Combining Machine Translation Output with Open Source

Kenneth Heafield

Language Technologies Institute
Carnegie Mellon University

January 26, 2010

Packages

MEMT **Core system combination code**

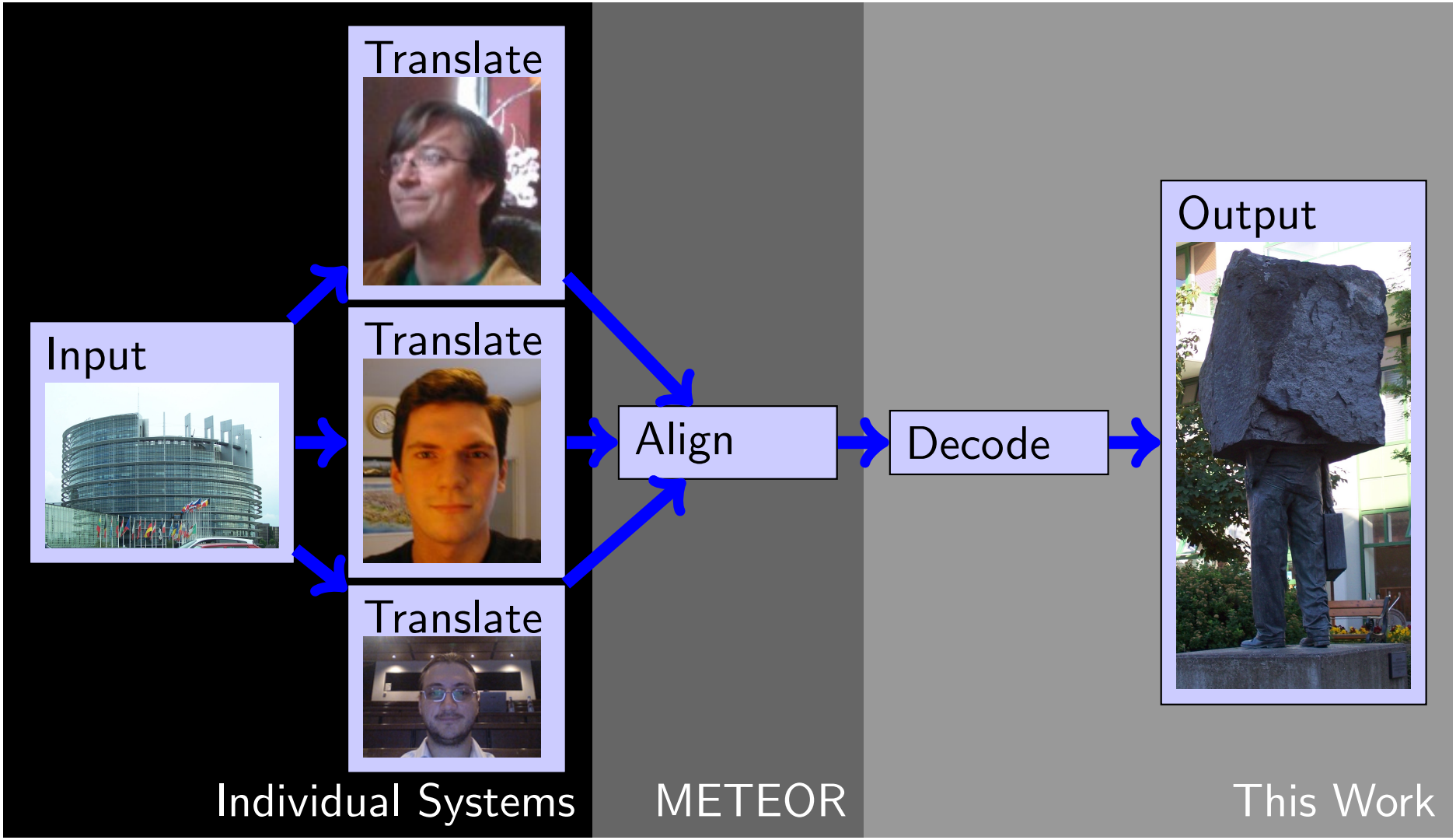
Filter Tighter language model filtering

Score Run several metrics against flat files

Download

- `http://kheafield.com/code/mt/`
- Lesser GNU Public License

Pipeline


Translate
Input
Translate
Translate
Align
Decode
Output
Individual Systems
METEOR
This Work

