



집현전 중급반

<https://github.com/jiphyeonjeon>

#1 beam search, uniform information density

If Beam Search is the Answer, What was the Question?

Table of Contents

❖ Cat Got Your Tongue?

- Exact Inference
- Model Errors & Search Errors
- Length Constraint
- Is Beam Search the Answer?

❖ What was the Question?

- Neural Probabilistic Text Generation
- Deriving Beam Search
- From Beam Search to UID

❖ Experiments

- Generalized UID Decoding
- Settings & Results

❖ Conclusions & Reference

SGNMT – A Flexible NMT Decoding Platform for Quick Prototyping of New Models and Search Strategies

Felix Stahlberg¹ and Eva Hasler² and Danielle Saunders¹ and Bill Byrne^{2†}

¹Department of Engineering, University of Cambridge, UK

²SDL Research, Cambridge, UK

EMNLP 2017

Abstract

This paper introduces SGNMT, our experimental platform for machine translation research. SGNMT provides a generic interface to neural and symbolic scoring modules (*predictors*) with left-to-right semantic such as translation models like NMT, language models, translation lattices, *n*-best lists or other kinds of scores and constraints. Predictors can be combined with other predictors to form complex decoding tasks. SGNMT implements a number of search strategies for traversing the space spanned by the predictors which are appropriate for different predictor constellations. Adding new predictors is easy.

SGNMT tool are *predictors* and *decoders*. Predictors are scoring modules which define scores over the target language vocabulary given the current internal predictor state, the history, the source sentence, and external side information. Scores from multiple, diverse predictors can be combined for use in decoding.

Decoders are search strategies which traverse the space spanned by the predictors. SGNMT provides implementations of common search tree traversal algorithms like beam search. Since decoders differ in runtime complexity and the kind of search errors they make, different decoders are appropriate for different predictor constellations.

The strict separation of scoring module and search strategy and the decoupling of scoring modules from each other makes SGNMT a very flexible decoding tool for neural and symbolic models which is applicable not only to machine translation. SGNMT is based on the OpenFST-based Cambridge SMT system (Allauzen et al., 2014). Although the system is less than a year old, we have found it to be very flexible and easy for new researchers to adopt. Our group has already integrated SGNMT into most of its research work.

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

Felix Stahlberg* and Bill Byrne

University of Cambridge
Department of Engineering
Trumpington St, Cambridge CB2 1PZ, UK
{fs439,wjb31}@cam.ac.uk

EMNLP 2019

Abstract

We report on search errors and model errors in neural machine translation (NMT). We present an exact inference procedure for neural sequence models based on a combination of beam search and depth-first search. We use our exact search to find the global best model scores under a Transformer base model for the entire WMT15 English-German test set. Surprisingly, beam search fails to find these global best model scores in most cases, even with a very large beam size of 100. For more than 50% of the sentences, the model in fact assigns its global best score to the empty translation, revealing a massive failure of neural models in properly accounting for adequacy. We show by constraining search with a minimum translation length that at the root of the problem of empty translations lies an inherent bias towards shorter translations. We conclude that vanilla NMT in its current form requires just the right amount of beam search errors, which, from a modelling perspective, is a highly unsatisfactory conclusion indeed, as the model often prefers an empty translation.

decoding or inference problem:

$$\hat{y} = \arg \max_{y \in T^*} P(y|x). \quad (2)$$

The NMT search space is vast as it grows exponentially with the sequence length. For example, for a common vocabulary size of $|T| = 32,000$, there are already more possible translations with 20 words or less than atoms in the observable universe ($32,000^{20} \gg 10^{52}$). Thus, complete enumeration of the search space is impossible. The size of the NMT search space is perhaps the main reason why – besides some preliminary studies (Niehues et al., 2017; Stahlberg et al., 2018b; Ott et al., 2018) – analyzing search errors in NMT has received only limited attention. To the best of our knowledge, none of the previous studies were able to quantify the number of search errors in unconstrained NMT due to the lack of an exact inference scheme that – although too slow for practical MT – guarantees to find the global best model score for analysis purposes.

In this work we propose such an exact decod-

If Beam Search is the Answer, What was the Question?

Clara Meister

Ryan Cotterell

Tim Vieira

ETH Zürich

University of Cambridge

Johns Hopkins University

clara.meister@inf.ethz.ch tim.vieira@gmail.com

ryan.cotterell@inf.ethz.ch

Abstract

EMNLP 2020

IWSLT'14 (De-En)

Quite surprisingly, exact maximum a posteriori (MAP) decoding of neural language generators frequently leads to low-quality results (Stahlberg and Byrne, 2019). Rather, most state-of-the-art results on language generation tasks are attained using beam search despite its overwhelmingly high search error rate. This implies that the MAP objective alone does not express the properties we desire in text, which merits the question: if beam search is the answer, what was the question? We frame beam search as the exact solution to a different decoding objective in order to gain insights into why high probability under a model alone may not indicate adequacy. We find that beam search enforces *uniform information density* in text, a property motivated by cognitive science. We suggest a set of decoding objectives that explicitly enforce this property and find that exact decoding with these objectives alleviates the problems encountered when decoding poorly calibrated language generation models. Additionally, we analyze the text produced using various decoding strategies and see that, in our neural machine translation experiments, the extent to which this property is adhered to strongly correlates with BLEU. Our code is publicly available at <https://github.com/rycolab/uid-decoding>.

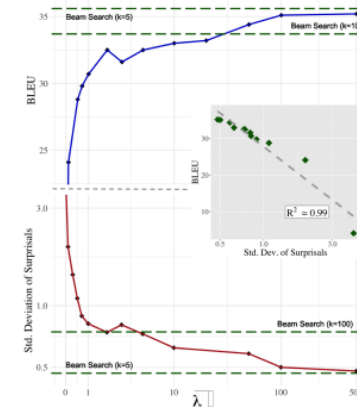


Figure 1: Average std. deviation σ of surprisals (per sentence) and corpus BLEU for translations generated using exact search over the MAP objective with a greedy regularizer (Eq. (11)) with varying degrees of λ . References for beam search ($k = 5$ and $k = 100$) are included. Sub-graph shows the explicit relationship between BLEU and σ . λ and σ axes are log-scaled.



Cat Got Your Tongue?



EMNLP 2017

SGNMT – A Flexible NMT Decoding Platform for Quick Prototyping of New Models and Search Strategies

Felix Stahlberg[†] and Eva Hasler[‡] and Danielle Saunders[†] and Bill Byrne^{††}

[†]Department of Engineering, University of Cambridge, UK

[‡]SDL Research, Cambridge, UK

Abstract

This paper introduces SGNMT, our experimental platform for machine translation research. SGNMT provides a generic interface to neural and symbolic scoring modules (*predictors*) with left-to-right semantic such as translation models like NMT, language models, translation lattices, *n*-best lists or other kinds of scores and constraints. Predictors can be combined with other predictors to form complex decoding tasks. SGNMT implements a number of search strategies for traversing the space spanned by the predictors which are appropriate for different predictor constellations. Adding new predictors or decoding strategies is particularly easy, making it a very efficient tool for prototyping new research ideas. SGNMT is actively being used by students in the MPhil program in Machine Learning, Speech and Language Technology at the University of Cambridge for course work and theses, as well as for most of the research work in our group.

SGNMT tool are *predictors* and *decoders*. Predictors are scoring modules which define scores over the target language vocabulary given the current internal predictor state, the history, the source sentence, and external side information. Scores from multiple, diverse predictors can be combined for use in decoding.

Decoders are search strategies which traverse the space spanned by the predictors. SGNMT provides implementations of common search tree traversal algorithms like beam search. Since decoders differ in runtime complexity and the kind of search errors they make, different decoders are appropriate for different predictor constellations.

The strict separation of scoring module and search strategy and the decoupling of scoring modules from each other makes SGNMT a very flexible decoding tool for neural and symbolic models which is applicable not only to machine translation. SGNMT is based on the OpenFST-based Cambridge SMT system (Allauzen et al., 2014). Although the system is less than a year old, we have found it to be very flexible and easy for new researchers to adopt. Our group has already integrated SGNMT into most of its research work.

<https://github.com/ucam-smt/sgnmt>

SGNMT

SGNMT is an open-source framework for neural machine translation (NMT) and other sequence prediction tasks. The tool provides a flexible platform which allows pairing NMT with various other models such as language models, length models, or bag2seq models. It supports rescoring both *n*-best lists and lattices. A wide variety of search strategies is available for complex decoding problems.

