'
"jusqu'	au	**matin**	*28*"	$\rightarrow$ voisin $k = -2$ 'jusqu'; voisin $k = -1$ 'au'

- preceding and succeeding neighbors

$k = -2$	jusqu', ...
$k = -1$	beau, au, ...
$k = +1$	de, a, ...
$k = +2$	été, ...

Word similarity

3. Compute the **KL divergence** between the neighbor distributions

$\rightarrow$ between each known word (**kW**) and the new word (**nW**)

Divergence computed on each k position:

$$D_{KL}\left(\, P_k(\bullet|\mathbf{kW}) \;||\; P_k(\bullet|\mathbf{nW}) \,\right) \;=\; \sum_{w \in V(\mathbf{nW})} P_k(w|\mathbf{kW}) \; log \left(\frac{P_k(w|\mathbf{kW})}{P_k(w|\mathbf{nW})}\right)$$

Global divergence: $D(\mathbf{kW}, \mathbf{nW}) \;=\; \sum_{k} D_k(\mathbf{kW}, \mathbf{nW})$

Word similarity

3. Compute the **KL divergence** between the neighbor distributions

→ between each known word (**kW**) and the new word (**nW**)

Divergence computed on each k position:

$$D_{KL}(\ P_k(\bullet|\mathbf{kW})\ ||\ P_k(\bullet|\mathbf{nW})\ ) = \sum_{w \in V(\mathbf{nW})} P_k(w|\mathbf{kW})\ log\left(\frac{P_k(w|\mathbf{kW})}{P_k(w|\mathbf{nW})}\right)$$

Global divergence: $D(\mathbf{kW}, \mathbf{nW}) = \sum_k D_k(\mathbf{kW}, \mathbf{nW})$

4. Select the **mots similar words** to the new word

→ those having minimal divergences

Word similarity

3. Compute the **KL divergence** between the neighbor distributions

$\rightarrow$ between each known word (**kW**) and the new word (**nW**)

Divergence computed on each k position:

$$D_{KL}\left(P_k(\bullet|\mathbf{kW}) \;||\; P_k(\bullet|\mathbf{nW}) \right) = \sum_{w \in V(\mathbf{nW})} P_k(w|\mathbf{kW}) \; log\left(\frac{P_k(w|\mathbf{kW})}{P_k(w|\mathbf{nW})}\right)$$

Global divergence: $D(\mathbf{kW}, \mathbf{nW}) = \sum_k D_k(\mathbf{kW}, \mathbf{nW})$

4. Select the **mots similar words** to the new word

$\rightarrow$ those having minimal divergences

Examples of similar words:

soir $\rightarrow$ matin, midi, dimanche, samedi, vendredi
soirs $\rightarrow$ temps, joueurs, matchs, pays, matches

Adding new n-grams

5. **Transpose the n-gram probabilities** of similar words onto the new word
   - → seek the n-grams that contain similar words
   - → replace the 'similar words' with the 'new word'
   - → add the new n-grams into the new language model

Adding new n-grams

5. **Transpose the n-gram probabilities** of similar words onto the new word

→ seek the n-grams that contain similar words

→ replace the 'similar words' with the 'new word'

→ add the new n-grams into the new language model

- new word "**soir**" similar to known word "**matin**"
- known n-grams (in the language model)

  `"-1.48214 possible ce` **`matin`**`"`

  `"-1.404164` **`matin`** `ajoute que"`

- new n-grams (to add into the new language model)

  `"-1.48214 possible ce` **`soir`**`"`

  `"-1.404164` **`soir`** `ajoute que"`

Setup for experiments

- **44 new words** selected
- Search for similar words
  - sentences based on "**word**|POS" units

    qui|PRO:REL **vient**|VER:pres **dîner**|VER:infi **ce**|PRO:DEM **soir**|NOM

  - 4 neighbors for each word: $k \in \{-2, -1, +1, +2\}$
- Evaluate the impact of
  - number of examples of sentences for each new word (5, 10, 20 or 50)
  - nomber of similar words for each new word (5, 10, 20 or 50)

Setup for experiments

- **BASELINE** language model
  * large vocabulary language model trained by interpolation
  * the 44 new words are absent in this model

- **ORACLE** language model
  * large vocabulary language model trained by interpolation
  * the 44 new words are present in this model

- 4 language models **LM-INTERP-1,-2,-3,-4**
  * large vocabulary language models trained by interpolation
    - on the same data as 'BASELINE'
    - plus the examples of sentences for each new word (5, 10, 20 or 50)
  * the 44 new words are present in these models

⚠ the optimal interpolation weights estimated on the ETAPE dev corpus
$\rightarrow$ the 44 new words have an occurrence frequency of 0,93%

