TTIC 31190: Natural Language Processing

Kevin Gimpel

Winter 2016

Lecture 15: Introduction to Machine Translation

Announcements

- Assignment 3 due Monday
- email me to sign up for your (10-minute) class presentation on 3/3 or 3/8

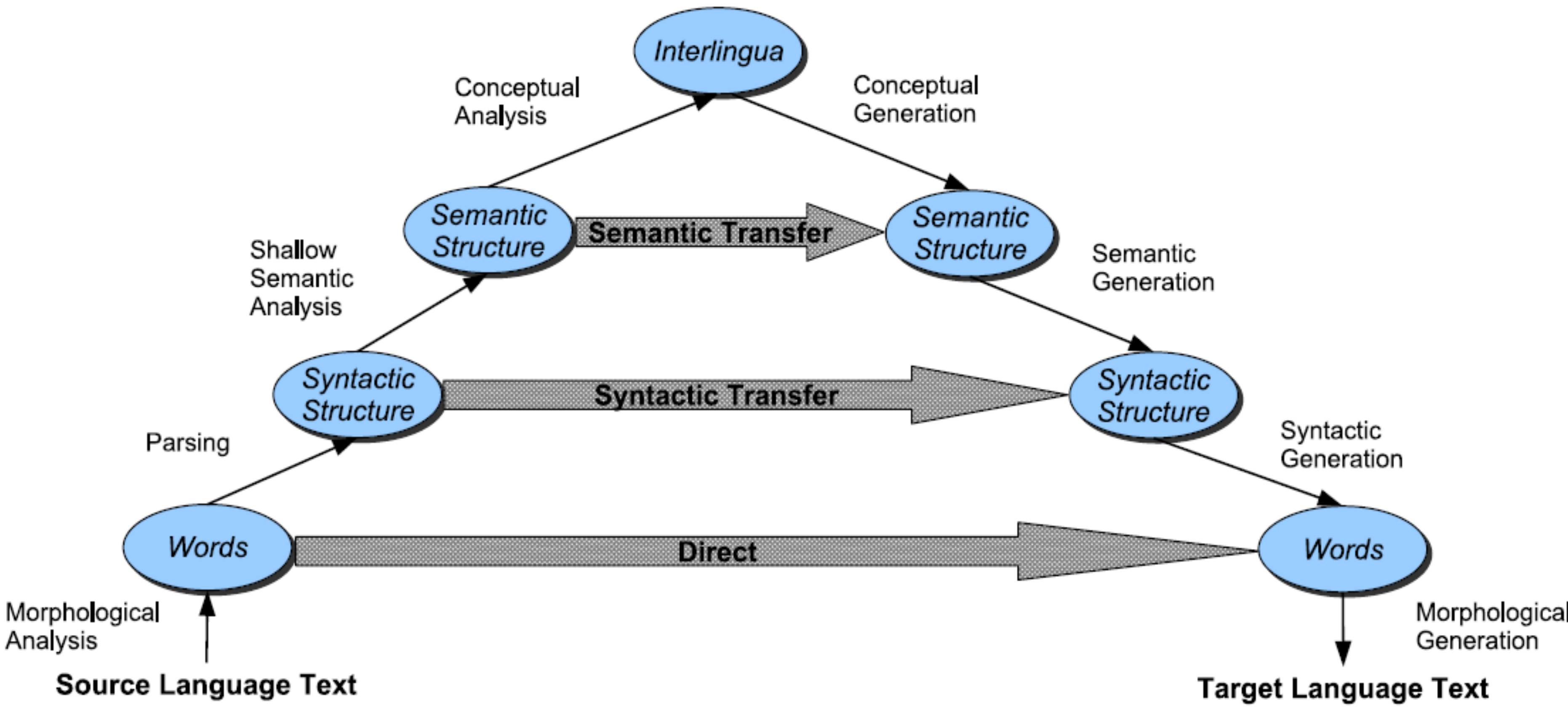
Roadmap

- classification
- words
- lexical semantics
- language modeling
- sequence labeling
- neural network methods in NLP
- syntax and syntactic parsing
- computational semantics
- machine translation
- other NLP applications

People rely on machine translation!

People rely on machine translation!

Approaches to Machine Translation: The Vauquois Triangle



Interlingua Example

$$
\begin{bmatrix}
\text{EVENT} & \text{SLAPPING} \\
\text{AGENT} & \text{MARY} \\
\text{TENSE} & \text{PAST} \\
\text{POLARITY} & \text{NEGATIVE} \\
\text{THEME} & \begin{bmatrix}
\text{WITCH} & \\
\text{DEFINITENESS} & \text{DEF} \\
\text{ATTRIBUTES} & \begin{bmatrix} \text{HAS-COLOR} & \text{GREEN} \end{bmatrix}
\end{bmatrix}
\end{bmatrix}
$$

Interlingual representation of *Mary did not slap the green witch*.

Classification Framework for Machine Translation

inference: solve argmax

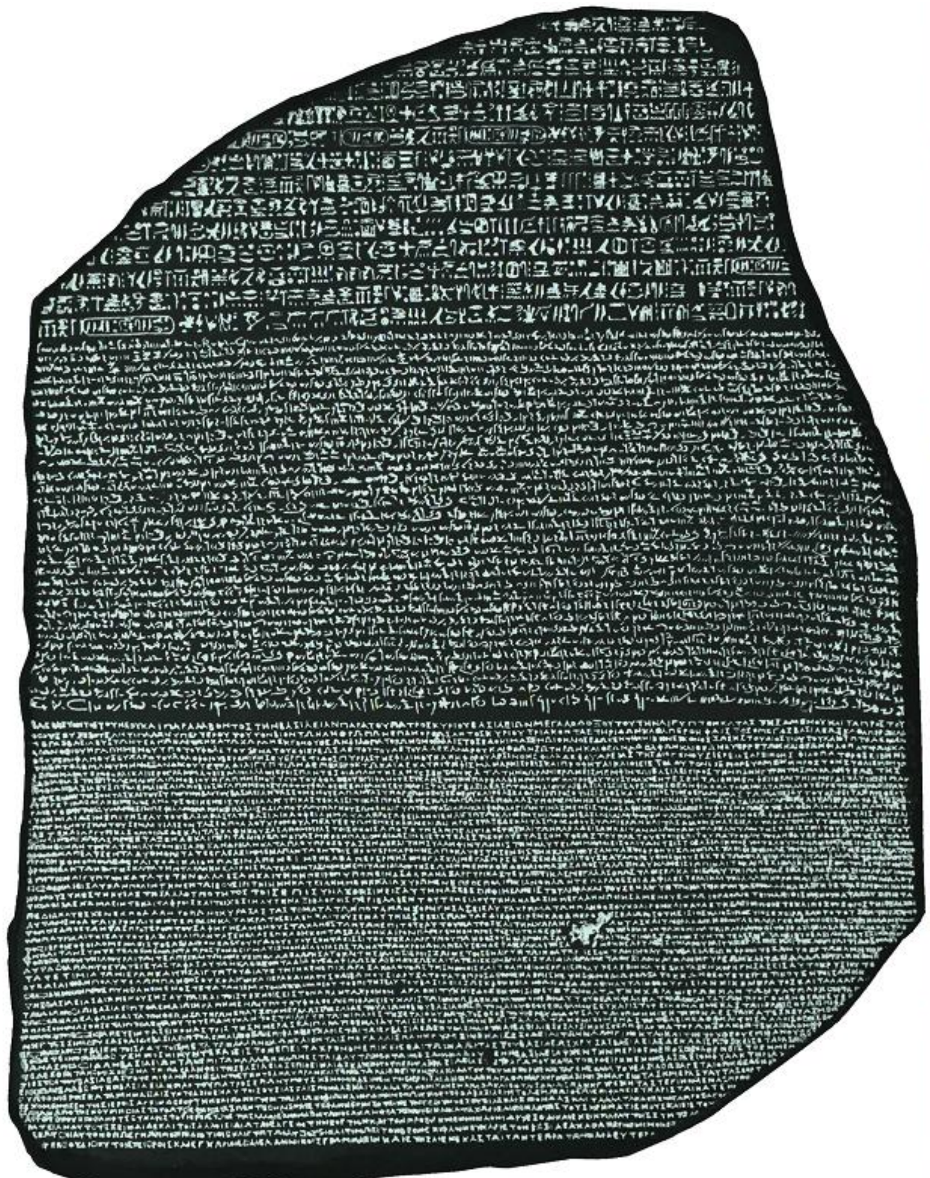
modeling: define score function

$$\mathrm{classify}(\boldsymbol{x}, \boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{y}} \ \mathrm{score}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})$$

learning: choose $\boldsymbol{\theta}$

- modern systems are **data-driven**
- first we need data!

Data?



Data?

碟頭飯 RICE PLATE

揚州炒飯	Yang Chow Fried Rice	7.95
咸魚雞粒炒飯	Salted Fish w/ Chicken Fried Rice	8.95
油雞飯	Soy Chicken Rice	5.95
滑雞菜遠飯	Chicken with Vegetable on Rice	5.95
粟米雞扒飯	Chicken with Cream Corn on Rice	5.95
豉椒雞球飯	Chicken W/ Black Bean Sauce	5.95
涼瓜牛肉飯	Beef with Bitter Melon on Rice	5.95
菜遠牛肉飯	Beef with Vegetable on Rice	5.95
牛腩飯	Beef Stew on Rice	6.95
滑蛋牛肉飯	Beef with Egg on Rice	5.95
滑蛋蝦仁飯	Shrimp with Egg on Rice	6.95
鮮蝦菜遠飯	Shrimp with Vegetable on Rice	6.95
魚片菜遠飯	Fish with Vegetable on Rice	6.95
咖哩尤魚飯	Curry Squid on Rice	6.95
滑蛋叉燒飯	BBQ Pork with Egg on Rice	5.95
肉片豆腐飯	Pork with Tofu on Rice	5.95

粥品 CONGEE

白粥	Plain Congee	2.50
皮蛋肉片粥	Preserve Egg w/ Pork Congee	5.50
生滾牛肉粥	Beef Congee	5.50
魚片粥	Fish Congee	5.95
滑雞粥	Chicken Congee	5.50

Chinese Menu

Kings Garden

球記-皇家園

Authentic Chinese Food

TEL: (614) 793-2234

7726 Sawmill Rd.
Dublin, Ohio 43017
(Old Sawmill Sq. Shopping Center)

OPEN HOUR

Mon	Close
Tues - Sat	11:00 am to 10:00 pm
Sun	11:00 am to 9:00 pm

Catering available.

Data?


302 云南芫爆松茸
Sauteed trichdoma matsutake with coriander
303 白油爆鸡枞
Stir-fried wikipedia
肉质细嫩，洁白如玉，或炒或蒸、串汤作菜，清香
云南皱椒鸡枞
Stir-fried wikipedia with pimientos
304 香油鸡枞蒸水蛋
Steam eggs with wikipedia


濃湯 Gream of pumpkin soup ¥ 15/
薩角 India samosa ¥ 25/
6. 意式火腿面包棒 Italian ham bread ¥ 15/
7. 田園沙拉 Garden salad ¥ 15/
8. 三明治 Sand wiches ¥ 15/
(培根/薩拉米/吞拿魚/火腿) (Bacon/Salami/Tuna/Ham)
9. 香烤魷魚圈 BBQ wikipedia ¥ 20/
10. 串烤牛小排 BBQ beef and vegetables ¥ 20/
11. 水牛城香辣鷄翅 Kookaburra wings ¥ 25/
12. 德國烤腸 German BBQ Sausage ¥ 30/
13. 香蒜面包 Garlic butter bread ¥ 10/

Data?

United Nations

EUROPEAN PARLIAMENT

Also:

- news articles
- company websites
- laws & patents
- subtitles

Parallel Data

- **parallel data**: bilingual data that is naturally aligned at some level
- usually aligned at the document level
- sentence-level alignments are generated automatically
  - how might you design an algorithm for this?
  - it can be done well without dictionaries!
  - can throw out sentences that don’t align with anything

Learning from Parallel Sentences

Chickasaw	English
1. Ofi 'at kowi 'ã lhiyohli	1. The dog chases the cat
2. Kowi 'at ofi 'ã lhiyohli	2. The cat chases the dog
3. Ofi 'at shoha	3. The dog stinks

Learning from Parallel Sentences

Chickasaw	English
1. Ofi 'at kowi 'ã lhiyohli	1. The dog chases the cat
2. Kowi 'at ofi 'ã lhiyohli	2. The cat chases the dog
3. Ofi 'at shoha	3. The dog stinks

Machine Translation Evaluation

- human judgments are ideal, but expensive
  - what other problems are there with human judgments?

- we need automatic evaluation metrics
  - BLEU (BiLingual Evaluation Understudy), Papineni et al. (2002)
    - compare *n*-gram overlap between system output and human-produced translation
    - correlates with human judgments surprisingly well, but only at the document level (not sentence level!)
  - other metrics do soft matching based on stemming and synonyms from WordNet
  - this is not a solved problem!

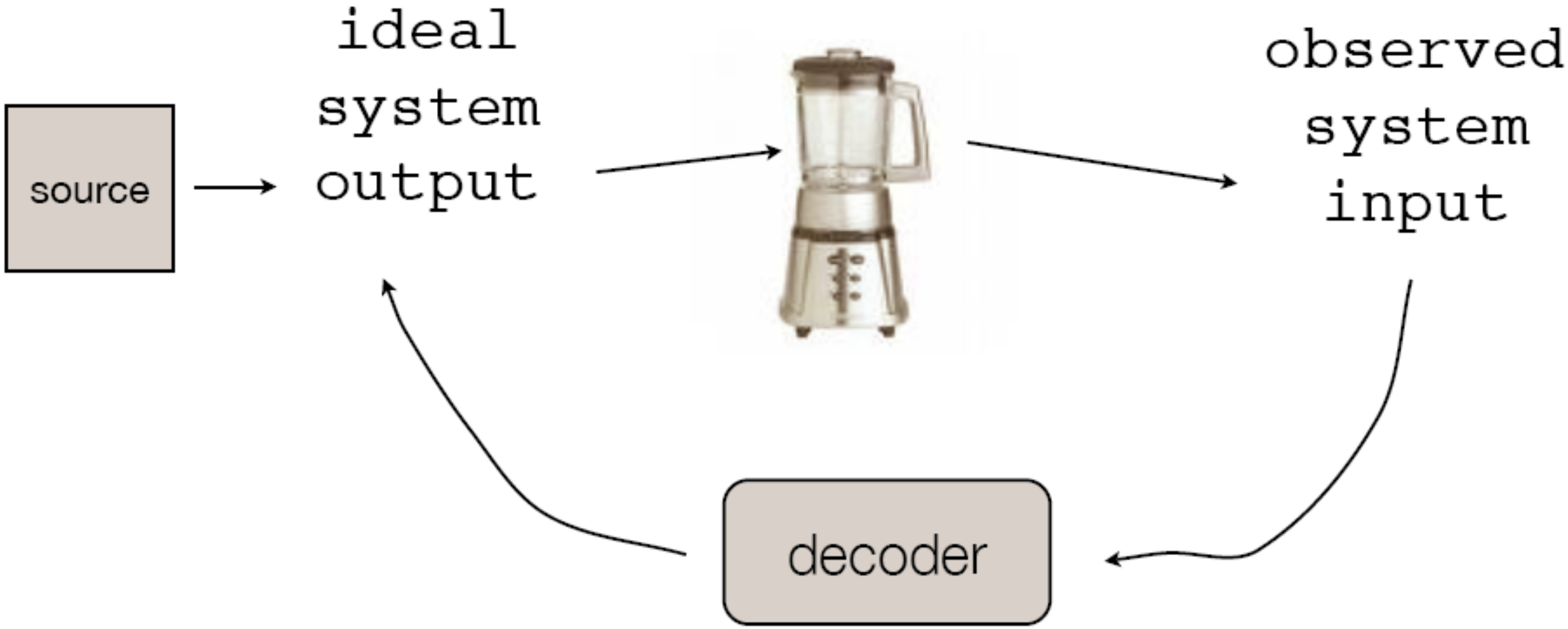
Statistical Machine Translation

*One naturally wonders if the problem of translation could conceivably be treated as a problem in **cryptography**.*

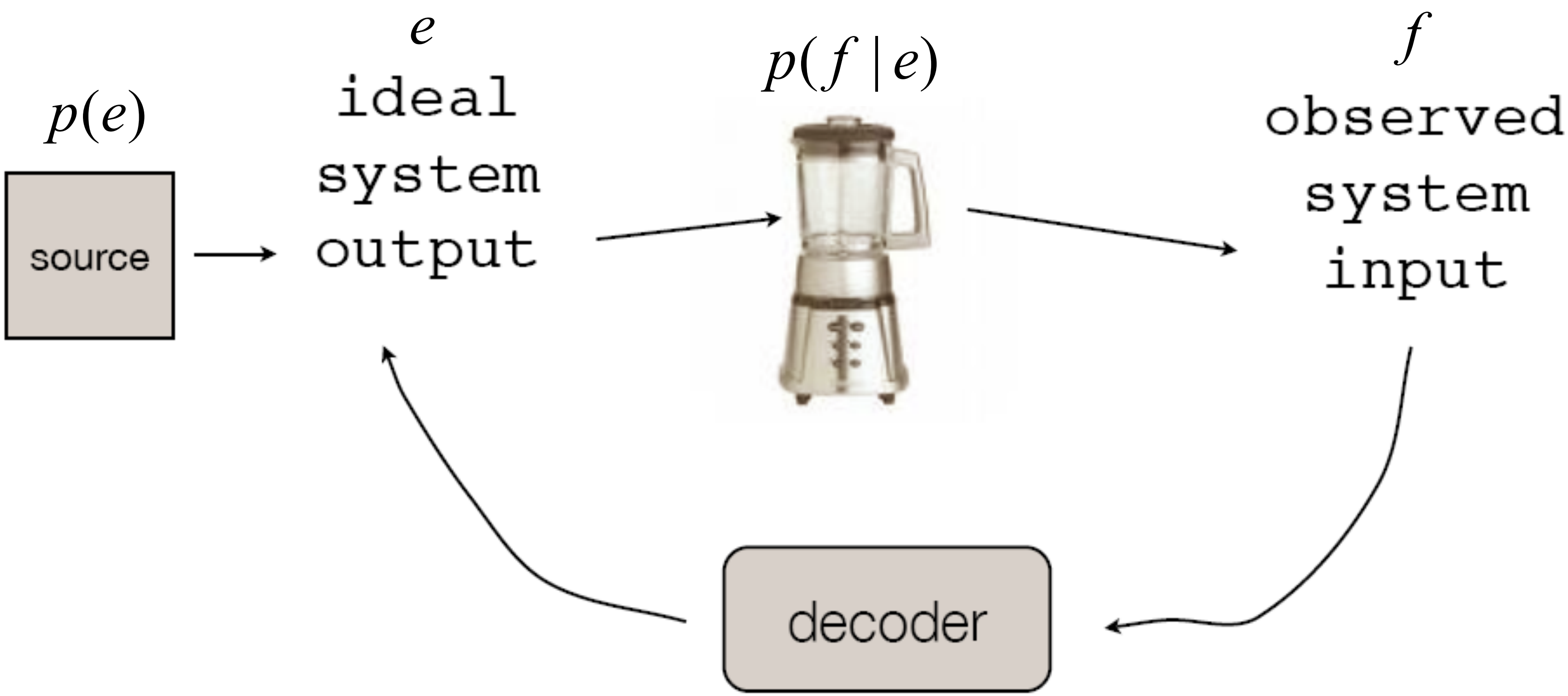
When I look at an article in Arabic, I say: "This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode."

Warren Weaver, 1947

Noisy Channel Model



Noisy Channel Model for Translating French (f) to English (e)



$$
\begin{aligned}
\hat{e} &= \arg\max_{e} p(e \mid f) \\
&= \arg\max_{e} \frac{p(f \mid e)\, p(e)}{p(f)} \\
&= \arg\max_{e} p(f \mid e)\, p(e)
\end{aligned}
$$

Modeling for the Noisy Channel

- We need to model two probability distributions: $P(e)$ and $P(f \mid e)$
  - $P(e)$ should favor fluent translations
  - $P(f \mid e)$ should favor accurate/faithful translations

Modeling for the Noisy Channel

- We need to model two probability distributions: $P(e)$ and $P(f \mid e)$
  - $P(e)$ should favor fluent translations
  - $P(f \mid e)$ should favor accurate/faithful translations

- Let's start with $P(e)$
  - How do we compute the probability of an English sentence?
  - This is an important part of MT (e.g., Google)

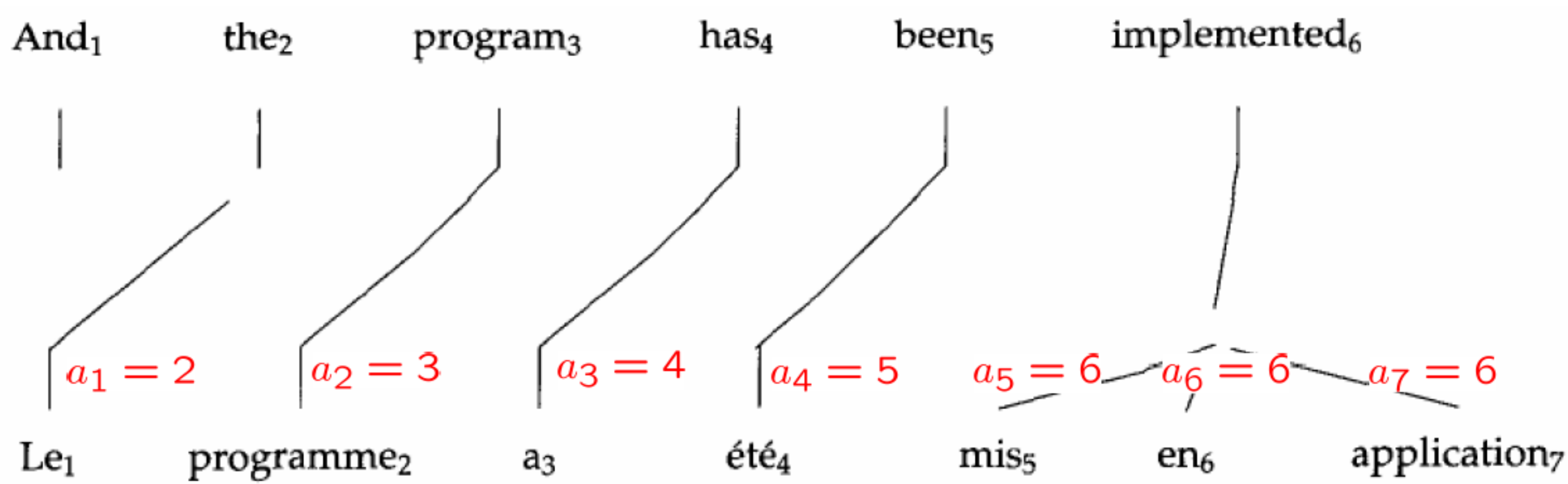
Word Alignments

And$_1$ the$_2$ program$_3$ has$_4$ been$_5$ implemented$_6$

Le$_1$ programme$_2$ a$_3$ été$_4$ mis$_5$ en$_6$ application$_7$

Word Alignments

$$a = a_1 \ldots a_{|f|}$$



- a is a "hidden" variable (not part of training data)
- for each French word, it holds the index of the aligned English word (or NULL)

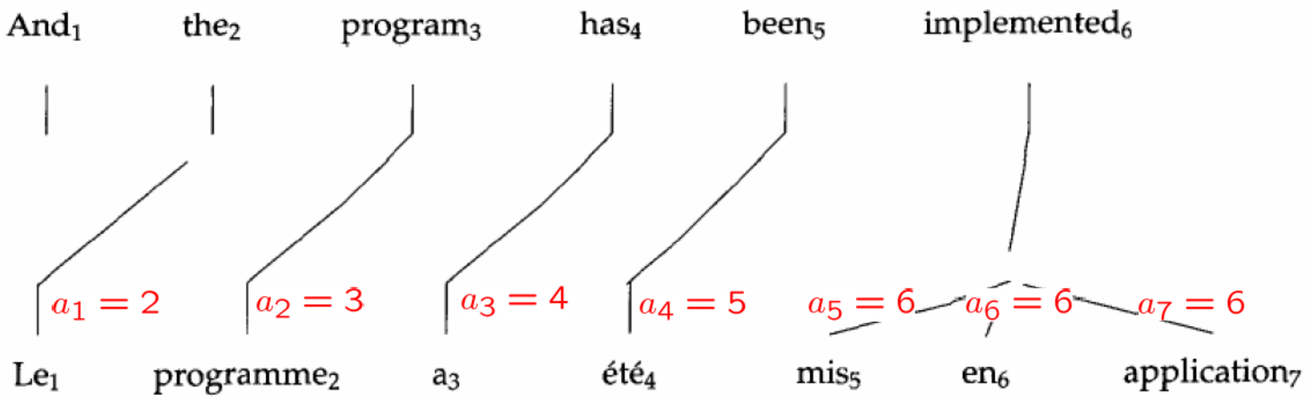
- remember: our goal was to model $P(f \mid e)$
- why would we introduce a hidden variable?
  - to make it “easier” to define the model
  - we often want to share certain types of information across multiple instances in our data
    - latent variables are a natural way to capture this
    - think of clustering (some of the points come from the same cluster)

Alignments as Hidden Variables

- for simplicity, assume that each French word aligns to 1 English word (or to NULL)

- analogy to clustering:
  - each data point has 1 vote which it can distribute among all the clusters
  - here, each French word has 1 vote which it can distribute among all the English words or NULL

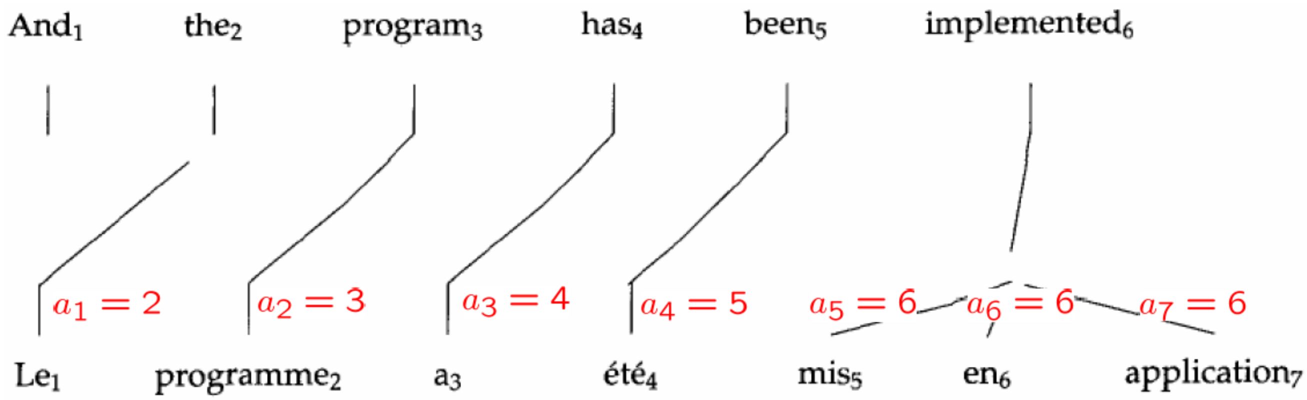
Modeling Alignments: IBM Model 1



$$P(f, a \mid e) = \prod_{j=1}^{|f|} P(a_j) P(f_j | e_{a_j})$$

$$= \prod_{j=1}^{|f|} \frac{1}{|e| + 1} P(f_j | e_{a_j})$$

Modeling Alignments: IBM Model 1

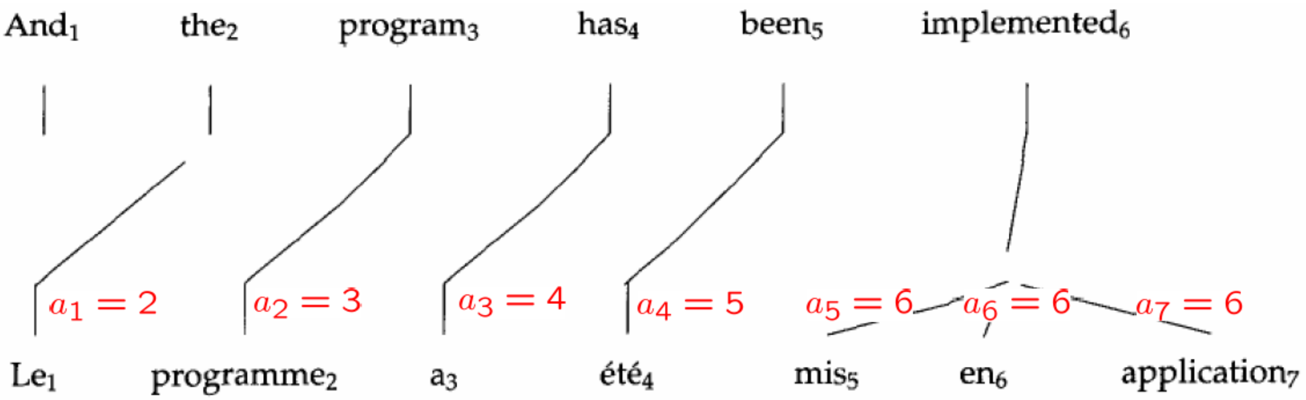


$$P(f, a \mid e) = \prod_{j=1}^{|f|} P(a_j) P(f_j | e_{a_j})$$

$$= \prod_{j=1}^{|f|} \frac{1}{|e| + 1} P(f_j | e_{a_j})$$

- How do we obtain $P(f \mid e)$?

