On NMT Search Errors and Model Errors: Cat Got Your Tongue?

By Felix Stahlberg and Bill Byrne

Michitatsu Sato

What is this paper about

NMT fails to find the globally best model score in most of the cases with beam search.

- Even with the large beam search size of 100!

For more than 50%, the model assigns the global best score to Empty Translation.

- The main factor of causes is an inherent bias toward shorter translation.

What is NMT?

Machine translation approach usnig Neural Networks.

Before NMT, there are 2 major approach which are;

- RMT: Rule based machine translation
  - It works without large corpus.
  - generate translation from rule by analysing syntactic aspect of sentences.
- SMT: Statistical machine translation
  - generate trnaslation based on large parallel translated data.

Now, NMT become very popular approach due to its quality of translation.

NMT (Neural Machine Translation)

Task: Decoding or inference problem;

Find the most likely translation y;

$$\hat{\mathbf{y}} = \underset{\mathbf{y} \in \mathcal{T}^*}{\arg\max} P(\mathbf{y} \mid \mathbf{x})$$

where **y** is transrated sentence of target langugages, **x** is original sentence of source languages.

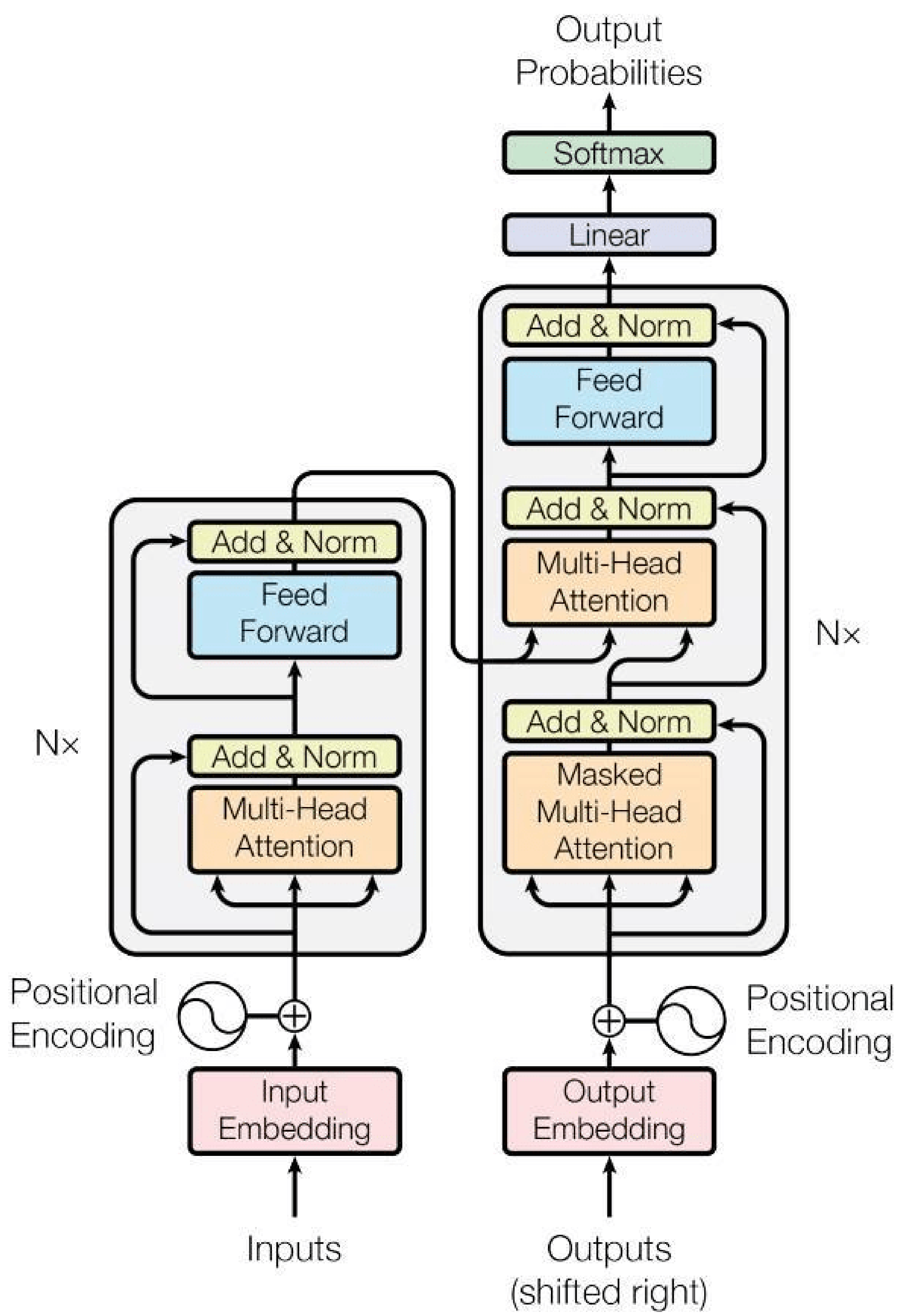
By left right factorization using chain rule;