Size of language models

- **New language models** ('baseline+1-,2-,3-grams')
  - add 1-,2-,3-grams of new words into the BASELINE model
  - new n-grams chosen according to the
    - number of examples of senteneces for each new word (5, 10, 20 or 50)
    - number of similar words for each new word (5, 10, 20 or 50)

Size of language models

- **New language models** ('baseline+1-,2-,3-grams')
  - add 1-,2-,3-grams of new words into the BASELINE model
  - new n-grams chosen according to the
    - number of examples of senteneces for each new word (5, 10, 20 or 50)
    - number of similar words for each new word (5, 10, 20 or 50)

	baseline	**'baseline+1-,2-,3-grams'** 5 examples of sentences 5 similar words	**'baseline+1-,2-,3-grams'** 50 examples of sentences 50 similar words	**ORACLE**
#2-grams	37,1	38,0 [+2%]		43,3
#3-grams	63,1	67,2 [+6%]		80,1

Number [in milions] of 2-grams and 3-grams

Size of language models

- **New language models** ('baseline+1-,2-,3-grams')
  - ∗ add 1-,2-,3-grams of new words into the BASELINE model
  - ∗ new n-grams chosen according to the
    - ▷ number of examples of senteneces for each new word (5, 10, 20 or 50)
    - ▷ number of similar words for each new word (5, 10, 20 or 50)

	baseline	'baseline+1-,2-,3-grams' 5 examples of sentences 5 similar words		'baseline+1-,2-,3-grams' 50 examples of sentences 50 similar words		ORACLE
#2-grams	37,1	38,0	[+2%]	40,7	[+10%]	43,3
#3-grams	63,1	67,2	[+6%]	94,2	[+49%]	80,1

Number [in milions] of 2-grams and 3-grams

Evaluation

- Setup for evaluations
  - the LMs are evaluated over the ESTER2 development set
  - the 44 new words have an occurrence frequency of 1.33%

- Compare the performance of new LMs with baseline LM
  - word error rate (WER)
  - percentage of new words correctly recognized

The WER performances

BASELINE	**26.97%**
ORACLE	**24.80%**

⚠ 1,33% occurrences of 44 new words

The WER performances

BASELINE **26.97%**
ORACLE **24.80%**

# examples of sentences		LM-INTERP	'baseline+1-,2-,3-grams' # similar words 5	10	20	50
	5		**25.78**	25.83	25.96	26.01
	10		**25.74**	25.84	25.96	26.05
	20		**25.63**	25.68	25.92	25.95
	50		**25.68**	25.75	25.82	25.99

⇒ better performances are obtained with **few similar words** (5) and with a **reasonable number of examples of sentences** (20-50)

⇒ adding n-grams of new words provides an **absolute improvement of 1.3%** on WER

The WER performances

BASELINE **26.97%**
ORACLE **24.80%**

# examples of sentences	LM-INTERP	‘baseline+1-,2-,3-grams’ # similar words 5	10	20	50
5	26.12	**25.78**	25.83	25.96	26.01
10	26.02	**25.74**	25.84	25.96	26.05
20	25.81	**25.63**	25.68	25.92	25.95
50	25.68	**25.68**	25.75	25.82	25.99

⇒ the new models ‘baseline+1-,2-,3-grams’ outperform the ’LM-INTERP’ models

Percentage of new words correctly recognized

BASELINE	**0.00%**
ORACLE	**85.45%**

Percentage of new words correctly recognized

BASELINE	**0.00%**
ORACLE	**85.45%**

	LM-INTERP	‘baseline+1-,2-,3-grams’			
		# similar words			
# examples of sentences		5	10	20	50
5		**64.90**	61.09	58.36	56.72
10		**63.09**	61.09	57.09	55.27
20		**68.72**	65.81	61.27	58.18
50		**68.54**	63.45	61.81	57.09

⇒ better performances are obtained with **few similar words** (5) and with a **reasonable number of examples of sentences** (20-50)

⇒ adding n-grams of new words allows to **correcly recognize 69% of new words**

Percentage of new words correctly recognized

BASELINE	**0.00%**
ORACLE	**85.45%**

# examples of sentences	LM-INTERP	'baseline+1-,2-,3-grams' # similar words 5	10	20	50
5	44.72	**64.90**	61.09	58.36	56.72
10	47.45	**63.09**	61.09	57.09	55.27
20	54.18	**68.72**	65.81	61.27	58.18
50	59.63	**68.54**	63.45	61.81	57.09

⇒ the new models 'baseline+1-,2-,3-grams' outperform the 'LM-INTERP' models

Conclusions

- our approach based on the word similarity to
  add new n-grams in a language model is efficient
- adding n-grams of new words provides
  an absolute improvement of **1.3%** on the WER
  and allows to correctly recognize **69%** of new words
- the new language models outperform the interpolated models

Thank you
for your attention!