Modeling Alignments: IBM Model 1

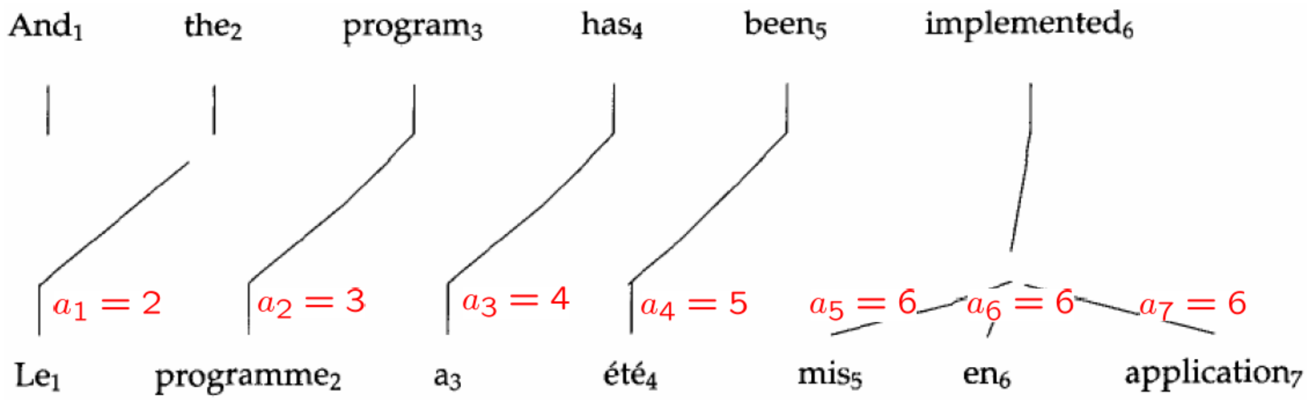


$$P(f, a \mid e) = \prod_{j=1}^{|f|} P(a_j) P(f_j | e_{a_j})$$

$$= \prod_{j=1}^{|f|} \frac{1}{|e| + 1} P(f_j | e_{a_j})$$

- How do we obtain $P(f \mid e)$?
- Sum over all alignments: $P(f \mid e) = \sum_a P(f, a \mid e)$

Modeling Alignments: IBM Model 1



$$P(f, a \mid e) = \prod_{j=1}^{|f|} P(a_j) P(f_j | e_{a_j})$$

$$= \prod_{j=1}^{|f|} \frac{1}{|e| + 1} P(f_j | e_{a_j})$$

Parameters in the model,
learned using expectation maximization

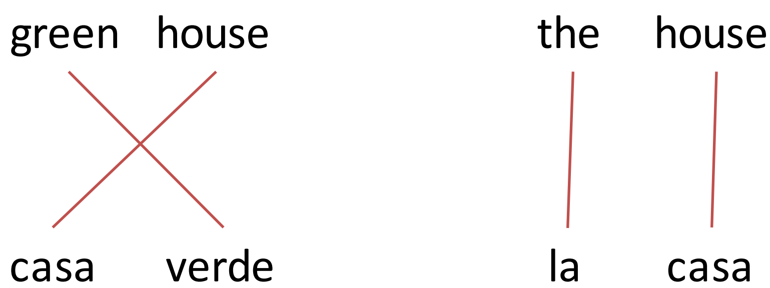
Aside: are alignments always hidden?

- certain small parallel corpora have been hand-aligned
- issues with this?
  - annotators don't agree
  - we have lots of parallel text, very little is hand-aligned
  - for some language pairs, we will never have manual alignments

- word alignment has become a fundamental part of MT, and we need unsupervised learning to solve it!

IBM Model 1 Example

- Consider a training set of two sentence pairs:



Initial Parameter Estimates:

$\text{t(casa}\mid\text{green)} = \frac{1}{3}$	$\text{t(verde}\mid\text{green)} = \frac{1}{3}$	$\text{t(la}\mid\text{green)} = \frac{1}{3}$
$\text{t(casa}\mid\text{house)} = \frac{1}{3}$	$\text{t(verde}\mid\text{house)} = \frac{1}{3}$	$\text{t(la}\mid\text{house)} = \frac{1}{3}$
$\text{t(casa}\mid\text{the)} = \frac{1}{3}$	$\text{t(verde}\mid\text{the)} = \frac{1}{3}$	$\text{t(la}\mid\text{the)} = \frac{1}{3}$

$t(f \mid e)$

= probability of translating *e* into *f*

After 1 iteration of EM:

$\text{t(casa}\mid\text{green)} = \frac{1/2}{1} = \frac{1}{2}$	$\text{t(verde}\mid\text{green)} = \frac{1/2}{1} = \frac{1}{2}$	$\text{t(la}\mid\text{green)} = \frac{0}{1} = 0$
$\text{t(casa}\mid\text{house)} = \frac{1}{2} = \frac{1}{2}$	$\text{t(verde}\mid\text{house)} = \frac{1/2}{2} = \frac{1}{4}$	$\text{t(la}\mid\text{house)} = \frac{1/2}{2} = \frac{1}{4}$
$\text{t(casa}\mid\text{the)} = \frac{1/2}{1} = \frac{1}{2}$	$\text{t(verde}\mid\text{the)} = \frac{0}{1} = 0$	$\text{t(la}\mid\text{the)} = \frac{1/2}{1} = \frac{1}{2}$

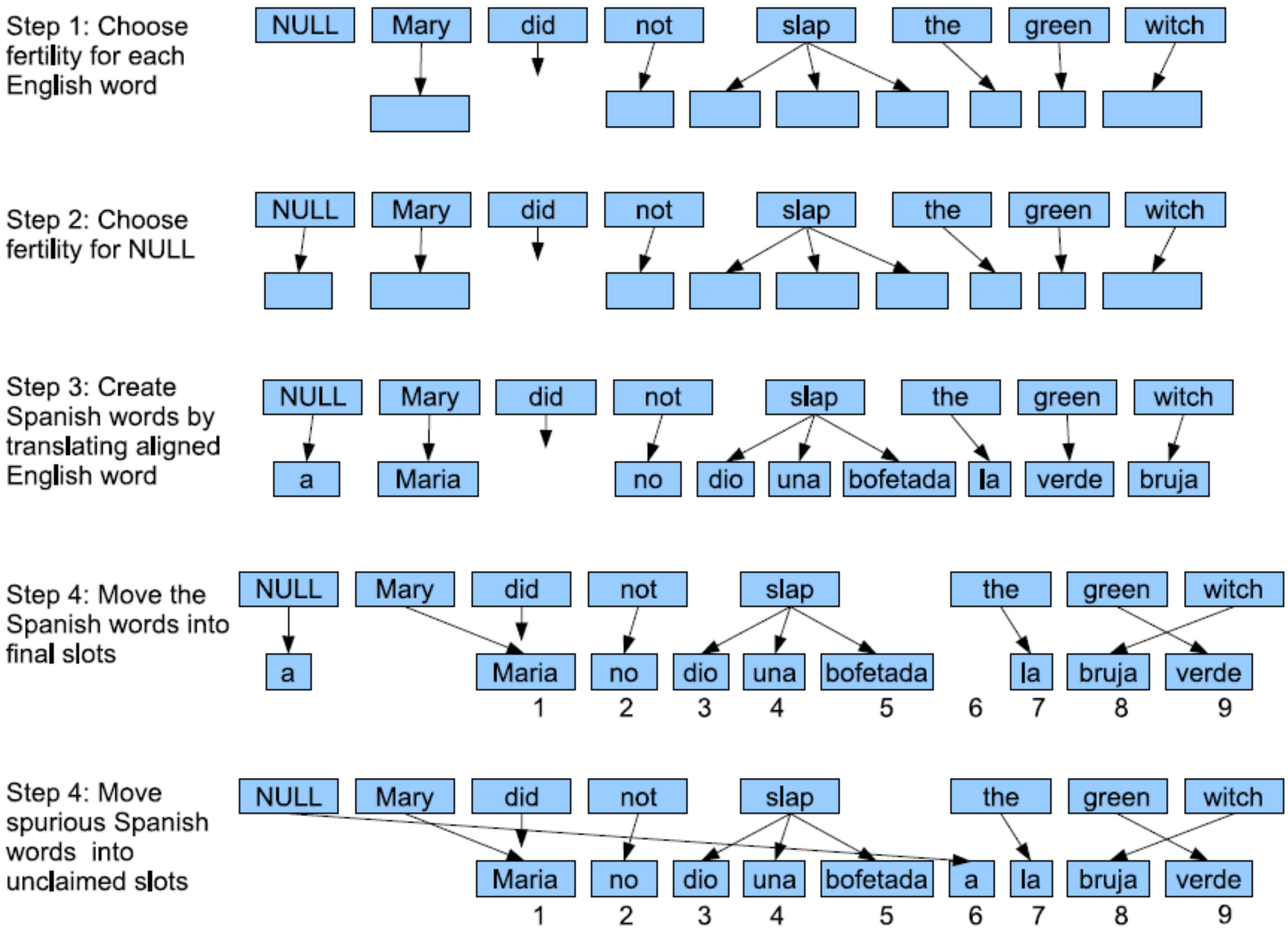
IBM Model 1

$$P(f, a \mid e) = \prod_{j=1}^{|f|} \frac{1}{|e| + 1} P(f_j | e_{a_j})$$

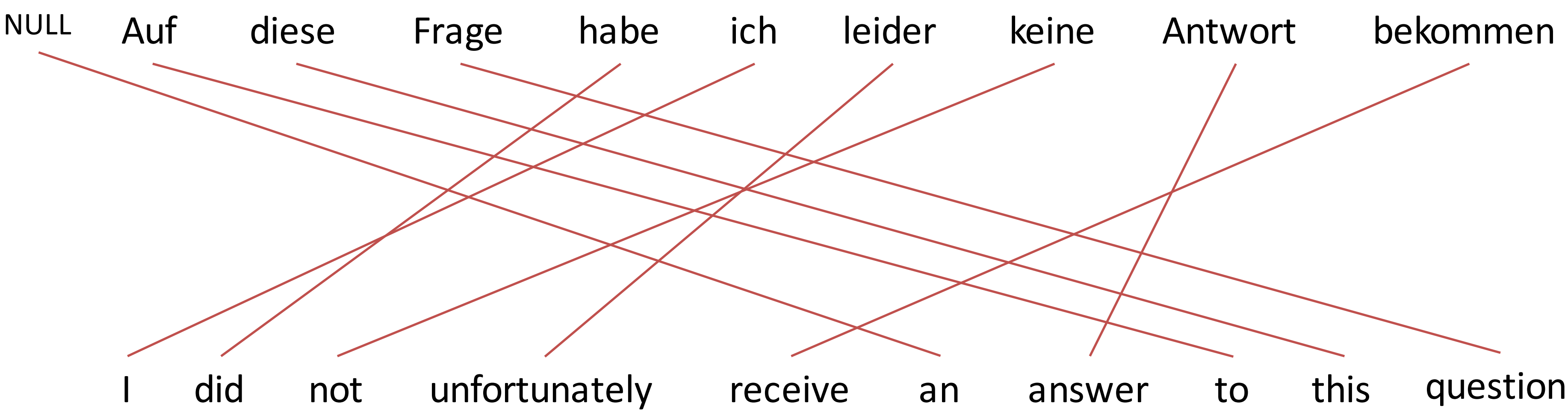
IBM Model 2

$$P(f, a \mid e) = \prod_{j=1}^{|f|} P(a_j \mid j, |f|, |e|) P(f_j | e_{a_j})$$

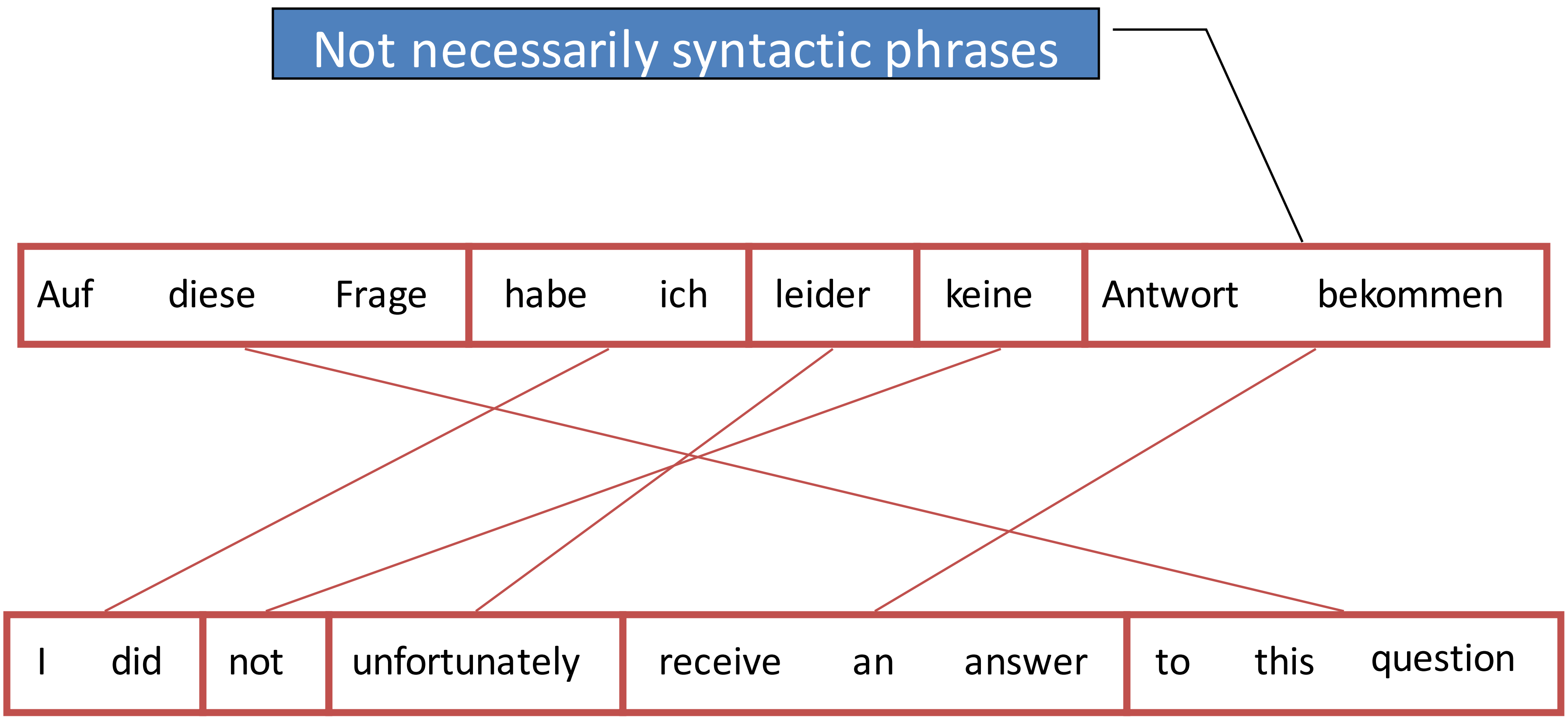
IBM Model 3


Step 1: Choose fertility for each English word
NULL
Mary
did
not
slap
the
green
witch
Step 2: Choose fertility for NULL
Step 3: Create Spanish words by translating aligned English word
a
Maria
no
dio
una
bofetada
la
verde
bruja
Step 4: Move the Spanish words into final slots
1
2
3
4
5
6
7
8
9
Step 4: Move spurious Spanish words into unclaimed slots

Moving to Phrases



Moving to Phrases

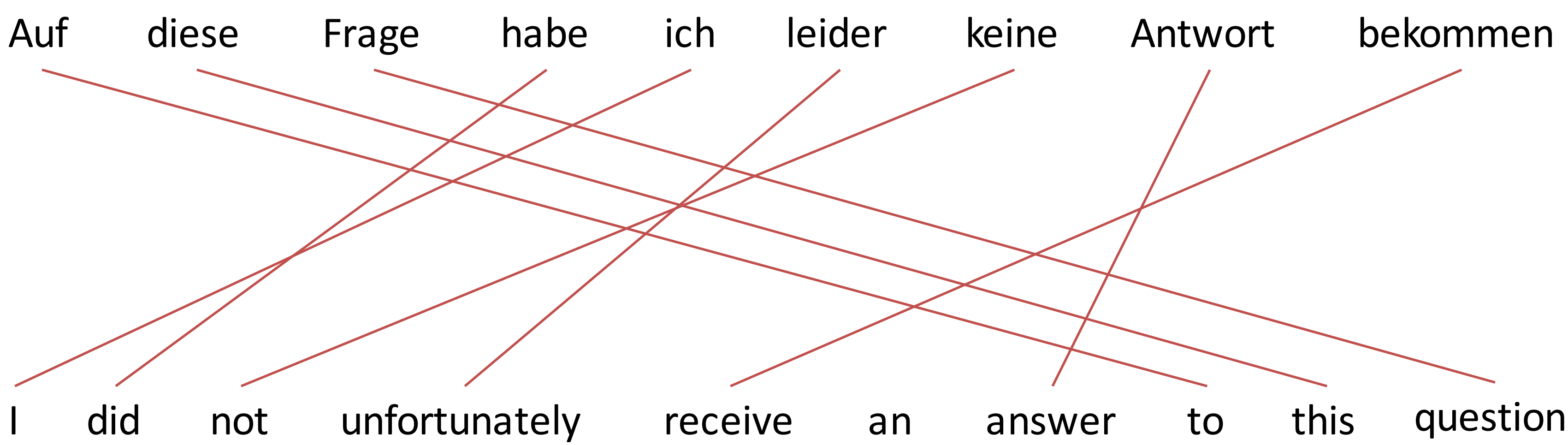


“Phrase-Based” Translation

- Relies on a **phrase table**
  - massive bilingual phrase dictionary, with probabilities
- To build:
  - Find the best word alignment for each sentence pair
  - Extract all phrase pairs **consistent** with the word alignment
  - Compute probabilities using relative frequency estimation

Phrase-Based Translation

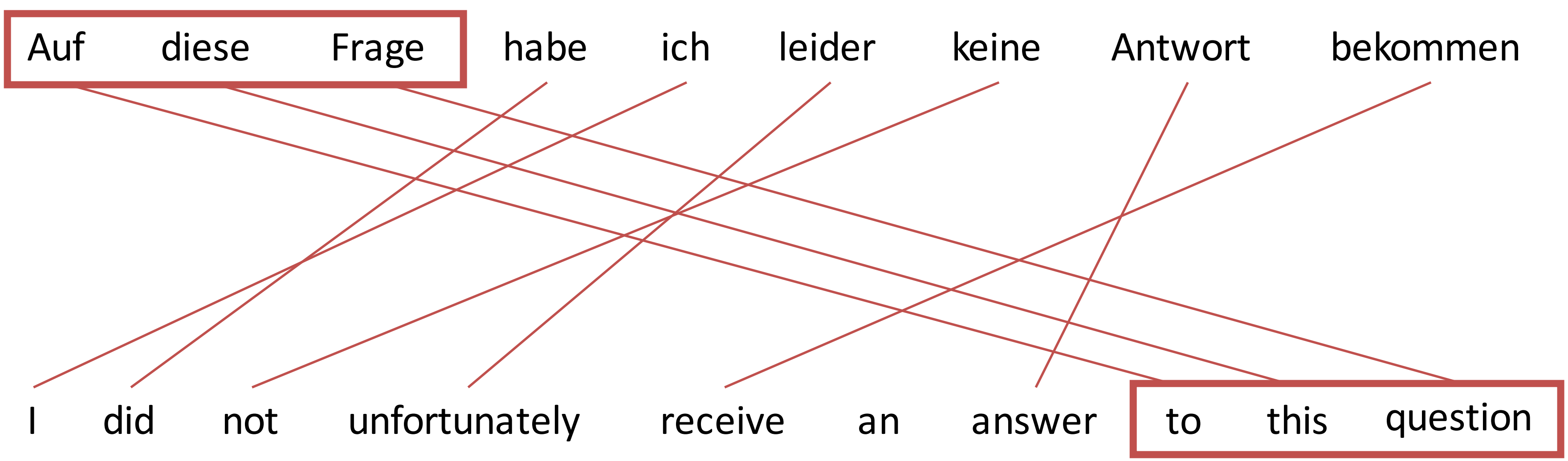
- Relies on a **phrase table**
  - massive bilingual phrase dictionary, with probabilities
- To build:
  - Find the best word alignment for each sentence pair
  - Extract all phrase pairs **consistent** with the word alignment
  - Compute probabilities using relative frequency estimation



Phrase-Based Translation

- Relies on a **phrase table**
  - massive bilingual phrase dictionary, with probabilities
- To build:
  - Find the best word alignment for each sentence pair
  - Extract all phrase pairs **consistent** with the word alignment
  - Compute probabilitie

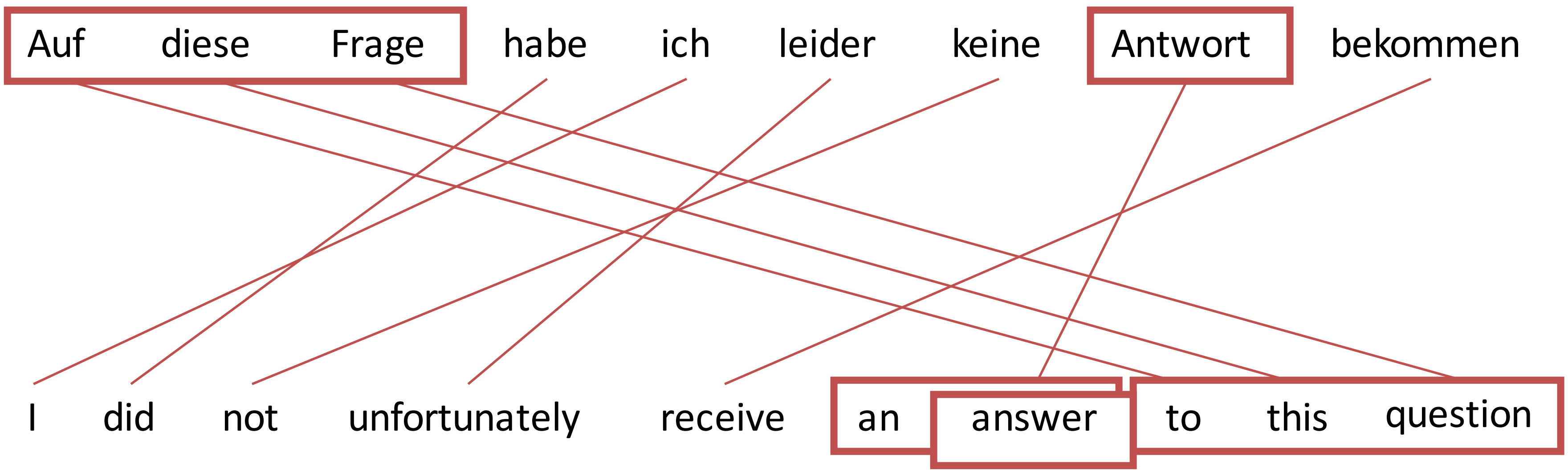
Auf diese Frage	to this question	1.0



Phrase-Based Translation

- Relies on a **phrase table**
  - massive bilingual phrase dictionary, with probabilities
- To build:
  - Find the best word alignment for each sentence pair
  - Extract all phrase pairs **consistent** with the word alignment
  - Compute probabilities

Auf diese Frage	to this question	1.0
Antwort	an answer	1.0
Antwort	answer	1.0
	...	