Arabic-English Example Combination

System 1: So even if that was meaningful , it is because you were late

System 2: Even if feasible , it is because you have been delayed

↓ Combine

Combined: Even if feasible , it is because you were late

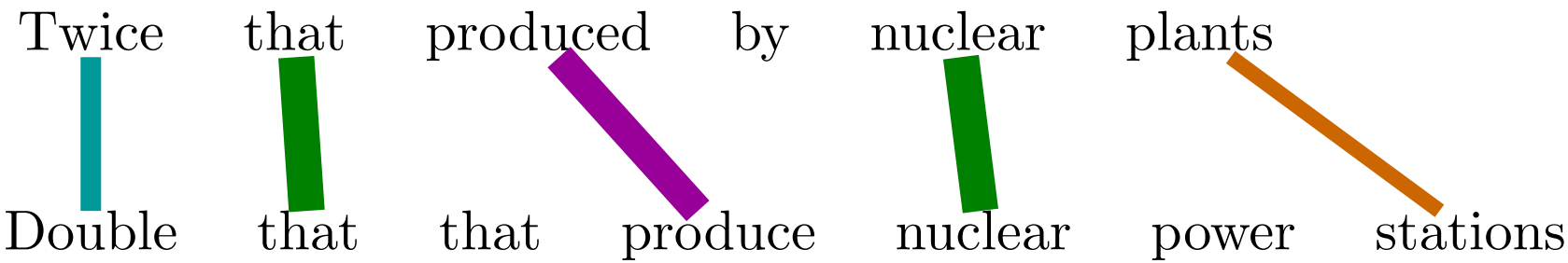
≠ Compare

Reference: And even if that was useful , it was because you were late

Outline

Sentence Pair Alignment

Match surface, stems, WordNet synsets, and TERp unigrams
Minimize crossing alignments



Lavie and Agarwal, METEOR: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments, WMT 2007.

Search Space

Algorithm

Start at the beginning of each sentence

Branch by appending the first unused word from a system

Example

System 1: Now can know why .

System 2: Now we can now know why .