$$\log P(\mathbf{y} \mid \mathbf{x}) = \sum_{j=1}^{J} \log P\left(y_j \mid y_1^{j-1}, \mathbf{x}\right)$$

where **J** = length of translated sentence.

Search Space: very large. If vocaburaly size is V and translation with 20 words, then there is V^20 possibility.

NMT architecture with Transformer



retireved from https://pytorch.org/tutorials/beginner/transformer_tutorial.html

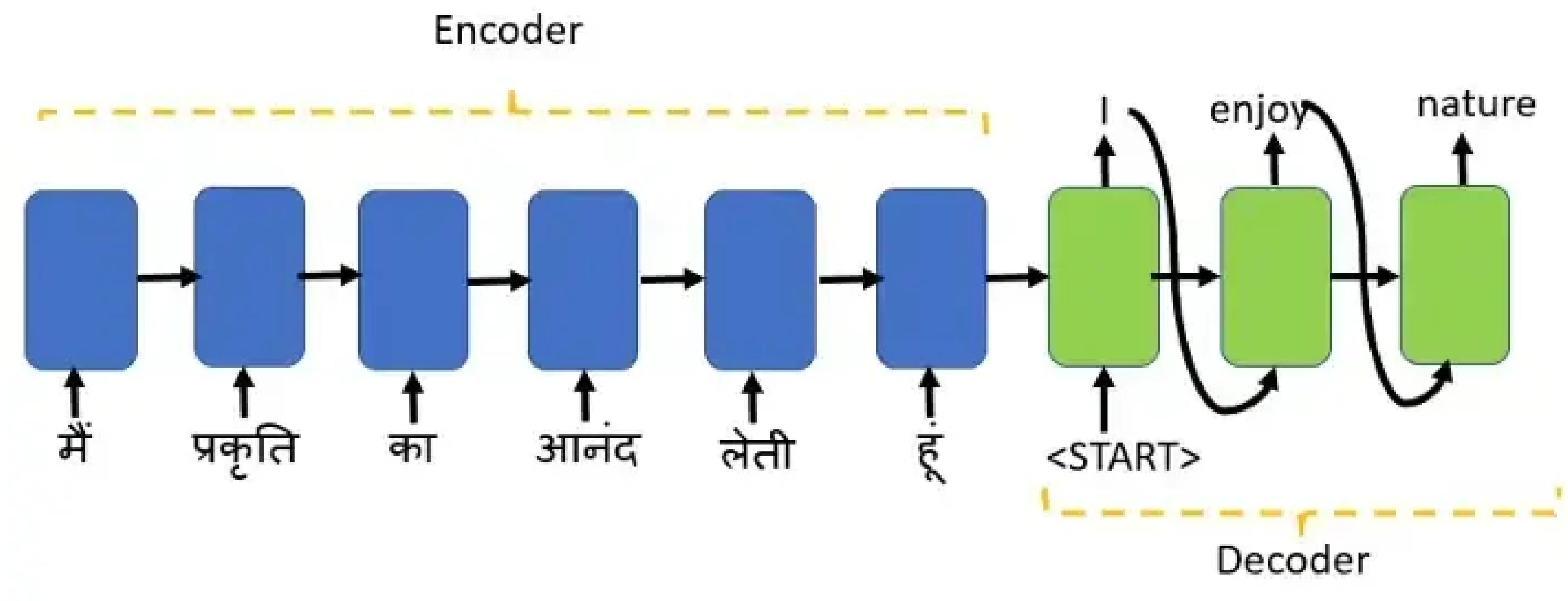
Beam search

Algorithm 1 BeamSearch$(\mathbf{x}, n \in \mathbb{N}_+)$

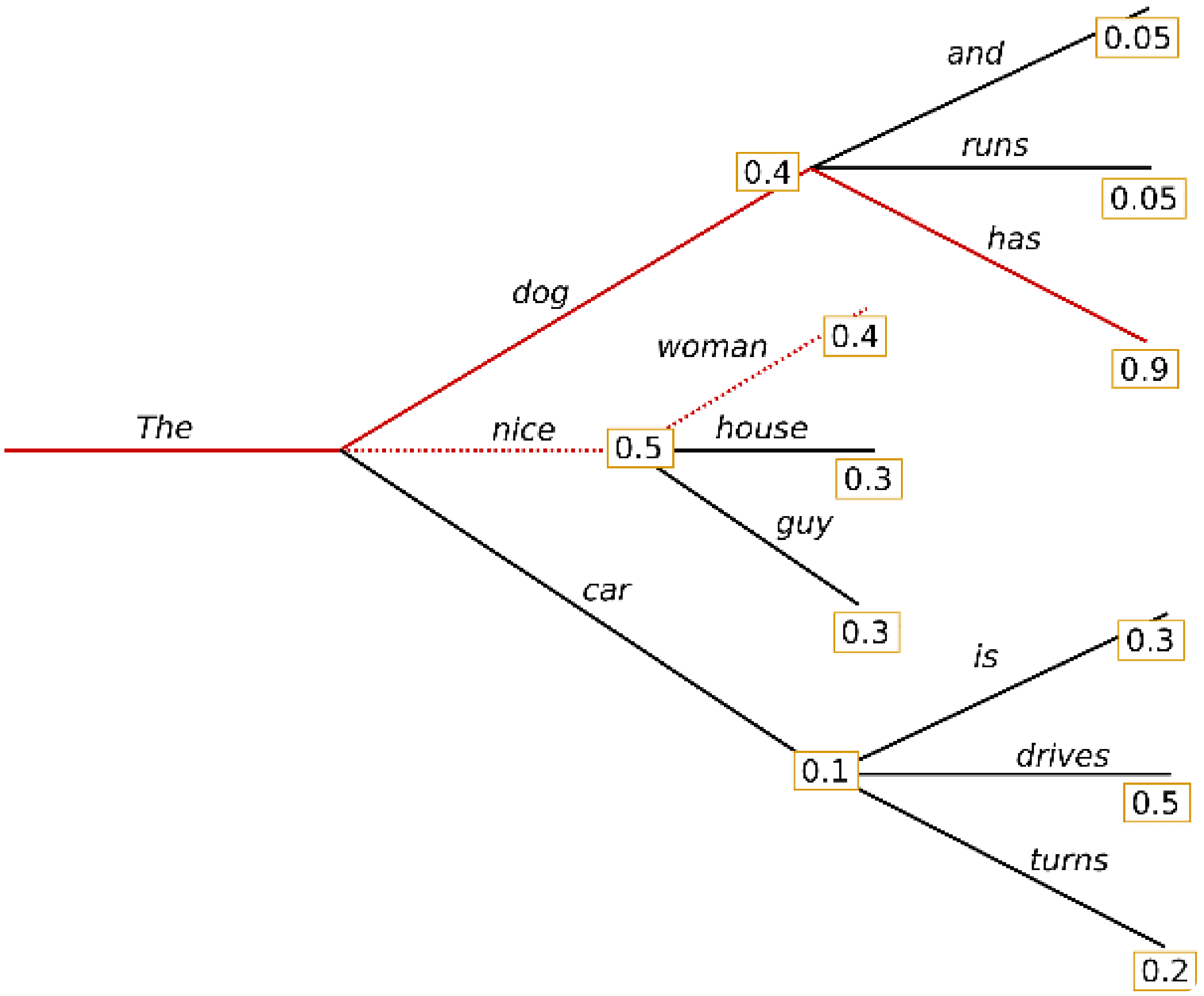
Input: $\mathbf{x}$: Source sentence, n: Beam size

