Federal State Autonomous Educational Institution for Higher Education
National Research University Higher School of Economics

Faculty of Computer Science
EducationalProgram
Applied Mathematics and Information Science

TERM PAPER

Research project

"Uncertainty estimation for Machine Translation"

Prepared by the student of group 171, 3rd year of study
Kuznetsov Dmitriy Sergeevich

Supervisor
research fellow Lobacheva Ekaterina Maksimovna

Moscow 2020

Contents

1 Annotation

In the field of machine translation *beam search* is the one of the main methods for improving the quality of the final prediction. However, when using neural network technologies, there is a problem of larger beam size, which limits the richness of beam search. Problem solution will significantly improve the quality of predictions by considering more hypotheses in the process of making predictions. Many factors in the model infrastructure can cause the appearance of the beam problem. In this paper we explore the *larger beam size problem* and its correlation with *model uncertainty*. We also study the propensity of models with an uncertainty estimation to cause beam problem.

2 Keywords

machine translation, uncertainty estimation, beam search, larger beam size problem, model uncertainty

3 Introduction

The task of machine translation is to provide a sentence equivalent in meaning in the target language for the original sentence in the source language. Let $e_1, \ldots, e_n$ be a sequence of input sentence tokens' vector representation in the source language. The machine translation model is required to build a sequence of tokens $f_1, \ldots, f_n$ in the target language, moreover, in this language, this sequence must have the original meaning.

Most modern neuromachine translation models follow the paradigm Sutskever et al. (2014) (seq2seq). Seq2seq models often consist of two recurrent neural networks or groups of networks. One network that processes input tokens is called *encoder*, the second network that builds output predictions is called *decoder*. The encoder task is to encode the input sequence into some hidden representation $h_1, \ldots, h_p$, which is fed to the decoder as hidden initialization before building the output prediction. The decoder output is a vector set $\{(p_{i1}, \ldots, p_{id})\}_{i=1}^{m}$, where d - target language dictionary power, m - maximum possible prediction sequence length, p_{ij} - the model's confidence that the ith token of the predicted sentence will be the jth token from the target language dictionary. As a result, the translation model builds a "probability distribution" in the space of the cartesian product of the target dictionary.

The simplest and most obvious way to build a prediction (final translation) is to select the token that the model is most confident in at each step. If $f_1, \ldots, f_m$ is a sequence of model hypotheses' tokens, then $f_i := argmax \, \{p_{ij}\}_{j=1}^{d}$. However, for an optimal prediction $f_1^*, \ldots, f_m^*$ it is not always true that in terms of model confidence $f_i^* = argmax \, \{p_{ij}\}_{j=1}^{d}$. This phenomenon is also related to the beam problem, which we will discuss in more detail in the following chapters.

Beam search is used as a more effective solution for building predictions. Its main idea is that instead of selecting the most likely token at each step, we require the model to store b the most likely predictions' prefixes $f_{b1}, \ldots, f_{bt}$. Using this approach, we do not choose the locally optimal token, but in general the optimal prefix, which will increase the chances of building a correct prediction.

It is easy to understand that with the increase of b (this hyperparameter is called *beam size*), the quality of the prediction should also increase, since the number of greedy searches is growing. In practice, it turns out that in neuromachine translation, starting from a certain threshold, the quality decreases significantly with increasing beam size. An illustration of this phenomenon can be seen in Fig. 1. The model used in the experiment is described in the Jonas Gehring (2017).

This problem is called *larger beam size problem*. Next, we will look at the possible causes of this phenomenon in details, until we note that the following situation is not excluded. Suppose on k_1 iteration step beam search stores the prefix $f_1, \ldots, f_{k_1}$ among b locally optimal, which has the least model confidence among all b prefixes. Let this prefix be constructed in such a way that on the next steps model will find a suffix that complements it in such a way that the final sentence is considered the optimal translation from the model's point of

