Refactored Moses

Hieu Hoang

MT Marathon

Prague

August 2013

What did you Refactor?

- Decoder
  - Feature Function Framework
    - moses.ini format
  - Simplify class structure
  - Deleted functionality
  - NOT decoding algorithms
- Tuning
  - NOT tuning algorithm
- EMS (Experiment Management System)

Why did you Refactor?

- Feature Function Framework
  - easier to implement new features
  - use sparse features
- Simplify class structure
  - easier to develop with Moses
- Delete functionality
  - easier to refactor code
  - very little deletion

Why did you Refactor?

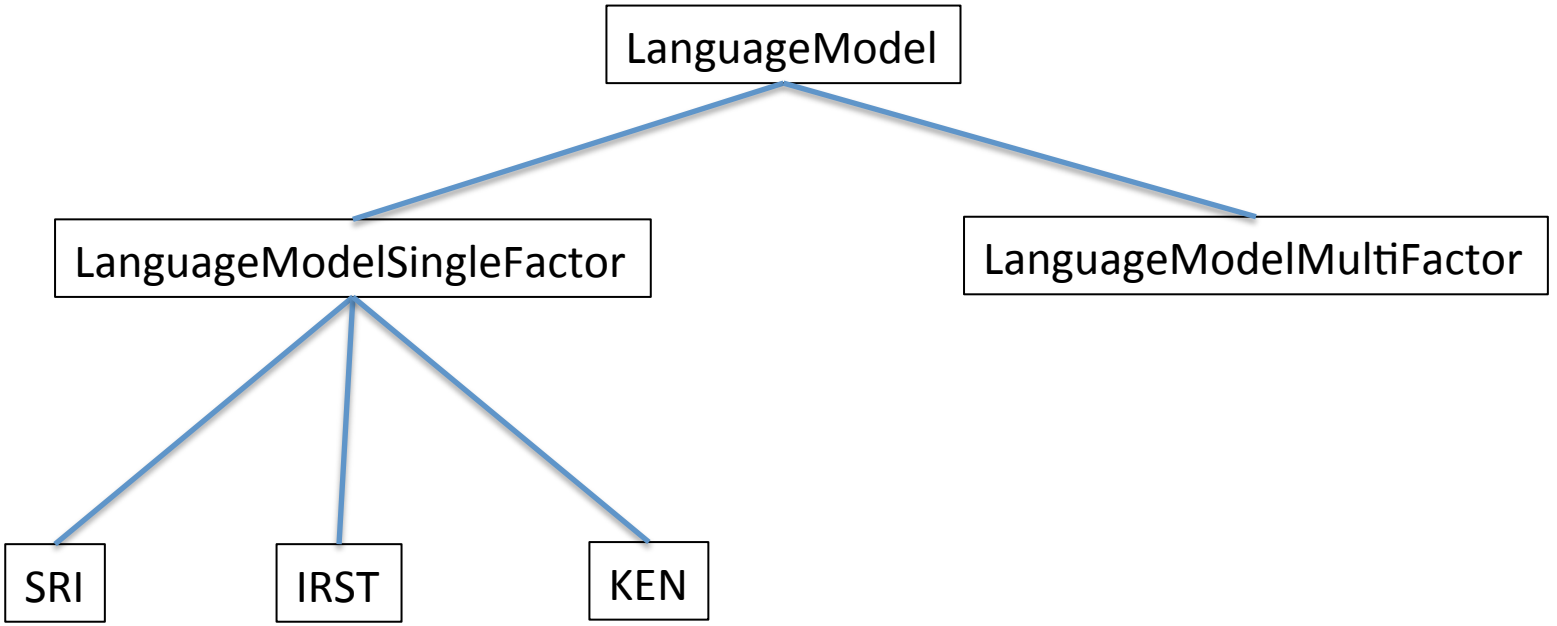
- Feature Function Framework
  - easier to implement new features
  - use sparse features
- Simplify class structure
  - easier to develop with Moses
- Delete functionality
  - easier to refactor code
  - very little deletion

Why did you Refactor?

