Feature Functions

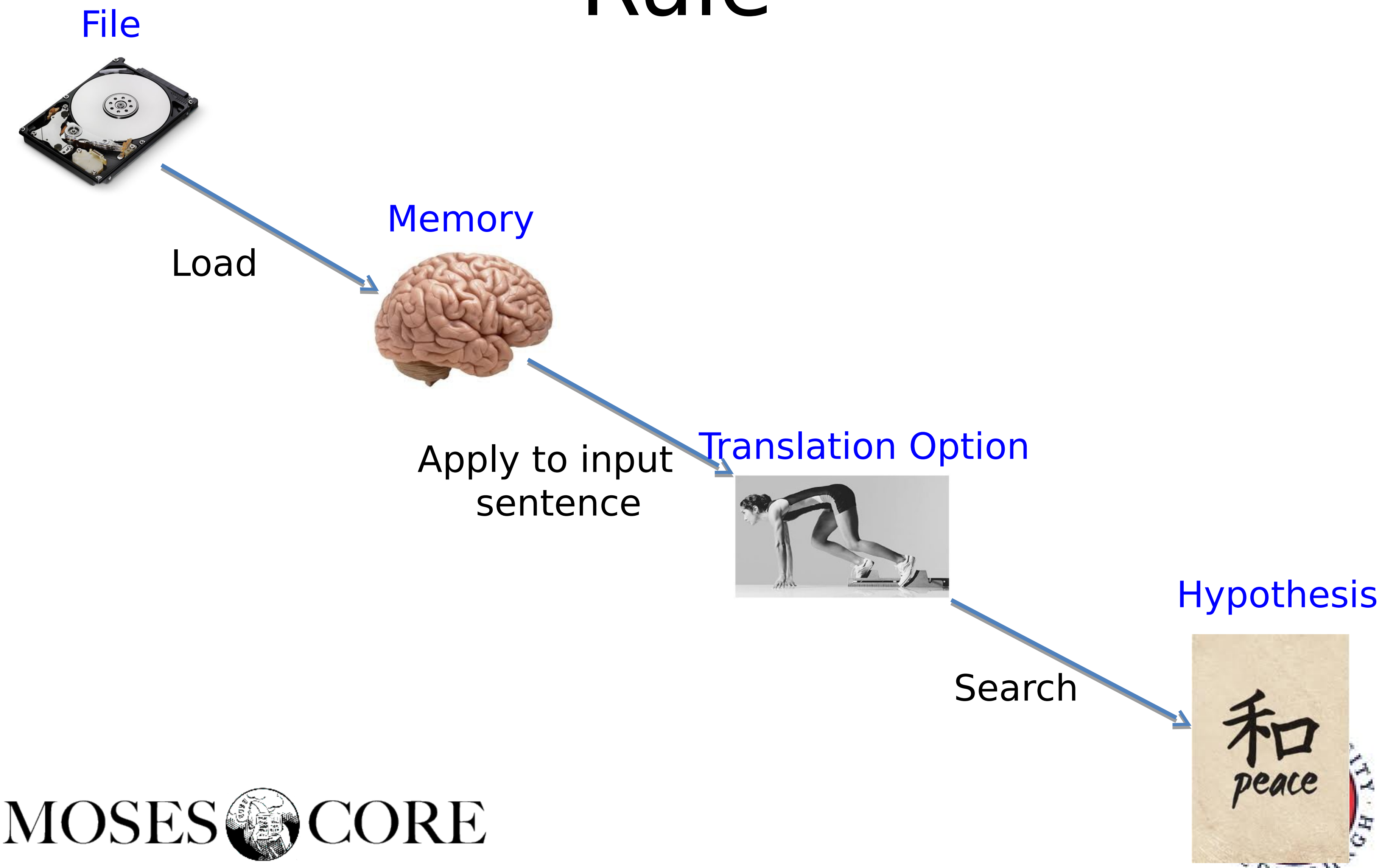
Hieu Hoang
Matthias Huck

December 2014

Feature Function

- Examples
  - Phrase table
  - Language model
  - Word penalty
  - Phrase penalty
- Gives a score to a translation
  - Partial translation
  - Completed translation
- Many feature functions
  - Weighted linear combination
- What is a translation?
  - Made of multiple ***translation rules***