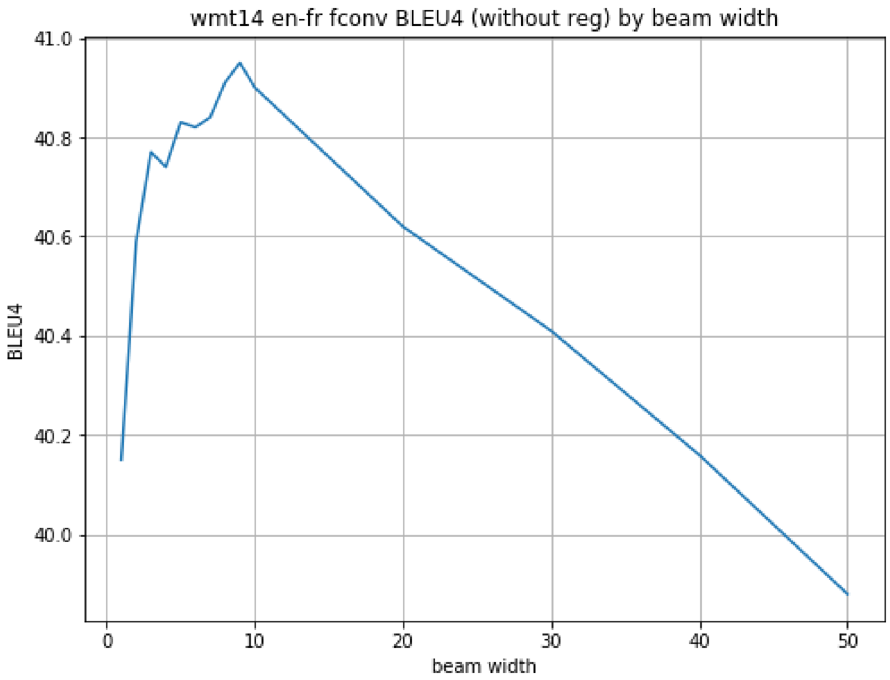


Figure 1 – Convolutional Sequence to sequence model BLEU4 evaluation by beam size. Training set: *WMT'14 English-France*

view. There are cases when such a translation can be incorrect. Consider following example. Let us train a translation model on a fairy tale corpus, the source language is Russian and the target language is English. Consider the following sentence: *Жульничество на экзамене однажды может закончиться проблемами.* Suppose following sentence is a model prediction: *At the examination once upon a time.* It is obvious that the meaning of the translation does not correspond to the original sentence, but this sentence is optimal from the point of view of our hypothetical model. Beam problem can cause such situations. The prefix *At the examination once* may have been included in a broad beam search and have been earned a low model confidence. However, due to the fact that we train on a large corpus of fairy tales, where there are many examples of sentences like *Once upon a time...* our model decides that after *Once* it is probably to continue prefix with suffix *upon a time*, which lead to a bad translation.

Note that in the example, some features of the model lead to a bad translation. Our model gave low probable in general predictions as a result under the influence of certain factors. *Uncertainty* can cause such phenomena, in particular *model uncertainty*, the uncertainty of predictions caused by the model. What if we can construct some probability measure on the space of output sentences $f_1, \dots, f_m$. In this case, we will be able to estimate in total how much it is typical to get a particular prediction and adjust the output of the model. The task of explicitly or implicitly constructing such a distribution on outputs is called *uncertainty estimation.*

There are articles that consider different types of uncertainties and their impact on beam search. In this paper, we investigate the influence of model uncertainty on the beam problem. We also investigate the applicability of the uncertainty estimation methods in the neural machine translation problem and their impact on the effectiveness of beam search.

4 Beam Search

4.1 Method definition

Suggest k-long prediction prefix:

$$f^i_{:k} := (f^i_1, \dots, f^i_k), \quad \text{where i - beam search branch index} \tag{1}$$

Let us introduce following iterative method:

1 Let us submit a special token of the beginning of the sentence and the encoder hidden state to the decoder input. We get a certain tokens' confidence vector $(p^{00}_1, \dots, p^{00}_d)$, where d - target dictionary power.

2 Take tokens whose confidence values are *beam* highest among $(p^{00}_1, \dots, p^{00}_d)$. Here *beam* is the hyperparameter of the method and is called *beam size.* As a result, we will remember: $f^1_{:1}, \dots, f^{beam}_{:1}$.

3 In the next iteration, we will submit to the decoder $(f^{00}_1, \dots, f^{00}_d)$ as inputs and the hidden state of the previous step. For each token, we get our output distribution $(p^{1i}_1, \dots, p^{1i}_d)$ (here 1 is the iteration number, i is the token number).