- Feature Function Framework
  - easier to implement new features
  - use sparse features
- Simplify class structure
  - easier to develop with Moses
- Delete functionality
  - easier to refactor code
  - very little deletion

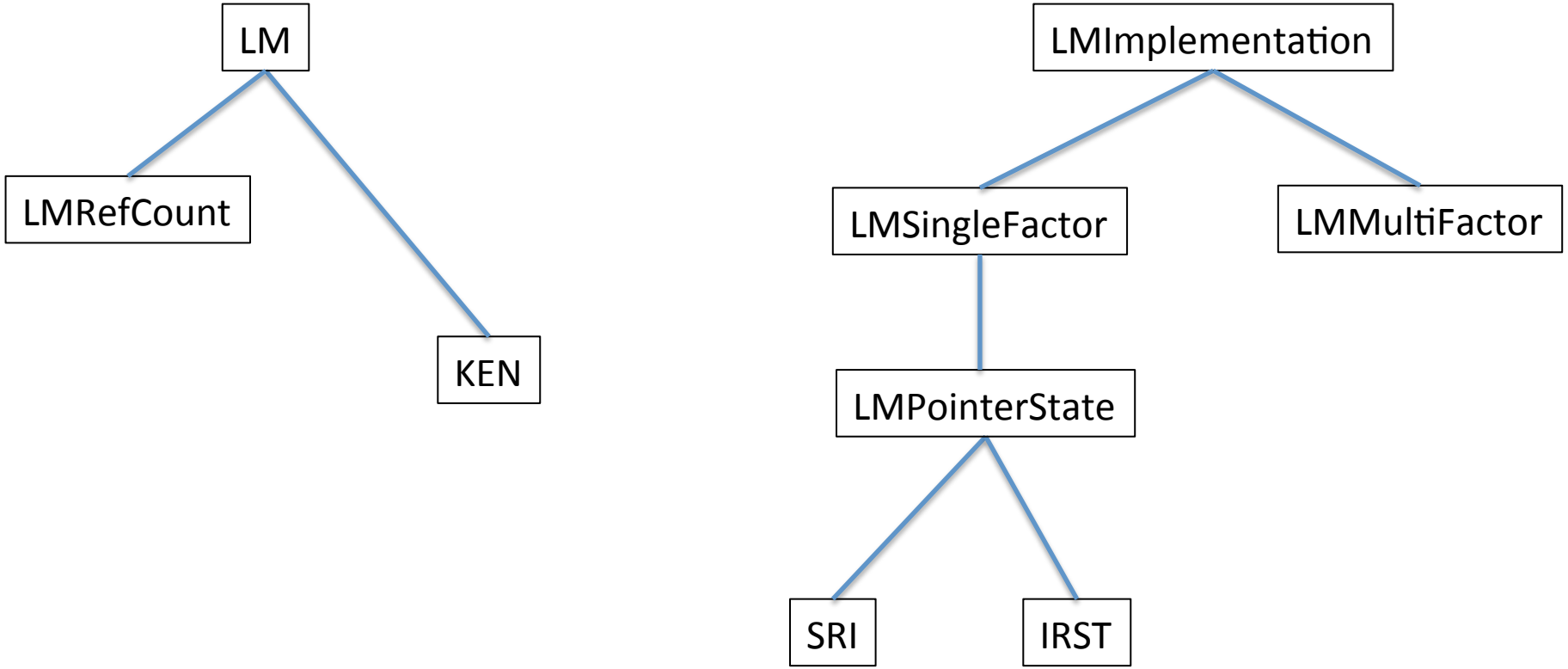
Language Model

IN THE BEGINNING



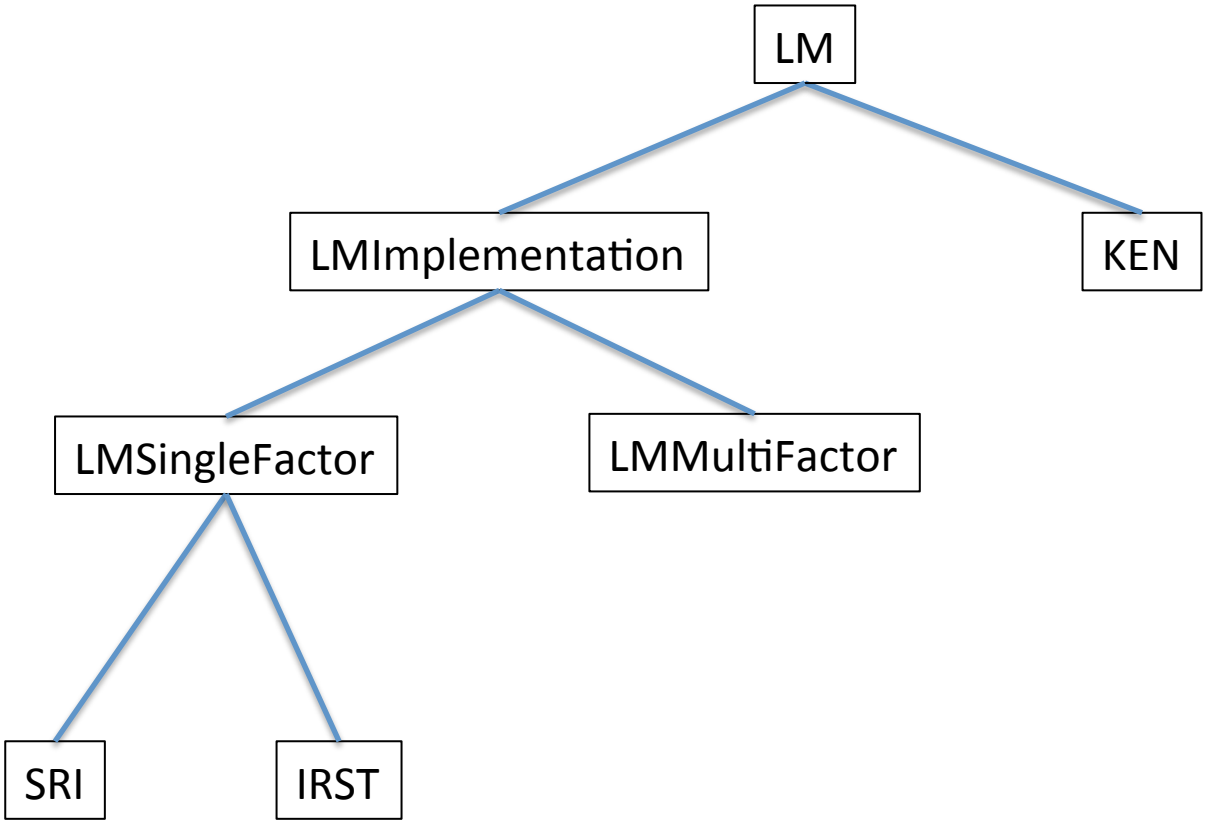
Language Model

THEN….