Timeline of a Translation Rule



File

Load

Memory

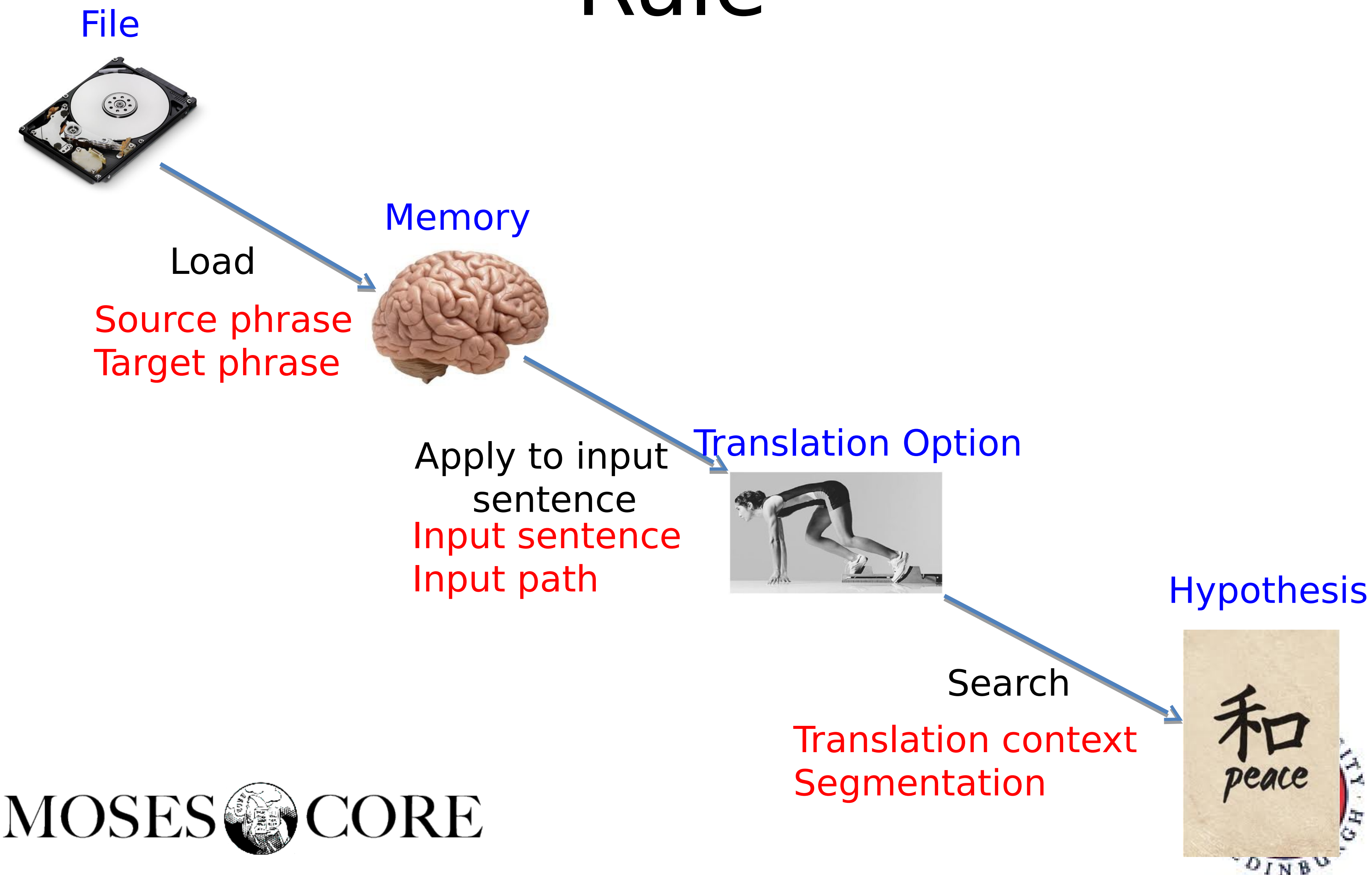
Apply to input sentence

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

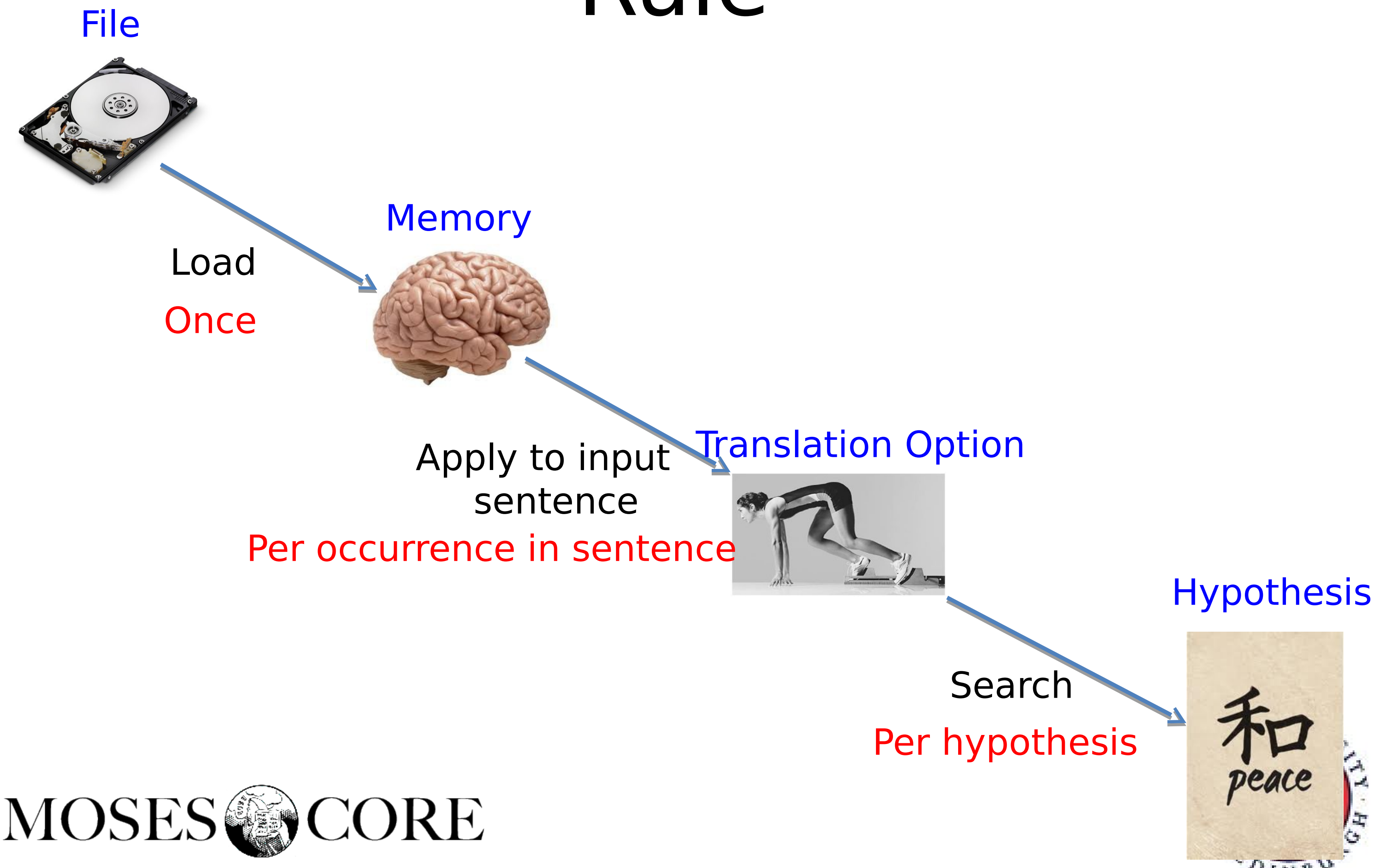
Search

Translation context
Segmentation

Hypothesis

MOSES CORE

Timeline of a Translation Rule



File

Load

Once

Memory

Apply to input sentence

Per occurrence in sentence