4 For all token $i \in \overline{1, beam}$ consider $\forall j \in \overline{1, d}$: $p(f^i_{:1}) * p^{1i}_j$. We get a set of prefixes of length 2. Let us choose among all $beam * d$ prefixes $beam$ with the greatest confidence. Remember them: $f^1_{:2}, \dots, f^{beam}_{:2}$

5 Iteratively, we will continue the operation $\forall k \in \overline{3, m}$.

We will get: $f^1_{:k}, \dots, f^{beam}_{:k}$

6 As the final prediction select $f^i_{:m}, \dots, f^i_{:m}$ such that its confidence is greatest $\forall i \in \overline{1, beam}$

In summary, this method stores *beam* of the "most likely" prefixes at each decoding iteration, in contrast to the naive approach, which greedily selects 1 locally optimal hypothesis at each iteration.

4.2 Large Beam Size Problem

In General, in neuromachine translation tasks large beam size problem is a phenomenon, as a result of which, starting from a certain threshold value, the quality metric decreases with the growth of the beam size. In this section, we will look at some approaches that do not solve the problem completely, but significantly reduce the effect of metric degradation.

The article Murray et al. (2018) raises two issues: the beam problem and the tendency of NMT models to make short predictions. According to the author, solving the short prediction problem involves solving the beam problem.

As a demonstration of the reasons listed in this article that cause the beam problem, consider the following example from the article.

On the Fig.2. we can see a tree of Beam Search predictions for the word *un hélicoptère*. Let the size of the beam search be 2 and on the first iteration we saved two hypotheses: *a* and *an*. Note that all 4 following hypotheses can be potential translation options, but *a helicopter* in this example is a correct translation. However, *autogyro* is an only one continuation for prefix *an*, thefore the model confidence for this suffix is 1, and as a result, the model confidence

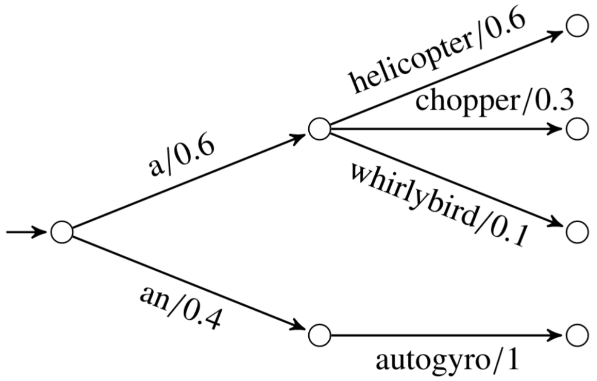


Figure 2 – Label bias causes this toy word-by-word translation model to translate French helicopter incorrectly. This figure is taken from [4].

to predict *an autogyro* is equal to 0.4. At the same time, the probability mass for the *a* continuation suffixes is divided among the words: *helicopter, chopper, whirlybird.* As a result, the confidence for the correct translation of *a helicopter* from the model's point of view is 0.36. As a result, the model will give a bad prediction. Larger beam size we take, higher probability we get in this situation, hence the loss of quality.

Note that this example demonstrates the fact that if a certain prefix involves a low-entropy suffix, the model tends to ignore the second input token and predict with high confidence one of several highly probable low-entropy suffixes. As a result, the low entropy of the suffix leads to overestimation of this prediction branch and as a result, poor quality. In addition, there is a high probability of getting short predictions due to low entropy, since the model begins to ignore some tokens in favor of high confidence of low-entropy suffixes. For this reason, the authors of the article consider the short prediction problem and the beam problem together.

Standard quality functional for a machine translation task is *cross-entropy*:

$$s(f) = \Sigma_{i=1}^{L} \log p(f_i|f_{:i}), \text{ where } L \text{ - translation length} \tag{2}$$

The authors consider the following quality functional adjustments as solutions:

1 Length normalization

$$s'(f) = s(f)/L \tag{3}$$

2 Google's NMT system (2016). Length normalization.

$$s'(f) = s(f) \Big/ \frac{(5+L)^\alpha}{(5+1)^\alpha} \tag{4}$$

3 Word reward

$$s'(f) = s(f) + \gamma L \tag{5}$$

Let us look at the results presented in the article for the *WMT'17 Russian-English* model. The architecture presented in the article Bahdanau et al. (2015) was chosen as the baseline. Appealing to experiments results from the paper we observe that the solution of the short prediction problem using methods proposed above significantly decreases the influence of the beam problem on the considered dataset. Comparing *word reward* and length normalization, we can see that they earn quite close BLEU values and lager beam size problem affects models in a same way, when baseline model suffers from beam problem significantly.