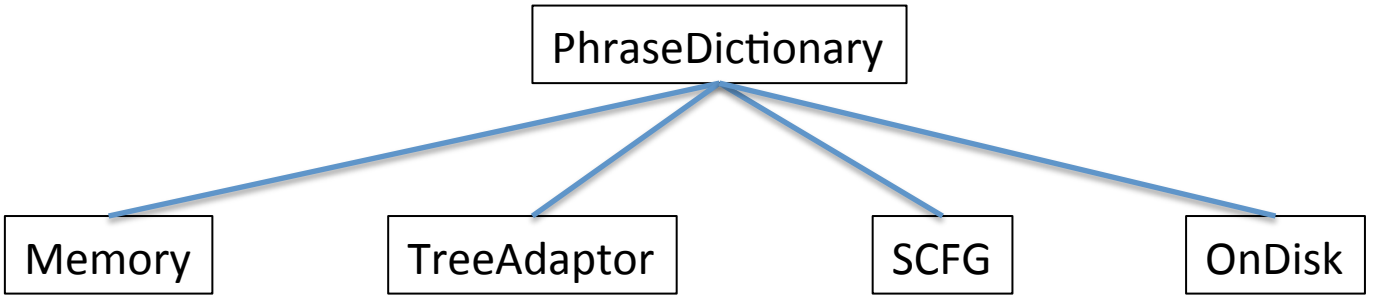
Language Model

NOW



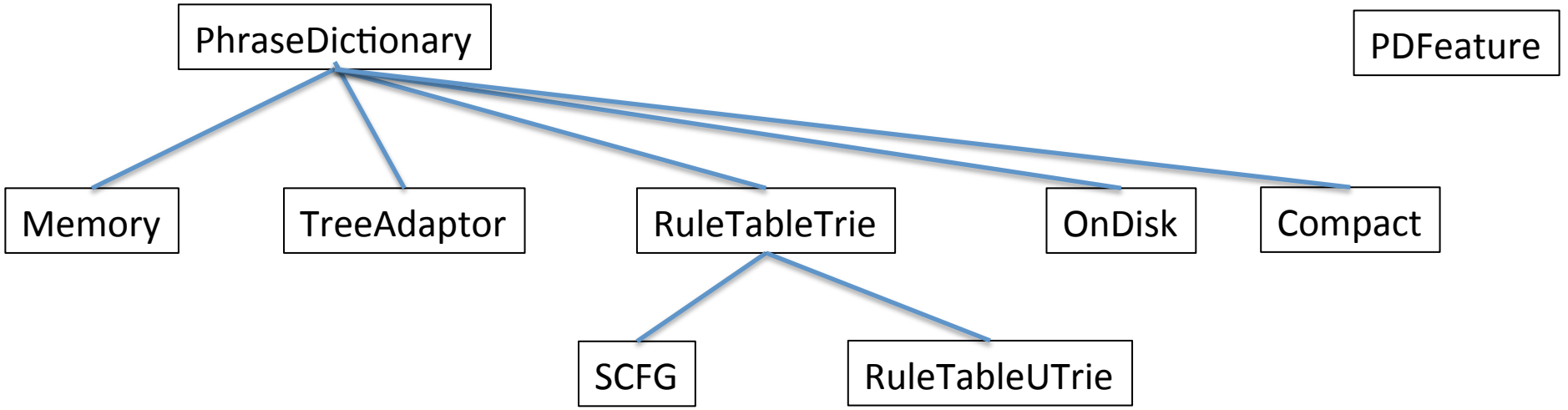
Phrase Tables

IN THE BEGINNING



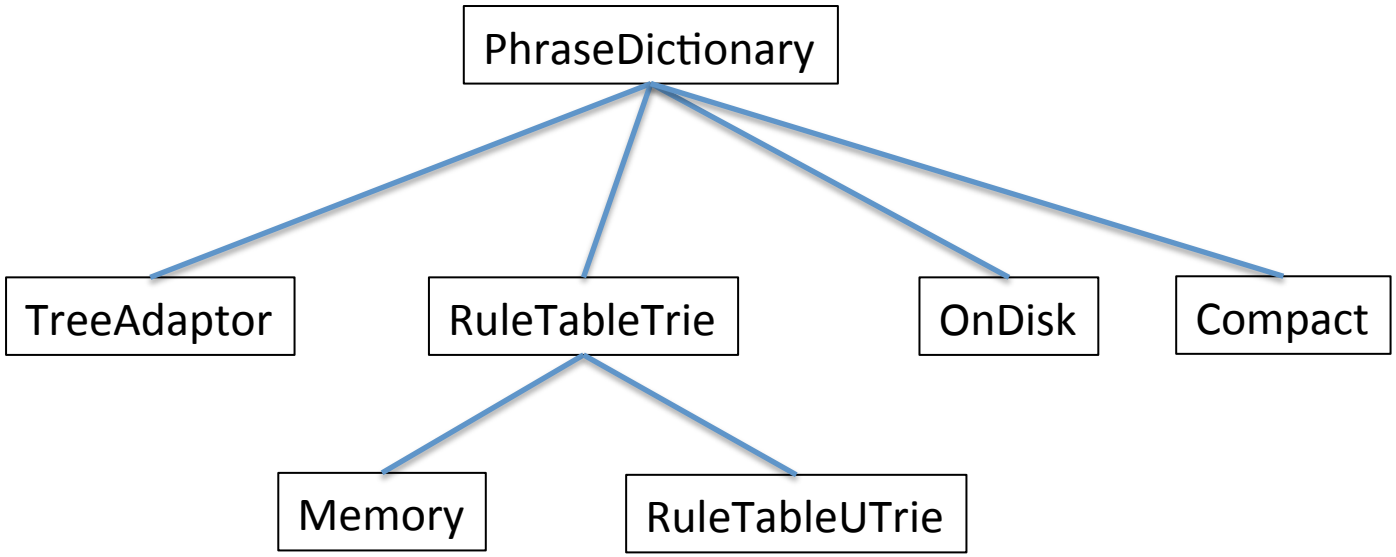
Phrase Tables

THEN….



Phrase Tables

NOW



Why did you Refactor?

- Feature Function Framework
  - easier to implement new features
  - use sparse features
- Simplify class structure
  - easier to develop with Moses
- Delete functionality
  - easier to refactor code
  - very little deletion

Deleted

1. Translation Systems
   - multiple engines in 1 decoding process
   - replaced with alternative weights function
2. Distributed Language Model
   - send LM queries to different machines
   - replace with multipass/asynchronous decoding?
3. Continue Partial Translation
   - start from non-empty hypothesis

Why did you Refactor?

- Feature Function Framework
  - easier to implement new features
  - use sparse features
- Simplify class structure
  - easier to develop with Moses
- Delete functionality
  - easier to refactor code
  - very little deletion

Adding new Feature Function

THEN….

- Add entry to
  - Parameter class
- Read parameter in StaticData