Phrase-Based Translation

- Relies on a **phrase table**
  - massive bilingual phrase dictionary, with probabilities
- To build:
  - Find the best word alignment for each sentence pair
  - Extract all phrase pairs **consistent** with the word alignment
  - Compute probabilities using relative frequency estimation:

$$p(e \mid f) = \frac{count(e, f)}{\sum_{e'} count(e', f)}$$

German	English	Count
Auf diese Frage	to this question	1.0
Antwort	an answer	1.0
Antwort	answer	1.0
…		

German	English	P(e | f)
Auf diese Frage	to this question	1.0
Antwort	an answer	0.5
Antwort	answer	0.5
…		

Adding Syntax: Synchronous Context-Free Grammars

CFG

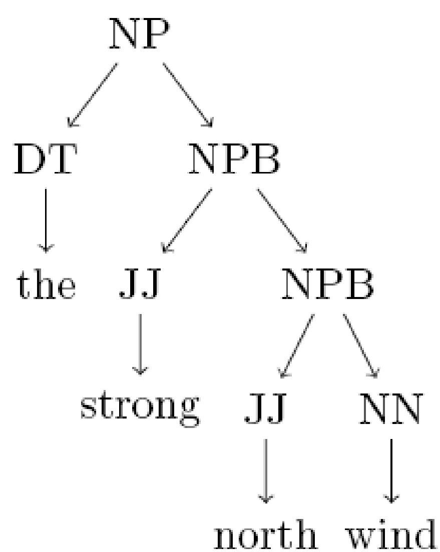
$$\begin{aligned}
\mathrm{NP} &\longrightarrow \mathrm{DT}\ \mathrm{NPB}\\
\mathrm{NPB} &\longrightarrow \mathrm{JJ}\ \mathrm{NPB}\\
\mathrm{NPB} &\longrightarrow \mathbf{NN}\\
\mathrm{DT} &\longrightarrow \text{the}\\
\mathrm{JJ} &\longrightarrow \text{strong}\\
\mathrm{JJ} &\longrightarrow \text{north}\\
\mathrm{NN} &\longrightarrow \text{wind}
\end{aligned}$$

SCFG

$$\begin{aligned}
\mathrm{NP} &\longrightarrow \mathrm{DT}_{\boxed{1}}\mathrm{NPB}_{\boxed{2}} \ /\ \mathrm{DT}_{\boxed{1}}\mathrm{NPB}_{\boxed{2}}\\
\mathrm{NPB} &\longrightarrow \mathrm{JJ}_{\boxed{1}}\mathrm{NPB}_{\boxed{2}} \ /\ \mathrm{JJ}_{\boxed{1}}\mathrm{NPB}_{\boxed{2}}\\
\mathrm{NPB} &\longrightarrow \mathrm{JJ}_{\boxed{1}}\mathrm{NPB}_{\boxed{2}} \ /\ \mathrm{NPB}_{\boxed{2}}\mathrm{JJ}_{\boxed{1}}\\
\mathrm{NPB} &\longrightarrow \mathrm{NP}_{\boxed{1}} \ /\ \mathrm{NP}_{\boxed{1}}\\
\mathrm{DT} &\longrightarrow \text{the} \ /\ \varepsilon\\
\mathrm{JJ} &\longrightarrow \text{strong} \ /\ \text{呼啸}\\
\mathrm{JJ} &\longrightarrow \text{north} \ /\ \text{北}\\
\mathrm{NN} &\longrightarrow \text{wind} \ /\ \text{风}
\end{aligned}$$

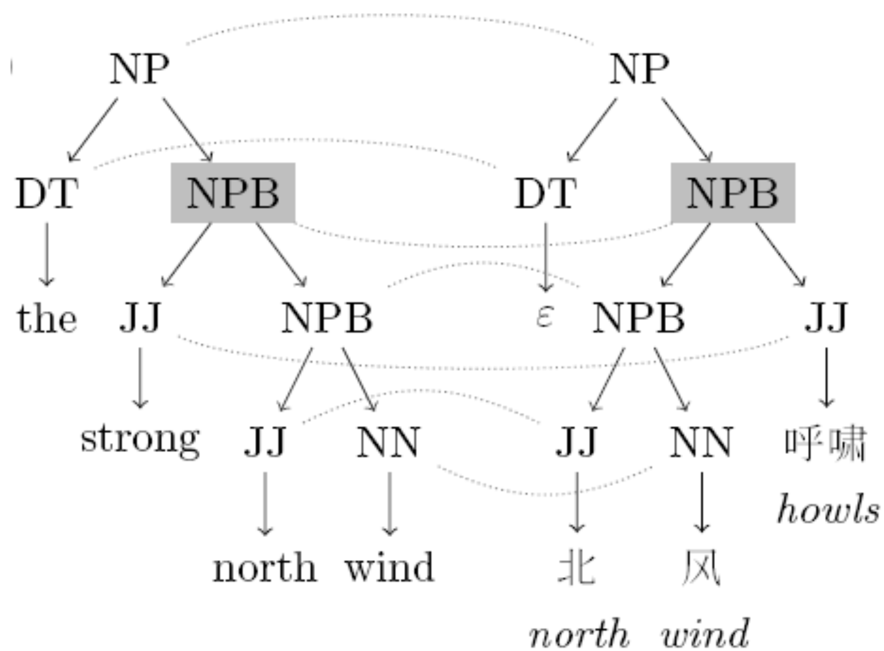
CFG

$$
\begin{aligned}
\mathrm{NP} &\longrightarrow \mathrm{DT\ NPB} \\
\mathrm{NPB} &\longrightarrow \mathrm{JJ\ NPB} \\
\mathrm{NPB} &\longrightarrow \mathbf{NN} \\
\mathrm{DT} &\longrightarrow \text{the} \\
\mathrm{JJ} &\longrightarrow \text{strong} \\
\mathrm{JJ} &\longrightarrow \text{north} \\
\mathrm{NN} &\longrightarrow \text{wind}
\end{aligned}
$$



SCFG

$$
\begin{aligned}
\mathrm{NP} &\longrightarrow \mathrm{DT}_{\boxed{1}}\mathrm{NPB}_{\boxed{2}} \ / \ \mathrm{DT}_{\boxed{1}}\mathrm{NPB}_{\boxed{2}} \\
\mathrm{NPB} &\longrightarrow \mathrm{JJ}_{\boxed{1}}\mathrm{NPB}_{\boxed{2}} \ / \ \mathrm{JJ}_{\boxed{1}}\mathrm{NPB}_{\boxed{2}} \\
\mathrm{NPB} &\longrightarrow \mathrm{JJ}_{\boxed{1}}\mathrm{NPB}_{\boxed{2}} \ / \ \mathrm{NPB}_{\boxed{2}}\mathrm{JJ}_{\boxed{1}} \\
\mathrm{NPB} &\longrightarrow \mathrm{NP}_{\boxed{1}} \ / \ \mathrm{NP}_{\boxed{1}} \\
\mathrm{DT} &\longrightarrow \text{the} \ / \ \varepsilon \\
\mathrm{JJ} &\longrightarrow \text{strong} \ / \ \text{呼啸} \\
\mathrm{JJ} &\longrightarrow \text{north} \ / \ \text{北} \\
\mathrm{NN} &\longrightarrow \text{wind} \ / \ \text{风}
\end{aligned}
$$



Noisy Channel

$$\boldsymbol{y}^* = \operatorname*{argmax}_{\boldsymbol{y}} \; P(\boldsymbol{x} \mid \boldsymbol{y}) \, P(\boldsymbol{y})$$

Noisy Channel

$$\boldsymbol{y}^* = \operatorname*{argmax}_{\boldsymbol{y}} P(\boldsymbol{x} \mid \boldsymbol{y})\, P(\boldsymbol{y})$$

predicted translation

source sentence

Noisy Channel

$$\boldsymbol{y}^* = \underset{\boldsymbol{y}}{\operatorname{argmax}} \; P(\boldsymbol{x} \mid \boldsymbol{y}) \, P(\boldsymbol{y})$$

assumes we have the right model, and that we estimate it perfectly

Noisy Channel

$$\boldsymbol{y}^* = \operatorname*{argmax}_{\boldsymbol{y}} P(\boldsymbol{x} \mid \boldsymbol{y}) \, P(\boldsymbol{y})$$

assumes we have the right model, and that we estimate it perfectly

$$\boldsymbol{y}^* = \operatorname*{argmax}_{\boldsymbol{y}} P(\boldsymbol{x} \mid \boldsymbol{y})^{\alpha} \, P(\boldsymbol{y})^{\beta}$$

Noisy Channel

$$\boldsymbol{y}^* = \operatorname*{argmax}_{\boldsymbol{y}} P(\boldsymbol{x} \mid \boldsymbol{y})\, P(\boldsymbol{y})$$

assumes we have the right model, and that we estimate it perfectly

$$\begin{aligned}\boldsymbol{y}^* &= \operatorname*{argmax}_{\boldsymbol{y}} P(\boldsymbol{x} \mid \boldsymbol{y})^{\alpha}\, P(\boldsymbol{y})^{\beta} \\ &= \operatorname*{argmax}_{\boldsymbol{y}}\ \alpha \log P(\boldsymbol{x} \mid \boldsymbol{y}) + \beta \log P(\boldsymbol{y})\end{aligned}$$

extra parameters to tune, can tune to optimize BLEU

Noisy Channel

$$\boldsymbol{y}^* = \operatorname*{argmax}_{\boldsymbol{y}} P(\boldsymbol{x} \mid \boldsymbol{y})\, P(\boldsymbol{y})$$

assumes we have the right model, and that we estimate it perfectly

$$\begin{aligned} \boldsymbol{y}^* &= \operatorname*{argmax}_{\boldsymbol{y}} P(\boldsymbol{x} \mid \boldsymbol{y})^{\alpha}\, P(\boldsymbol{y})^{\beta} \\ &= \operatorname*{argmax}_{\boldsymbol{y}} \ \alpha \log P(\boldsymbol{x} \mid \boldsymbol{y}) + \beta \log P(\boldsymbol{y}) \end{aligned}$$

extra parameters to tune, can tune to optimize BLEU

“tuning”

Noisy Channel → Linear Model?

$$\boldsymbol{y}^* = \operatorname*{argmax}_{\boldsymbol{y}} \ \alpha \log P(\boldsymbol{x} \mid \boldsymbol{y}) + \beta \log P(\boldsymbol{y})$$

since we're not using idealized decoding rule anymore,
why not add more feature functions?

Noisy Channel → Linear Model?

$$\boldsymbol{y}^* = \underset{\boldsymbol{y}}{\operatorname{argmax}} \; \alpha \log P(\boldsymbol{x} \mid \boldsymbol{y}) + \beta \log P(\boldsymbol{y})$$

since we're not using idealized decoding rule anymore,
why not add more feature functions?

"word count feature":

$$\boldsymbol{y}^* = \underset{\boldsymbol{y}}{\operatorname{argmax}} \; \alpha \log P(\boldsymbol{x} \mid \boldsymbol{y}) + \beta \log P(\boldsymbol{y}) + \boxed{\gamma \, |\boldsymbol{y}|}$$

Noisy Channel → Linear Model?

$$\boldsymbol{y}^* = \operatorname*{argmax}_{\boldsymbol{y}} \; \alpha \log P(\boldsymbol{x} \mid \boldsymbol{y}) + \beta \log P(\boldsymbol{y})$$

since we're not using idealized decoding rule anymore,
why not add more feature functions?

"word count feature":

$$\boldsymbol{y}^* = \operatorname*{argmax}_{\boldsymbol{y}} \; \alpha \log P(\boldsymbol{x} \mid \boldsymbol{y}) + \beta \log P(\boldsymbol{y}) + \boxed{\gamma \, |\boldsymbol{y}|}$$

"reverse translation model feature":

$$\boldsymbol{y}^* = \operatorname*{argmax}_{\boldsymbol{y}} \; \alpha \log P(\boldsymbol{x} \mid \boldsymbol{y}) + \beta \log P(\boldsymbol{y}) + \gamma \, |\boldsymbol{y}| + \boxed{\delta \, \log P(\boldsymbol{y} \mid \boldsymbol{x})}$$

African National Congress	opposition	sanction	Zimbabwe
非国大	反对	制裁	津巴布韦

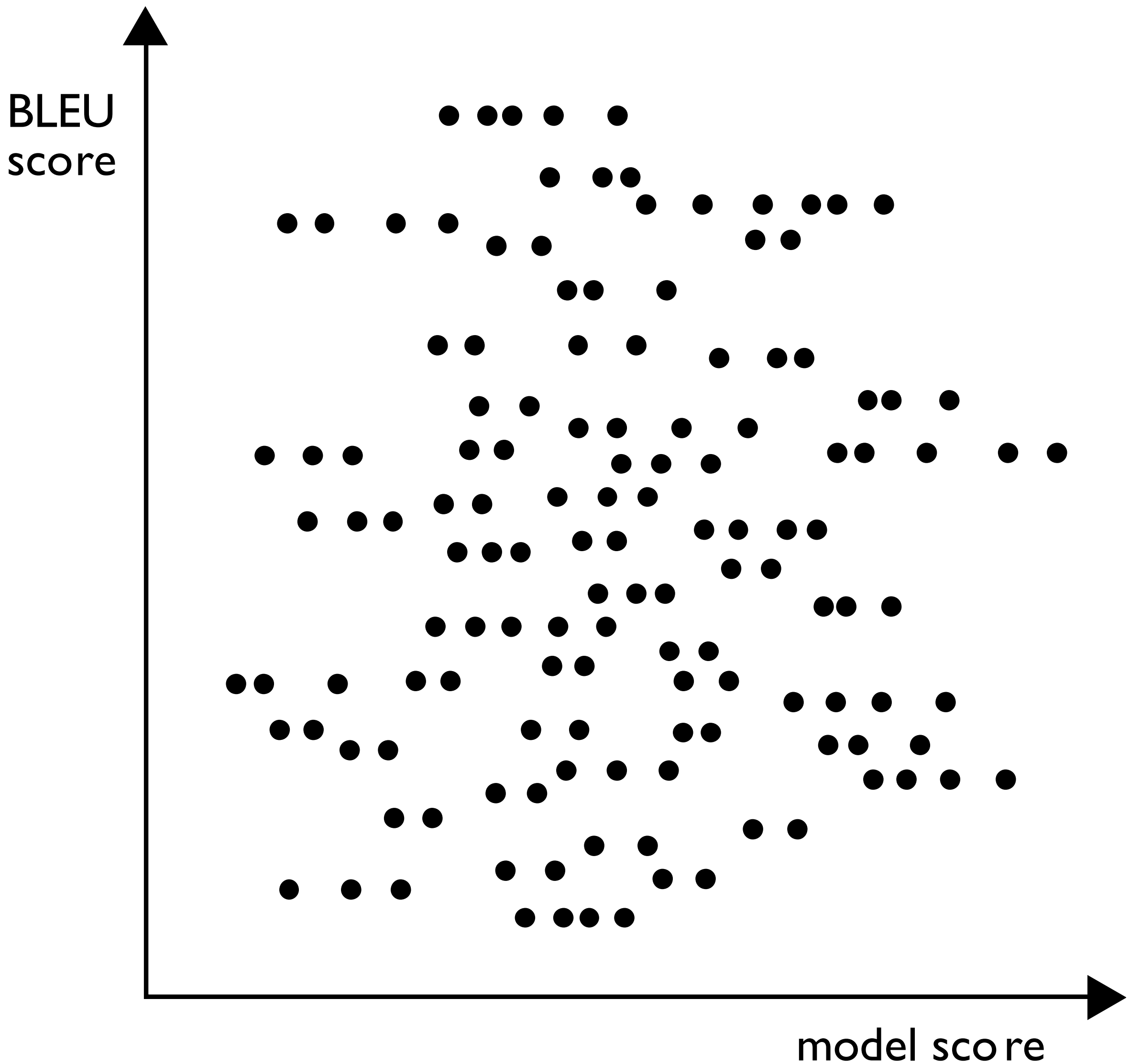
African National Congress	opposition	sanction	Zimbabwe
非国大	反对	制裁	津巴布韦

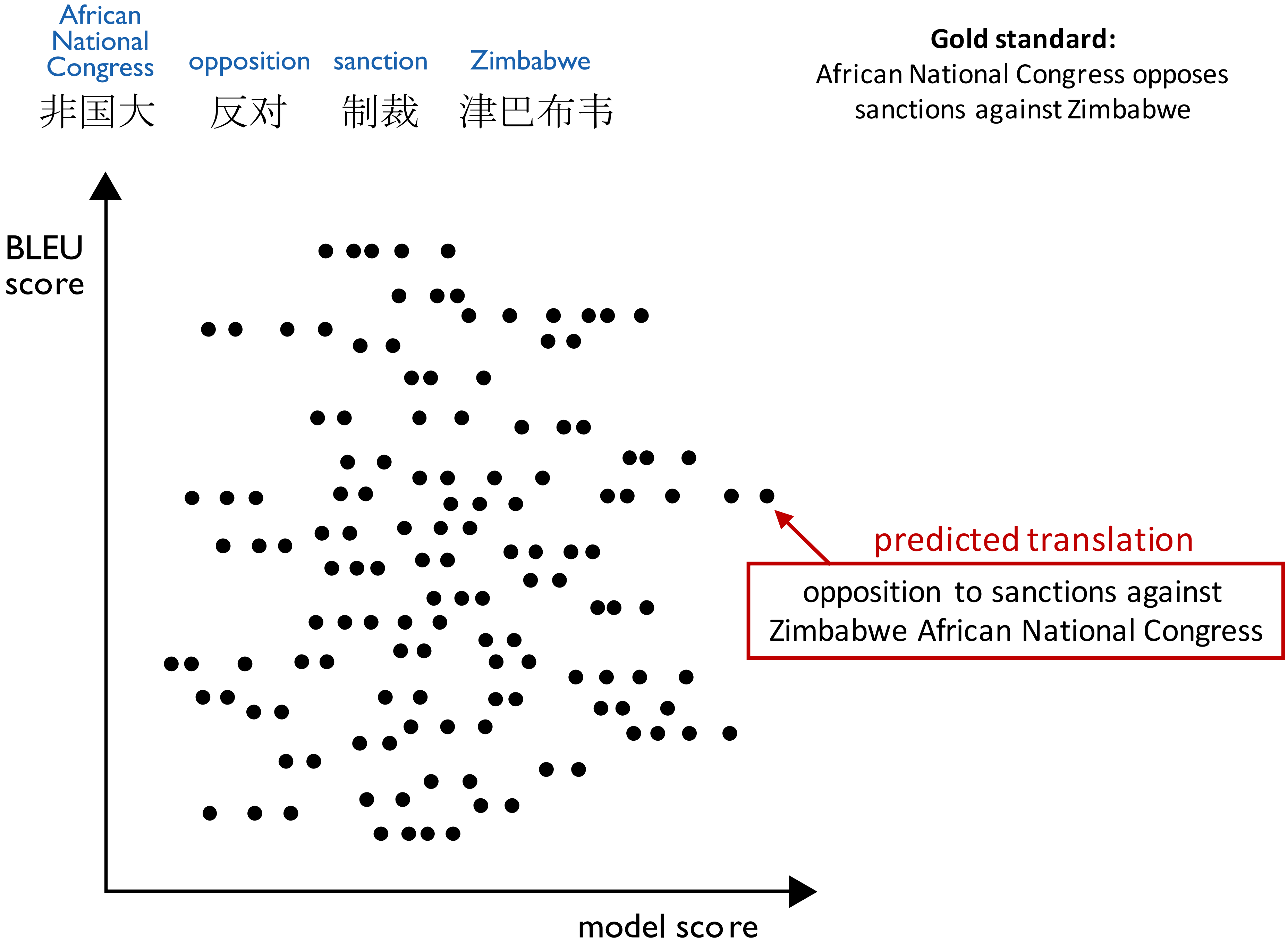
Gold standard:

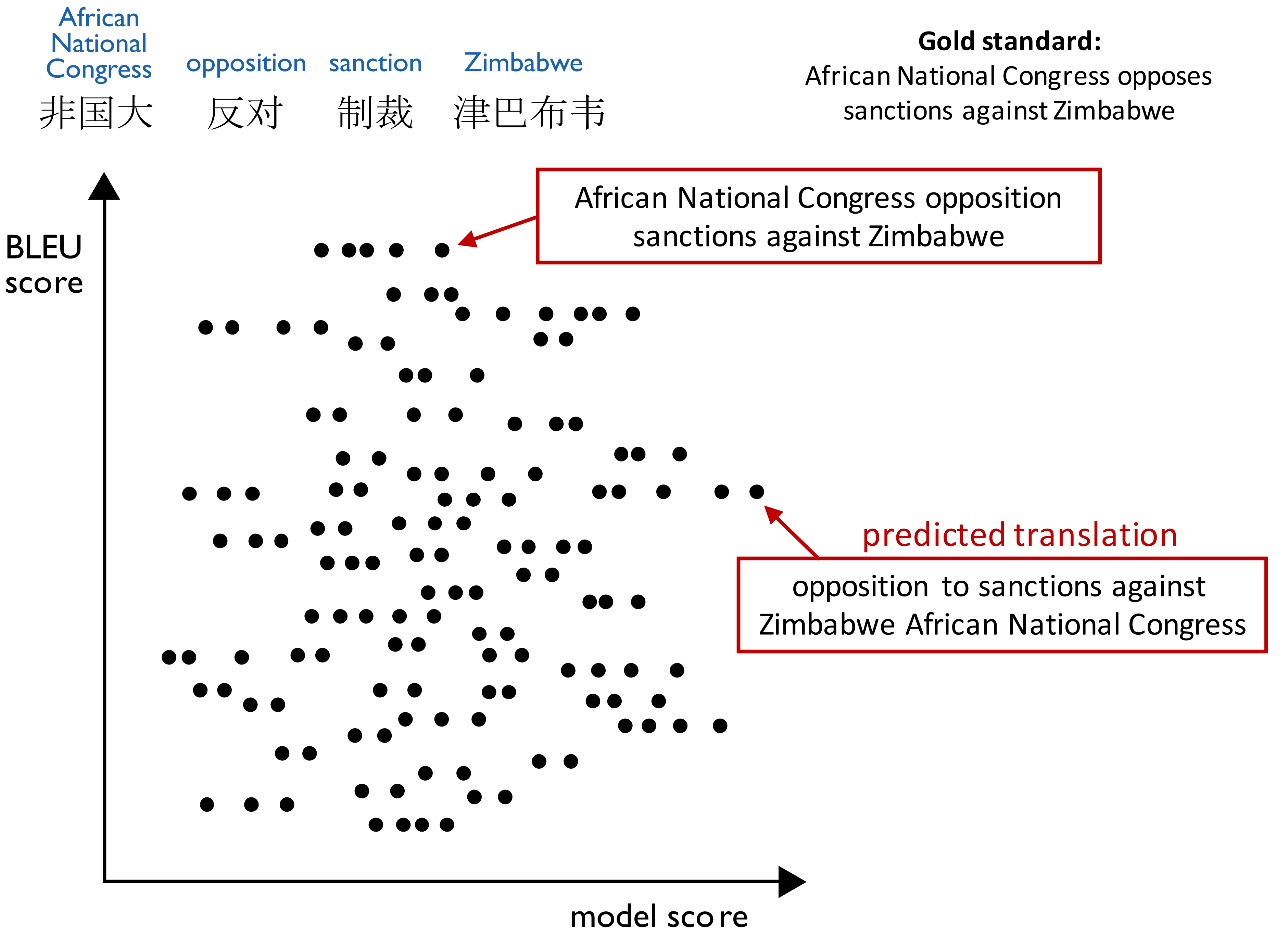
African National Congress opposes sanctions against Zimbabwe