Let us now look at the article Koehn et al. (2017). This is a review article on the main problems of neuromachine translation, including the beam problem.

The authors of the article do not present new solutions for the problem, but conduct a review study. As a baseline, they use the same neural network as in the previous article: attention-based encoder-decoder. Fig. 3 presents their results for various *WMT* datasets. Here, normalization is the Length normalization from the previous article.

The results obtained in this article are consistent with the results and conclusions of the previous article.

Consider an article that offers a different method for solving the problem Kumar et al. (2019).

In this article the beam problem solution is based on the calibration of the probability distribution $p_{it} := p(f_{i,t}|f_{i,:t})$, here i is the index of the sample exam-

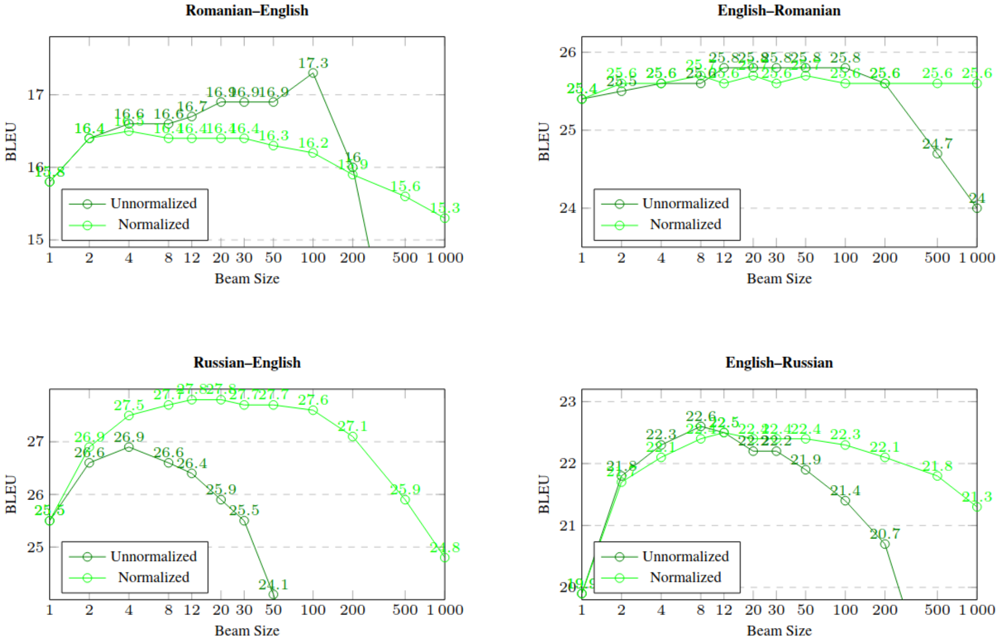


Figure 3 – Translation quality with varying beam sizes. For large beams, quality decreases, especially when not normalizing scores by sentence length. This figure is taken from [7].

ple. Before starting the review, let us introduce the definition of "well-calibrated" distribution. Output distribution $p(f_{i,t}|f_{i,:t})$ *well calibrated* if:

$$\forall\beta \in [0,1] : \frac{|\{y \in V \,|\, p(y) = \ beta\}|}{d} = \beta$$

where V is the target language dictionary.

Let us denote calibration metrics **expected calibration error (ECE)**:

Let $\mathbf{x}_i$ reference sentece is $\mathbf{y}_i = (y_{i1}, \ldots, y_{im})$.

Model predictions: $\mathbf{f}_{it} := argmax_f p(f|f_{i,:t})$.

$C_{it}(f) := \delta(y_{it} = f)$, where δ - Kronecker delta.

Let us divide probability space $[0, 1]$ by M equal bins: $I_1, \ldots, I_M$.