  - add variable to hold new feature
  - load feature

Bespoke

ini file:

```
[new-feature-section]
0 0 1-1 file.name

[weight-new-feature]
```

Adding new Feature Function

Now….

- StaticData
  - Simple framework to load feature

ini file:

```
[feature]
WordPenalty
Distortion
…
NEW-FEATURE file=path factor=0

[weight]
WordPenalty0= -0.27
Distortion0= 0.14
…
NEW-FEATURE0= 0.54
```

MERT

THEN….

- Only know about certain feature functions
  - Hardcoded array

```
my @ABBR_FULL_MAP = qw(d=weight-d lm=weight-l tm=weight-t w=weight-w
  g=weight-generation lex=weight-lex I=weight-i dlm=weight-dlm pp=weight-pp
wt=weight-wt pb=weight-pb lex=weight-lex glm=weight-glm);
```

  - requires feature name and abbreviation

MERT

THEN….

- Only know about certain feature functions
  - Hardcoded array

```
my @ABBR_FULL_MAP = qw(d=weight-d lm=weight-l tm=weight-t w=weight-w
  g=weight-generation lex=weight-lex I=weight-i dlm=weight-dlm pp=weight-pp
wt=weight-wt pb=weight-pb lex=weight-lex glm=weight-glm);
```

  - requires feature name and abbreviation

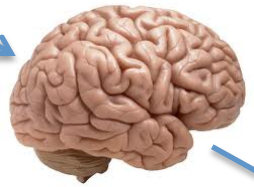
- Deleted array
  - Ask decoder for feature function
  - abbreviations deleted