African National Congress | opposition | sanction | Zimbabwe
非国大 反对 制裁 津巴布韦

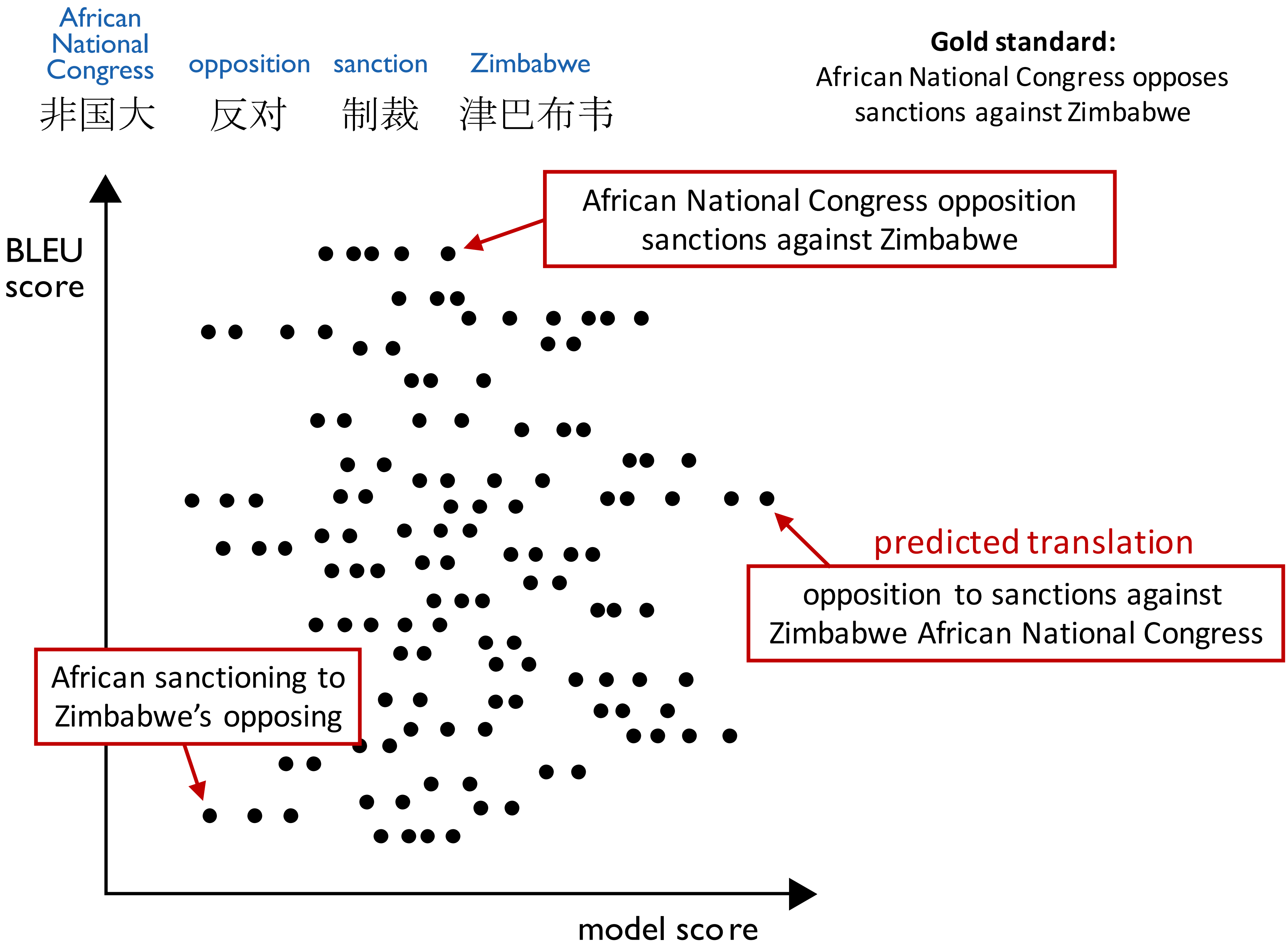
Gold standard:
African National Congress opposes sanctions against Zimbabwe




African National Congress
opposition
sanction
Zimbabwe
非国大
反对
制裁
津巴布韦
Gold standard:
African National Congress opposes sanctions against Zimbabwe
BLEU score
model score
predicted translation
opposition to sanctions against Zimbabwe African National Congress



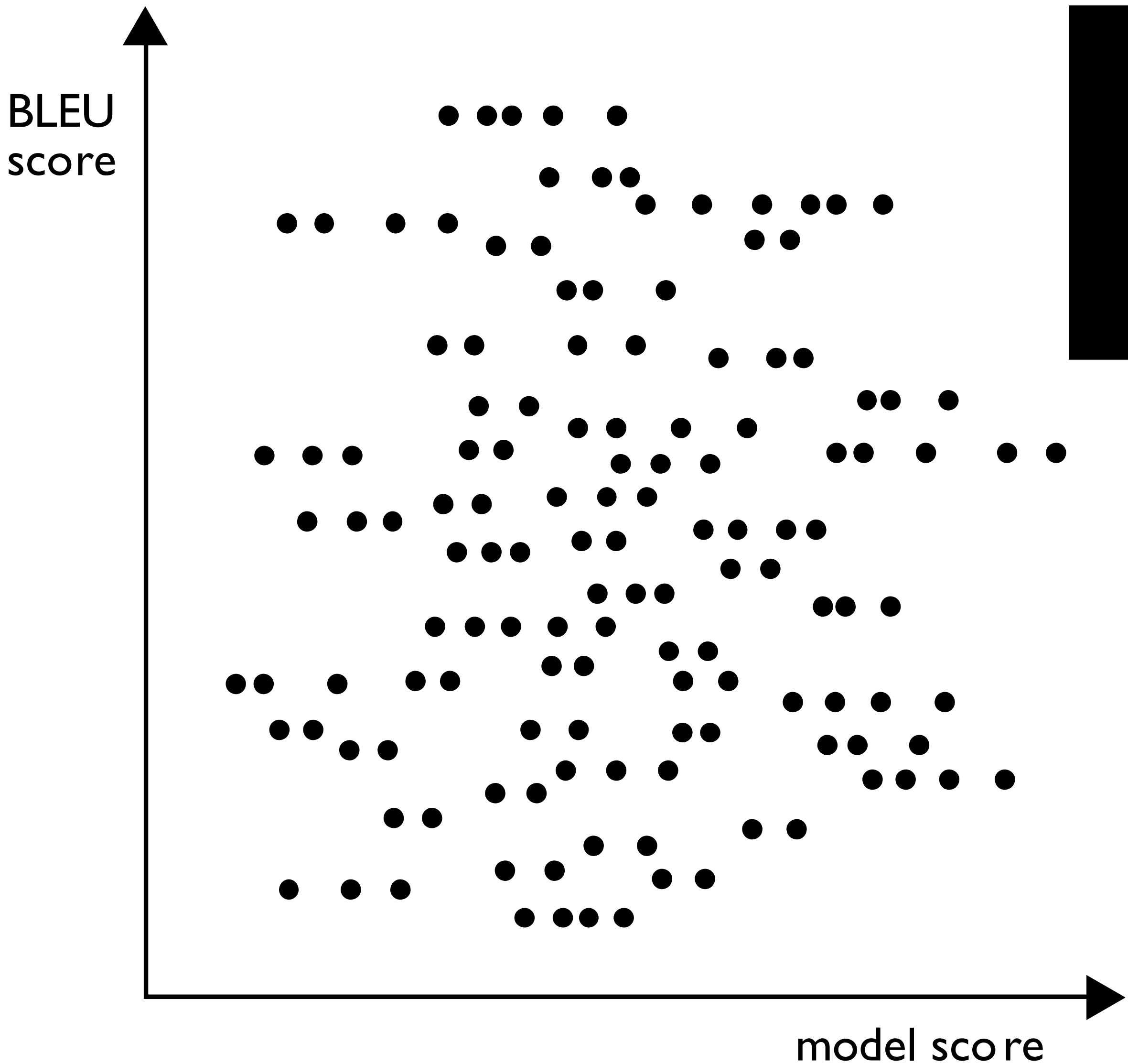
African National Congress
opposition
sanction
Zimbabwe
非国大
反对
制裁
津巴布韦
Gold standard:
African National Congress opposes sanctions against Zimbabwe
African National Congress opposition sanctions against Zimbabwe
BLEU score
predicted translation
opposition to sanctions against Zimbabwe African National Congress
model score


African National Congress
opposition
sanction
Zimbabwe
非国大
反对
制裁
津巴布韦
Gold standard:
African National Congress opposes sanctions against Zimbabwe
African National Congress opposition sanctions against Zimbabwe
BLEU score
predicted translation
opposition to sanctions against Zimbabwe African National Congress
African sanctioning to Zimbabwe's opposing
model score

African National Congress　opposition　sanction　Zimbabwe

非国大　反对　制裁　津巴布韦

Gold standard:
African National Congress opposes sanctions against Zimbabwe

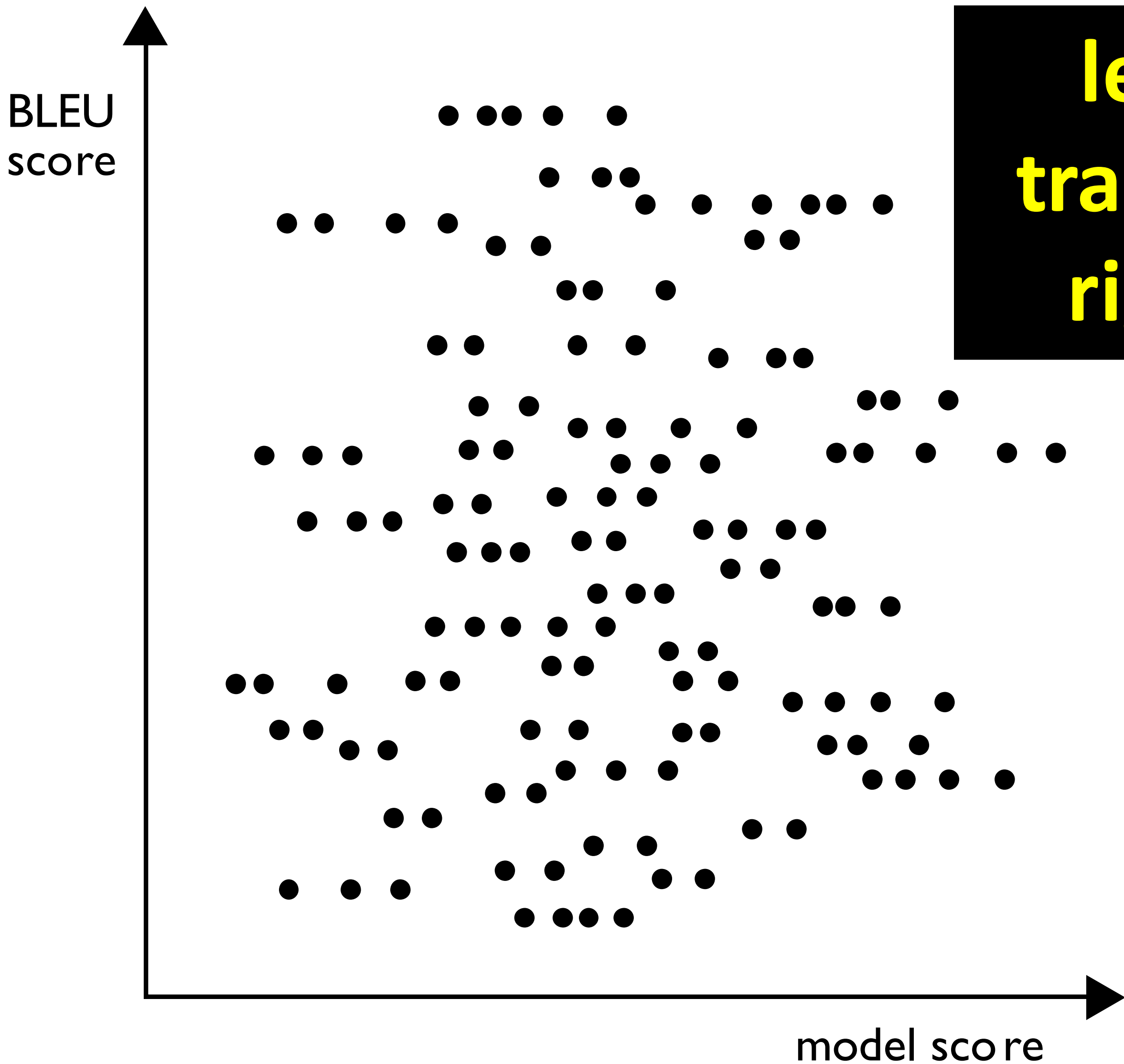


learning moves translations in this plot

African National Congress	opposition	sanction	Zimbabwe
非国大	反对	制裁	津巴布韦

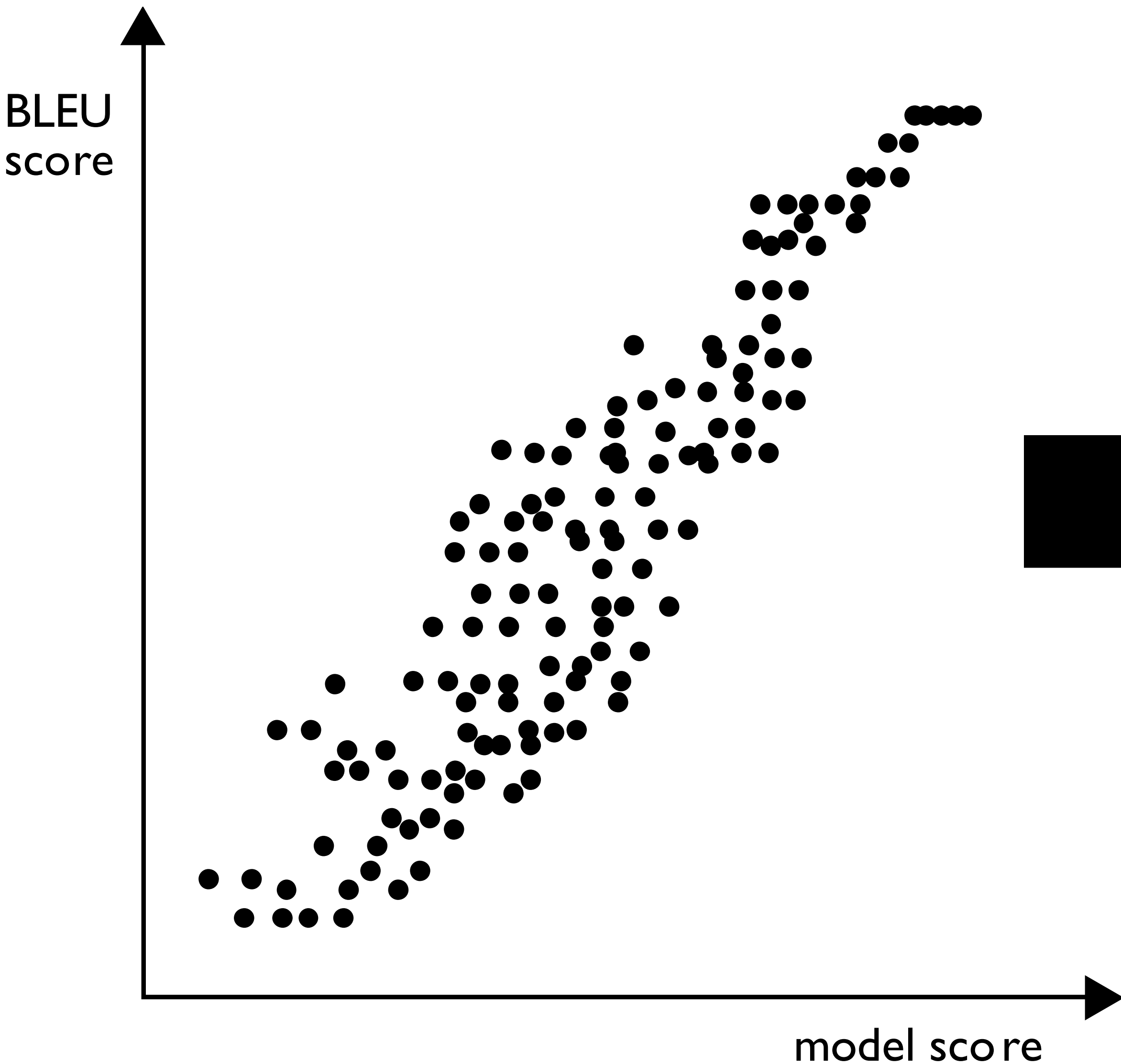
Gold standard:
African National Congress opposes sanctions against Zimbabwe

learning moves translations left or right in this plot



African National Congress	opposition	sanction	Zimbabwe
非国大	反对	制裁	津巴布韦

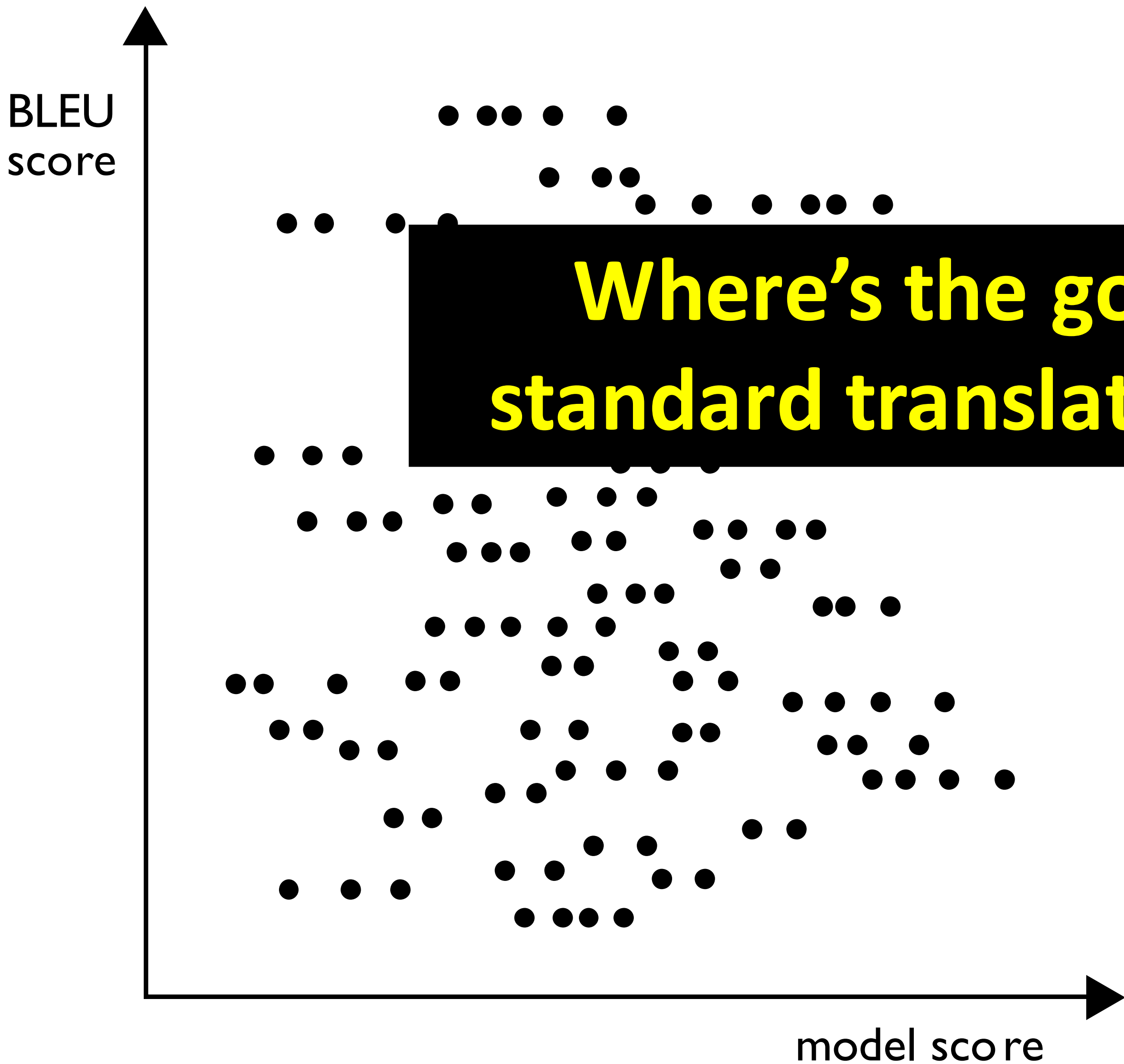
Gold standard:
African National Congress opposes sanctions against Zimbabwe



“ideal” model

African National Congress	opposition	sanction	Zimbabwe
非国大	反对	制裁	津巴布韦

Gold standard:
African National Congress opposes sanctions against Zimbabwe

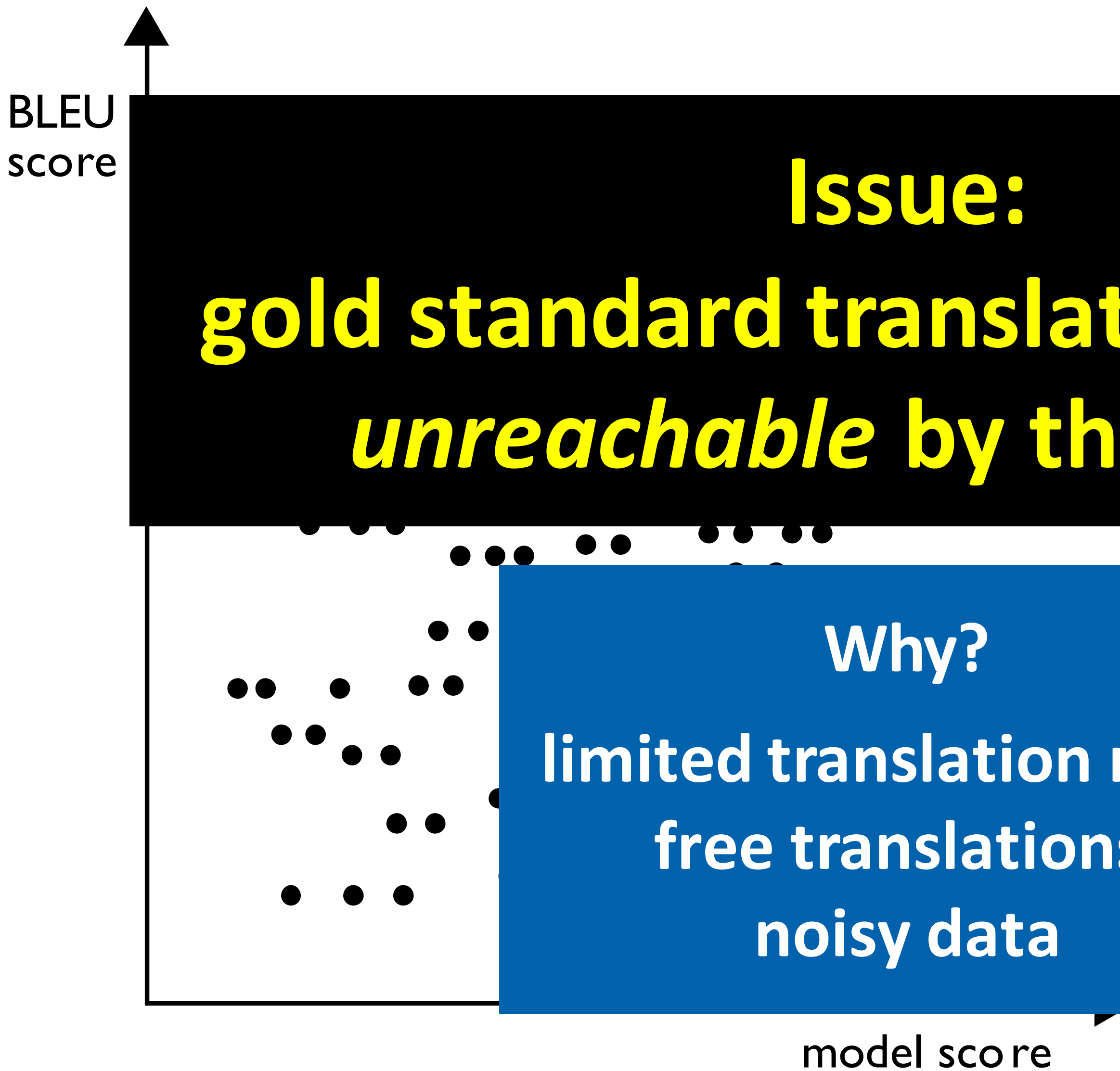


Where's the gold standard translation?

African National Congress | opposition | sanction | Zimbabwe

非国大 反对 制裁 津巴布韦

Gold standard:
African National Congress opposes sanctions against Zimbabwe



Issue:
gold standard translation is often *unreachable* by the model

Why?

limited translation rules,
free translations,
noisy data

Free Translations

Machine translation:
Sharon's office said, leader of the main opposition Labor Party has admitted defeat and congratulatory telephone calls to Sharon.

Human-generated translation:
According to a representative of Sharon's office, the leader of the main opposition Labor Party has admitted defeat and made the obligatory congratulating telephone call to Sharon.

Even if gold standard translation was reachable by model, we might not want to learn from it directly

to Sharon.