SGNMT is compatible with the following NMT toolkits:

- [Tensor2Tensor \(TensorFlow\)](#)
- [fairseq \(PyTorch\)](#)

Old SGNMT versions (0.x) are compatible with:

- [\(extended\) TF seq2seq tutorial \(TensorFlow\)](#)
- [Blocks \(Theano\)](#)




Features:

- Syntactically guided neural machine translation (NMT lattice rescoring)
- *n*-best list rescoring with NMT
- Integrating external *n*-gram posterior probabilities used in MBR
- Ensemble NMT decoding
- Forced NMT decoding
- Integrating language models
- Different search algorithms (beam, A*, depth first search, greedy...)
- Target sentence length modelling
- Bag2Sequence models and decoding algorithms
- Joint decoding with word- and subword/character-level models
- Hypothesis recombination
- Heuristic search
- ...

Packages

No packages published


Contributors 3

-  [fstahlberg](#)
-  [DCSaunders](#) D C Saunders
-  [ehasler](#) Eva Hasler

Languages

Python 99.5% Shell 0.5%

Releases 13

 **v1.1** Latest
on 24 Aug 2019

[+ 12 releases](#)



EMNLP 2019

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

Felix Stahlberg* and Bill Byrne

University of Cambridge
Department of Engineering
Trumpington St, Cambridge CB2 1PZ, UK
{fs439, wjb31}@cam.ac.uk

Abstract

We report on search errors and model errors in neural machine translation (NMT). We present an exact inference procedure for neural sequence models based on a combination of beam search and depth-first search. We use our exact search to find the global best model scores under a Transformer base model for the entire WMT15 English-German test set. Surprisingly, beam search fails to find these global best model scores in most cases, even with a very large beam size of 100. For more than 50% of the sentences, the model in fact assigns its global best score to the empty translation, revealing a massive failure of neural models in properly accounting for adequacy. We show by constraining search with a minimum translation length that at the root of the problem of empty translations lies an inherent bias towards shorter translations. We conclude that vanilla NMT in its current form requires just the right amount of beam search errors, which, from a modelling perspective, is a highly unsatisfactory conclusion indeed, as the model often prefers an empty translation.

decoding or inference problem:

$$\hat{y} = \arg \max_{y \in \mathcal{T}^*} P(y|x). \quad (2)$$

The NMT search space is vast as it grows exponentially with the sequence length. For example, for a common vocabulary size of $|\mathcal{T}| = 32,000$, there are already more possible translations with 20 words or less than atoms in the observable universe ($32,000^{20} \gg 10^{82}$). Thus, complete enumeration of the search space is impossible. The size of the NMT search space is perhaps the main reason why – besides some preliminary studies (Niehues et al., 2017; Stahlberg et al., 2018b; Ott et al., 2018) – analyzing search errors in NMT has received only limited attention. To the best of our knowledge, none of the previous studies were able to quantify the number of search errors in unconstrained NMT due to the lack of an exact inference scheme that – although too slow for practical MT – guarantees to find the global best model score for analysis purposes.

In this work we propose such an exact decod-





Exact Inference Procedure

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

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y} \in \mathcal{T}^*} P(\mathbf{y}|\mathbf{x})$$

$$|\mathcal{T}| = 32,000 \quad 32,000^{20} \gg 10^{82}$$

- Complete enumeration of the search space is impossible
- NMT의 search space가 굉장히 크기 때문에 Search Error에 대한 연구는 제한적이었음.
Why not be Versatile? Applications of the SGNMT Decoder for Machine Translation (<https://www.aclweb.org/anthology/W18-1821/>)
Analyzing Neural MT Search and Model Performance (<https://www.aclweb.org/anthology/W17-3202/>)
Analyzing Uncertainty in Neural Machine Translation (<https://arxiv.org/abs/1803.00047>)
- 본 논문에서 Exact Decoding을 추론하는 알고리즘을 제안하여 Search Error를 정량적으로 분석할 수 있는 scheme를 제안 !! (Contribution)



Exact Inference Procedure

Time synchronous approximate search algorithm! (가설을 left-to-right로 build)

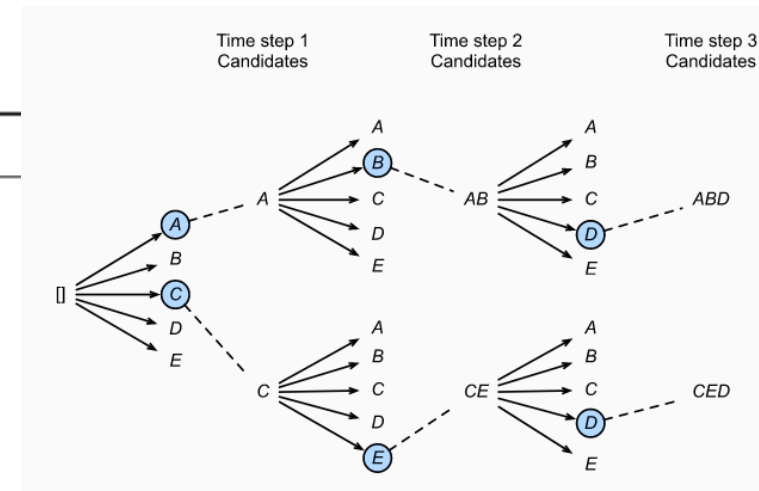
Algorithm 1 BeamSearch($\mathbf{x}, n \in \mathbb{N}_+$)

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

```

1:  $\mathcal{H}_{cur} \leftarrow \{(\epsilon, 0.0)\}$  {Initialize with empty translation prefix and zero score}
2: repeat
3:    $\mathcal{H}_{next} \leftarrow \emptyset$ 
4:   for all  $(\mathbf{y}, p) \in \mathcal{H}_{cur}$  do
5:     if  $y_{|\mathbf{y}|} = </s>$  then EOS 토큰 등장 시 다른 가설을 탐색하지 않음
6:        $\mathcal{H}_{next} \leftarrow \mathcal{H}_{next} \cup \{(\mathbf{y}, p)\}$  {Hypotheses ending with  $</s>$  are not expanded}
7:     else
8:        $\mathcal{H}_{next} \leftarrow \mathcal{H}_{next} \cup \bigcup_{w \in \mathcal{T}} (\mathbf{y} \cdot w, p + \log P(w|\mathbf{x}, \mathbf{y}))$  {Add all possible continuations}
9:     end if 모든 target word에 대해 가능한 continuation들을 탐색
10:  end for
11:   $\mathcal{H}_{cur} \leftarrow \{(\mathbf{y}, p) \in \mathcal{H}_{next} : |\{(\mathbf{y}', p') \in \mathcal{H}_{next} : p' > p\}| < n\}$  {Select  $n$ -best}
12:   $(\tilde{\mathbf{y}}, \tilde{p}) \leftarrow \arg \max_{(\mathbf{y}, p) \in \mathcal{H}_{cur}} p$  Beam Size
13: until  $\tilde{y}_{|\tilde{\mathbf{y}}|} = </s>$ 
14: return  $\tilde{\mathbf{y}}$ 

```



$$\gamma \leq \log P(\hat{\mathbf{y}}|\mathbf{x})$$

$$\tilde{p}_{beam\ search} := \gamma$$

Note That> Beam Search Score는 Global Score의 Lower Bound!



Exact Inference Procedure

Eq (3) Partial hypothesis is guaranteed to result in a lower model score

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

proof) $\log p(y_j | y_1^{j-1}, x) < 0$

$$\log \left(\frac{p(y_j, y_1^{j-1} | x)}{p(y_1^{j-1} | x)} \right) < 0$$

$$\frac{p(y_j, y_1^{j-1} | x)}{p(y_1^{j-1} | x)} < 1$$

양변에 로그를 취해 증명 완료 \square

- ✓ 왜 j는 2보다 큰가? $\rightarrow j=1$ 이면 length 0 sentence
- ✓ 왜 Equality를 고려하지 않는가? \rightarrow 확률이 NN에 의해 모델링 되기 때문, Softmax로 인해 확률이 정확히 1이 나오지 않음

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

We only need to consider partial hypotheses with scores greater than γ



Exact Inference Procedure

Algorithm 2 DFS($\mathbf{x}, \mathbf{y}, p \in \mathbb{R}, \gamma \in \mathbb{R}$)

Input: \mathbf{x} : Source sentence

\mathbf{y} : Translation prefix (default: ϵ)

p : $\log P(\mathbf{y}|\mathbf{x})$ (default: 0.0)