Timeline of a Translation Rule

File

Load

Memory



Apply to input sentence

Translation Option

Search

Hypothesis

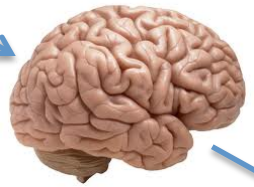
Timeline of a Translation Rule

File

Load

Source phrase
Target phrase

Memory



Apply to input sentence

Input sentence
Input path

Translation Option

Search

Translation context
Segmentation

Hypothesis

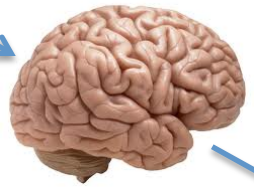
Timeline of a Translation Rule

File

Load

Once

Memory



Apply to input sentence

Per occurrence in sentence

Translation Option

Search

Per hypothesis

Hypothesis

Feature Function API

Loading

File

Memory

X→ je suis X_1 ||| I am X_1

Access to:

Source phrase: je suis X_1

Target phrase: I am X_1

```
void Evaluate(source,
              target,
              scores,
              estimated future scores)
```

Feature functions that use this:

- Word Penalty
- Phrase penalty
- Language model (partial)

Feature Function API

Apply to input sentence

Memory

Translation Option

Input: je|PRO suis|VB 25|NUM ans|NNS .|.

Access to:

Input subphrase: je|PRO suis|VB 25|NUM

Input scores: 0.3 0.2 0.1

```
void Evaluate(input,
              input path,
              scores)
```

Feature functions that uses this:

Input feature

Bag-of-word features….

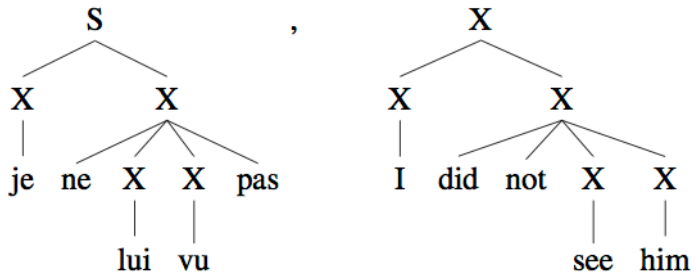
Feature Function API

Search

Translation Option

Hypothesis

Hypothesis:



Access to:

Current hypothesis
Previous hypotheses
Segmentation

Stateless features:

```
void Evaluate(hypo, scores)
void EvaluateChart(hypo, scores)
```

Stateful features:

```
state Evaluate(hypo,
               previous state,
               scores)
state EvaluateChart(hypo
                    previous state,
                    scores)
```

Feature Function API

Decoding

Translation Option

Hypothesis

Feature functions that uses this:

- All stateful features
  - Language models
  - Distortion model
  - Lexicalized distortion
  - …
- Some stateless features
  - Global lexicalized model
  - Word translation

Feature Function

Loading:

```
void Evaluate(source,
              target,
              scores,
              estimated future scores)
```

Apply to Input:

```
void Evaluate(input,
              input path,
              scores)
```

Search:

Stateless features:

```
void Evaluate(hypo, scores)
void EvaluateChart(hypo, scores)
```

Stateful features:

```
state Evaluate(hypo,
               previous state,
               scores)
state EvaluateChart(hypo
                    previous state,
                    scores)
```

Strange Features functions (1)

- Language model
  - implement 2 Evaluate()
  1. Loading
     - evaluate full n-grams

       reprise de la session ||| resumption of the session
     - estimate future cost
       - partial n-grams
  2. Decoding
     - evaluate overlapping n-grams

Strange Feature Functions (2)

- Phrase-tables
- Unknown Word Penalty
- Generation Model
  - integral part of decoding process
  - Uses no Evaluate()
    - scores assign by decoder

Register a Feature Function

- Register
  - in moses/FF/Factory.cpp
  - add entry
    - MOSES_FNAME(ClassName);