Applicable to other tasks:
summarization
image caption generation

Loss Functions

name	loss	where used
cost ("0-1")	$\text{cost}(y, \text{classify}(\boldsymbol{x}, \boldsymbol{\theta}))$	intractable, but underlies "direct error minimization"
perceptron	$-\text{score}(\boldsymbol{x}, y, \boldsymbol{\theta}) + \max_{y' \in \mathcal{L}} \ \text{score}(\boldsymbol{x}, y', \boldsymbol{\theta})$	perceptron algorithm (Rosenblatt, 1958)
hinge	$-\text{score}(\boldsymbol{x}, y, \boldsymbol{\theta}) + \max_{y' \in \mathcal{L}} (\text{score}(\boldsymbol{x}, y', \boldsymbol{\theta}) + \text{cost}(y, y'))$	support vector machines, other large-margin algorithms
log	$-\log p_{\boldsymbol{\theta}}(y \mid \boldsymbol{x})$ $= \text{score}(\boldsymbol{x}, y, \boldsymbol{\theta}) + \log \sum_{y' \in \mathcal{L}} \exp\{\text{score}(\boldsymbol{x}, y', \boldsymbol{\theta})\}$	logistic regression, conditional random fields, maximum entropy models

issue: gold standard translation is often unreachable by the model

name		where used
cost ("0-1")	$\text{cost}(y, \text{classify}(\boldsymbol{x}, \boldsymbol{\theta}))$	intractable, but underlies "direct error minimization"
perceptron	$-\text{score}(\boldsymbol{x}, y, \boldsymbol{\theta}) + \max_{y' \in \mathcal{L}} \text{score}(\boldsymbol{x}, y', \boldsymbol{\theta})$	perceptron algorithm (Rosenblatt, 1958)
hinge	$-\text{score}(\boldsymbol{x}, y, \boldsymbol{\theta}) + \max_{y' \in \mathcal{L}} (\text{score}(\boldsymbol{x}, y', \boldsymbol{\theta}) + \text{cost}(y, y'))$	support vector machines, other large-margin algorithms
log	$-\log p_{\boldsymbol{\theta}}(y \mid \boldsymbol{x})$ $= -\text{score}(\boldsymbol{x}, y, \boldsymbol{\theta}) + \log \sum_{y' \in \mathcal{L}} \exp\{\text{score}(\boldsymbol{x}, y', \boldsymbol{\theta})\}$	logistic regression, conditional random fields, maximum entropy models

intractable, but it doesn't need to compute model score of gold standard!

name		where used
cost ("0-1")	$\text{cost}(y, \text{classify}(\boldsymbol{x}, \boldsymbol{\theta}))$	intractable, but underlies "direct error minimization"
perceptron	$-\text{score}(\boldsymbol{x}, y, \boldsymbol{\theta}) + \max_{y' \in \mathcal{L}} \text{score}(\boldsymbol{x}, y', \boldsymbol{\theta})$	perceptron algorithm (Rosenblatt, 1958)
hinge	$-\text{score}(\boldsymbol{x}, y, \boldsymbol{\theta}) + \max_{y' \in \mathcal{L}} (\text{score}(\boldsymbol{x}, y', \boldsymbol{\theta}) + \text{cost}(y, y'))$	support vector machines, other large-margin algorithms
log	$- \log p_{\boldsymbol{\theta}}(y \mid \boldsymbol{x}) = \text{score}(\boldsymbol{x}, y, \boldsymbol{\theta}) + \log \sum_{y' \in \mathcal{L}} \exp\{\text{score}(\boldsymbol{x}, y', \boldsymbol{\theta})\}$	logistic regression, conditional random fields, maximum entropy models

MERT, Och (2003)

Minimum Error Rate Training in Statistical Machine Translation

Franz Josef Och
Information Sciences Institute
University of Southern California
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292
och@isi.edu

Notation

feature weights

feature vector

$$\langle \boldsymbol{y}^*, \boldsymbol{h}^* \rangle = \underset{\langle \boldsymbol{y}, \boldsymbol{h} \rangle \in \mathcal{T}_{\boldsymbol{x}}}{\operatorname{argmax}} \; \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{h})$$

source sentence

translation

latent derivation

Minimum Error Rate Training (MERT)

$$\min_{\boldsymbol{\theta}} \quad \text{cost}\left(\{\boldsymbol{y}^{(i)}\}_{i=1}^{N}, \left\{ \operatorname*{argmax}_{\langle \boldsymbol{y}, \boldsymbol{h} \rangle \in \mathcal{T}_{\boldsymbol{x}^{(i)}}} \quad \boldsymbol{\theta}^{\top} \boldsymbol{f}(\boldsymbol{x}^{(i)}, \boldsymbol{y}, \boldsymbol{h}) \right\}_{i=1}^{N} \right)$$

Minimum Error Rate Training (MERT)

$$\min_{\boldsymbol{\theta}} \ \text{cost}\left(\underbrace{\{\boldsymbol{y}^{(i)}\}_{i=1}^{N}}_{\text{references}}, \underbrace{\left\{\operatorname*{argmax}_{\langle \boldsymbol{y},\boldsymbol{h}\rangle \in \mathcal{T}_{\overbrace{\boldsymbol{x}^{(i)}}^{\text{set of source sentences}}}} \boldsymbol{\theta}^{\top}\boldsymbol{f}(\boldsymbol{x}^{(i)},\boldsymbol{y},\boldsymbol{h})\right\}_{i=1}^{N}}_{\text{decoder outputs}}\right)$$

Minimum Error Rate Training (MERT)

“how bad are these translations?”
e.g., negative BLEU

set of source sentences

$$\min_{\boldsymbol{\theta}} \ \text{cost}\left(\underbrace{\{\boldsymbol{y}^{(i)}\}_{i=1}^{N}}_{\text{references}}, \underbrace{\left\{ \operatorname*{argmax}_{\langle \boldsymbol{y},\boldsymbol{h}\rangle \in \mathcal{T}_{\boldsymbol{x}^{(i)}}} \ \boldsymbol{\theta}^{\top}\boldsymbol{f}(\boldsymbol{x}^{(i)},\boldsymbol{y},\boldsymbol{h}) \right.}_{\text{decoder outputs}} \left. \right\}_{i=1}^{N} \right)$$

Minimum Error Rate Training (MERT)

minimize the cost of the decoder output

intractable in general – how can we solve it?

“h

$$\min_{\boldsymbol{\theta}} \ \text{cost}\left(\underbrace{\{\boldsymbol{y}^{(i)}\}_{i=1}^{N}}_{\text{references}}, \underbrace{\left\{\operatorname*{argmax}_{\langle \boldsymbol{y},\boldsymbol{h}\rangle \in \mathcal{T}_{\boldsymbol{x}^{(i)}}} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}^{(i)}, \boldsymbol{y}, \boldsymbol{h})\right\}_{i=1}^{N}}_{\text{decoder outputs}}\right)$$

Minimum Error Rate Training (MERT)

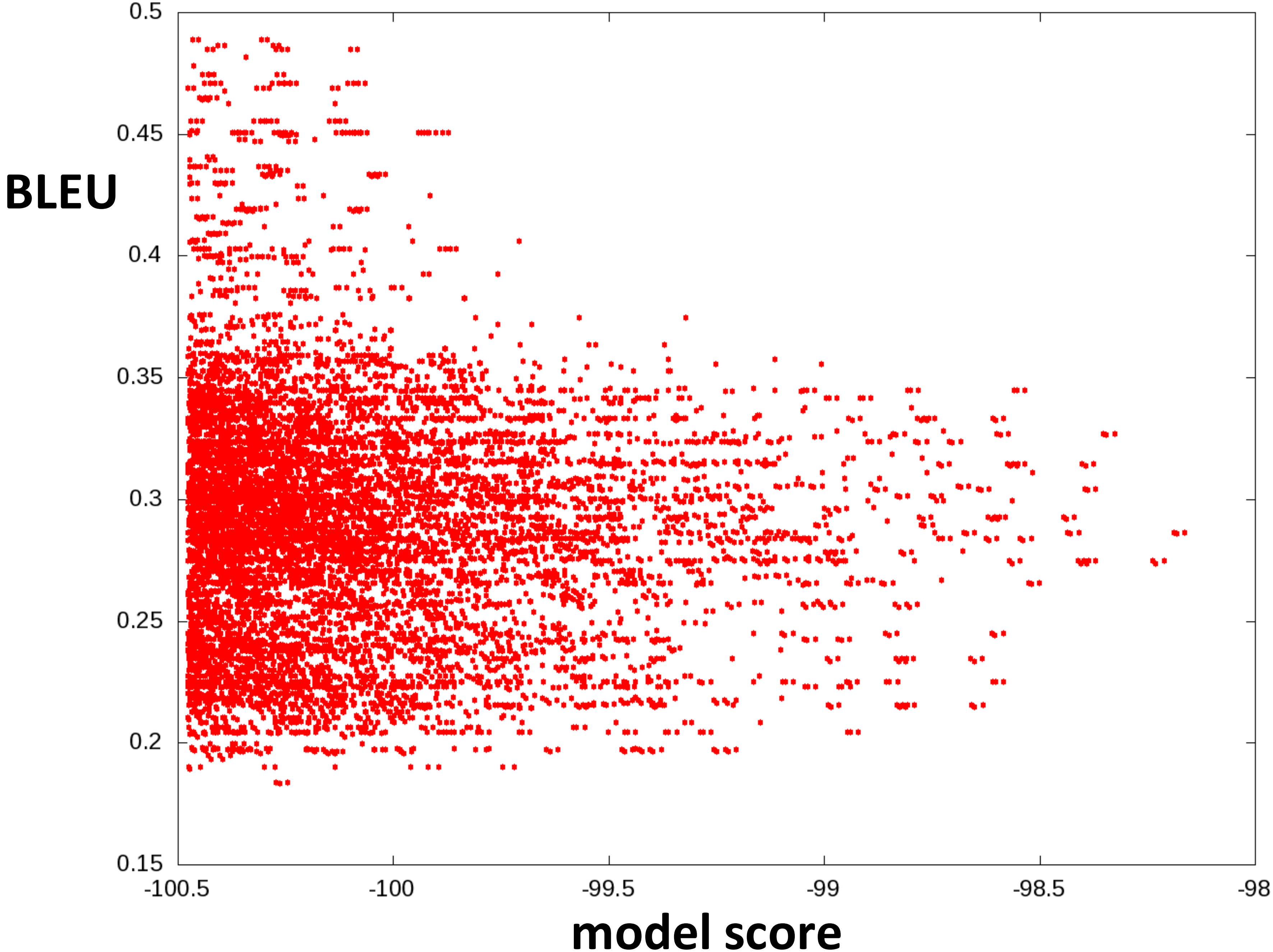
“h

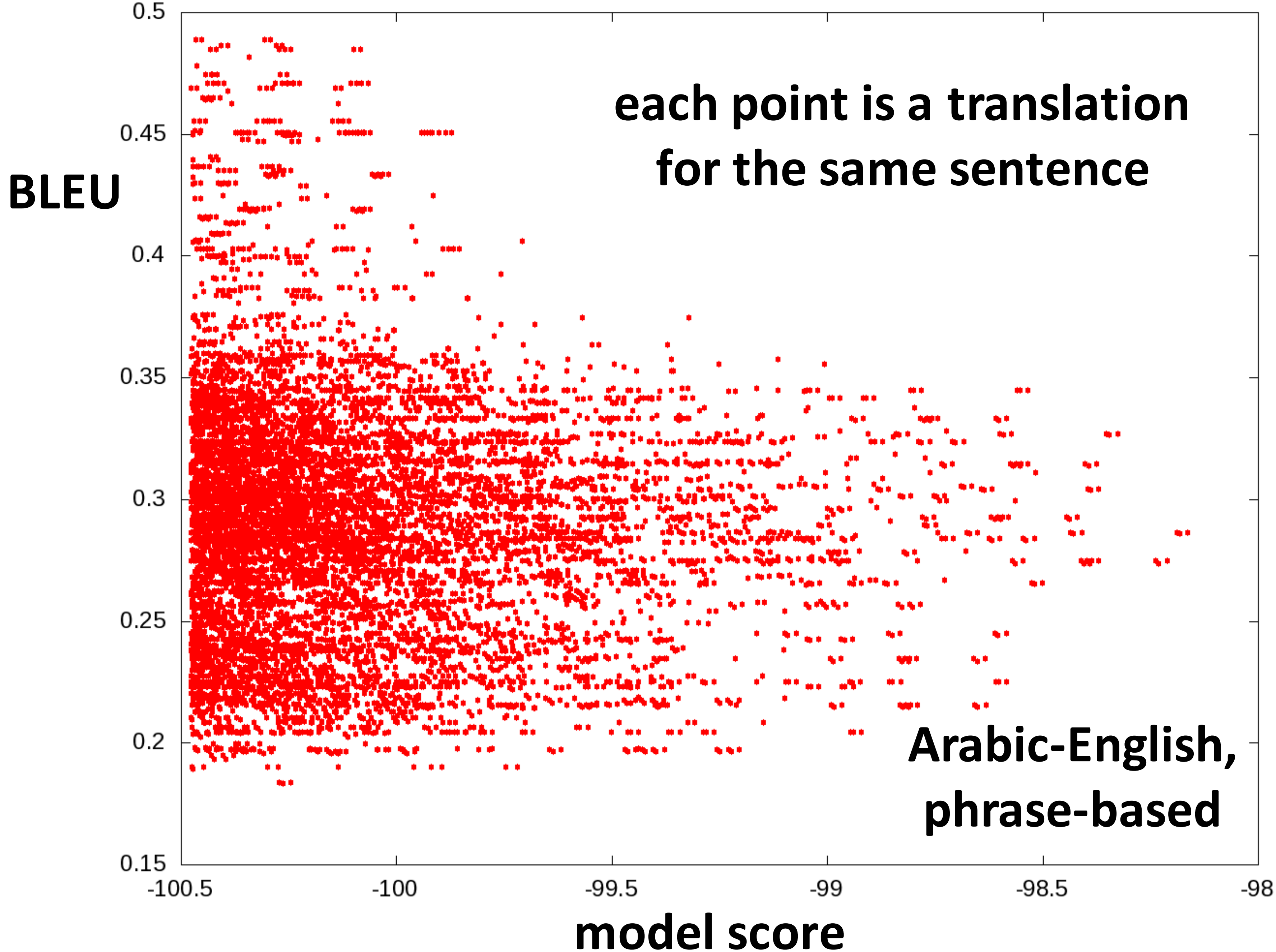
minimize the cost of the decoder output

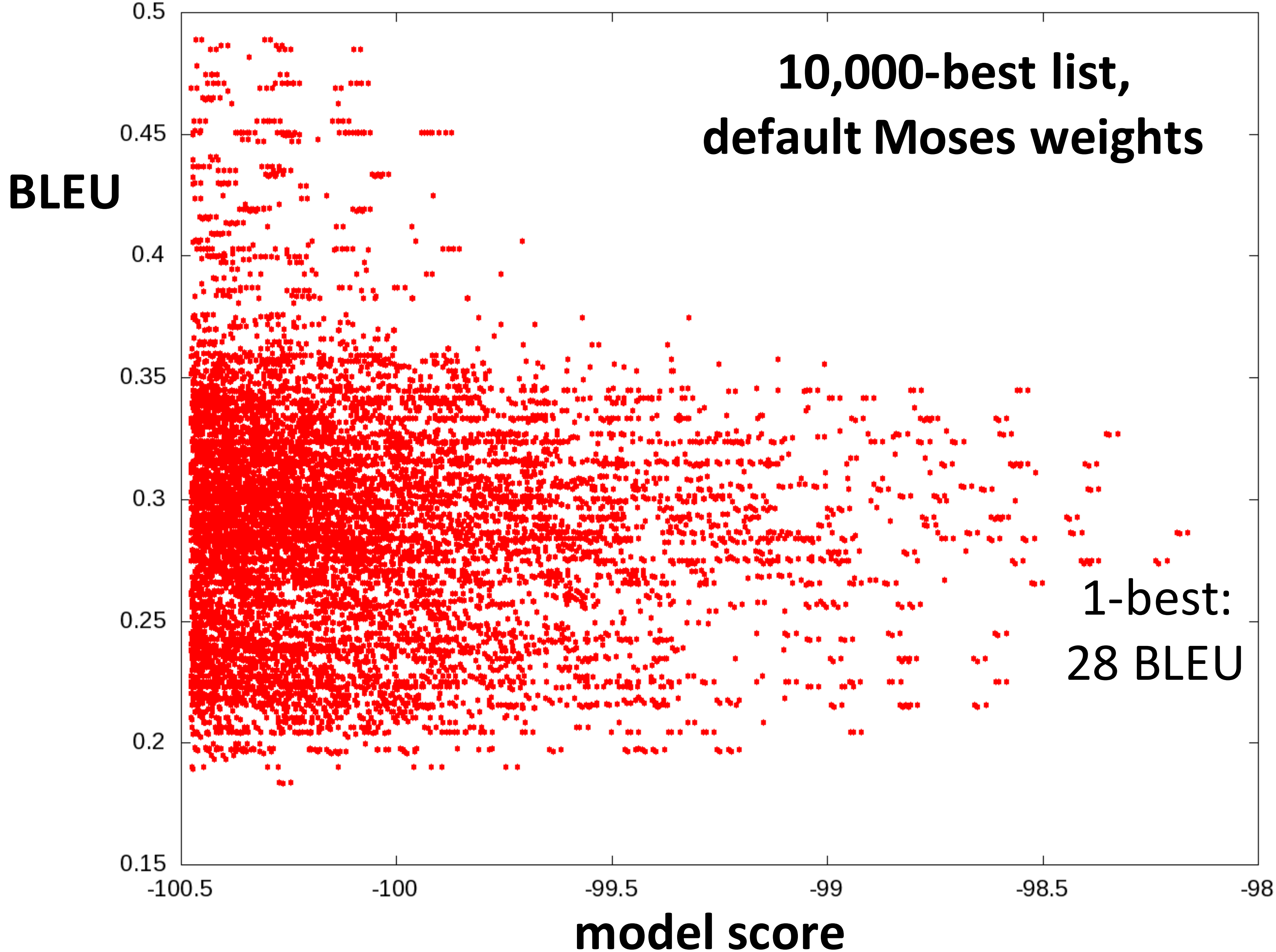
intractable in general – how can we solve it?

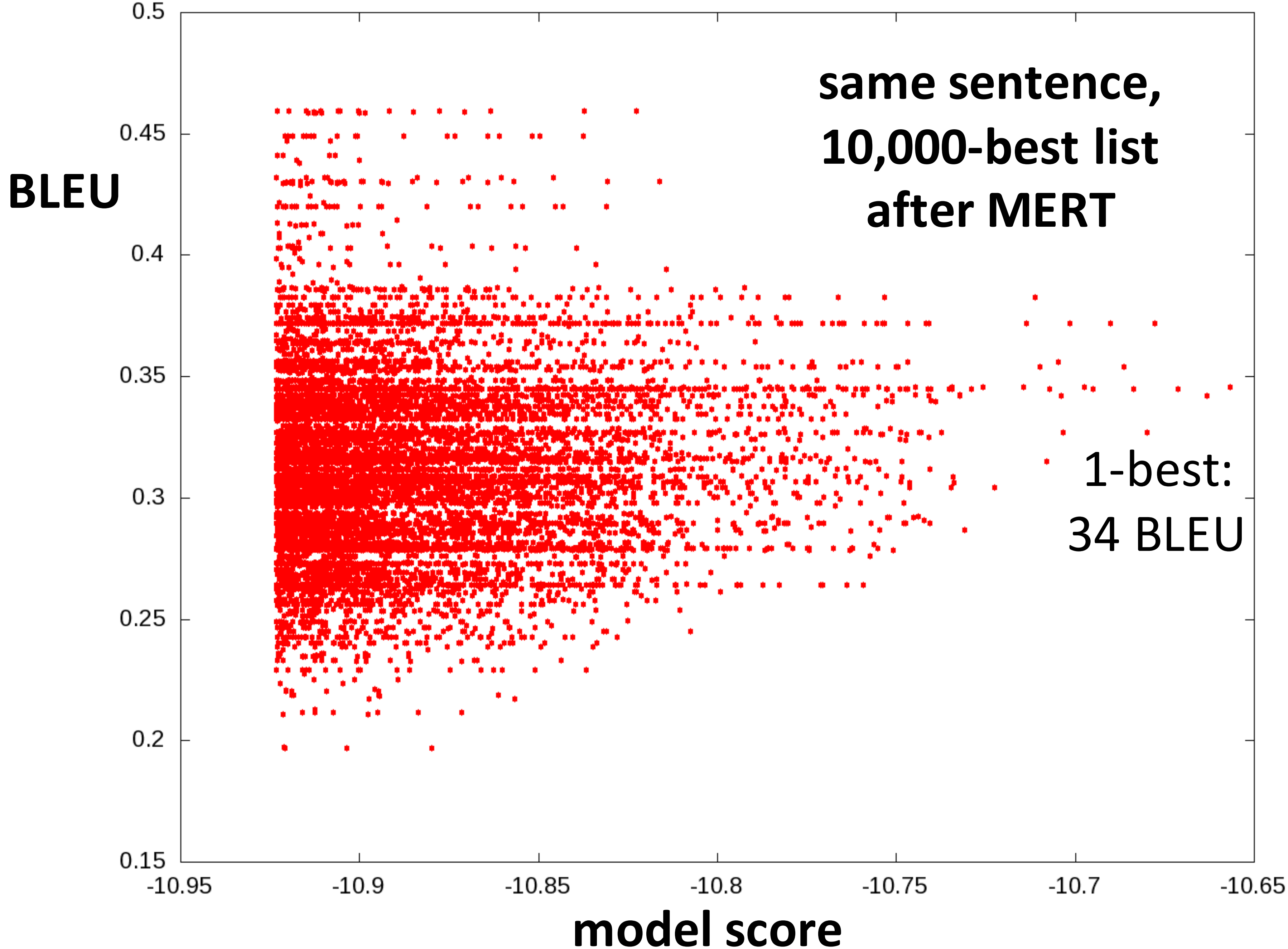
$$\min_{\theta} \; \text{cost}\left(\{\boldsymbol{y}^{(i)}\}_{i=1}^{N}, \left\{ \operatorname{argmax} \; \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}^{(i)}, \boldsymbol{y}, \boldsymbol{h}) \right\}^{N} \right)$$

generate k-best lists of translations,
approximately minimize cost on k-best lists,
repeat with new parameters
(pool k-best lists across iterates)

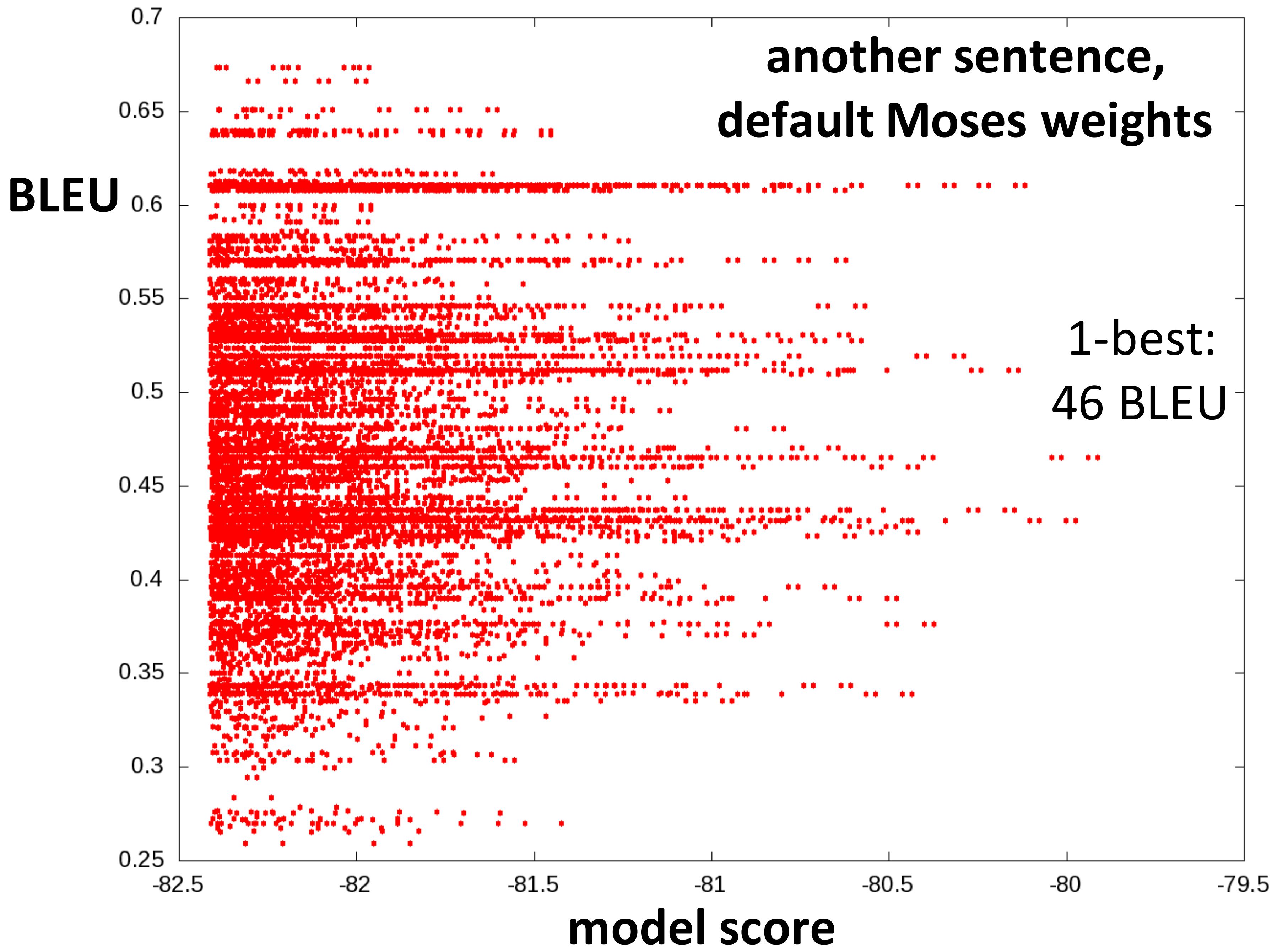

BLEU
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
-100.5
-100
-99.5
-99
-98.5
-98
model score


each point is a translation
for the same sentence
BLEU
Arabic-English,
phrase-based
model score
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
-100.5
-100
-99.5
-99
-98.5
-98