Translation Option

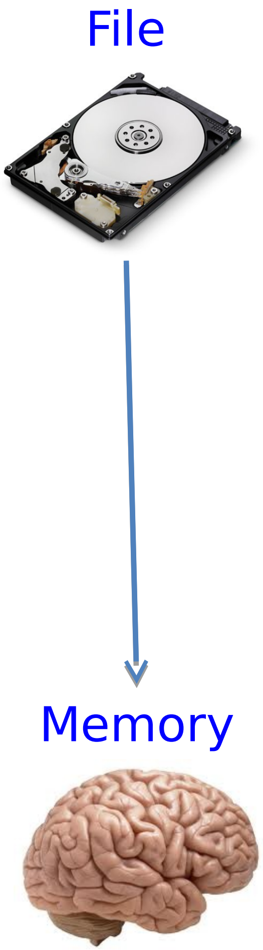
Search

Per hypothesis

Hypothesis

Feature Function API

Loading



je suis ||| I am

Access to: Source phrase: je suis

Target phrase: I am

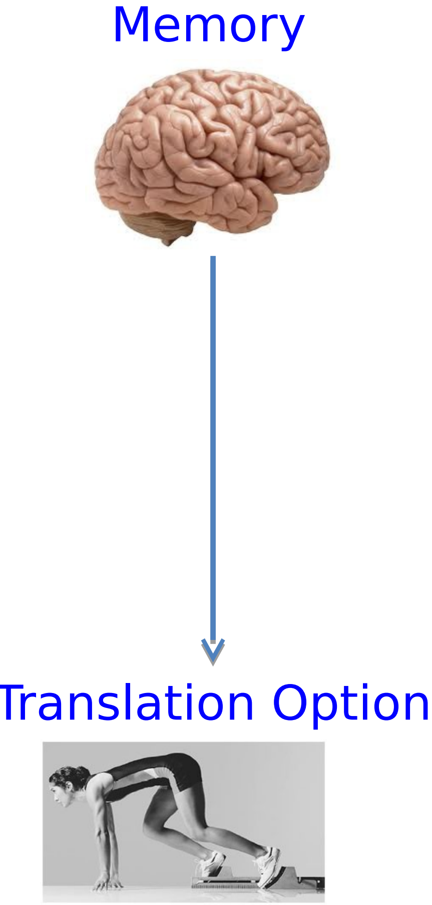
```
void EvaluateInIsolation(
            source,
            target,
            scores,
            estimated future scores)
```

Feature functions that use this:

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

Feature Function API

Apply to input sentence



Memory

Translation Option

Access to: Input sentence: je suis 25 ans .

Input subphrase: je suis 25

```
void EvaluateWithSourceContext(
           input,
          input path,
          target,
           scores,
           estimated future scores)
```

