Beyond Parallel Corpora

Philipp Koehn

presented by Huda Khayrallah

1 November 2018

data and machine learning

Supervised and Unsupervised

- We framed machine translation as a supervised machine learning task
  - training examples with labels
  - here: input sentences with translation
  - structured prediction: output has to be constructed in several steps

Supervised and Unsupervised

- We framed machine translation as a supervised machine learning task
  - training examples with labels
  - here: input sentences with translation
  - structured prediction: output has to be constructed in several steps

- Unsupervised learning
  - training examples without labels
  - here: just sentences in the input language
  - we will also look at using just sentences output language

Supervised and Unsupervised

- We framed machine translation as a supervised machine learning task
  - training examples with labels
  - here: input sentences with translation
  - structured prediction: output has to be constructed in several steps

- Unsupervised learning
  - training examples without labels
  - here: just sentences in the input language
  - we will also look at using just sentences output language

- Semi-supervised learning
  - some labeled training data
  - some unlabeled training data (usually more)

Supervised and Unsupervised

- We framed machine translation as a supervised machine learning task
  - training examples with labels
  - here: input sentences with translation
  - structured prediction: output has to be constructed in several steps

- Unsupervised learning
  - training examples without labels
  - here: just sentences in the input language
  - we will also look at using just sentences output language

- Semi-supervised learning
  - some labeled training data
  - some unlabeled training data (usually more)

- Self-training
  - make predictions on unlabeled training data
  - use predicted labeled as supervised translation data

Transfer Learning

- Learning from data similar to our task

Transfer Learning

- Learning from data similar to our task
- Other language pairs
  - first, train a model on different language pair
  - then, train on the targeted language pair
  - or: train jointly on both

Transfer Learning

- Learning from data similar to our task

- Other language pairs
  - first, train a model on different language pair
  - then, train on the targeted language pair
  - or: train jointly on both

- Multi-Task training
  - train on a related task first
  - e.g., part-of-speeh tagging

- Share some or all of the components

using monolingual data

Using Monolingual Data

- Language model
  - trained on large amounts of target language data
  - better fluency of output
- Key to success of statistical machine translation
- Neural machine translation
  - integrate neural language model into model
  - create artificial data with backtranslation

Adding a Language Model

- Train a separate language model

- Add as conditioning context to the decoder

Adding a Language Model

- Train a separate language model
- Add as conditioning context to the decoder
- Recall state progression in the decoder
  - decoder state s_i
  - embedding of previous output word Ey_{i-1}
  - input context c_i

$$s_i = f(s_{i-1},\ Ey_{i-1}, c_i)$$

Adding a Language Model

- Train a separate language model

- Add as conditioning context to the decoder

- Recall state progression in the decoder
  - decoder state s_i
  - embedding of previous output word Ey_{i-1}
  - input context c_i

$$s_i = f(s_{i-1},\ Ey_{i-1}, c_i)$$

- Add hidden state of neural language model $s_i^{\textsf{LM}}$

$$s_i = f(s_{i-1},\ Ey_{i-1}, c_i, s_i^{\textsf{LM}})$$

Adding a Language Model

- Train a separate language model

- Add as conditioning context to the decoder

- Recall state progression in the decoder
  - decoder state s_i
  - embedding of previous output word Ey_{i-1}
  - input context c_i

$$s_i = f(s_{i-1},\ Ey_{i-1}, c_i)$$

- Add hidden state of neural language model s_i^{LM}

$$s_i = f(s_{i-1},\ Ey_{i-1}, c_i, s_i^{\mathsf{LM}})$$

- Pre-train language model

- Leave its parameters fixed during translation model training

Refinements

- Balance impact of language model vs. translation model

Refinements

- Balance impact of language model vs. translation model
- Learn a scaling factor (gate) $\text{gate}_i^{\mathsf{LM}} = f(s_i^{\mathsf{LM}})$

Refinements

- Balance impact of language model vs. translation model

- Learn a scaling factor (gate) $\text{gate}_i^{\mathsf{LM}} = f(s_i^{\mathsf{LM}})$