Let $L = \sum_i^N |\mathbf{y}_i|$ be a total output token length.

Then:

$$ECE = \frac{1}{L}\sum_{b=1}^{M}\Big|\sum_{i,t:\, p(\mathbf{f}_{it})\in I_b} C_{it}(\mathbf{f}_{it}) - p(\mathbf{f}_{it})\Big| \qquad (6)$$

The reasons why the probability distribution calibration can solve problems with beam size follow from the examples we discussed above.

The authors consider the classic *temperature scaling as the basic method*:

The method is quite simple. It is necessary to raise all probability distribution to the power of $\frac{1}{T}$, where T is the hyperparameter of the method:

$$p_{it} \to p_{it}^{\frac{1}{T}}$$

This approach does not change the relative values of probabilities, i.e. it does not locally change the optimality of a particular choice, but it helps to smooth out the consequences of overestimating or underestimating certain prefixes or suffixes.

The authors consider 3 models trained on *WMT* as a baseline: attention-based encoder-decoder (2015), GNMT Wu et al. (2016), transformer Vaswani et al. (2017). Authors build probability plots for each model output distribution with and without classic temperature scaling. Results show that transformer model output distribution is closer to perfect calibration than other models, even without scaling. Classic attention-based encoder-decoder, trained on English-Vietnamese WMT dataset, shows the worst calibrated output distribution without using temperature scaling. Other model in a list, trained on other sets of data, shows quite similar results without calibration method and with temerature scaling, this results differ a little from transformer values. Graphs show that using classic temperature scaling probability plot goes to perfect, ECE decreases even in the worst case scenarios.

In order not to overweight the literary review, we will not formally introduce the method proposed by the authors. We will consider conceptual differences from the default temperature scaling. To study the formal definition of the method, you can refer to the article Kumar et al. (2019).

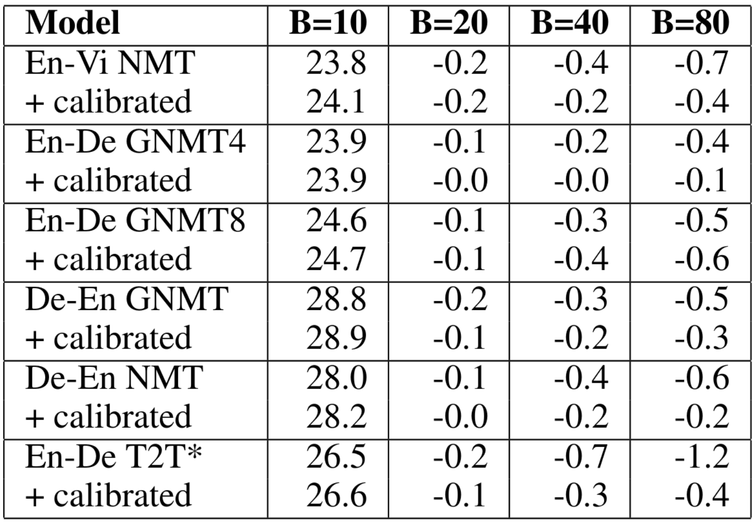
As a result of experiments, the authors found out that end of sequence (EOS) token probability is the worst calibrated among all other tokens. Therefore, first of all, the authors calibrate the distribution of the EOS token. This observation strongly correlates with the conclusions from the articles reviewed earlier. In contrast to the classic temperature scaling and the hyperparameter selection on a validation set, the authors propose to make the hyperparameter flexible, different for each example and token index, i.e. not to smooth the entire probability mass, but to smooth it depending on the offset of the probability density on the token. To find the values of hyperparameters, it is proposed to use two fully connected two-layer neural networks with 3 neurons on the hidden layer. The composition of two neural networks with optimization *Negative Log Likelihood (NLL)* is proposed to train to predict the value of the hyperparameter T for a given example and a token in the prediction.

Authors present experiments results, that comparing ECE and BLEU for baseline models with authors calibration method, classic temperature scaling and without any calibration. In generall we can say that temperature scaling gives significant fall comparing to models without any calibration (about several points of number). Authors' calibration method shows the best results except only one case, ECE of output distribution is less than ECE of temperature scaling models (about several points of number after the dot). It can also be seen that the authors' calibration method achieves the best results on the most part of the baseline models from the BLEU point of view.