$\mathcal{H}_{cur} \leftarrow \{(\epsilon, 0.0)\}$ {Initialize with empty translation prefix and zero score}
repeat
  $\mathcal{H}_{next} \leftarrow \emptyset$
  for all $(\mathbf{y}, p) \in \mathcal{H}_{cur}$ **do**
    if $y_{|\mathbf{y}|} = < /\mathbf{s} >$ **then**
      $\mathcal{H}_{next} \leftarrow \mathcal{H}_{next} \cup \{(\mathbf{y}, p)\}$ {Hypotheses ending with $< /\mathbf{s} >$ are not expanded}
    else
      $\mathcal{H}_{next} \leftarrow \mathcal{H}_{next} \cup \bigcup_{w \in \mathcal{T}} (\mathbf{y} \cdot w, p + \log P(w|\mathbf{x}, \mathbf{y}))$ {Add all possible continuations}
    end if
  end for
  $\mathcal{H}_{cur} \leftarrow \{(\mathbf{y}, p) \in \mathcal{H}_{next} : |\{(\mathbf{y}', p') \in \mathcal{H}_{next} : p' > p\}| < n\}$ {Select n-best}
  $(\tilde{\mathbf{y}}, \tilde{p}) \leftarrow \arg\max_{(\mathbf{y}, p) \in \mathcal{H}_{cur}} p$
until $\tilde{y}_{|\tilde{\mathbf{y}}|} = < /\mathbf{s} >$
return $\tilde{\mathbf{y}}$

Beam Search in Machine Translation



Retrieved from https://towardsdatascience.com/an-intuitive-explanation-of-beam-search-9b1d744e7a0f



Retrieved from https://huggingface.co/blog/constrained-beam-search

Problem of Beam Search

- prone to search errors as the number of active hypotheses is limited by n.
- never compares partial hypotheses of different lengths with each other.
  - partial hypotheses: possible sentences not ending with symbol </s>, which means the sentence is not ending.

Exact Decoding Scheme

In the paper, for the evaluation of NMT's search and model errors, the new method, **Exact Decoding** is introduced.

- Travel search space of the translated sentence in DFS (Depth First Search) order, but did not conduct exhausitive search.
- Cut off the branch which has lower model score than *threashold value*.
- The *threshold value* will be updated when it find better **complete hypothesis**.
    - complete hypotheses: possible sentences ending with symbol </s>

Basically, for avoiding searching error by normal NMT model.

Why it works?

From

$$\log P(\mathbf{y} \mid \mathbf{x}) = \sum_{j=1}^{J} \log P\left(y_j \mid y_1^{j-1}, \mathbf{x}\right)$$

follwing relationship is true;

$$\forall j \in [2, J] : \log P\left(y_1^{j-1} \mid \mathbf{x}\right) > \log P\left(y_1^{j} \mid \mathbf{x}\right)$$

- Expanding a partial hypothesis is guaranteed to result in a lower model score.
- Setting *threshaold value* as lower bound of global best score.
- Then the algorithm only need to consider partial hypothesis with score grater than *threashold value* for finding better translation.

Experiment setting

- Evaluation Data: English-German WMT news-test2015 test set (2,169 sentences)
- Model: Transformer base (Vaswani et al., 2017) model trained with Tensor2Tensor (Vaswani et al., 2018) on parallel WMT18 data excluding ParaCrawl.

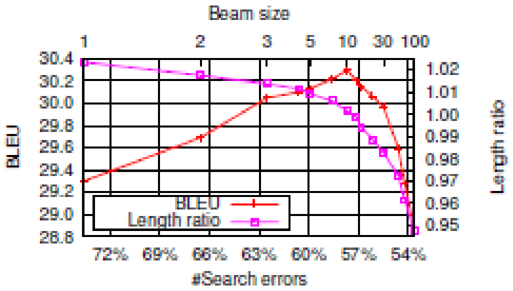
What is found out?

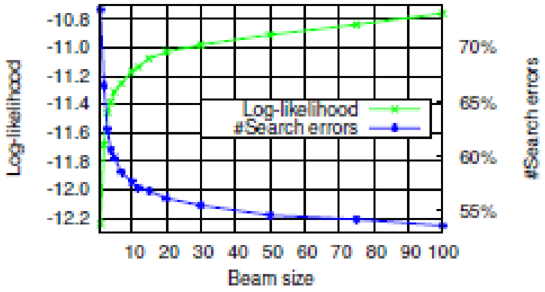
Search errors

Search	BLEU	Ratio	#Search errors	#Empty
Greedy	29.3	1.02	73.6%	0.0%
Beam-10	30.3	1.00	57.7%	0.0%
Exact	2.1	0.06	0.0%	51.8%