- Use it to scale values of language model state

$$\bar{s}_i^{\mathsf{LM}} = \text{gate}_i^{\mathsf{LM}} \times s_i^{\mathsf{LM}}$$

Refinements

- Balance impact of language model vs. translation model
- Learn a scaling factor (gate)

$$\text{gate}_i^{\mathsf{LM}} = f(s_i^{\mathsf{LM}})$$

- Use it to scale values of language model state

$$\bar{s}_i^{\mathsf{LM}} = \text{gate}_i^{\mathsf{LM}} \times s_i^{\mathsf{LM}}$$

- Use this scaled language model state for decoder state

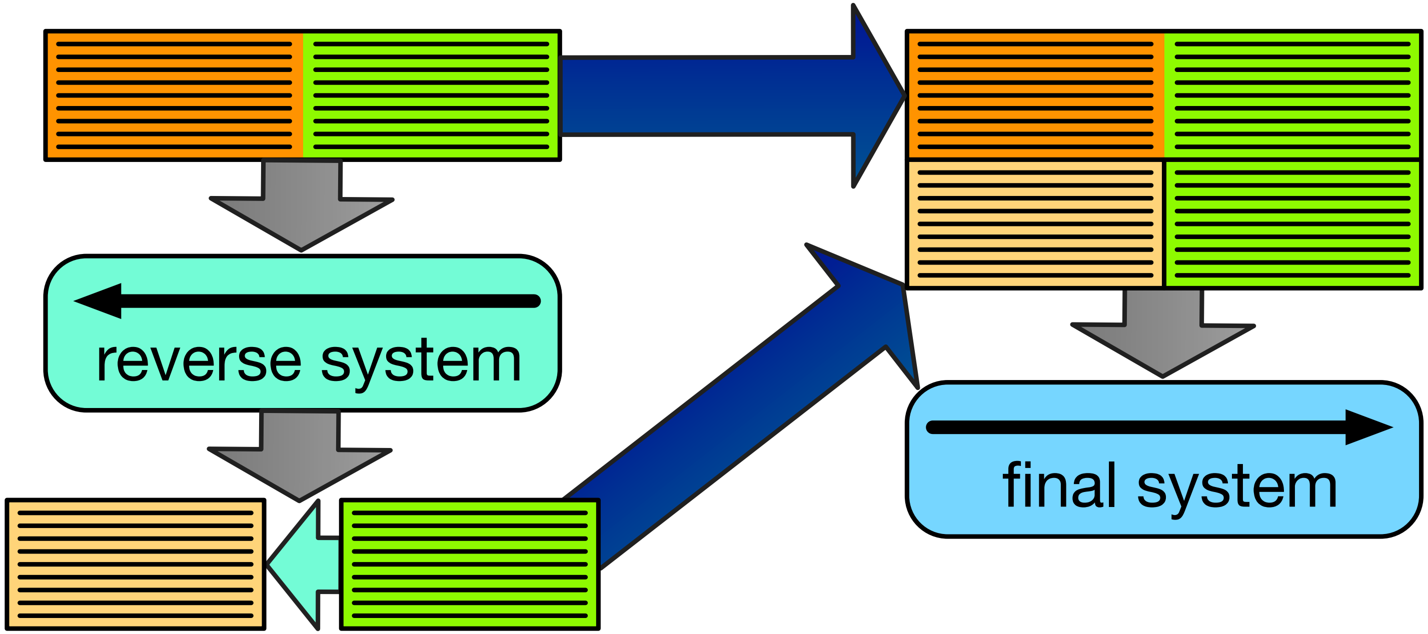
$$s_i = f(s_{i-1},\ Ey_{i-1}, c_i, \bar{s}_i^{\mathsf{LM}})$$

Back Translation

- Monolingual data is parallel data that misses its other half

Back Translation

- Monolingual data is parallel data that misses its other half
- Let’s synthesize that half



Back Translation

- Steps

  1. train a system in reverse language translation
  2. use this system to translate translate target side monolingual data
     → synthetic parallel corpus
  3. combine generated synthetic parallel data with real parallel data to build the final system