Now, turning to Fig. 4, let us look at the effect of the authors' calibration on the beam size problem. It is easy to see that calibration significantly reduces the impact of the beam problem.

In this section, we investigated the possible causes of the beam problem, as well as methods of counteraction. Unfortunately, it is not possible to draw unambiguous conclusions from the results of the article about which method shows the best results due to the fact that the experiments in the articles were carried out in different configurations. According to approximate estimates, both methods show good results, are easy to implement, but do not solve the problem completely.

In the next section, we look at a different view of the beam problem, from the perspective of uncertainty.



Model	B=10	B=20	B=40	B=80
En-Vi NMT	23.8	-0.2	-0.4	-0.7
+ calibrated	24.1	-0.2	-0.2	-0.4
En-De GNMT4	23.9	-0.1	-0.2	-0.4
+ calibrated	23.9	-0.0	-0.0	-0.1
En-De GNMT8	24.6	-0.1	-0.3	-0.5
+ calibrated	24.7	-0.1	-0.4	-0.6
De-En GNMT	28.8	-0.2	-0.3	-0.5
+ calibrated	28.9	-0.1	-0.2	-0.3
De-En NMT	28.0	-0.1	-0.4	-0.6
+ calibrated	28.2	-0.0	-0.2	-0.2
En-De T2T*	26.5	-0.2	-0.7	-1.2
+ calibrated	26.6	-0.1	-0.3	-0.4

Figure 4 – BLEU with increasing beam on the devset. *Beam sizes for Transformer/T2T: 4, 8, 10 and 12. This figure is taken from [8].

5 Uncertainty

5.1 Definition

This information was obtained from the article Malinin et al. (2018)

Uncertainty in machine learning is usually understood as a measure of the nondegeneration of the distribution on the model predictions at a fixed input. Less formally, if we were able to estimate uncertainty, we could tell how much we can trust the model's predictions.

There are several types of uncertainties: data uncertainty, model uncertainty, and distributive uncertainty. *Data uncertainty (aleatoric uncertainty)* - type of uncertainty caused by the nature of the data, including noise, class balance, etc. *Model uncertainty (epistemic uncertainty)* measures uncertainty caused by a model specifity, how good a model understands given data. *Distributional uncertainty* measures similarity between training and test data, caused by unknown or strange test examples.

It is important for the model to evaluate all types of uncertainty, because they are caused by different reasons and have different negative effects.

5.2 Analyzing Uncertainty in Neural Machine Translation

In the article Ott et al. (2018) *data uncertainty* is being investigated.

Let us introduce several base definition from the article. *Intrinsic uncertainty* is the result of existence of several semantically equivalent translations of the same source sentence, for instance there are several ways to express the same meaning. Also situations when target language is more complex than a source language, it can cause an uncertainty. In such situations it could be impossible to earn additional information, such as gender or tense, which is necessary to build correct prediction in a source language. Web data crawling to construct train corpus can cause *extrinsic uncertainty*. For example, authors notice that at least 1% sentences in WMT datasets is just a copy of a source sentences.

The architecture described in the article Gehring et al. (2017) is specified as the baseline of the experiments.

One of the main problems that is raised in this article is an effect of *uncertainty* on the degradation of quality with the growth of beam size. Authors notices copies (extrinsic uncertainty) are overrepresented in the output of beam search. Also they say that bigger beam size we take, more copies appear in a beam search tree, and this growth is quite significant.

The authors conducted an experiment, the results of which can be seen in Fig. 5, in which WMT'17 added explicitly replaced random examples from the training with copies of the original sentence. It is clearly seen that this leads to quality degradation with increasing noise. Moreover, models with a larger beam size suffer more from extrinsic uncertainty, which confirms the thesis put forward in the previous paragraph.