Language Model

- Inherit from
  - LanguageModelSingleFactor

```
class LanguageModelSingleFactor : …
{
  LMResult GetValue(
       const std::vector<const Word*> &contextFactor,
       State* finalState = NULL) const = 0;
}
```

Phrase Table

- Inherit from
  - PhraseDictionary
- Legacy API:

```
class PhraseDictionary :  . . .
{
public:
  TargetPhraseCollection *GetTargetPhraseCollectionLEGACY(
                Phrase &src) const;

protected:
   const TargetPhraseCollection
         *GetTargetPhraseCollectionNonCacheLEGACY(Phrase &src);

}
```

Phrase Table

- New API:

```
class PhraseDictionary : . . .
{
public:
  void GetTargetPhraseCollectionBatch(InputPathList &);
}
```

- InputPathList

Input Sentence: je suis 25 ans .

Input Path List:

je
je suis
je suis 25
…
suis
suis 25
…

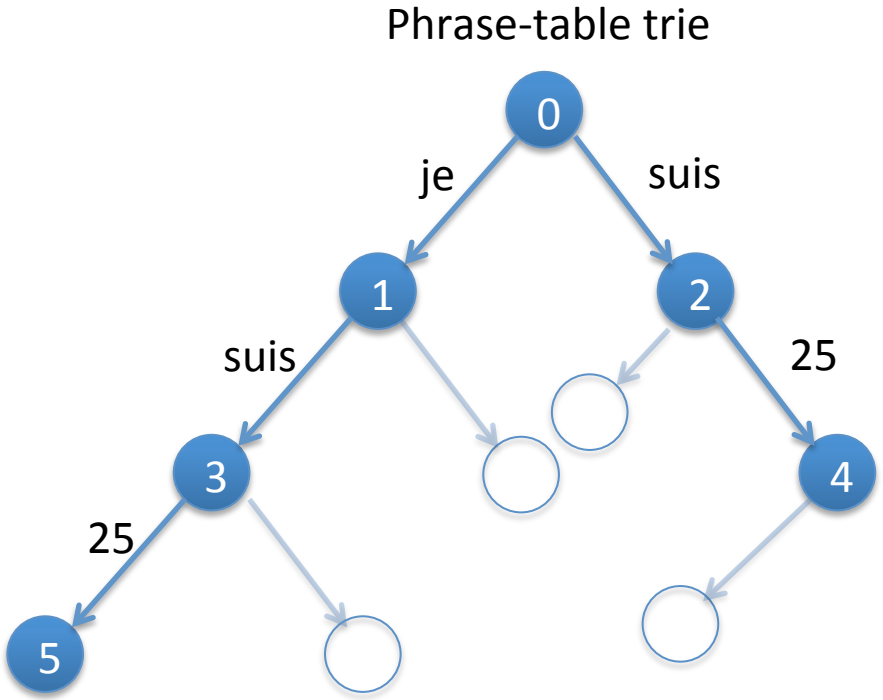
Phrase-table

Optimization

Input Sentence: je suis 25 ans .

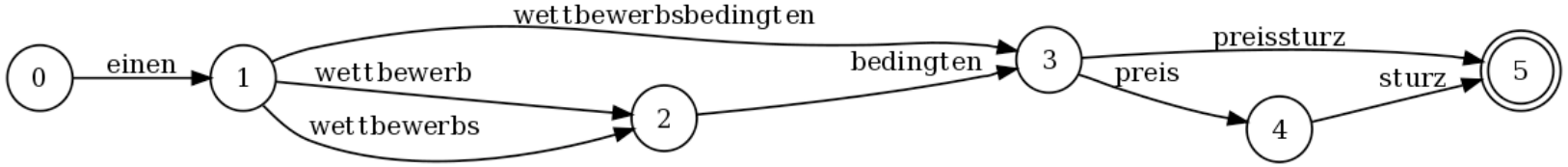
Input Path List:

je
je suis
je suis 25
...
suis
suis 25
...