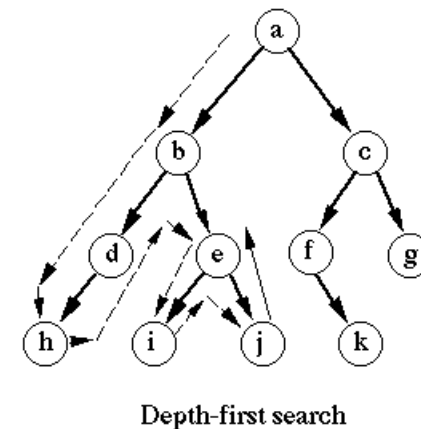
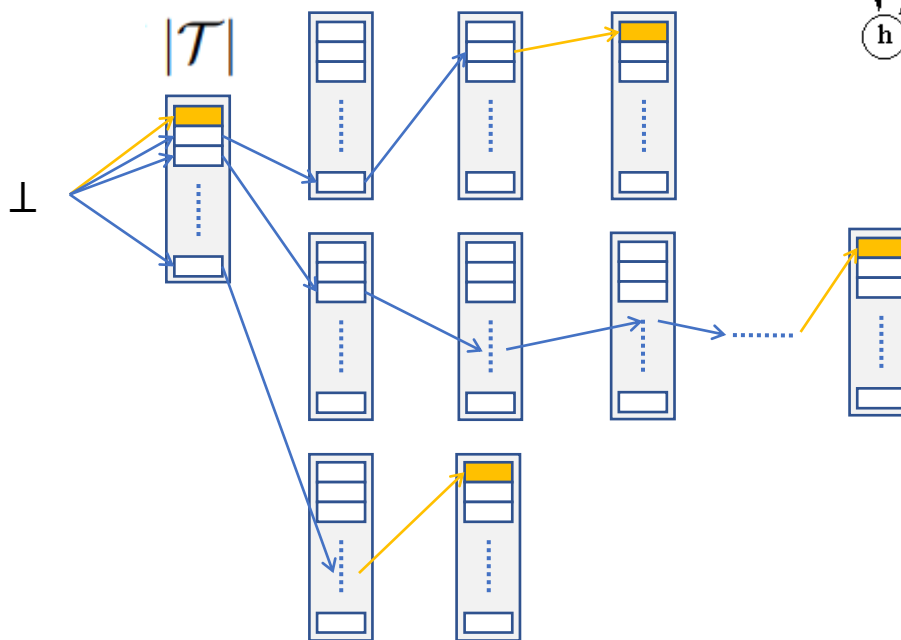
γ : Lower bound $\tilde{p}_{beam\ search} := \gamma$

```

1: if  $y_{|\mathbf{y}|} = </s>$  then
2:   return  $(\mathbf{y}, p)$  {Trigger  $\gamma$  update}
3: end if
4:  $\tilde{\mathbf{y}} \leftarrow \perp$  {Initialize  $\tilde{\mathbf{y}}$  with dummy value}
5: for all  $w \in \mathcal{T}$  do EOS부터 고려!
6:    $p' \leftarrow p + \log P(w|\mathbf{x}, \mathbf{y})$ 
7:   if  $p' \geq \gamma$  then
8:      $(\mathbf{y}', \gamma') \leftarrow \text{DFS}(\mathbf{x}, \mathbf{y} \cdot w, p', \gamma)$ 
9:     if  $\gamma' > \gamma$  then
10:       $(\tilde{\mathbf{y}}, \gamma) \leftarrow (\mathbf{y}', \gamma')$ 
11:    end if
12:   end if
13: end for
14: return  $(\tilde{\mathbf{y}}, \gamma)$  Global Best Score

```

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





Model Errors & Search Errors

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

Felix Stahlberg* and Bill Byrne
University of Cambridge
Department of Engineering
Trumpington St, Cambridge CB2 1PZ, UK
{fs439, wjb31}@cam.ac.uk

- Model Errors:
 $Global\ Best\ Model\ Score\ \log p(\hat{y}|x)$ 를 Empty Translation에 할당하는 경우
- Search Errors:
Decoder가 $Global\ Best\ Model\ Score\ \log p(\hat{y}|x)$ 를 찾지 못한 경우

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%

Table 1: NMT with exact inference. In the absence of search errors, NMT often prefers the empty translation, causing a dramatic drop in length ratio and BLEU.

- Exact Search에서 Search Errors는 0 (by def)
- Beam Search에선 큰 Search Errors!
- 그러나, Exact Search에선 Empty String 할당 문제가 있음
- Exact의 BLEU score도 너무 낮음
- Beam Search의 BLEU가 reasonable score!!
 - 투자 포트폴리오 제안 수익률을 생각해보자!



Large beam size

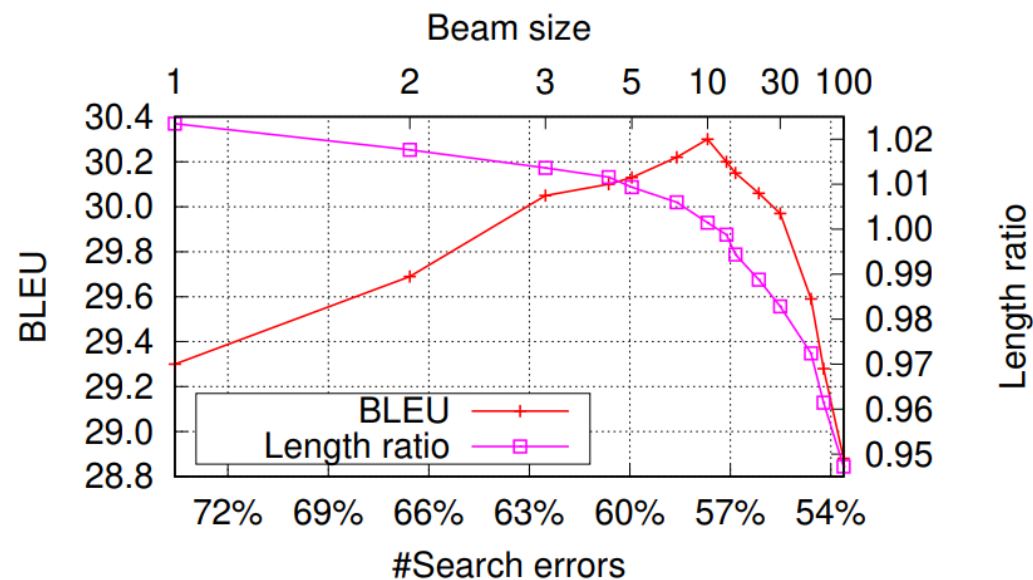


Figure 1: BLEU over the percentage of search errors. Large beam sizes yield fewer search errors but the BLEU score suffers from a length ratio below 1.

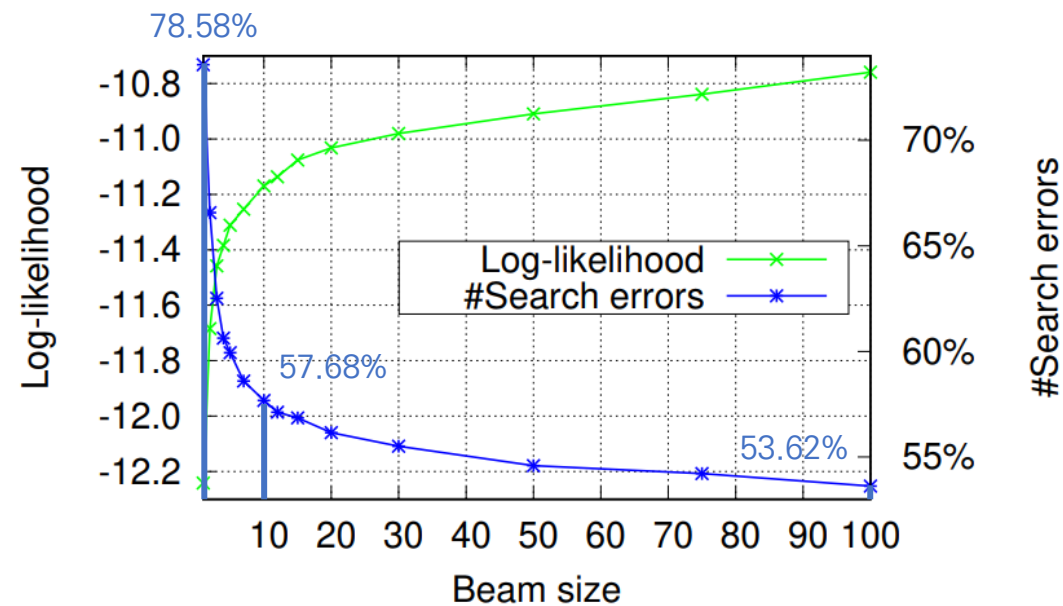


Figure 2: Even large beam sizes produce a large number of search errors.