↓ Partial Hypothesis

{ Now
  Now

Search Space

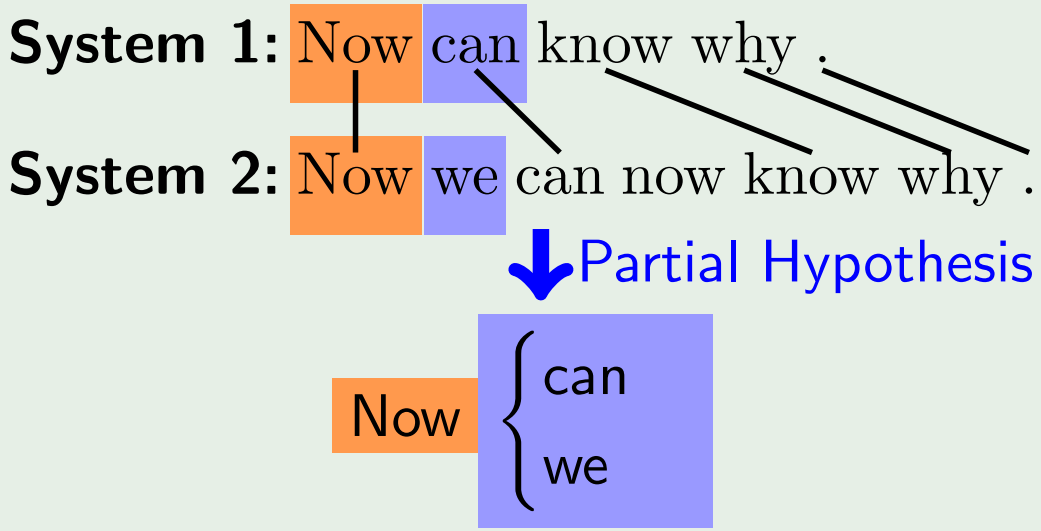
Algorithm

Start at the beginning of each sentence

Branch by appending the first unused word from a system

Use the appended word and those aligned with it

Example



Search Space

Algorithm

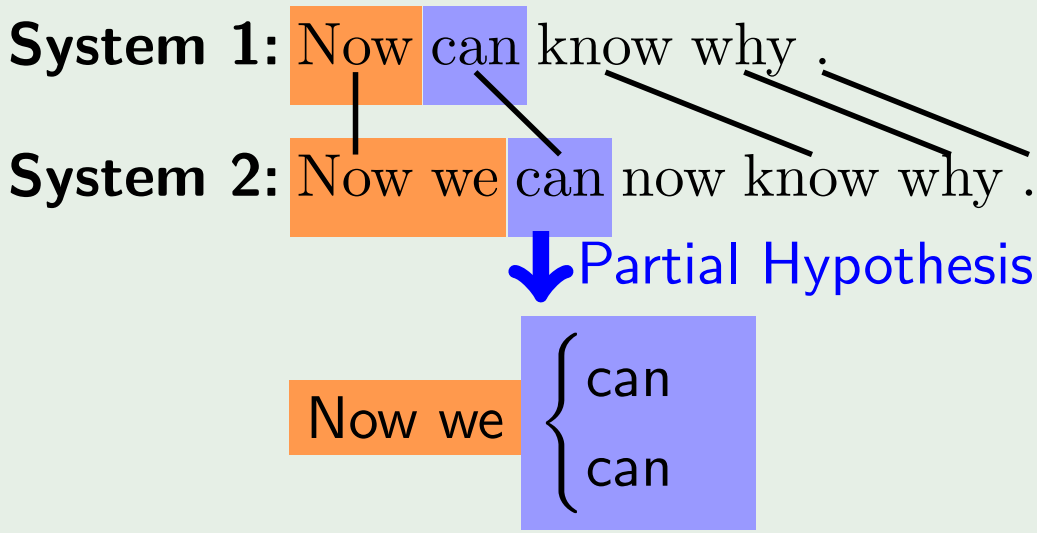
Start at the beginning of each sentence

Branch by appending the first unused word from a system

Use the appended word and those aligned with it

Loop until all hypotheses reach end of sentence

Example



Search Space

Algorithm

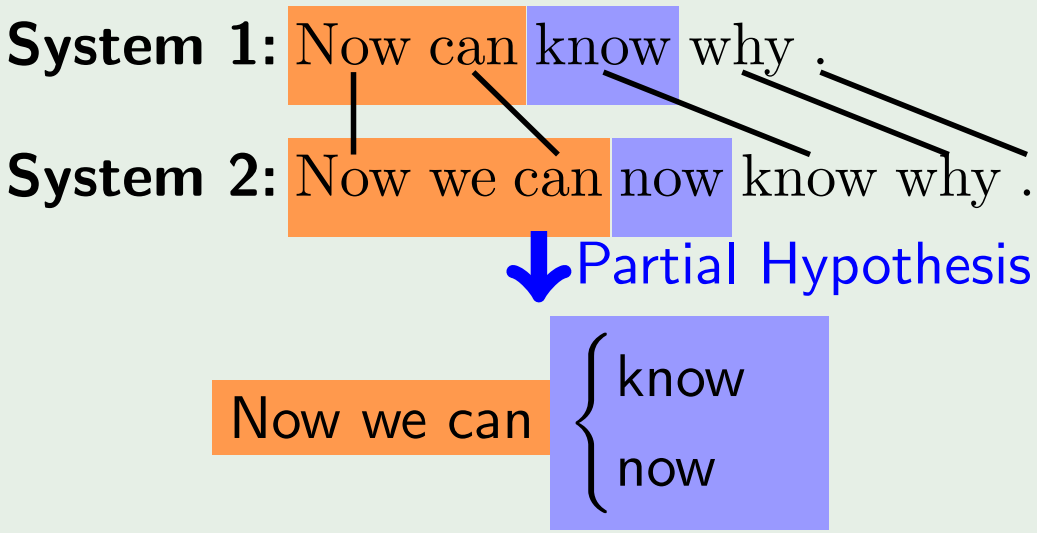
Start at the beginning of each sentence

Branch by appending the first unused word from a system

Use the appended word and those aligned with it

Loop until all hypotheses reach end of sentence

Example



Outline

Features

Length

Length of hypothesis

Language Model

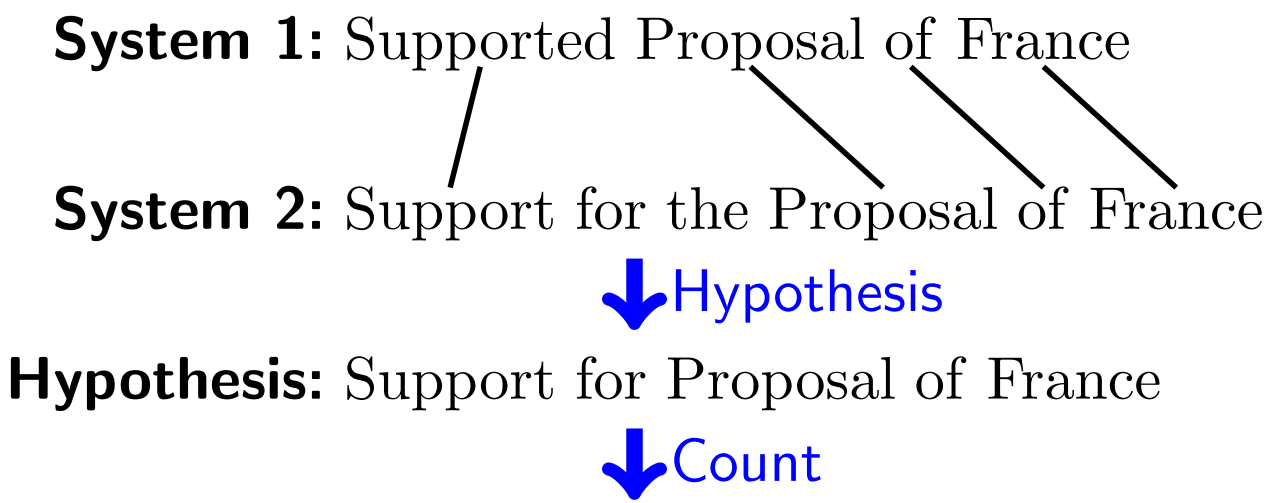
Model: log probability from an ARPA language model

n-**Gram:** average length *n*-gram found in model

Match

Count of *n*-grams matching each system

Match Features



	Unigram	Bigram	Trigram	Quadgram
System 1	4	2	1	0
System 2	5	3	1	0

Match Feature Configuration

Options

`individual`	Longest *n*-gram counted per system
`collective`	Longer *n*-gram length summed over systems
`mask`	Alignment types that count (exact, stem, synonym, terp)

Rationale

- Control feature count
- One copy counts exact alignments for lexical choice
- Another copy counts all alignments for word order

Parameter Tuning

Overall Score

Linear combination of length, language model, and match features

Tuning

- Minimum Error Rate Training
- Wrapper for Z-MERT is included

Outline

Software Requirements