Phrase-table

Lattice Input



Input Path List:

einen

einen wettbewerb
einen wettbewerbs
einen wettbewerbsbedingten

einen wettbewerb bedingten
einen wettbewerbs bedingten

einen wettbewerb bedingten preissturz
einen wettbewerbs bedingten preis
einen wettbewerbsbedingten preissturz
einen wettbewerbsbedingten preis
….

Phrase Table

Syntax Decoding

- New API:

```
class PhraseDictionary : . . .
{
  virtual ChartRuleLookupManager *CreateRuleLookupManager(
              const ChartParser &,
              const ChartCellCollectionBase &) = 0;
}
```

  - Active Parsing
  - CKY+

```
class ChartRuleLookupManager
{
   void GetChartRuleCollection(
              const WordsRange &range,
              ChartParserCallback &outColl) = 0;
}
```

Results

- Lattice decoding
  - use all phrase-table implementations
  - Syntax model (nearly…)
- One in-memory phrase-table implementation
- One on-disk phrase-table
  - Syntax models
  - AND phrase-based model
  - Zen's binary implementation
    - phrase-based only
    - deprecated – to be removed

Results

Translation Quality

- Pass regression tests
  - tests have also changed…
- BLEU unchanged
  - subject to random variability

	#	Description	v 1.0	29th July
En-es	1	pb	24.81	24.59
	2	hiero	24.2	24.2
	3	(1) + CreateOnDiskPt		24.58
Es-en	1	pb	23.01	23
	2	hiero	22.37	22.32
	3	(1) + CreateOnDiskPt		23.03
En-cs	1	pb	11.04	11.05
	2	hiero	10.93	10.9
	3	(1) + placeholders		9.23
	4	(1) + CreateOnDiskPt		11.19
	5	(2) + Ken's incre algo		10.84
Cs-en	1	pb	15.72	15.8
	2	hiero	15.68	15.66
	3	(1) + placeholders		14.01
	4	(1) + CreateOnDiskPt		15.66
	5	(2) + Ken's incre algo		15.44
En-de	1	pb	11.87	11.7
	2	hiero	11.62	11.58
	3	(1) + POS de	11.67	11.65
	4	(3) + POS en	11.75	11.65
	5	(1) + CreateOnDiskPt	15.75	11.62

	#	Description	v 1.0	29th July
De-en	1	pb	15.75	15.81
	2	hiero	15.53	15.71
	3	(1) + POS de	15.84	15.9
	4	(3) + POS en	15.93	15.78
	5	(1) + CreateOnDiskPt		15.91
	6	(3) + CreateOnDiskPt		15.7
	7	(4) + CreateOnDiskPt		15.78
En-fr	1	pb truecase	24.38	24.38
	2	pb recase	22.94	22.81
	3	(2) + hiero	22.28	22.36
	4	(2) + POS en	23.05	23.07
	5	(2) + kbmira		22.85
	6	(2) + pro		22.64
	7	(2) + CreateOnDiskPt		22.65
Fr-en	1	pb truecase	24.06	23.97
	2	pb recased	22.41	22.43
	3	(2) + hiero	18.05	17.75
	4	(2) + factors	22.55	22.47
	5	(2) + kbmira		22.44
	6	(2) + pro		22.49
	7	(2) + CreateOnDiskPt		22.46

Results

Speed

Phrase-based

		speed (sec)	speed (sec)	
1	v 1.0	784	857.12	
2	master + caching	1,154	1317.07	**-54%**
With compact pt:				
3	v1		840.57	**2%**
4	master (with caching)		934.82	**-9%**

Hierarchical

Pop Limit	**100**	**200**	**500**	**1000**	**Loading**
v 1.0	1,044	1,393	2,620	4,546	115
13th Aug	670	1034	2064	3743	115
Reduction (excl load)	**-65%**	**-26%**	**-21%**	**-18%**	

Comparison with the decoders