- Large K는 Search Error를 줄임
- 그러나 BLEU 다운 (왜냐? 문장 길이가 점점 짧아지거든)

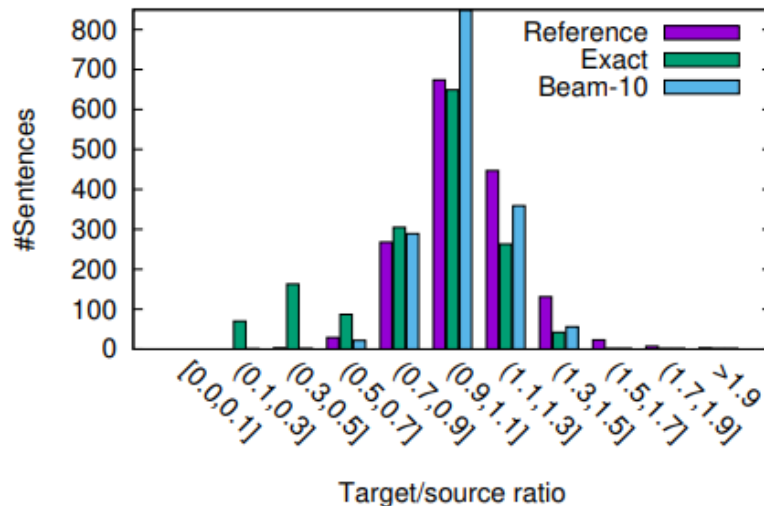
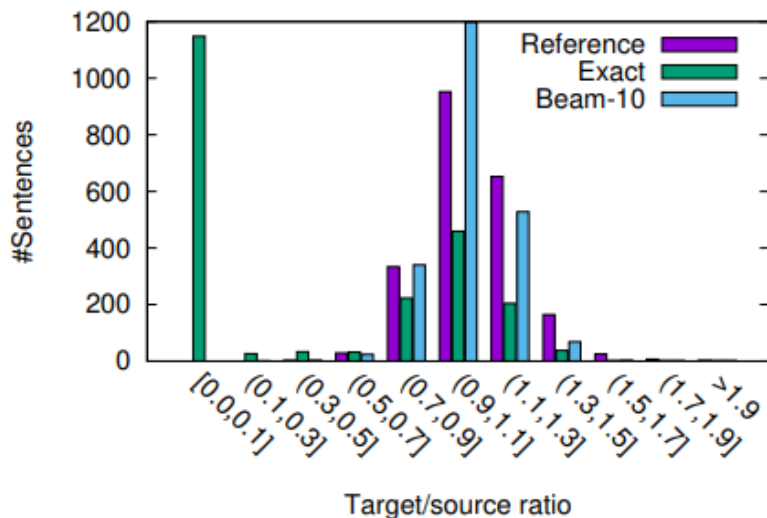


Model(Transformer)의 문제인가? Length를 조절해주면 해결될 문제인가?

Model	Beam-10		Exact #Empty
	BLEU	#Search err.	
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%

Table 2: *: The recurrent LSTM, the convolutional SliceNet (Kaiser et al., 2017), and the Transformer-Big systems are strong baselines from a WMT'18 shared task submission (Stahlberg et al., 2018a).

- Model의 문제 아님
- Length Constraint를 도입하여 문제 완화는 했으나 근본적인 해결책이 아님
- **Inherent Bias**가 문제로 보임... 찝찝한 결론으로 마무리





EMNLP 2019

On NMT Search Errors and Model

Felix Stahlberg* at
University of Cambridge
Department of Engineering
Trumpington St, Cambridge
{fs439, wjb31}@cam.ac.uk

Abstract

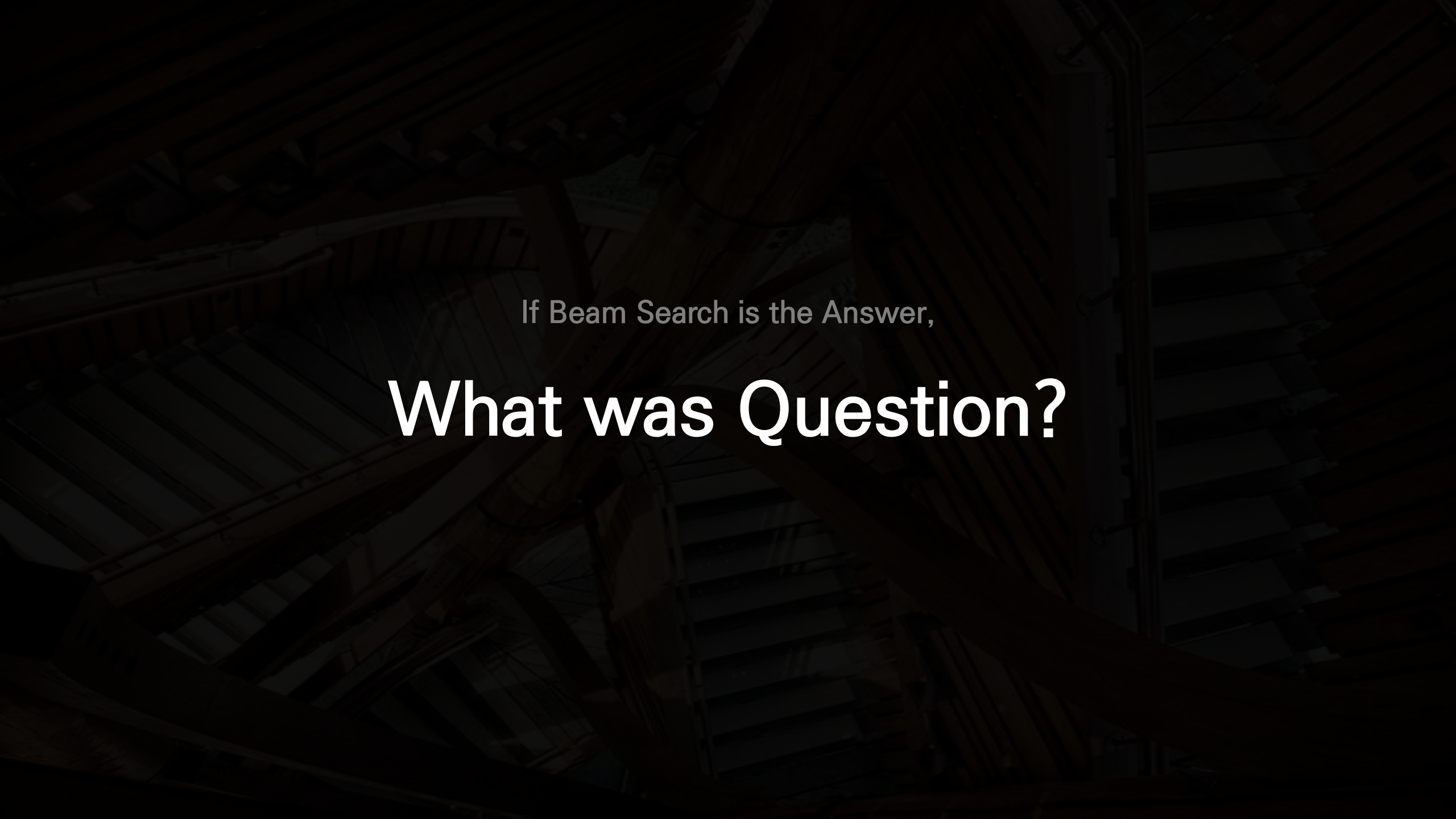
We report on search errors and model errors in neural machine translation (NMT). We present an exact inference procedure for neural sequence models based on a combination of beam search and depth-first search. We use our exact search to find the global best model scores under a Transformer base model for the entire WMT15 English-German test set. Surprisingly, beam search fails to find these global best model scores in most cases, even with a very large beam size of 100. For more than 50% of the sentences, the model in fact assigns its global best score to the empty translation, revealing a massive failure of neural models in properly accounting for adequacy. We show by constraining search with a minimum translation length that at the root of the problem of empty translations lies an inherent bias towards shorter translations. We conclude that vanilla NMT in its current form requires just the right amount of beam search errors, which, from a modelling perspective, is a highly unsatisfactory conclusion indeed, as the model often prefers an empty translation.

Abstract

We report on search errors and model errors in neural machine translation (NMT). We present an exact inference procedure for neural sequence models based on a combination of beam search and depth-first search. We use our exact search to find the global best model scores under a Transformer base model for the entire WMT15 English-German test set. Surprisingly, beam search fails to find these global best model scores in most cases, even with a very large beam size of 100. For more than 50% of the sentences, the model in fact assigns its global best score to the empty translation, revealing a massive failure of neural models in properly accounting for adequacy. We show by constraining search with a minimum translation length that at the root of the problem of empty translations lies an inherent bias towards shorter translations. We conclude that vanilla NMT in its current form requires just the right amount of beam search errors, which, from a modelling perspective, is a highly unsatisfactory conclusion indeed, as the model often prefers an empty translation.