Assumed

- UNIX
- C++, Java, and Python

Install Script Provided

- Boost and Boost Jam
- ICU
- Ruby
- MT evaluation metrics

No root required

Tuning Data

Requirements

- One-best translations from each system
- 400 segments with references
  - Held out from individual systems

Availability

- LoonyBin
- WMT and NIST evaluations

Hardware Requirements and Language Model

Types

- ARPA *n*-gram model; SRI generates these
- Suffix array

ARPA Filtering Options

- Corpus vocabulary
- Per-segment models
- Union of per-segment models

Rationale

Filtered model must fit in RAM

Available as a standalone package

Decoder Design

TCP Server Architecture

- Load resources once
- Clients can configure at the sentence level
- Parallel clients with different configurations
- Output streamed back to clients

Under the Hood

- Multithreaded producer-consumer
- C++ with Boost

Decoding Algorithm

Beam Search

Assemble combined sentence left to right

Recombination

- Detection by hash table over feature states
- Lazy k-best unpacking

Speed

1.1 combinations/second/core with beam size 500

Evaluation

Command

```
$ score.rb --hyp-tok test.txt --refs 1.txt 2.txt
```

Metrics

- BLEU
- NIST
- TER
- METEOR
- Precision
- Recall
- Length

Available as a standalone package

Results

Source	Top	Gain
Arabic	58.55	+6.67
Czech	21.98	+0.80
French	31.56	+0.42
German	23.88	+2.57
Hungarian	13.84	+1.09
Spanish	28.79	+0.10
Urdu	34.72	+1.84

Table: Post-evaluation uncased BLEU gains on NIST and WMT tasks.

Packages

MEMT Core system combination code

Filter Tighter language model filtering

Score Run several metrics against flat files

Download

- `http://kheafield.com/code/mt/`
- Lesser GNU Public License