- Greedy and beam search both achieve reasonable BLEU scores but rely on a high number of search errors

Beam serch size

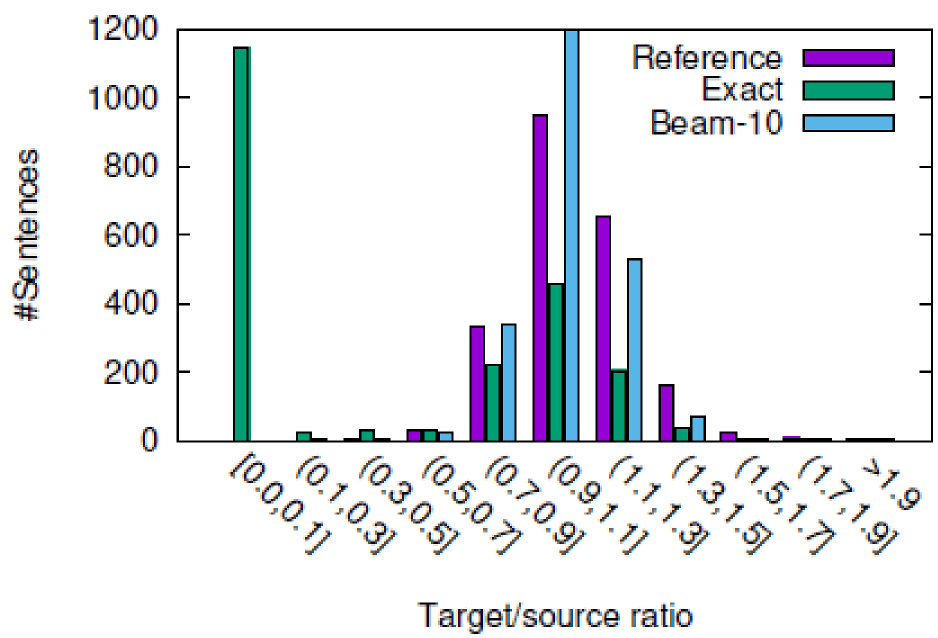




- Large beam sizes reduce the number of search errors, but the BLEU score drops because translations are too short.
- Even a large beam size of 100 produces 53.62% search errors.
- Beam search effectively reduces search errors with respect to greedy decoding to some degree, but is ineffective in reducing search errors even further.

Empty translation

- From previous results, for **51.8% of the sentences, NMT assigns the global best model score to the empty translation**, i.e. a single </s> token.

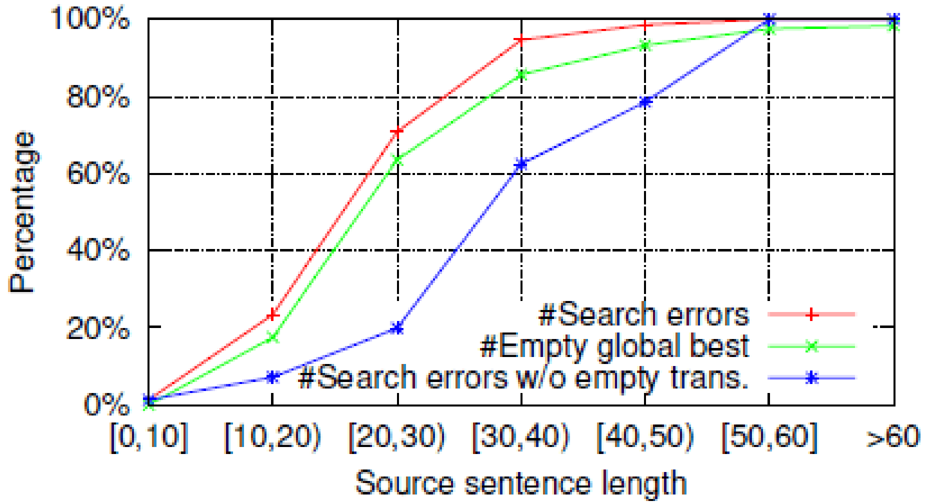


- Exact search has an isolated peak in [0.0, 0.1] from the empty translations.

Model	Beam-10		Exact
	BLEU	#Search err.	#Empty
LSTM*	28.6	58.4%	47.7%
SliceNet*	28.8	46.0%	41.2%
Transformer-Base	30.3	57.7%	51.8%
Transformer-Big*	31.7	32.1%	25.8%

- **Not specific problem for Transformer Based model, also observed in other NMT architecture**.