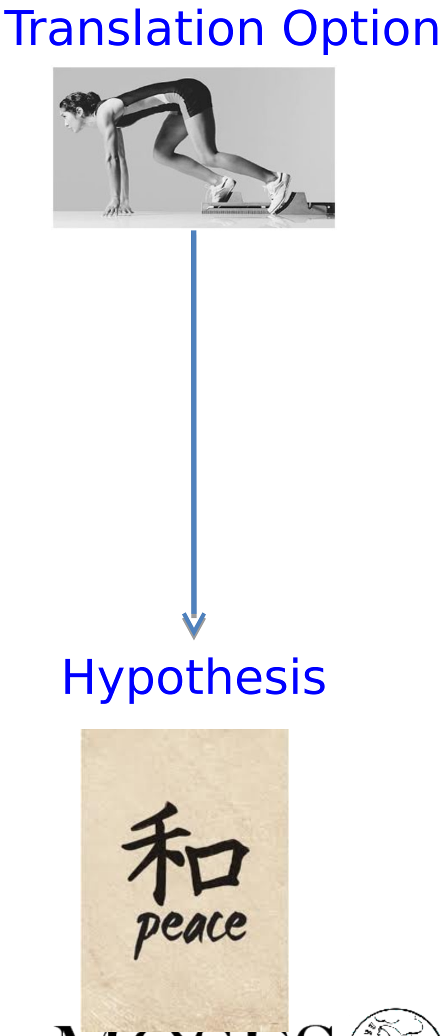
Feature functions that use this:

Input feature

Bag-of-word features….

Feature Function API

Search



Translation Option

Hypothesis

Access to: Current rule (hypothesis)
Previous rules
Segmentation

Stateful features:

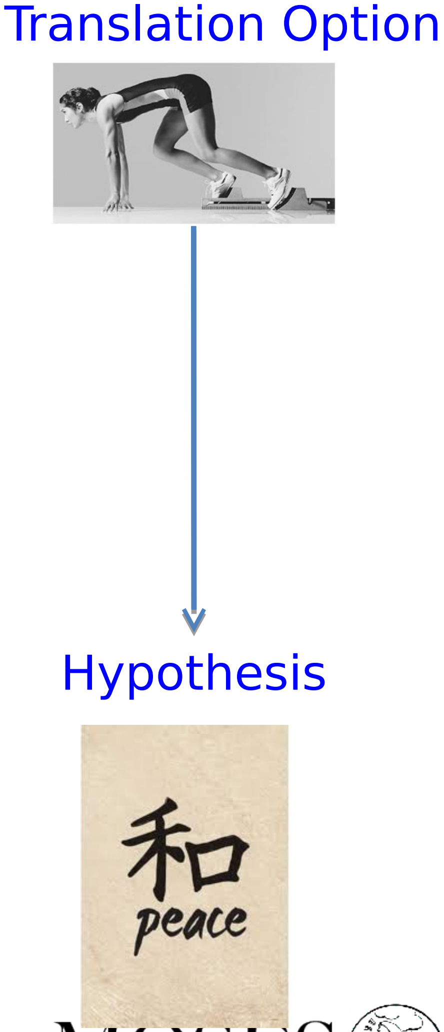
```
   state EvaluateWhenApplied(hypo,
          previous state,
          scores)
   state EvaluateWhenApplied(hypo
            previous state,
            scores)
```

Stateless features:

```
   void EvaluateWhenApplied(hypo, scores)
   void EvaluateWhenApplied(hypo, scores)
```

Feature Function API

Decoding


Translation Option

Hypothesis

Feature functions that use this:

- All stateful features
  - Language models
  - Distortion model
  - Lexicalized reordering model
  - …

Feature Function

Loading:

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

Apply to Input:

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

Search:

Stateful features:

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

Stateless features:

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

Strange Feature 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. Search
   - 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

Adding a New Feature Function

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

Extensions

- Change input
  - Add/delete word
  - Integrate parser/tagger
- Prune
  - Hard constraint
  - Negative infinity score

Properties

- Vanilla translation rule

```
je suis  ||| I am ||| 0.1 0.2 0.3 0.4
```

- With properties

```
je suis  ||| I am ||| 0.1 0.2 0.3 0.4||| ||| |||
                      ||| {{Key1 Value1}} {{Key2 Value2}}
```

- Example properties
  - Syntax structure
  - Document context
  - Orientation

Properties

- Implement class
  - Inherit from PhraseProperty
  - Override method
    - ProcessValue(string)
  - Register property
    - MOSES_PNAME2("KeyName", Property class);
- Use property
  - Inside feature function Evaluate()
  - targetPhrase.GetProperty("KeyName");