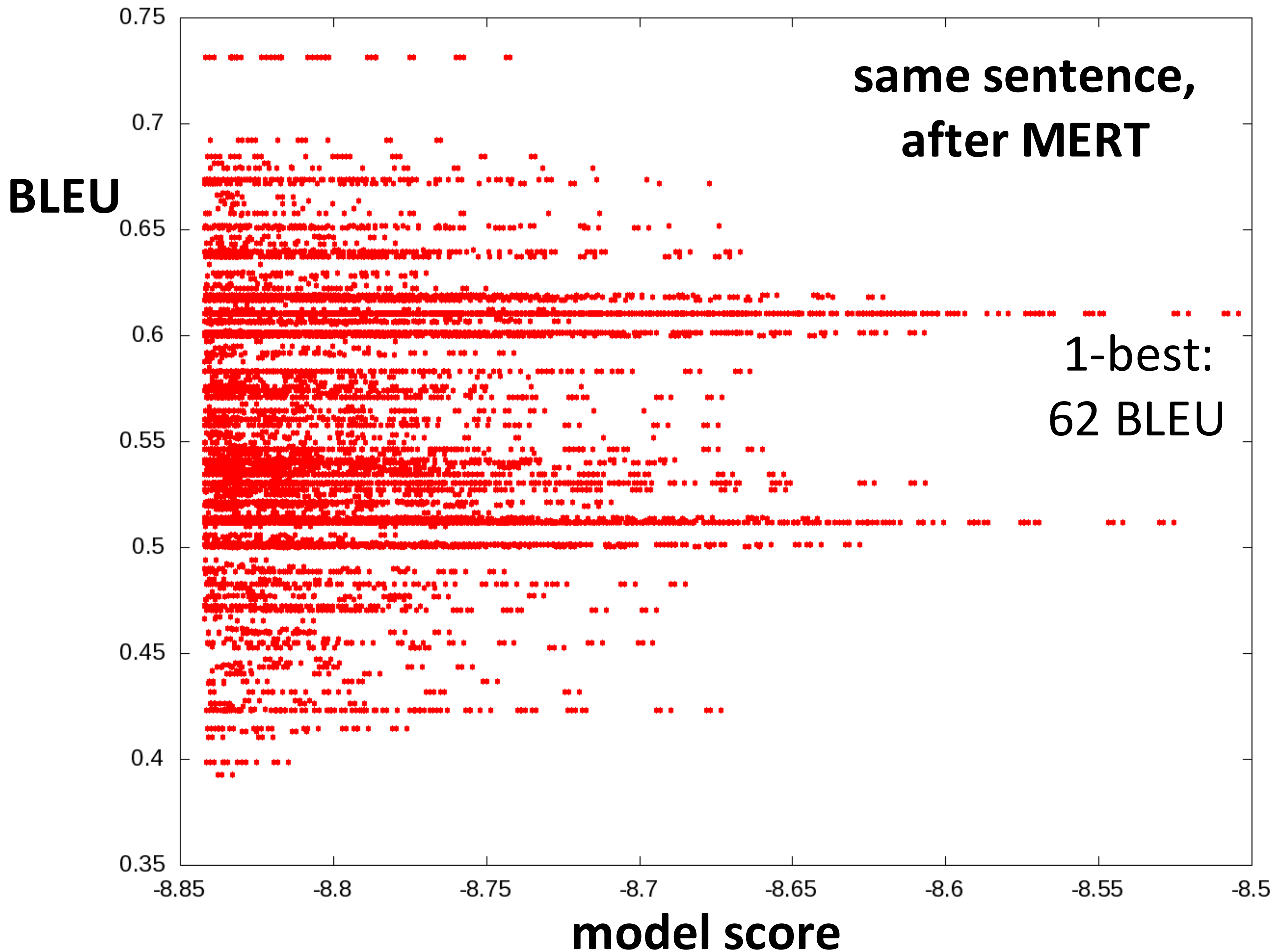

10,000-best list,
default Moses weights
BLEU
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
-100.5
-100
-99.5
-99
-98.5
-98
model score
1-best:
28 BLEU

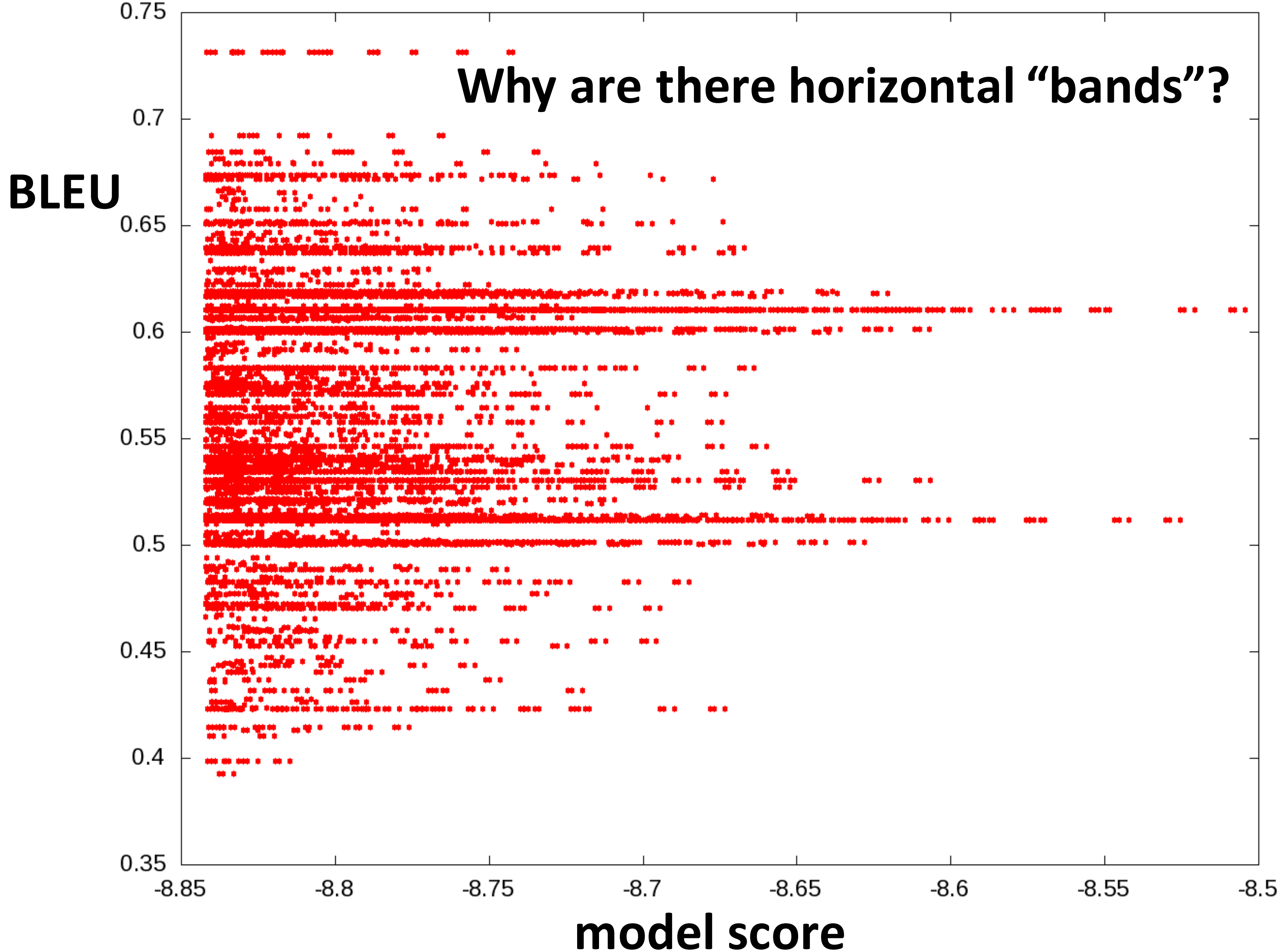


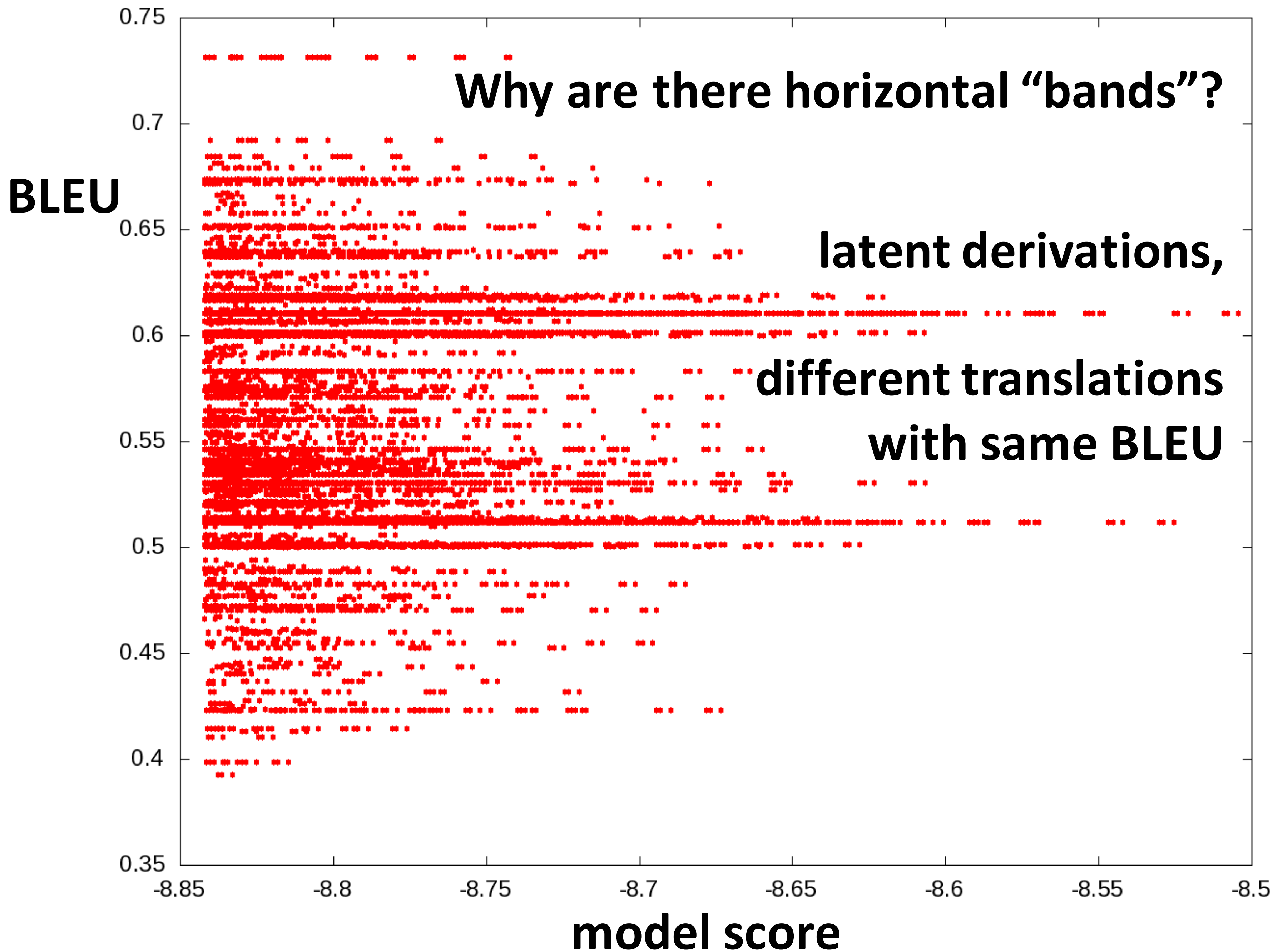
same sentence,
10,000-best list
after MERT
BLEU
1-best:
34 BLEU
model score
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
-10.95
-10.9
-10.85
-10.8
-10.75
-10.7
-10.65



another sentence,
default Moses weights
BLEU
1-best:
46 BLEU
model score
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
-82.5
-82
-81.5
-81
-80.5
-80
-79.5


same sentence,
after MERT
BLEU
1-best:
62 BLEU
model score
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0.35
-8.85
-8.8
-8.75
-8.7
-8.65
-8.6
-8.55
-8.5


Why are there horizontal "bands"?
BLEU
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0.35
-8.85
-8.8
-8.75
-8.7
-8.65
-8.6
-8.55
-8.5
model score


Why are there horizontal "bands"?
latent derivations,
different translations
with same BLEU
BLEU
model score
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0.35
-8.85
-8.8
-8.75
-8.7
-8.65
-8.6
-8.55
-8.5

$$\min_{\boldsymbol{\theta}} \ \text{cost}\left(\underbrace{\{\boldsymbol{y}^{(i)}\}_{i=1}^{N}}_{\text{references}}, \underbrace{\left\{ \operatorname*{argmax}_{\langle \boldsymbol{y}, \boldsymbol{h}\rangle \in \mathcal{T}_{\boldsymbol{x}^{(i)}}} \boldsymbol{\theta}^{\top} \boldsymbol{f}(\boldsymbol{x}^{(i)}, \boldsymbol{y}, \boldsymbol{h}) \right\}_{i=1}^{N}}_{\text{decoder outputs}} \right)$$

What are some issues with this loss function?

- Discontinuous & non-convex $\longrightarrow$ optimization relies on randomized search
- No regularization $\longrightarrow$ leads to overfitting

As a result, MERT is only effective for very small models (<40 parameters)

Many researchers tried to improve MERT:

Regularization and Search for MERT (Cer et al., 2008)

Random Restarts in MERT for MT (Moore & Quirk, 2008)

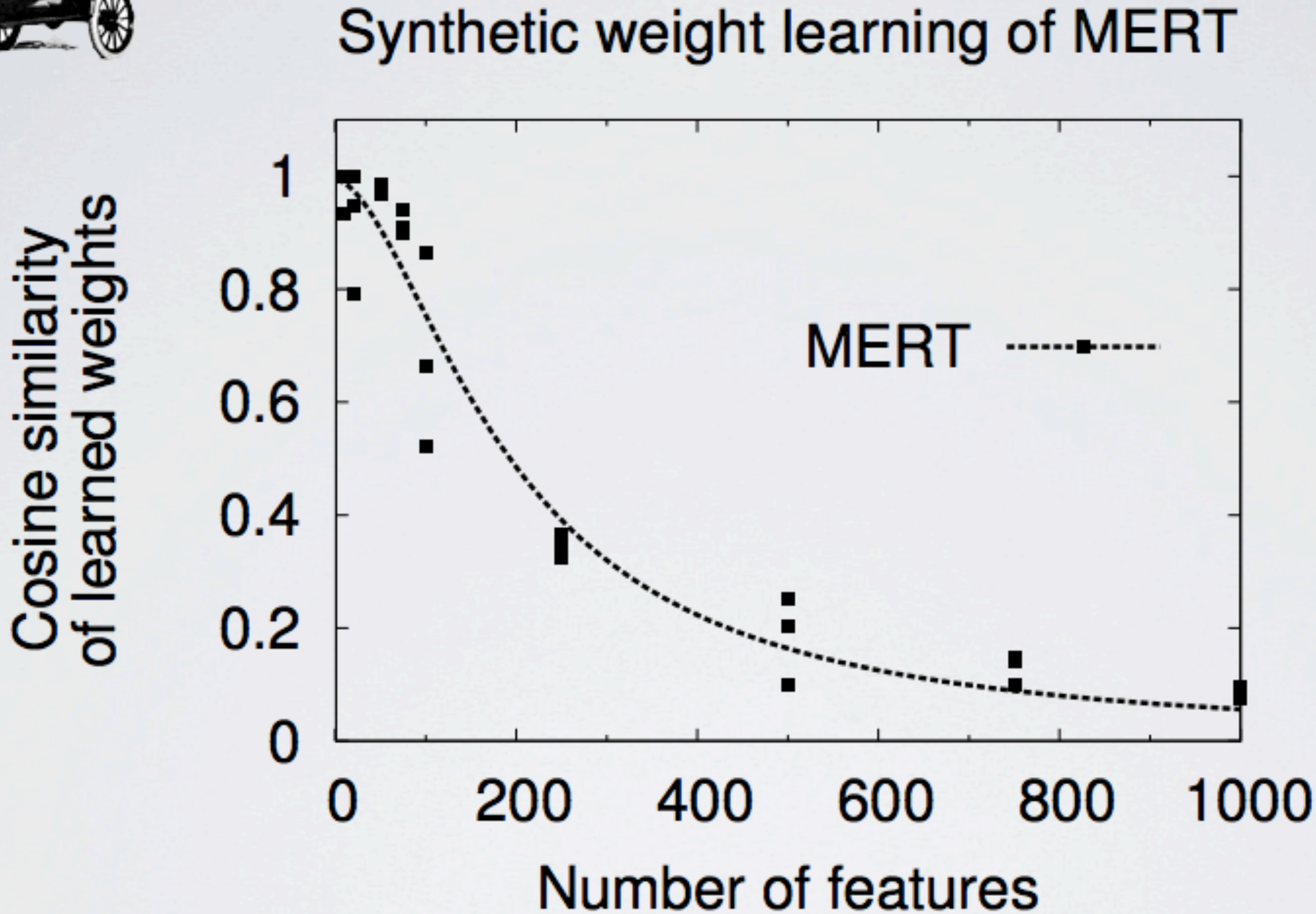
Stabilizing MERT (Foster & Kuhn, 2009)

Issues remain:

Better Hypothesis Testing for Statistical MT: Controlling for Optimizer Instability (Clark et al., 2011)

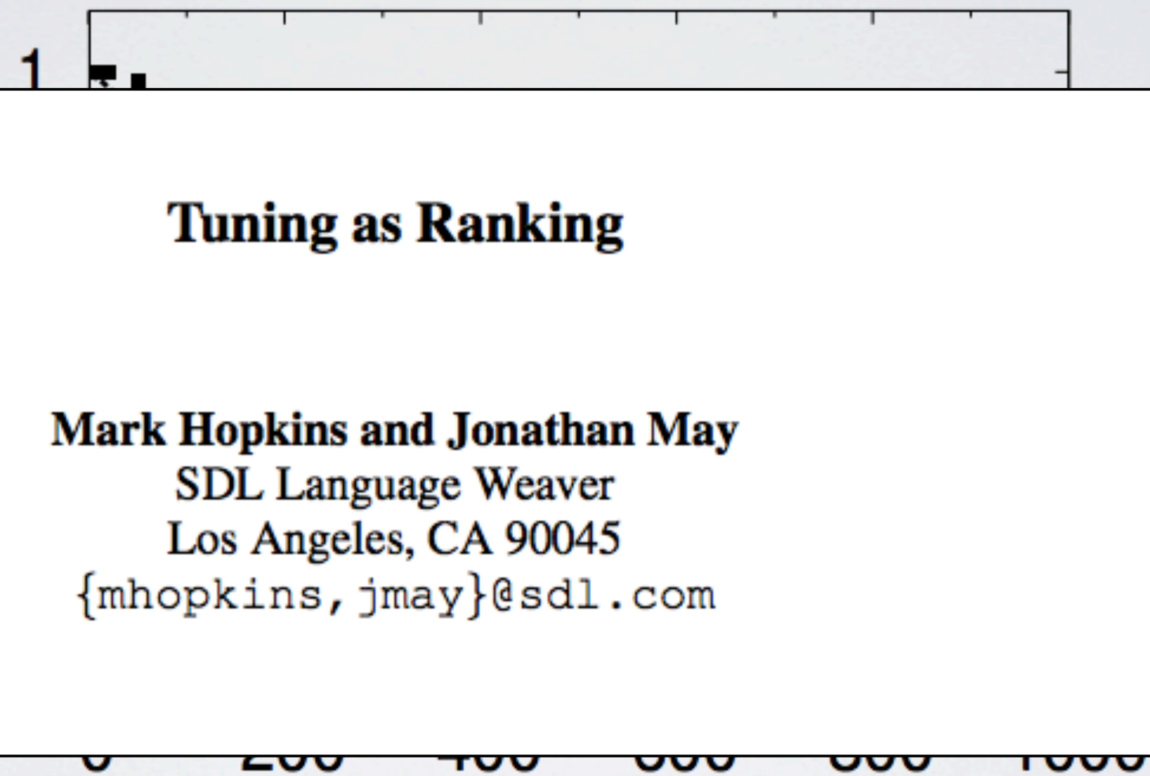
They suggest running MERT 3-5 times due to its instability

MERT ***doesn't scale***



The synthetic experiment in ideal conditions validates what has long been accepted as truth

MERT *doesn't scale*

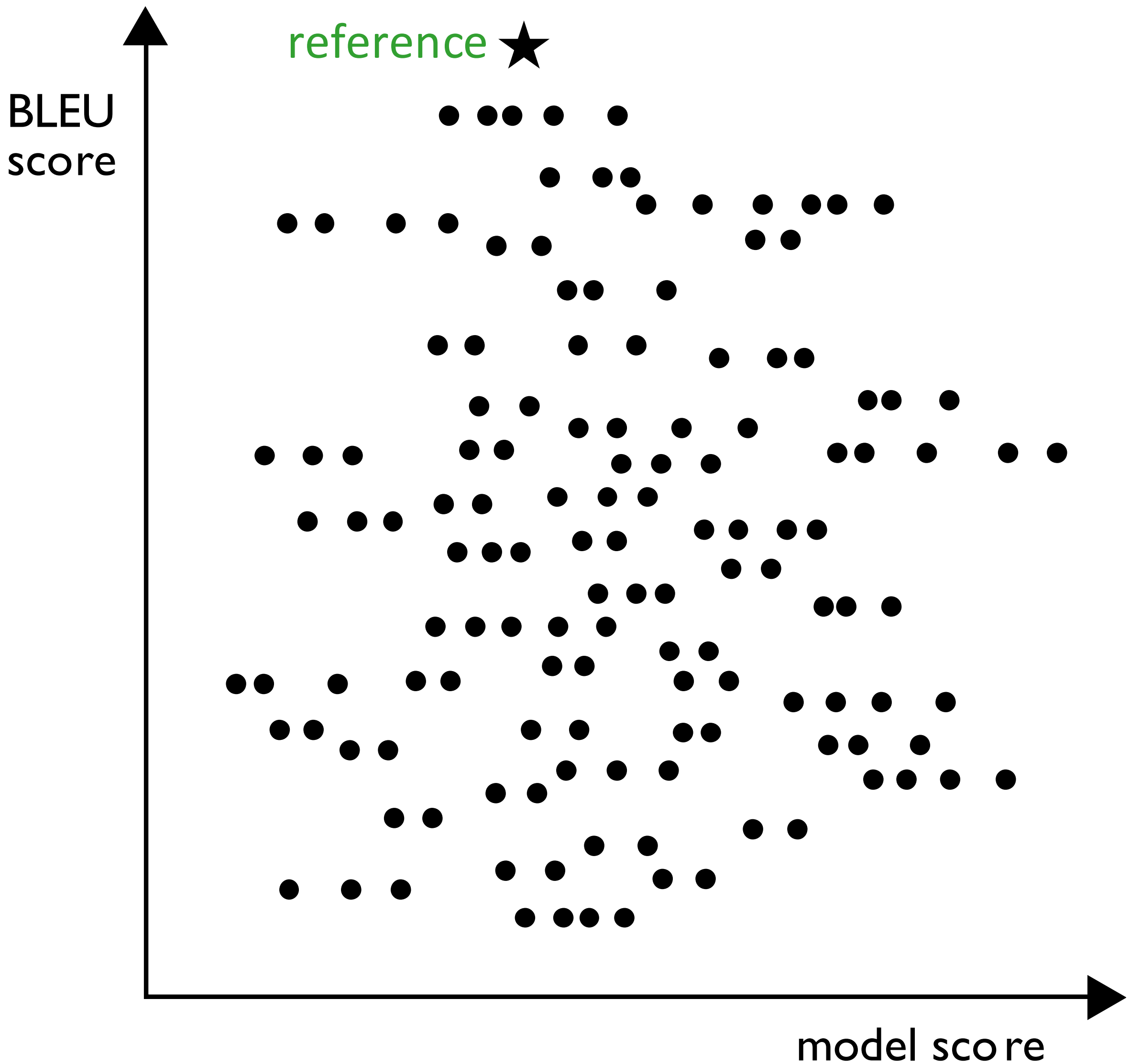


Tuning as Ranking

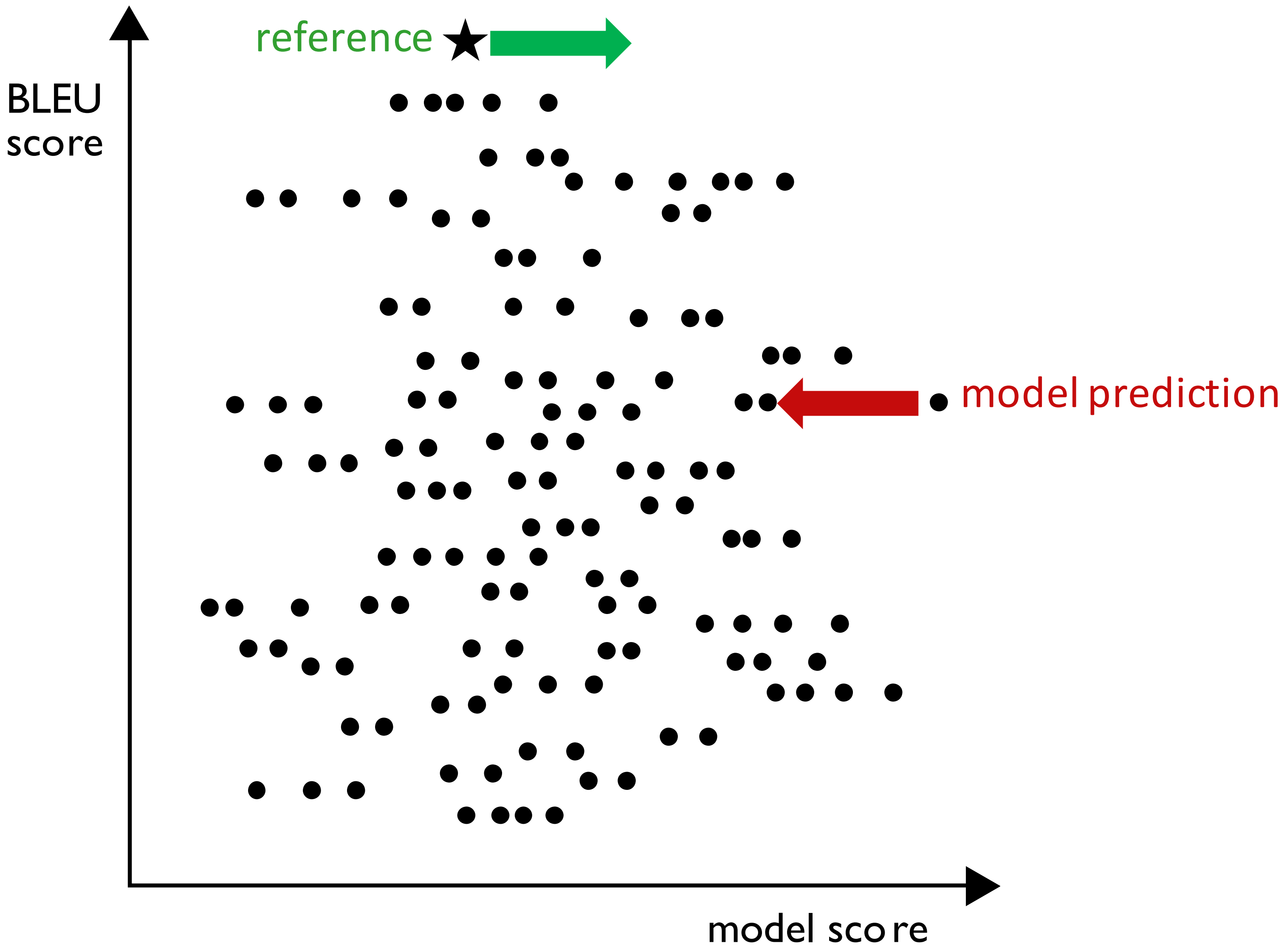
Mark Hopkins and Jonathan May
SDL Language Weaver
Los Angeles, CA 90045
{mhopkins, jmay}@sdl.com