- Roughly equal amounts of synthetic and real data

- Useful method of domain adaptation

Iterative Back Translation

- Quality of backtranslation system matters

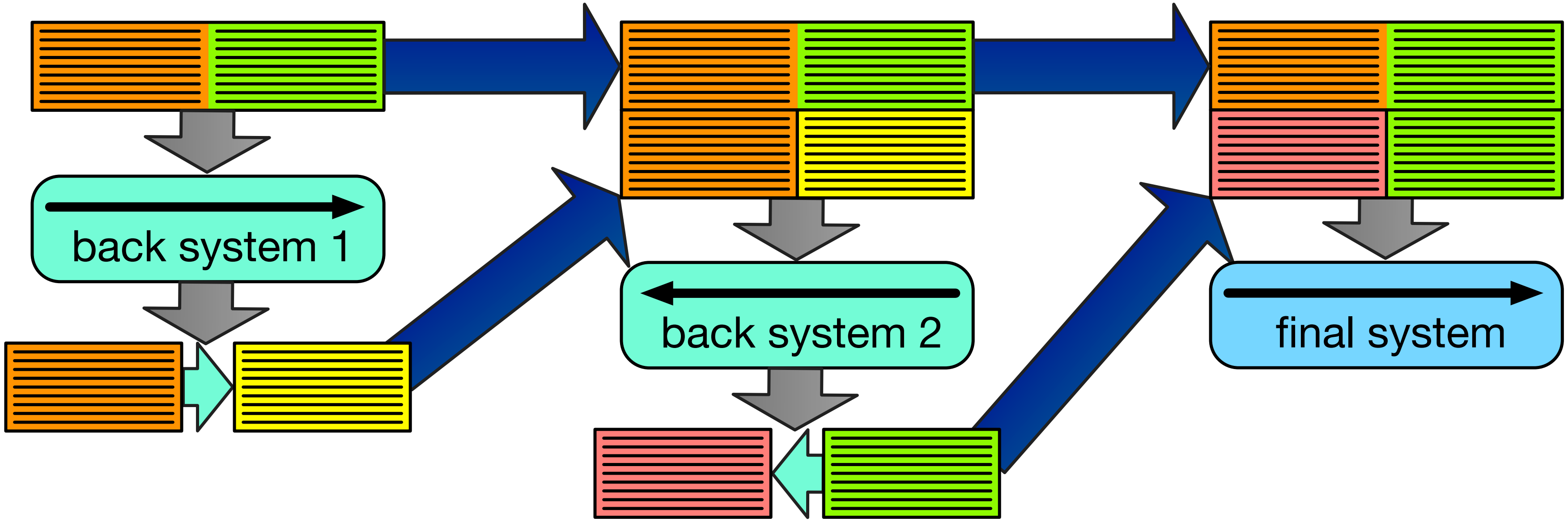
Iterative Back Translation

- Quality of backtranslation system matters
- Build a better backtranslation system ... with backtranslation

Iterative Back Translation

- Quality of backtranslation system matters

- Build a better backtranslation system ... with backtranslation



Iterative Back Translation

- Example

German–English	Back	Final
no back-translation	-	29.6
*10k iterations	10.6	29.6 (+0.0)
*100k iterations	21.0	31.1 (+1.5)
convergence	23.7	32.5 (+2.9)
re-back-translation	27.9	33.6 (+4.0)

* = limited training of back-translation system

Round Trip Training

- We could iterate through steps of
  - train system
  - create synthetic corpus

Round Trip Training

- We could iterate through steps of
  - train system
  - create synthetic corpus

- Dual learning: train models in both directions together
  - translation models $F \rightarrow E$ and $E \rightarrow F$
  - take sentence **f**
  - translate into sentence **e'**
  - translate that back into sentence **f'**
  - training objective: **f** should match **f'**