- Exact Inference Procedure를 제안, Decoding 성능의 정량적인 평가를 가능케 함
- Beam Size를 크게 늘려도 Global best score에는 도달하지 못함
- Exact Inference, > 50% Sentence를 Empty Translate에 할당
- 이는 짧은 번역에 대한 inherent bias 문제로 보임
- Model이 Empty Translation을 선호
- “적절한 Beam Search Error가 필요하다”로 결론





If Beam Search is the Answer,

What was Question?



EMNLP 2020

If Beam Search is the Answer, What was the Question?

Clara Meister

Ryan Cotterell

Tim Vieira

ETH Zürich

University of Cambridge

Johns Hopkins University

clara.meister@inf.ethz.ch

tim.vieira@gmail.com

ryan.cotterell@inf.ethz.ch

Abstract

Quite surprisingly, exact maximum a posteriori (MAP) decoding of neural language generators frequently leads to low-quality results (Stahlberg and Byrne, 2019). Rather, most state-of-the-art results on language generation tasks are attained using beam search despite its overwhelmingly high search error rate. This implies that the MAP objective alone does not express the properties we desire in text, which merits the question: if beam search is the answer, what was the question? We frame beam search as the exact solution to a different decoding objective in order to gain insights into why high probability under a model alone may not indicate adequacy. We find that beam search enforces *uniform information density* in text, a property motivated by cognitive science. We suggest a set of decoding objectives that explicitly enforce this property and find that exact decoding with these objectives alleviates the problems encountered when decoding poorly calibrated language generation models. Additionally, we analyze the text produced using various decoding strategies and see that, in our neural machine translation experiments, the extent to which this property is adhered to strongly correlates with BLEU. Our code is publicly available at <https://github.com/rycolab/uid-decoding>.

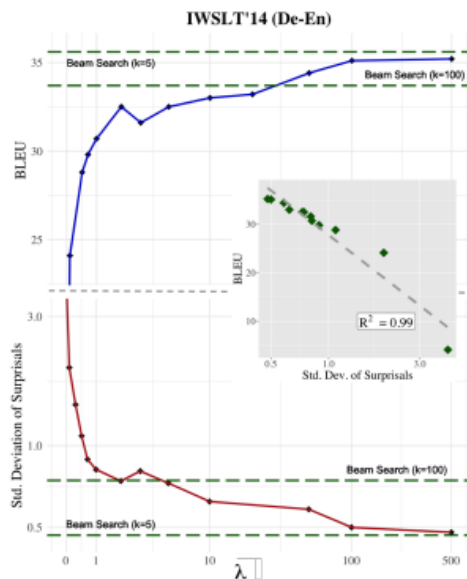


Figure 1: Average std. deviation σ of surprisals (per sentence) and corpus BLEU for translations generated using exact search over the MAP objective with a greedy regularizer (Eq. (11)) with varying degrees of λ . References for beam search ($k = 5$ and $k = 100$) are included. Sub-graph shows the explicit relationship between BLEU and σ . λ and σ axes are log-scaled.

Abstract

Quite surprisingly, exact maximum a posteriori (MAP) decoding of neural language generators frequently leads to low-quality results (Stahlberg and Byrne, 2019). Rather, most state-of-the-art results on language generation tasks are attained using beam search despite its overwhelmingly high search error rate. This implies that the MAP objective alone does not express the properties we desire in text, which merits the question: if beam search is the answer, what was the question? We frame beam search as the exact solution to a different decoding objective in order to gain insights into why high probability under a model alone may not indicate adequacy. We find that beam search enforces *uniform information density* in text, a property motivated by cognitive science. We suggest a set of decoding objectives that explicitly enforce this property and find that exact decoding with these objectives alleviates the problems encountered when decoding poorly calibrated language generation models. Additionally, we analyze the text produced using various decoding strategies and see that, in our neural machine translation experiments, the extent to which this property is adhered to strongly correlates with BLEU. Our code is publicly available at <https://github.com/rycolab/uid-decoding>.

Cogn Psychol. Author manuscript; available in PMC 2011 Aug 1.

PMCID: PMC2896231
NIHMSID: NIHMS185965

Published in final edited form as:

PMID: 20434141

Cogn Psychol. 2010 Aug; 61(1): 23–62.

doi: 10.1016/j.cogpsych.2010.02.002

Redundancy and reduction: Speakers manage syntactic information density

T. Florian Jaeger

Author information Copyright and License information Disclaimer

The publisher's final edited version of this article is available at Cogn Psychol.

See other articles in PMC that cite the published article.

Abstract

Go to:

A principle of efficient language production based on information theoretic considerations is proposed: Uniform Information Density predicts that language production is affected by a preference to distribute information uniformly across the linguistic signal. This prediction is tested against data from syntactic reduction. A single multilevel logit model analysis of naturally distributed data from a corpus of spontaneous speech is used to assess the effect of information density on complementizer *that*-mentioning, while simultaneously evaluating the predictions of several influential alternative accounts: availability, ambiguity



Neural Probabilistic Text Generation

- Neural probabilistic language generators provide a (conditional) probability distribution over all sequences of text y given some input x . We then “generate” text according to this distribution

our model, *e. g.*, a neural seq2seq model: $p_{\theta}(y|x)$ $\mathcal{Y} := \{\text{BOS} \circ v \circ \text{EOS} | v \in \mathcal{V}^*\}$

- In the case of neural generators, we typically model locally normalized distributions over words at each time steps:

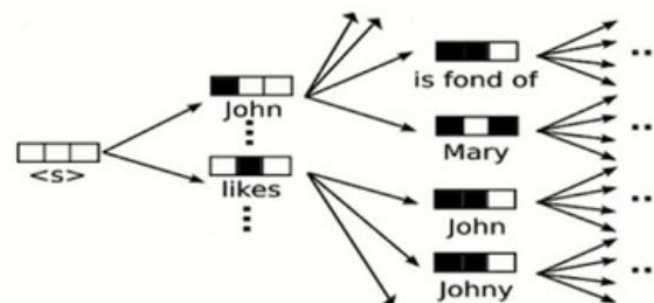
$$p_{\theta}(y|x) = \prod_{t=1}^{|y|} p_{\theta}(y_t|x, y_{<t})$$

$$\bar{\mathcal{V}} := \mathcal{V} \cup \{\text{EOS}\}$$

$$y_{<1} = y_0 := \text{BOS}$$

- The decoding problem (a.k.a. maximum a posteriori (MAP) inference)

$$y^* = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} (\log p_{\theta}(y|x))$$



Non-Markovian Structure! → DP method은 탐색이 효율적이지 못함...



Neural Probabilistic Text Generation

- How often does beam search find the global optimum in language generation tasks?

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%

Table 1: NMT with exact inference. In the absence of search errors, NMT often prefers the empty translation, causing a dramatic drop in length ratio and BLEU.

- The solution of MAP inference is clearly not desirable text...

$$y^* = \operatorname{argmax}_{y \in \mathcal{Y}} \log p_{\theta}(y|x)$$



Neural Probabilistic Text Generation

- But the solution provided by beam search is...?

$$y^* = \operatorname{argmax}_{y \in \mathcal{Y}} \quad ?$$

- Our (clunky) algorithm for beam search

$$Y_0 = \{\text{BOS}\}$$

$$Y_t = \operatorname{argmax}_{\substack{Y' \subseteq \mathcal{B}_t, \\ |Y'|=k}} \log p_\theta(Y' \mid \mathbf{x})$$

$$p_\theta(Y|x) = \prod_{y \in Y} p_\theta(y|x)$$

$$\mathcal{B}_t = \left\{ \mathbf{y}_{t-1} \circ y \mid y \in \bar{\mathcal{V}} \textbf{ and } \mathbf{y}_{t-1} \in Y_{t-1} \right\}$$

Return $Y_{n_{\max}}$

이 알고리즘을 최적화 문제로 쓸 수 있을까?