The synthetic experiment in ideal conditions validates what has long been accepted as truth

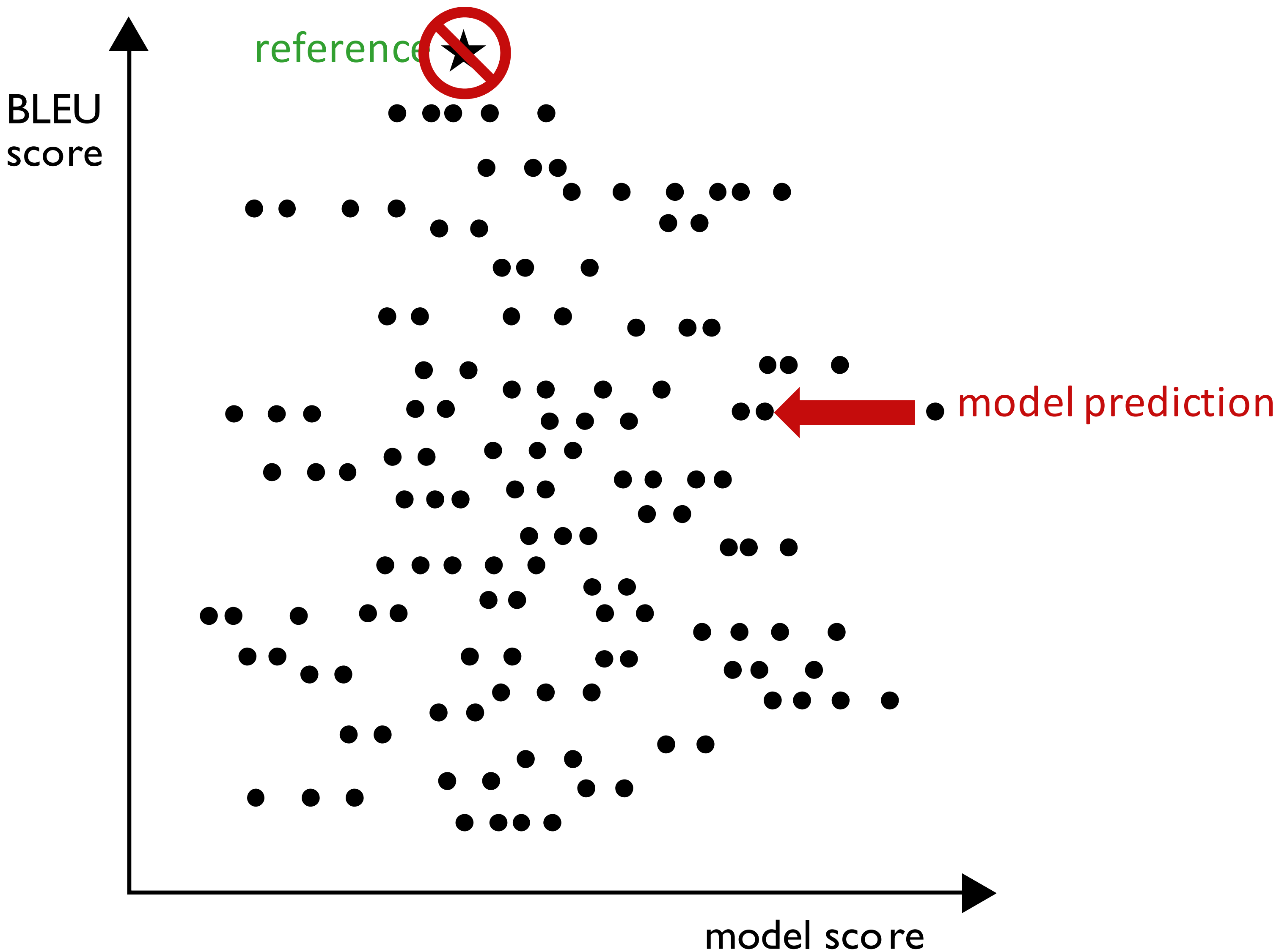
Perceptron Loss



Perceptron Loss

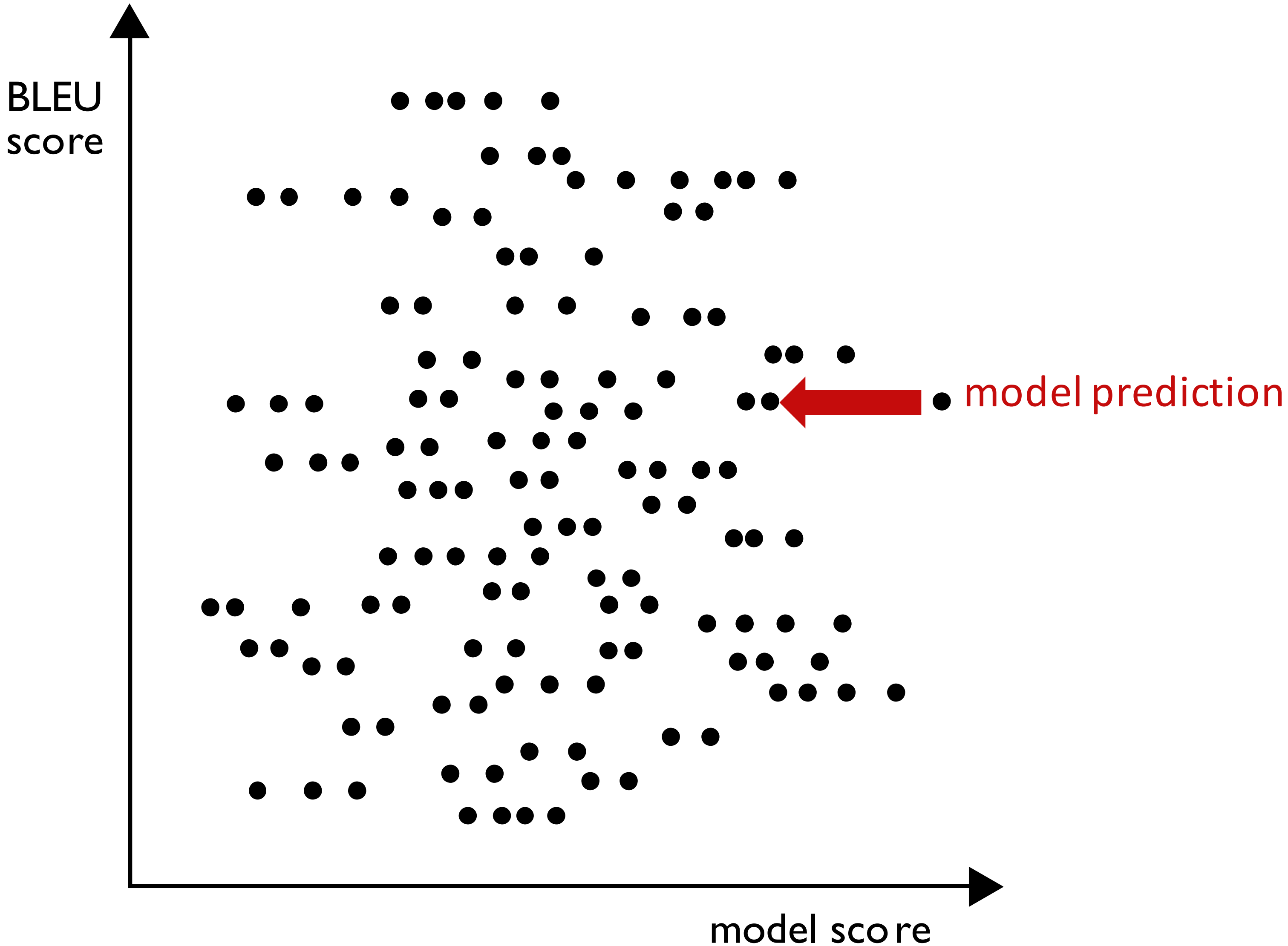


Perceptron Loss for MT?



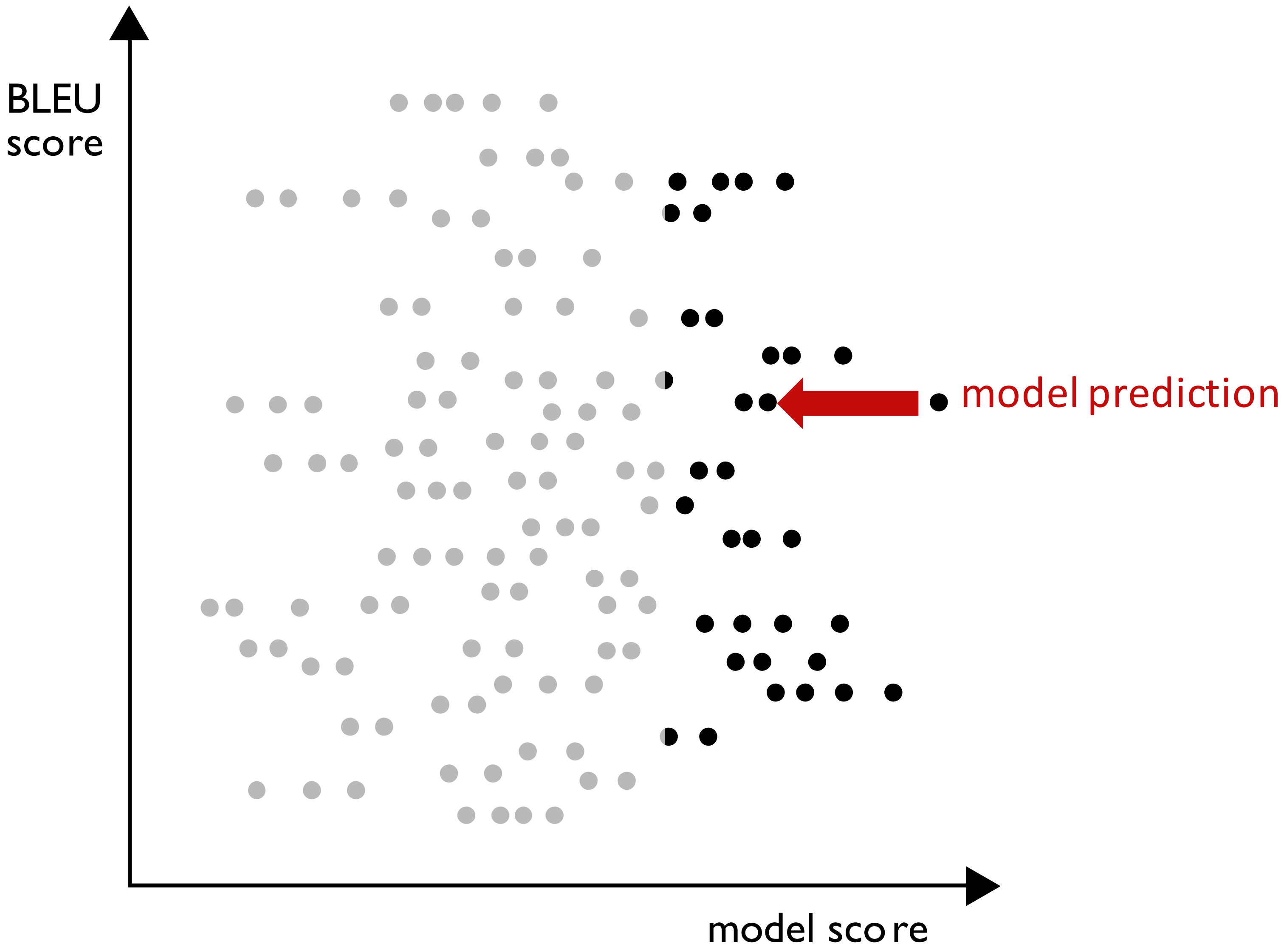
k-Best Perceptron for MT

(Liang et al., 2006)



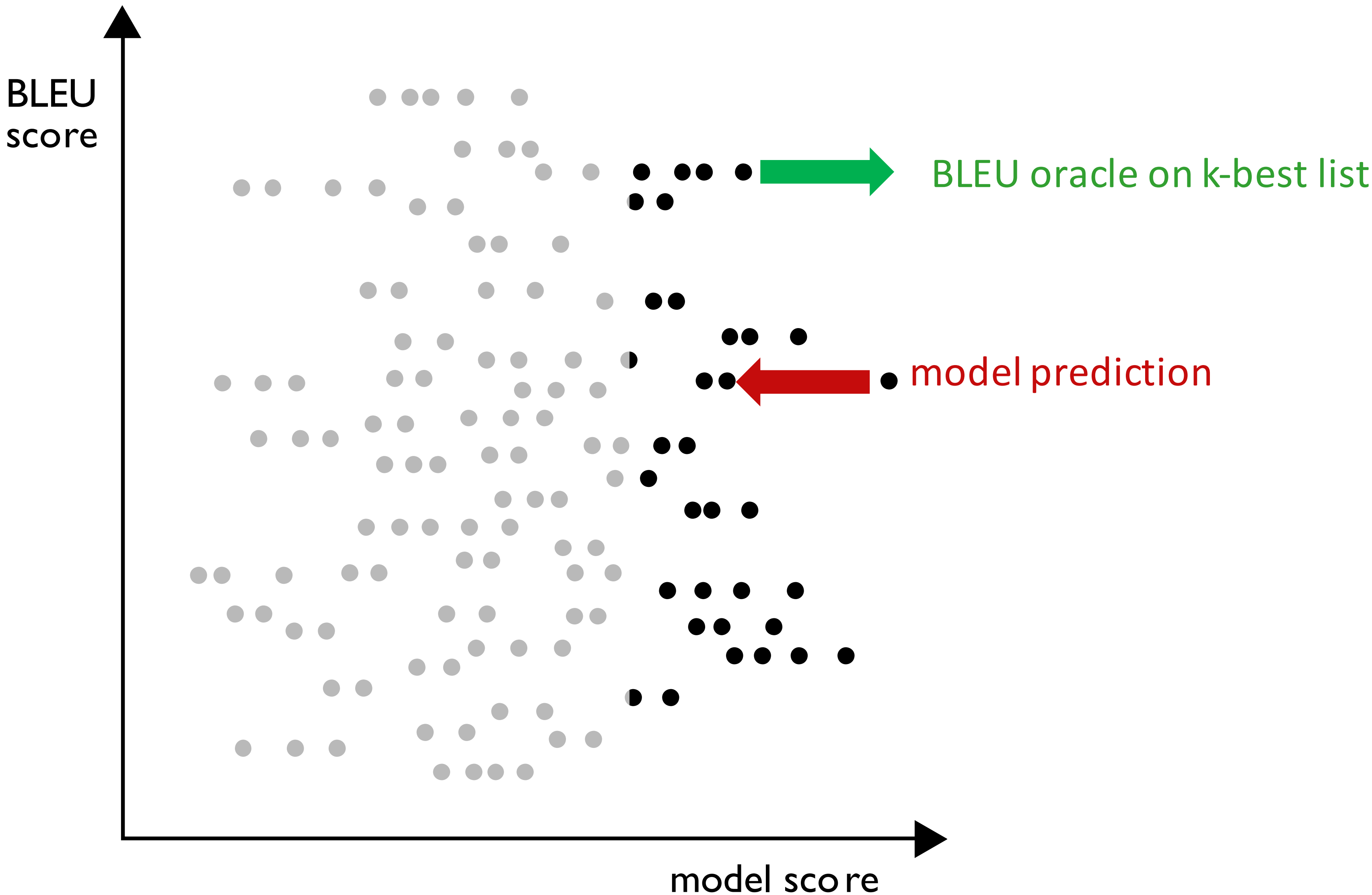
k-Best Perceptron for MT

(Liang et al., 2006)

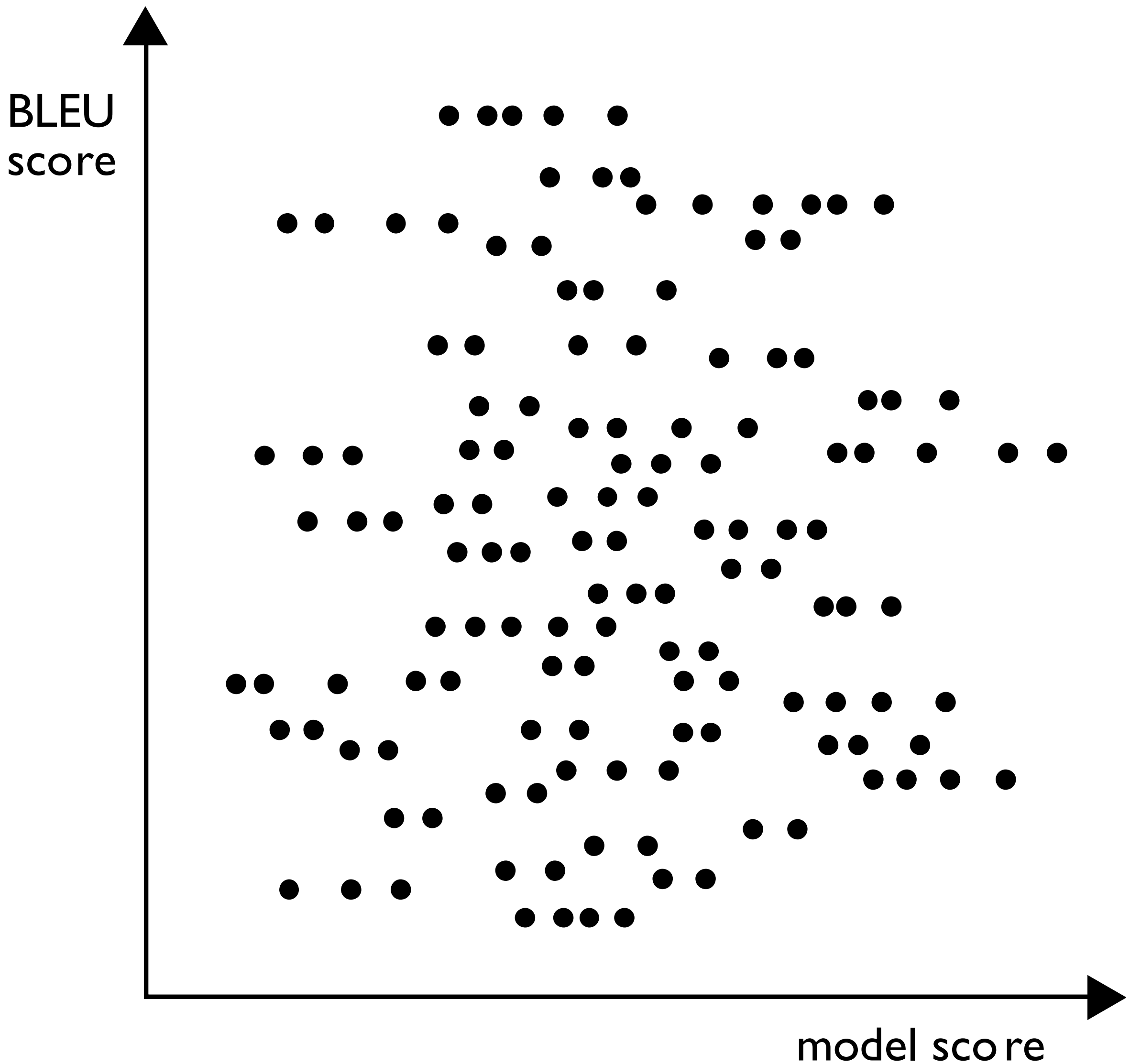


k-Best Perceptron for MT

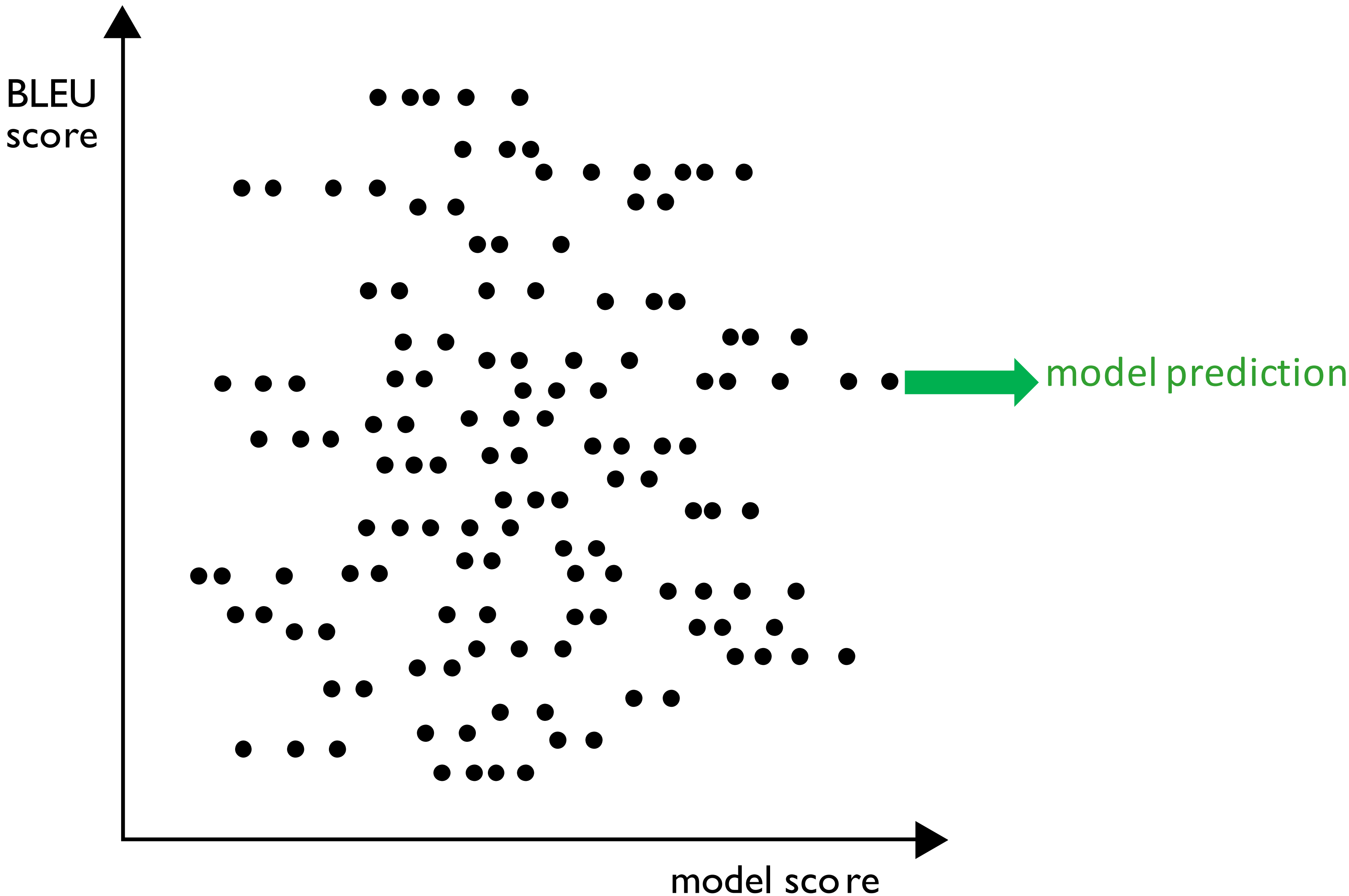
(Liang et al., 2006)