Round Trip Training

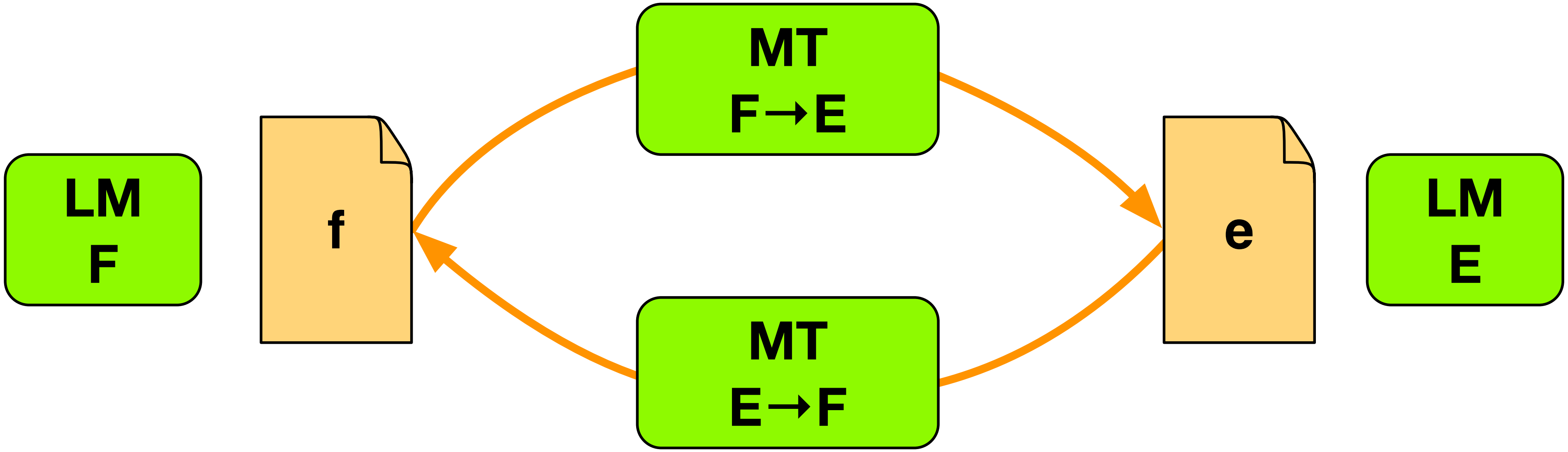
- We could iterate through steps of
  - train system
  - create synthetic corpus

- Dual learning: train models in both directions together
  - translation models $F \rightarrow E$ and $E \rightarrow F$
  - take sentence **f**
  - translate into sentence **e'**
  - translate that back into sentence **f'**
  - training objective: **f** should match **f'**

- Setup could be fooled by just copying (**e'** = **f**)

  ⇒ score **e'** with a language for language E
  add language model score as cost to training objective

Round Trip Training



multiple language pairs

Multiple Language Pairs

- There are more than two languages in the world
- We may want to build systems for many language pairs
- Typical: train separate models for each
- Joint training

Multiple Input Languages

- Example
  - German–English
  - French–English
- Concatenate training data
- Joint model benefits from exposure to more English data
- Shown beneficial in low resource conditions
- Do input languages have to be related? (maybe not)

Multiple Output Languages

- Example
  - French–English
  - French–Spanish

- Concatenate training data

- Given a French input sentence, how specify output language?

Multiple Output Languages

- Example
  - French–English
  - French–Spanish

- Concatenate training data

- Given a French input sentence, how specify output language?

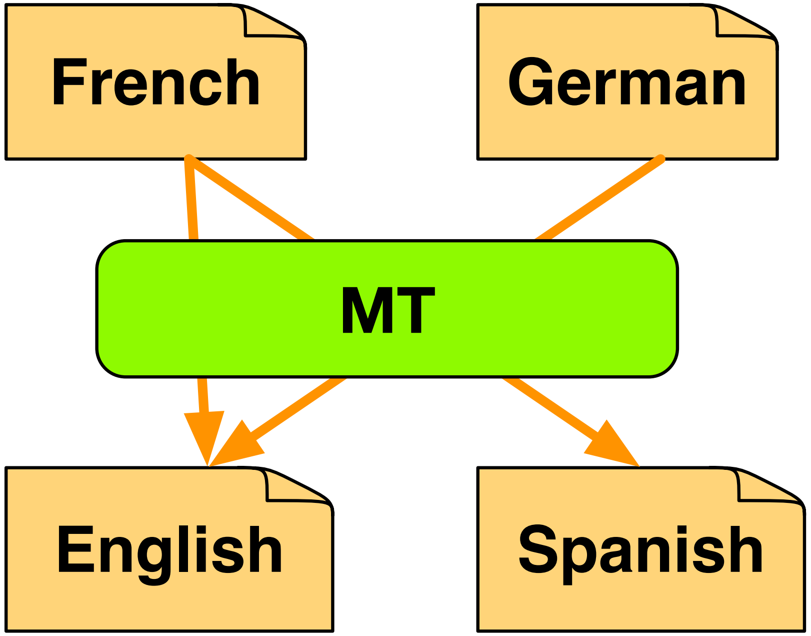
- Indicate output language with special tag