Deriving Beam Search

- $k=1$ (greedy search)

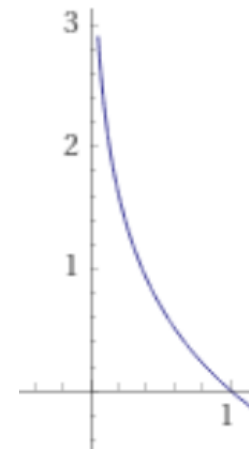
$$y^* = \operatorname{argmax}_{y \in \mathcal{Y}} (\underbrace{\log p_{\theta}(y|x)}_{\text{original objective}} - \underbrace{\lambda \cdot R(y)}_{\text{regularized term}})$$

$$R_{\text{greedy}}(y) = \sum_{t=1}^{|y|} \left(u_t(y_t) - \min_{y' \in \mathcal{V}} u_t(y') \right)^2$$

Locally Optimal
Decisions

Theorem 3.1. *The argmax of $\log p_{\theta}(y | \mathbf{x}) - \lambda \cdot \mathcal{R}_{\text{greedy}}(y)$ is exactly computed by greedy search in the limiting case as $\lambda \rightarrow \infty$.*

Surprisal



$$u_0(BOS) = 0$$

$$u_t(y) = -\log p_{\theta}(y|x, y_{<t}) \quad \forall t \geq 1$$

Three Axiom of Self-Information

- 100% 발생할 사건은 놀랍지 않고 (unsurprisal) 어떠한 정보도 제공하지 않는다
- 드물게 발생할 사건은 놀랍고 (surprisal) 더 많은 정보를 제공한다
- 독립 사건들의 총 정보량은 각 사건 self-information들의 합과 같다



Deriving Beam Search

- $k > 1$ (beam search)

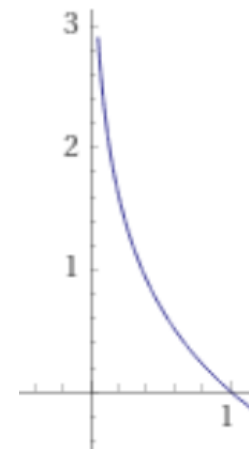
$$Y^* = \operatorname{argmax}_{Y \in \mathcal{Y}, |Y|=k} (\underbrace{\log p_{\theta}(Y|x)}_{\text{original objective}} - \underbrace{\lambda \cdot R(Y)}_{\text{regularized term}})$$

$$R_{\text{beam}}(Y) = \sum_{t=1}^{|n_{\max}|} \left(u_t(Y_t) - \min_{y' \subseteq B_t, |Y'|=k} u_t(Y) \right)^2$$

Locally Optimal
Decisions

Theorem 3.2. *The argmax of $\log p_{\theta}(Y | \mathbf{x}) - \lambda \cdot R(Y)$ is computed by beam search with beam size of $k = |Y|$ as $\lambda \rightarrow \infty$.*

Surprisal



$$u_0(BOS) = 0$$

$$u_t(y) = -\log p_{\theta}(y|x, y_{<t}) \quad \forall t \geq 1$$

Three Axiom of Self-Information

- 100% 발생할 사건은 놀랍지 않고 (unsurprisal) 어떠한 정보도 제공하지 않는다
- 드물게 발생할 사건은 놀랍고 (surprisal) 더 많은 정보를 제공한다
- 독립 사건들의 총 정보량은 각 사건 self-information들의 합과 같다



Deriving Beam Search

Theorem 3.1. *The argmax of $\log p_{\theta}(\mathbf{y} \mid \mathbf{x}) - \lambda \cdot \mathcal{R}_{\text{greedy}}(\mathbf{y})$ is exactly computed by greedy search in the limiting case as $\lambda \rightarrow \infty$.*

- Locally optimal decisions
- Greedy search는 locally하게 가장 높은 확률을 가지는 결정을 찾는 regularizer의 limiting case
- Optimal MAP의 경우, globally optimality를 위해 각 local에 **compensatory decision**을 요구하는 경우가 존재



From Beam Search to UID

The theoretical crux of this paper hinges on a proposed relationship between beam search and the **uniform information density** hypothesis (Levy, 2005; Levy and Jaeger, 2007), a concept from cognitive science:

Hypothesis 4.1. “*Within the bounds defined by grammar, speakers prefer utterances that distribute information uniformly across the signal (information density). Where speakers have a choice between several variants to encode their message, they prefer the variant with more uniform information density (ceteris paribus)*” (Jaeger, 2010).

Journal List > HHS Author Manuscripts > PMC2896231



[Cogn Psychol](#). Author manuscript; available in PMC 2011 Aug 1.

PMCID: PMC2896231

Published in final edited form as:

NIHMSID: NIHMS185965

[Cogn Psychol](#). 2010 Aug; 61(1): 23–62.

PMID: 20434141

doi: [10.1016/j.cogpsych.2010.02.002](#)

Redundancy and reduction: Speakers manage syntactic information density

[T. Florian Jaeger](#)

► Author information ► Copyright and License information [Disclaimer](#)

The publisher's final edited version of this article is available at [Cogn Psychol](#)

See other articles in PMC that [cite](#) the published article.

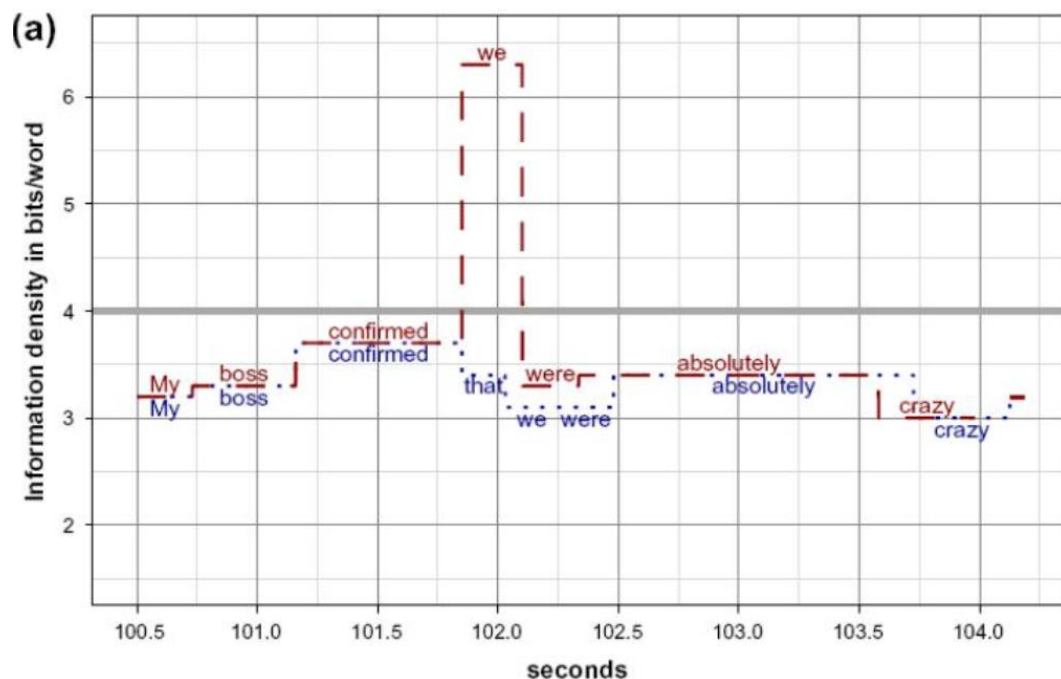
Abstract

Go to:

A principle of efficient language production based on information theoretic considerations is proposed: Uniform Information Density predicts that language production is affected by a preference to distribute information uniformly across the linguistic signal. This prediction is tested against data from syntactic reduction. A single multilevel logit model analysis of naturally distributed data from a corpus of spontaneous speech is used to assess the effect of information density on complementizer *that*-mentioning, while simultaneously evaluating the predictions of several influential alternative accounts: availability, ambiguity avoidance, and dependency processing accounts. Information density emerges as an important predictor of speakers' preferences during production. As information is defined in terms of probabilities, it follows that production is probability-sensitive, in that speakers' preferences are affected by the contextual probability of syntactic structures. The merits of a corpus-based approach to the study of language



From Beam Search to UID



- 빈번한 단어는 더 짧은 언어 형태를 가짐
- 단어의 길이는 예측 가능성과 연관이 깊음
- 정보는 context-dependent함 (가능성이 높을 수록 불필요함)
- 화자는 언어 신호 양 당 정보량을 관리함
- 축소 가능 단위가 많은 정보를 encode할 때 마다 언어적 신호가 적은 형식은 선호되지 않아야 함
- UID; 평균적으로 각 단어는 우리가 이미 알고있는 정보에 **동일한 양**을 추가
- My boss confirmed (that) we were absolutely crazy
- 관계 대명사 that을 포함하면, 정보를 두 단어로 퍼뜨려 문장 전체에 정보를 보다 **고르게** 배포하고 언어심리학적으로 불쾌감을 주는 **높은 surprisal**을 피할 수 있음