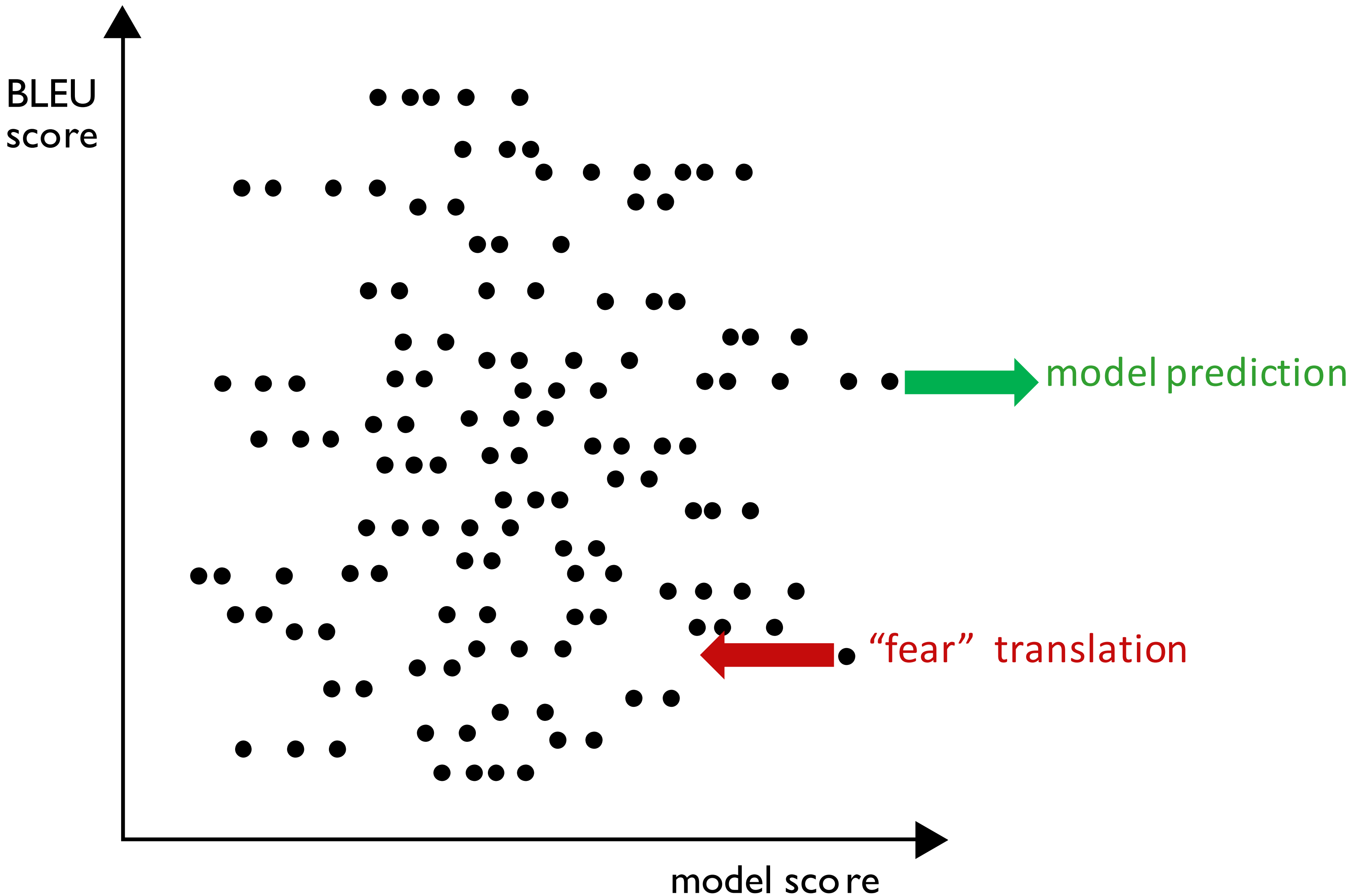
Ramp Loss Minimization



Ramp Loss Minimization

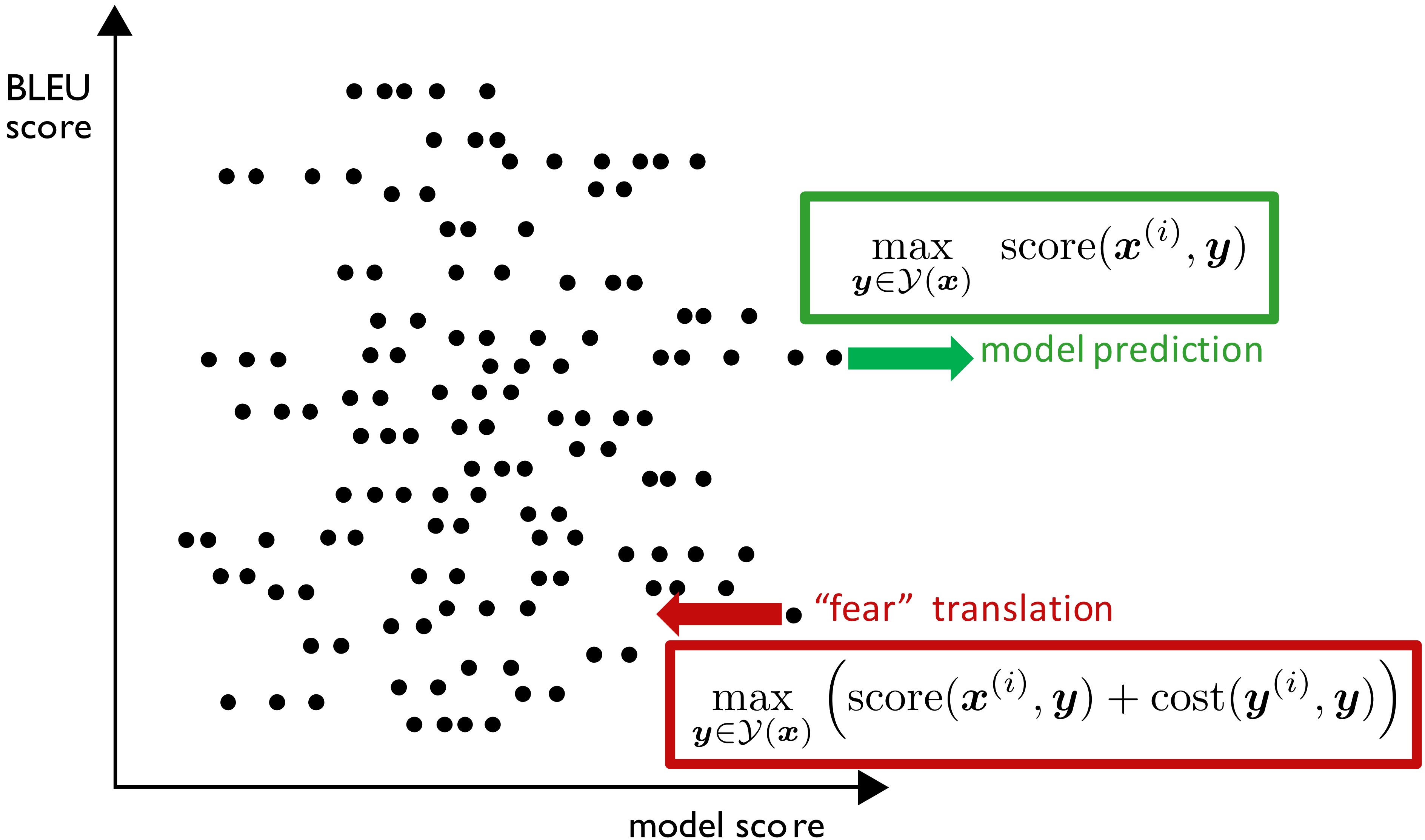


Ramp Loss Minimization



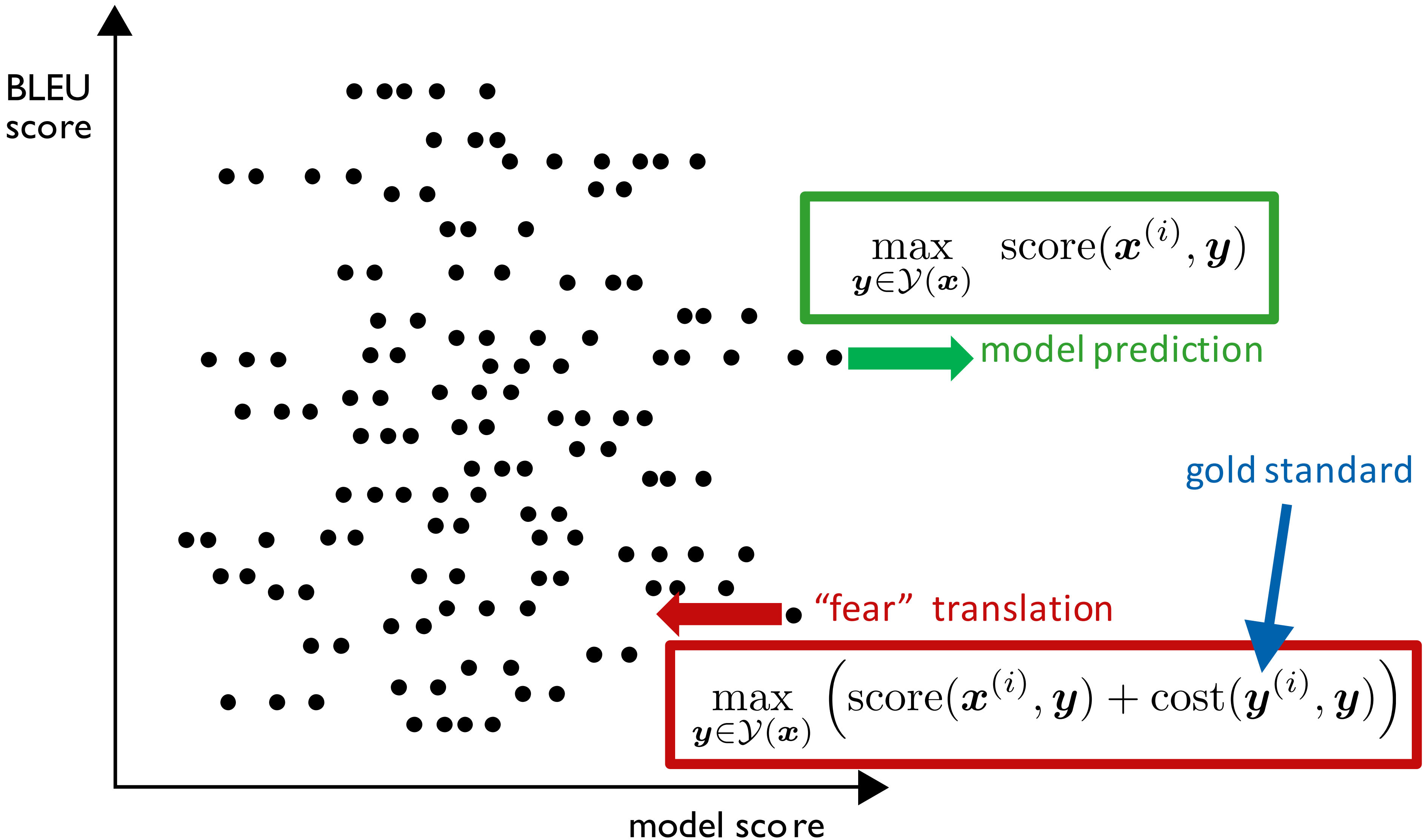
“Fear” Ramp Loss

(Do et al., 2008)



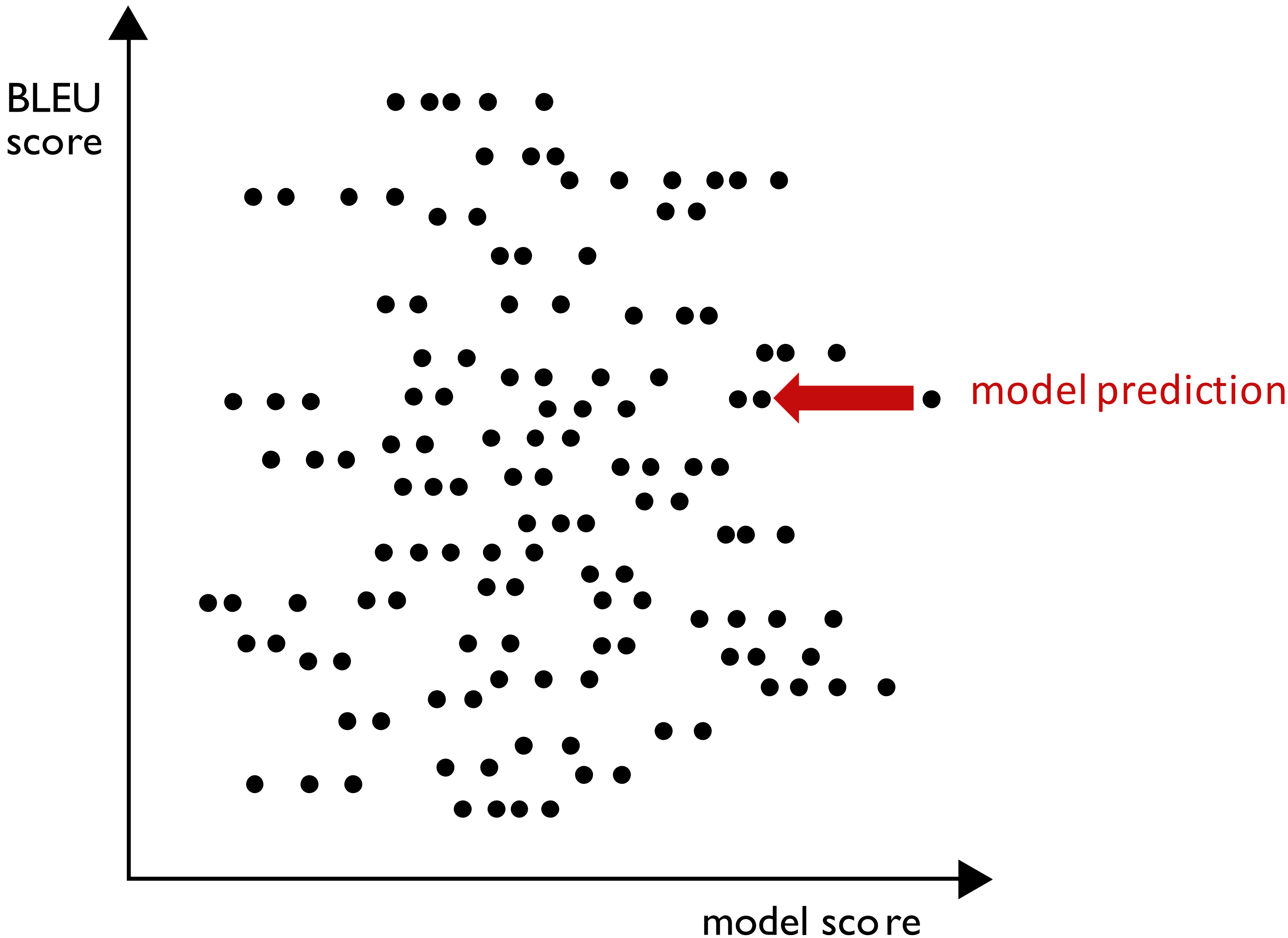
"Fear" Ramp Loss

(Do et al., 2008)



“Hope” Ramp Loss

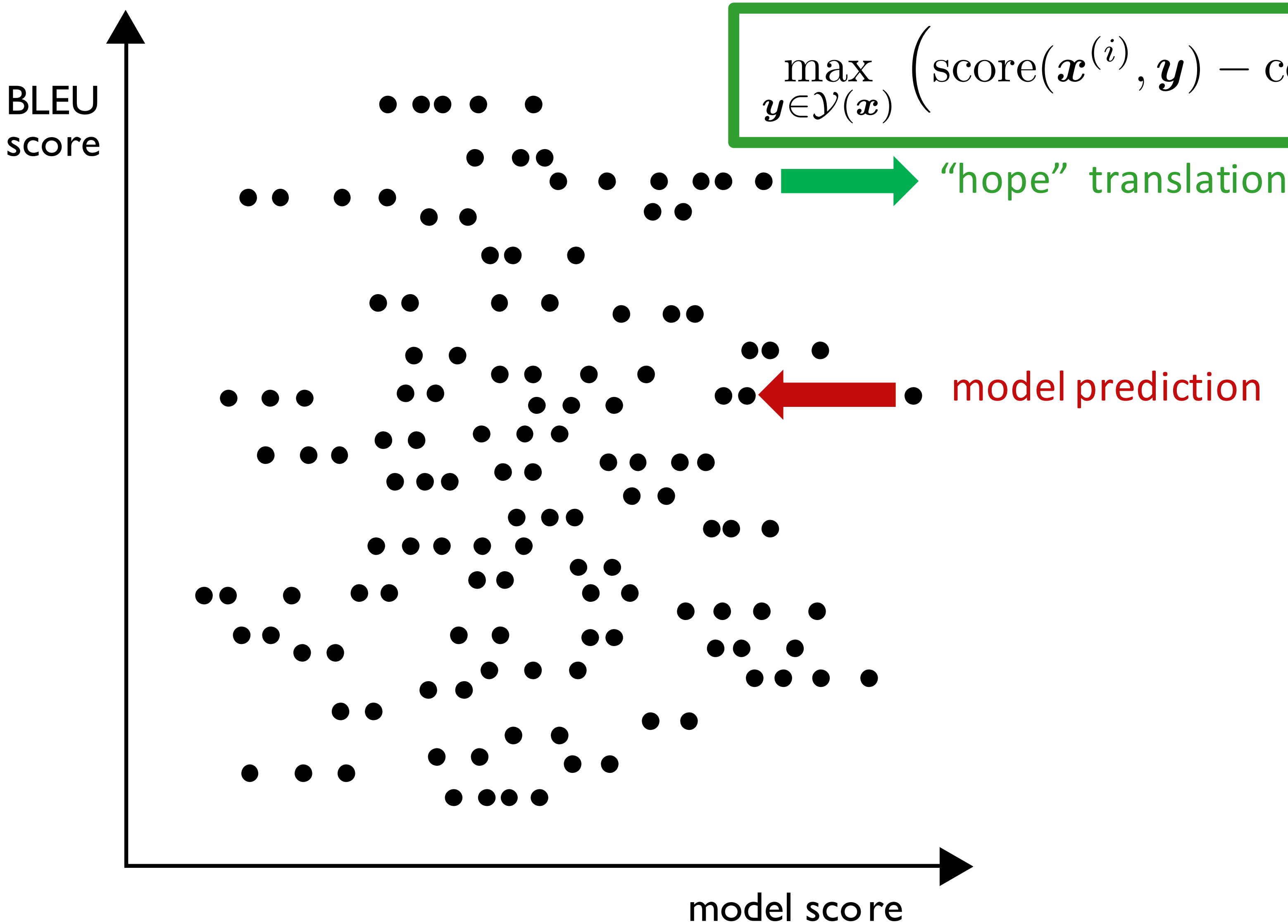
(McAllester & Keshet, 2011; Liang et al., 2006)



“Hope” Ramp Loss

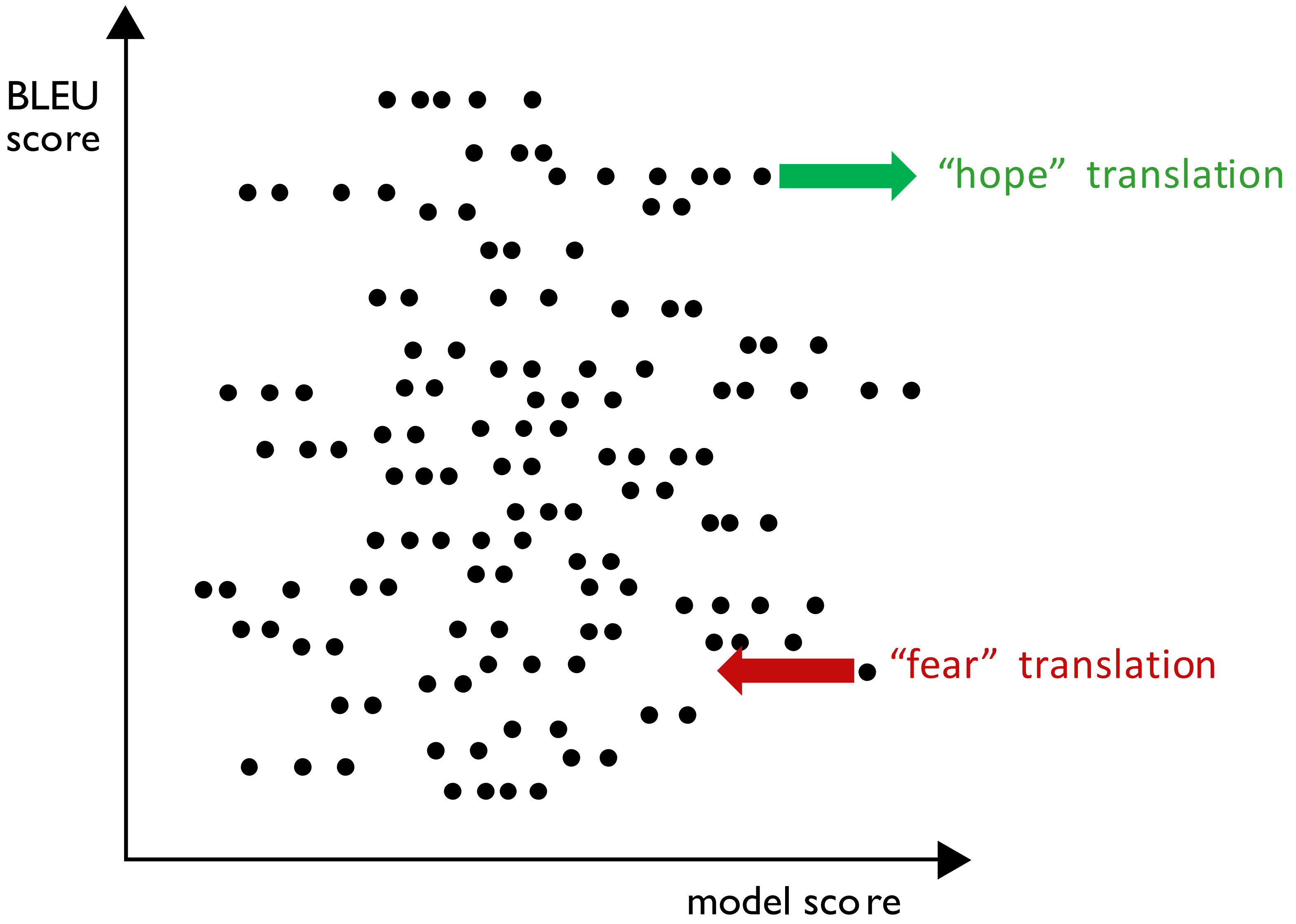
(McAllester & Keshet, 2011; Liang et al., 2006)

$$\max_{\boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x})} \Big(\mathrm{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}) - \mathrm{cost}(\boldsymbol{y}^{(i)}, \boldsymbol{y}) \Big)$$



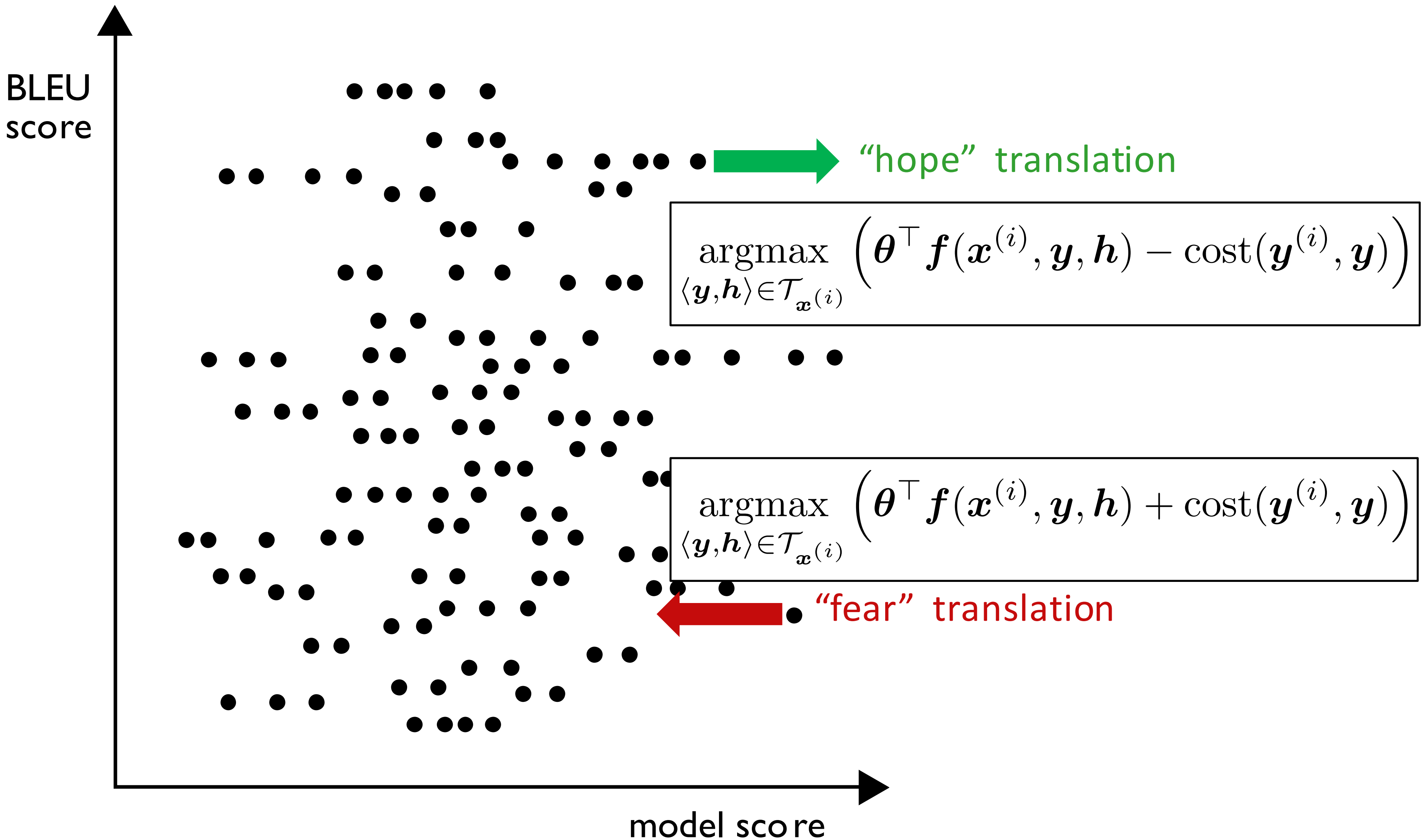
“Hope-Fear” Ramp Loss

(Chiang et al., 2008; 2009; Cherry & Foster, 2012; Chiang, 2012)



“Hope-Fear” Ramp Loss

(Chiang et al., 2008; 2009; Cherry & Foster, 2012; Chiang, 2012)



$$\underset{\langle \boldsymbol{y},\boldsymbol{h}\rangle \in \mathcal{T}_{\boldsymbol{x}^{(i)}}}{\operatorname{argmax}} \left(\boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}^{(i)}, \boldsymbol{y}, \boldsymbol{h}) - \operatorname{cost}(\boldsymbol{y}^{(i)}, \boldsymbol{y})\right)$$

$$\underset{\langle \boldsymbol{y},\boldsymbol{h}\rangle \in \mathcal{T}_{\boldsymbol{x}^{(i)}}}{\operatorname{argmax}} \left(\boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}^{(i)}, \boldsymbol{y}, \boldsymbol{h}) + \operatorname{cost}(\boldsymbol{y}^{(i)}, \boldsymbol{y})\right)$$

Experiments
(Gimpel, 2012)

averages over 8 test sets across 3 language pairs

	Moses	Hiero
	%BLEU	%BLEU
MERT	**35.9**	**37.0**
Fear Ramp (away from bad)	34.9	34.2
Hope Ramp (toward good)	35.2	36.0
Hope-Fear Ramp (toward good + away from bad)	35.7	**37.0**

Pairwise Ranking Optimization

(Hopkins & May, 2011)