[ENGLISH] *N'y a-t-il pas ici deux poids, deux mesures?*
⇒ *Is this not a case of double standards?*

[SPANISH] *N'y a-t-il pas ici deux poids, deux mesures?*
⇒ *¿No puede verse con toda claridad que estamos utilizando un doble rasero?*

Zero Shot Translation

- Example
  - German–English
  - French–English
  - French–Spanish
- We want to translate
  - German–Spanish



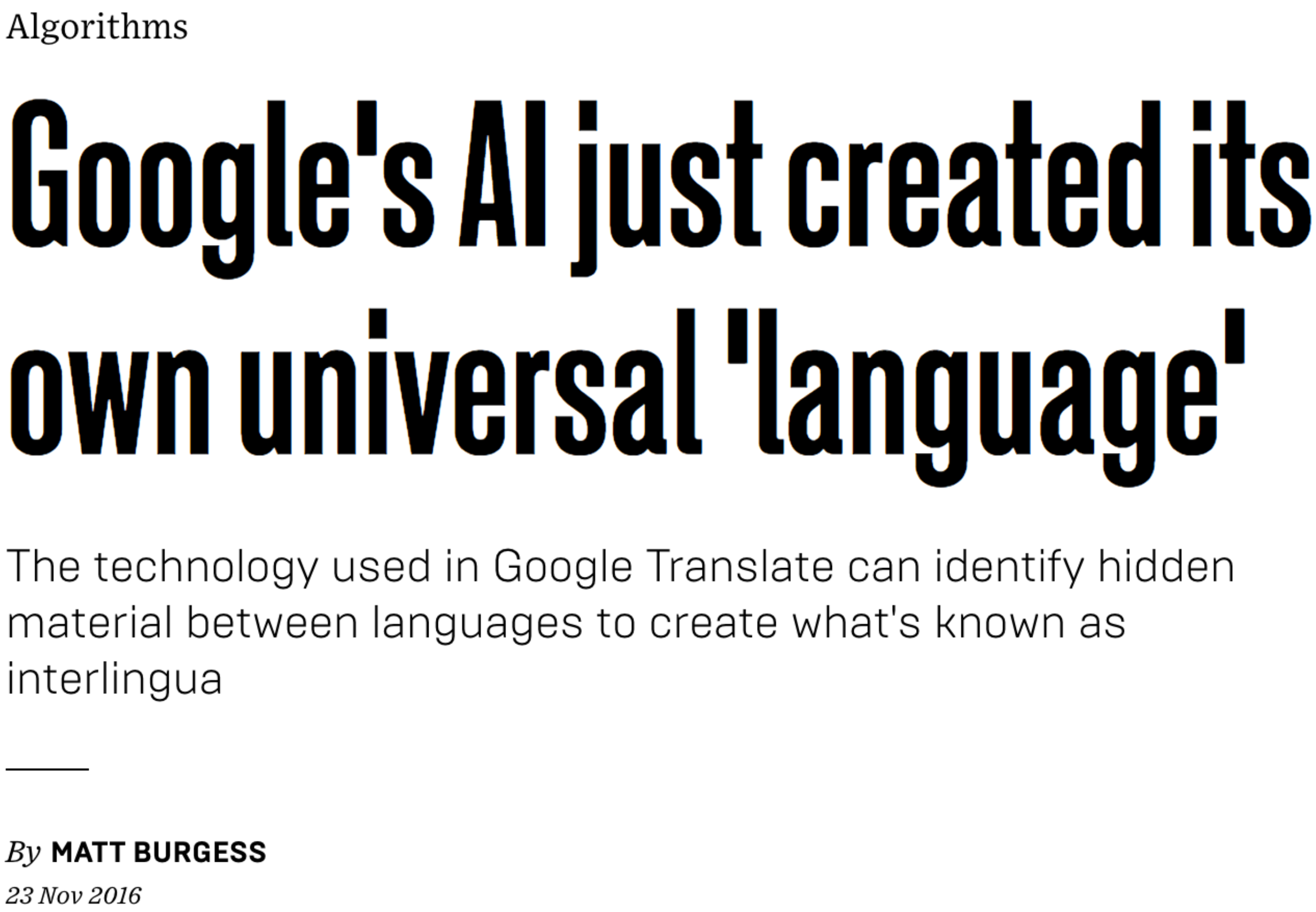
Zero Shot

- Train on
  - German–English
  - French–English
  - French–Spanish

- Specify translation