From Beam Search to UID

Importantly, the preference suggested by the UID hypothesis is between possible utterances (i.e., outputs) where grammaticality and information content are held constant. Any violation of these assumptions presents confounding factors when measuring, or optimizing, the information density of the generated text. In our setting, there is reason to believe that grammaticality and information content are approximately held constant while selecting between hypothesis. First, the high-probability

- Holtzman et al., 2020에 따르면, neural generation model들의 가장 높은 확률을 가지는 출력은 문법적인 경향이 많다.
- 조건부 확률 모델은 주어진 input sentence의 그럴 듯한 output sentence에 높은 확률을 할당
 - 다양한 output sentence들이 의미상 동일하도록 제한하는 것은 X
 - 최소한 동일한 input sentence에 관련이 있기 때문에
 - 유사한 정보를 표현하는 경향이 존재

$$y^* = \operatorname{argmax}_{y \in \mathcal{Y}} (\log p_{\theta}(y|x) - \lambda \cdot R(y))$$

Original objective;
Rewards grammatically and content relevance

Regularized term;
Encourages the human preferences posited by the UID hypothesis



From Beam Search to UID

To shorten path or 나중에 더 낮은 surprisal step을 취하기 위해 중간에 higher-surprisal을 취해버리는 현상 발생

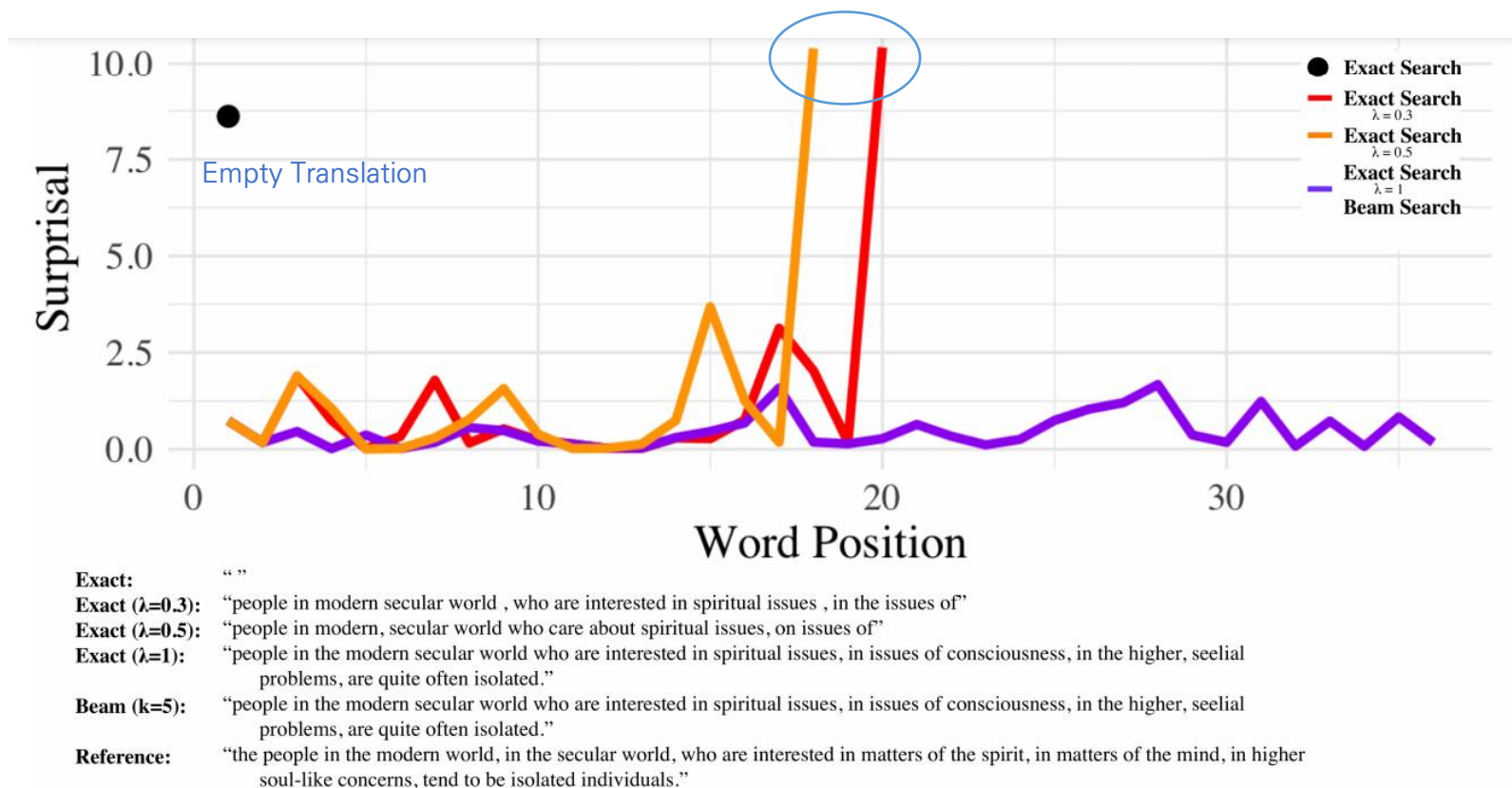


Figure 2: Surprisals (according to p_θ) by time step of sequences generated with various decoding strategies. Values of λ indicate the greedy regularizer was used with the corresponding λ value. Note that beam search (k=5) and exact search ($\lambda = 1.0$) return the same prediction in this example, and thus, are represented by the same line.



From Beam Search to UID

Hypothesis 4.2. *Beam Search is a cognitively motivated search heuristic for decoding language generation model. The success of beam search on such tasks is, in part, due to the fact that it inherently biases the search procedure towards text that human prefer.*



Experiments



Generalized UID Decoding

- Variance Regularizer

$$R_{var}(y) = \frac{1}{|y|} \sum_{t=1}^{|y|} (u_t(y_t) - \mu)^2 \quad \mu = \frac{1}{|y|} \sum_{t=1}^{|y|} u_t(y_t)$$

- Local Consistency

$$R_{local}(y) = \frac{1}{|y|} \sum_{t=1}^{|y|} (u_t(y_t) - u_{t-1}(y_{t-1}))^2$$

- Max Regularizer

$$R_{max}(y) = \max_{t=1 \sim |y|} u_t(y_t)$$

- Squared Regularizer

$$R_{square}(y) = \sum_{t=1}^{|y|} (u_t(y_t))^2$$



Dataset and Model Setting & Results

Experiments are performed using models trained on the IWSLT'14 De-En (Cettolo et al., 2012) and WMT'14 En-Fr (Bojar et al., 2014) datasets. For reproducibility, we use the model provided by fairseq (Ott et al., 2019) for the WMT'14 task;⁹ we use the data pre-processing scripts and recommended hyperparameter settings provided by fairseq for training a model on the IWSLT'14 De-En dataset. We use the Newstest'14 dataset as the test set for the WMT'14 model. All model and data information can be found on the fairseq NMT repository.¹⁰