Long source sentence

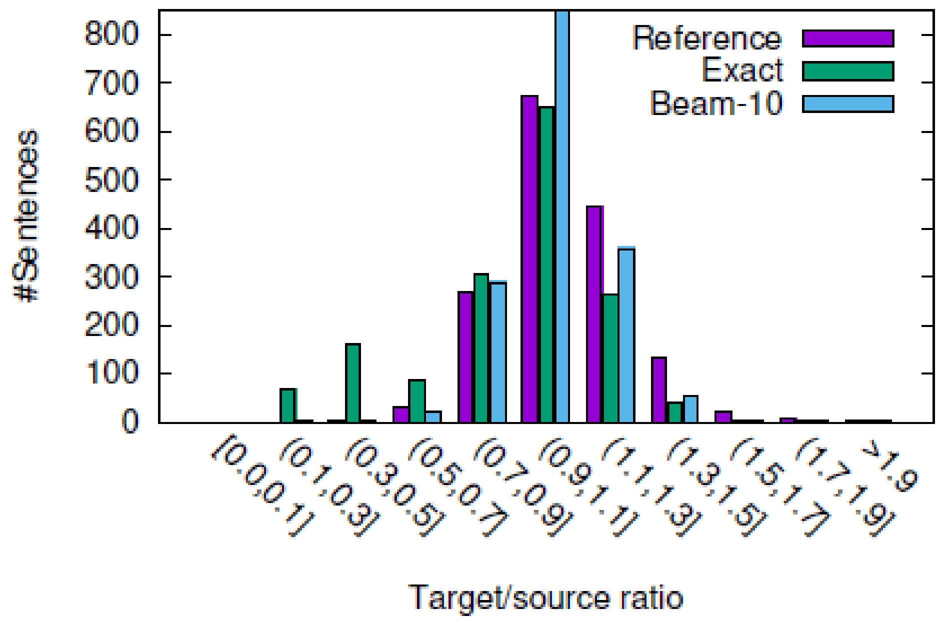


- Long source sentences are more affected by both beam search errors and the problem of empty translations.
- The global best translation is empty for almost all sentences longer than 40 tokens.
- Even without sentences where the model prefers the empty translation, a large amount of search errors remain

Results with Length Constraints

Constrained search to **translations longer than 0.25 times the source sentence** length.

- excluded the empty translation from the search space.



This solved the problem slightly.

- But, still results in a peak in the (0:3; 0:5] cluster.

This suggests that **the problem of empty translations is the consequence of an inherent model bias towards shorter hypotheses** and cannot be fixed with a length constraint.

Constrained exact search to **either the length of the best Beam-10 hypothesis or the reference length**.

Search	BLEU	Ratio
Beam-10	37.0	1.00
Exact for Beam-10 length	37.0	1.00
Exact for reference length	37.9	1.01

- Exact search constrained to the Beam-10 hypothesis length does not improve over beam search.
  - suggesting that any search errors between beam search score and global best score for that length are insignificant enough so as not to affect the BLEU score.
- Constrained exact search to the correct reference length improved the BLEU score by 0.9 points.

Possible solution

A popular method to counter the length bias in NMT is **length normalization**.

Length Normalization:

- divides the sentence score by the sentence length.
- since score is non-positive, deviding by length which is shorter will result in lower score.

Search	W/o length norm.		With length norm.	
	BLEU	Ratio	BLEU	Ratio
Beam-10	37.0	1.00	36.3	1.03
Beam-30	36.7	0.98	36.3	1.04
Exact	27.2	0.74	36.4	1.03

- **Exact search under length normalization does not suffer from the length deficiency anymore**.
  - But it is not able to match our best BLEU score under Beam-10 search.
- This suggests that **while length normalization biases search towards translations of roughly the correct length, it does not fix the fundamental modelling problem**.

Summary

- The paper shows NMT actually **assigns more than half of the global best score to Empty translation**.
- Even with exclusion of Empty Translation (by constrain), the model showed **inherent model bias towards shorter hypotheses.**
- This bias could be solved by Length Normalization, but not perfect.

Why model have such a bias?

- Training object?
- Encoder-Decoder architecture?
- Neural networks itself?
- Training data bias?

Thank you