[SPANISH] *Messen wir hier nicht mit zweierlei Maß?*
⇒ *¿No puede verse con toda claridad que estamos utilizando un doble rasero?*

Zero Shot: Hype



Algorithms

Google's AI just created its own universal 'language'

The technology used in Google Translate can identify hidden material between languages to create what's known as interlingua

By **MATT BURGESS**

23 Nov 2016

Zero Shot: Reality

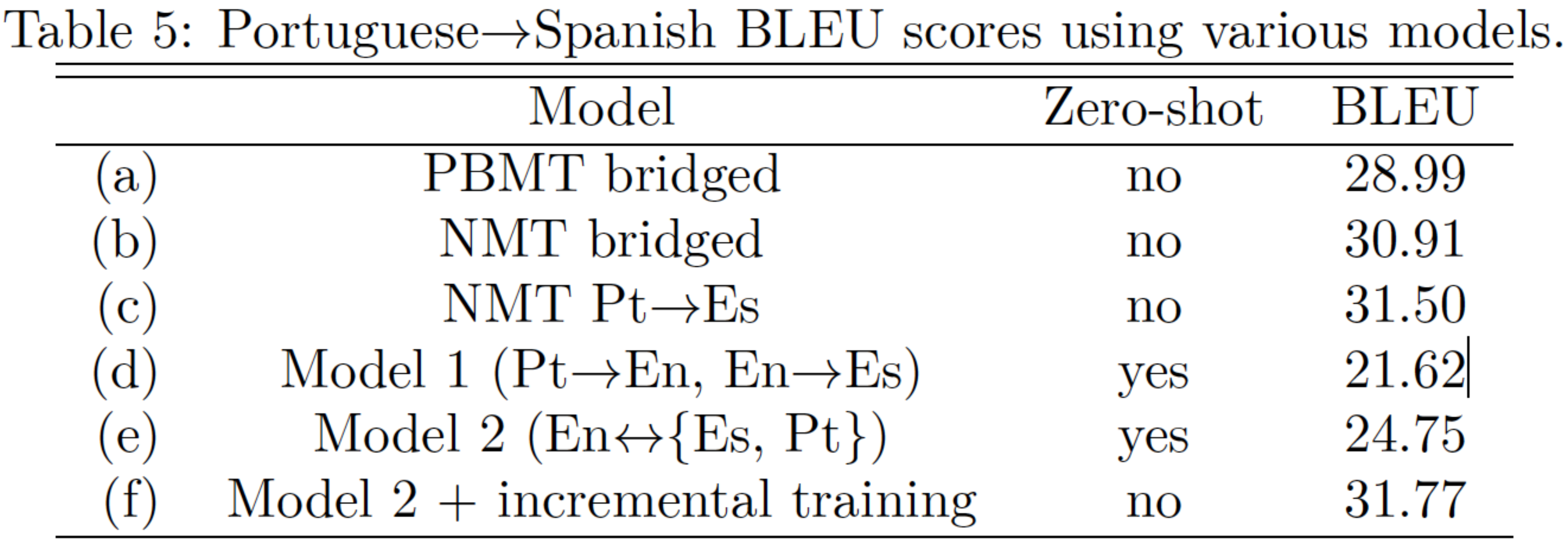


Table 5: Portuguese→Spanish BLEU scores using various models.