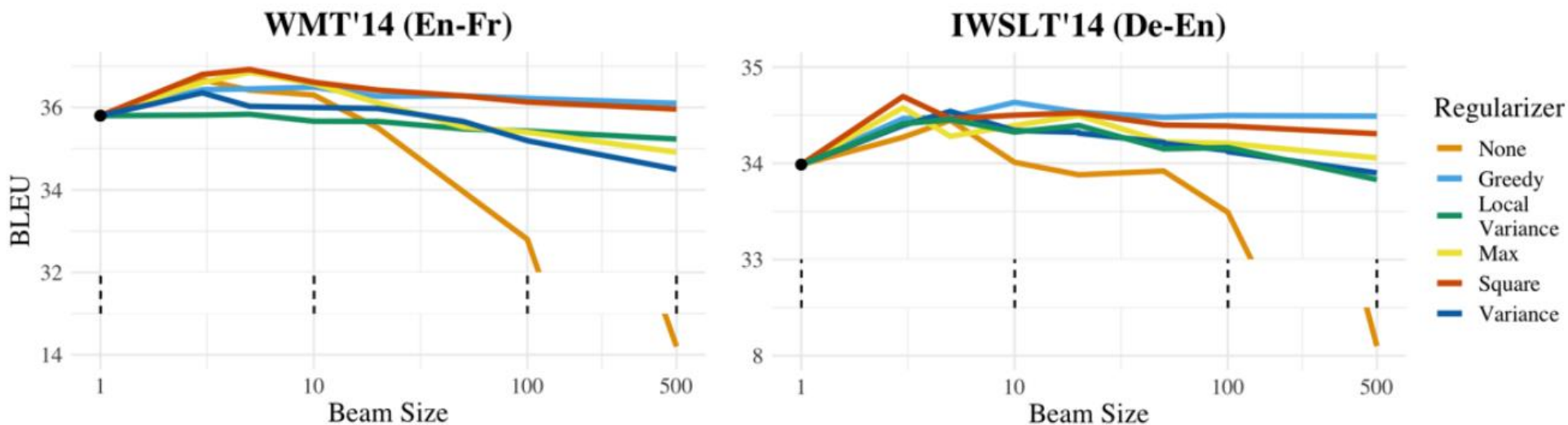
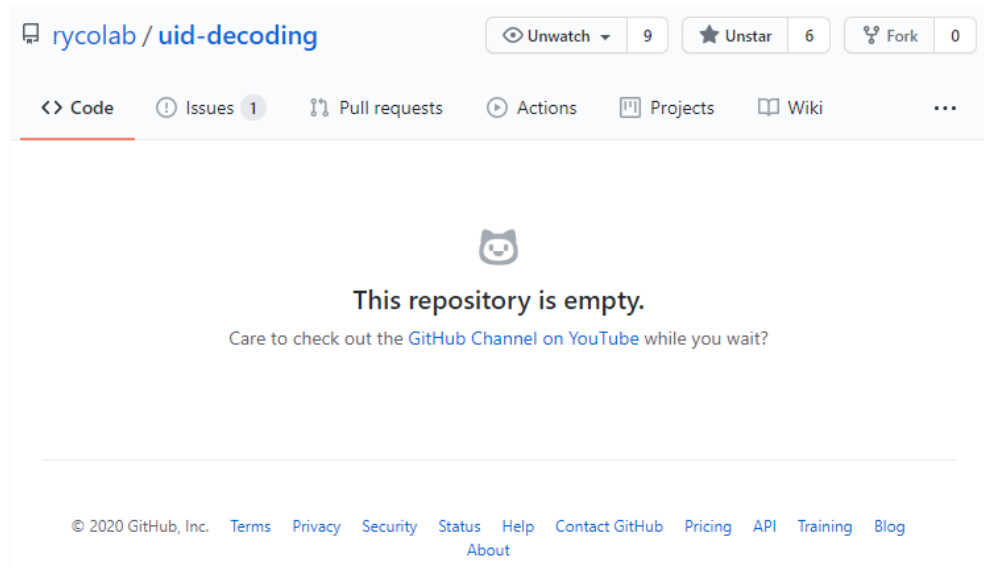


Figure 3: BLEU as a function of beam width for various regularizers. We choose λ for each regularizer by best performance on validation sets (see App. B). y -scales are broken to show minimum BLEU values. x -axis is log-scaled.

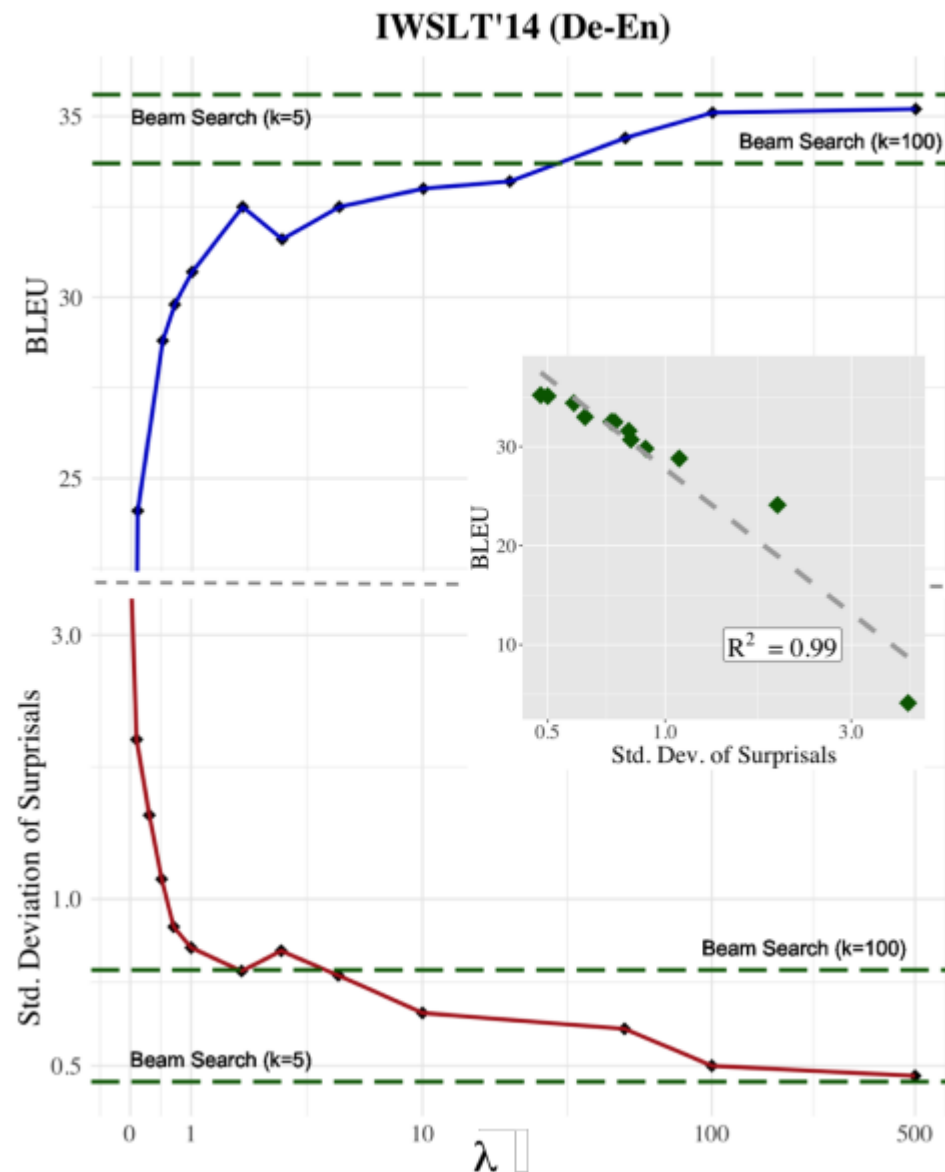


Conclusion

30

8 Conclusion

We analyze beam search as a decoding strategy for text generation models by framing it as the solution to an exact decoding problem. We hypothesize that beam search has an inductive bias which can be linked to the promotion of uniform information density (UID), a theory from cognitive science regarding even distribution of information in linguistic signals. We observe a strong relationship between variance of surprisals (an operationalization of UID) and BLEU in our experiments with NMT models. With the aim of further exploring decoding strategies for neural text generators in the context of UID, we design a set of objectives to explicitly encourage uniform information density in text generated from neural probabilistic models and find that they alleviate the quality degradation typically seen with increased beam widths.





Reference

- <https://www.youtube.com/watch?v=D00OqMJsgs4>
- <https://steemit.com/kr-english/@bree1042/english-42-1-feat-carrotcake-cat-got-your-tongue>
- <https://www.slideserve.com/dwayne/redundancy-and-reduction-speakers-manage-syntactic-information-density>
- <https://www.aclweb.org/anthology/W18-4605/>
- <https://kangbk0120.github.io/articles/2018-03/information-theory>
- https://en.wikipedia.org/wiki/Information_content
- <https://www.chemurope.com/en/encyclopedia/Self-information.html>
- <https://github.com/msu-coinlab/pymoo>
- http://pymoo.org/customization/subset_selection.html
- <https://arxiv.org/pdf/2010.02650.pdf>
- <https://www.aclweb.org/anthology/D19-1331.pdf>
- <https://gmlwjd9405.github.io/2018/08/14/algorithm-dfs.html>
- <https://velog.io/@nawnoes/%EC%9E%90%EC%97%B0%EC%96%B4%EC%B2%98%EB%A6%AC-Beam-Search>
- <https://www.aclweb.org/anthology/D19-1331/>
- <https://twitter.com/bryaneikema/status/1264929092551487489>
- <https://scholar.google.com/citations?user=quJhNH8AAAAJ&hl=en>
- <https://twitter.com/claraisabelmei1>
- <https://arxiv.org/pdf/2005.10283.pdf>
- <https://towardsdatascience.com/an-intuitive-explanation-of-beam-search-9b1d744e7a0f>
- <https://www.aclweb.org/anthology/D17-2005/>
- <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2896231/>
- <https://machinelearningmastery.com/beam-search-decoder-natural-language-processing/>
- https://d2l.ai/chapter_recurrent-modern/beam-search.html
- Clara Meister, Tim Vieira, Ryan Cotterell | If beam search is the answer, what was the question? · [SlidesLive](#)