	Model	Zero-shot	BLEU
(a)	PBMT bridged	no	28.99
(b)	NMT bridged	no	30.91
(c)	NMT Pt→Es	no	31.50
(d)	Model 1 (Pt→En, En→Es)	yes	21.62
(e)	Model 2 (En↔{Es, Pt})	yes	24.75
(f)	Model 2 + incremental training	no	31.77

- Bridged: pivot translation Portuguese → English → Spanish
- Model 1 and 2: Zero shot training
- Model 2 + incremental training: use of some training data in language pair

Sharing Components

- So far: generic neural machine translation model
- Maybe better: separate systems with shared components
  - encoder shared in models with same input language.
  - decoder shared in models with same output language.
  - attention mechanism shared in all models
- Sharing = same parameters, updates from any language pair training
- No need to mark output language

multi-task training

Related Tasks

- Our translation models: generic sequence-to-sequence models
- Same model used for many other tasks
  - sentiment detection
  - grammar correction
  - semantic inference
  - summarization
  - question answering
  - speech recognition
- For all these tasks, we need to learn basic properties of language
  - word embeddings
  - contextualize word representations in encoder
  - language model aspects of decoder
- Why re-invent the wheel each time?

Training on Related Tasks

- Train model on several tasks
- Maybe shared and task-specific components
- System learns general facts about language
  - informed by many different tasks
  - useful for many different tasks

Pre-Training Word Embeddings

- Let us keep it simple...

- Neural machine translation models use word embeddings
  - encoding of input words
  - encoding of output words

- Word embeddings can be trained on vast amounts of monolingual data

⇒ pre-train word embeddings and initialize model with them

Pre-Training Word Embeddings

- Let us keep it simple...

- Neural machine translation models use word embeddings
  - encoding of input words
  - encoding of output words

- Word embeddings can be trained on vast amounts of monolingual data

⇒ pre-train word embeddings and initialize model with them

- Not very successful so far
  - monolingual word embeddings trained on language model objectives
  - for machine translation, different similarity aspects may matter more
  - e.g., *teacher* and *teaching* similar in MT, not in LM

Pre-Training the Encoder and Decoder

- Pre-training other components of the translation model
- Decoder
  - language model, informed by input context
  - pre-train as language model on monolingual data
  - input context vector set to zero

Pre-Training the Encoder and Decoder

- Pre-training other components of the translation model

- Decoder
  - language model, informed by input context
  - pre-train as language model on monolingual data
  - input context vector set to zero

- Encoder
  - also structures like a language model
    (however, not optimized to predict following words)
  - pre-train as language model on monolingual data

Multi-Task Training

- Multiple end-to-end tasks that share common aspects
  - need to encode an input word sequence
  - produce an output word sequence

Multi-Task Training

- Multiple end-to-end tasks that share common aspects
  - need to encode an input word sequence
  - produce an output word sequence

- May have very different input/output
  - sentiment detection: output is sentiment value
  - part-of-speech tagging: output is tag sequence
  - syntactic parsing: output is recursive parse structure (may be linearized)
  - semantic parsing: output is logical form, database query, or AMR
  - grammar correction: input is error-prone text
  - question answering: needs to be also informed by knowledge base
  - speech recognition: input is sequence of acoustic features

Multi-Task Training

- Multiple end-to-end tasks that share common aspects
  - need to encode an input word sequence
  - produce an output word sequence
- May have very different input/output
  - sentiment detection: output is sentiment value
  - part-of-speech tagging: output is tag sequence
  - syntactic parsing: output is recursive parse structure (may be linearized)
  - semantic parsing: output is logical form, database query, or AMR
  - grammar correction: input is error-prone text
  - question answering: needs to be also informed by knowledge base
  - speech recognition: input is sequence of acoustic features
- Input and output in the same language, may be mostly copied
  - grammar correction, automatic post-editing
  - question answering, semantic inference

Multi-Task Training

- Train a single model for all tasks
- Positive results with joint training of
  - part-of-speech tagging
  - named entity recognition
  - syntactic parsing
  - semantic analysis.
- Tasks may share just some components