The authors suggest two approaches to preparing data for training. First, they remove train examples, which has low score in a model trained on the news-commentary part of WMT'17 En-De point of view (filtered). Second, they restrict beam search hypotheses, which has BLEU with source sentence more than 50% (no copy). The results can be seen in Fig. 6. They are fully consistent with the

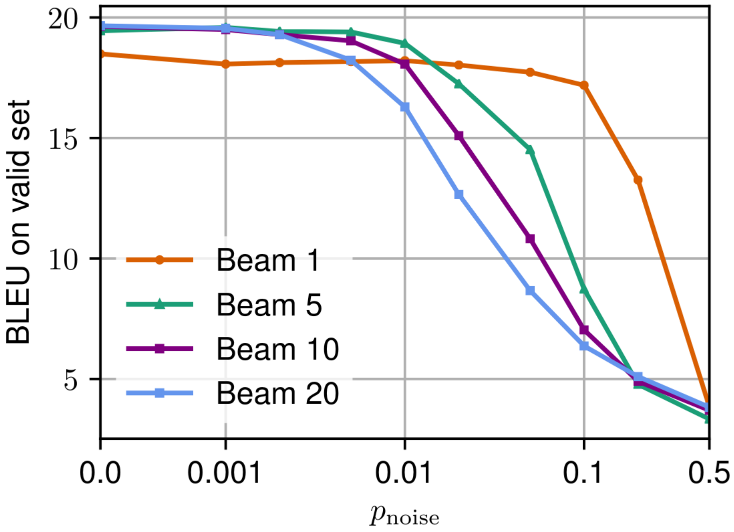


Figure 5 – Translation quality of models trained on WMT'17 English-German news-commentary data with added synthetic copynoise in the training data (x-axis) tested with various beam sizeson the validation set. This figure is obtained from [9].

conclusions obtained earlier.

6 Work Plan

We have studied the analysis of the raised problems and the existing solutions at the moment of work writing. The purpose of this work is to investigate the influence of *model uncertainty* on the larger beam size problem, and to try to apply existing methods in the field of *uncertainty estimation* in the problem of neuromachine translation.

The course work will be based on the following plan:

1 Study and analysis state-of-the-art models of neuromachine translation. Their applicability to the problem.

2 Selecting the appropriate set of training cases for problem analysis

3 Experiments results from the reviewed articles validation with different beam sizes on our own configurations.

4 Analysis of the results of the experiments from the point of view of uncertainty. Comparison of predictions for different beam sizes at the level of the

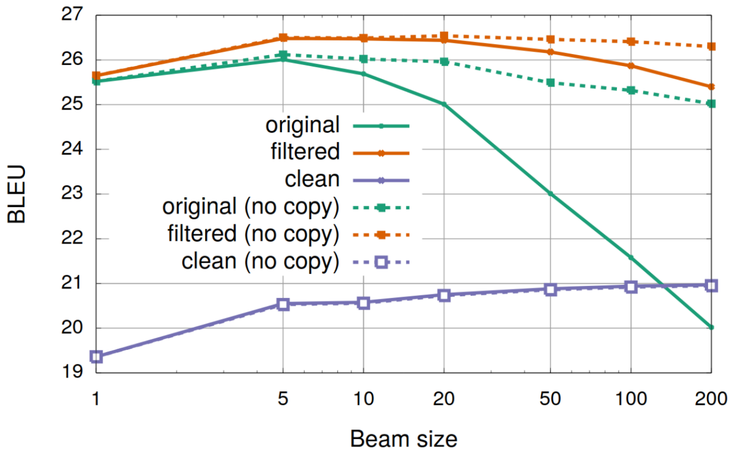


Figure 6 – BLEU on newstest2017 as a function of beam width for models trained on all of the WMT'17 En-De training data (original), a filtered version of the training data (filtered) and a small but clean subset of the training data (clean). They also show results when excluding copies as a post-processing step (no copy). This figure is obtained from [9].

beam search tree.

5 Research of existing solutions in the field of Uncertainty estimation

6 Selecting a set of potentially effective models for uncertainty estimation

7 Conducting experiments to study the effect of uncertainty estimation on the beam problem with models from a potentially effective set.

8 Implementation of the final configuration

9 Final test

10 Course work writing complition

References

1. Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, Yann N. Dauphin. Facebook AI Research. Convolutional Sequence to Sequence Learning. PMLR 70, 2017.

2. Ilya Sutskever, Oriol Vinyals, Quoc V.Le. Sequence to Sequence Learning with Neural Networks. Google. NIPS 2014.

3. Dzmitry Bahdanau, KyungHyun Cho, Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. ICLR 2015.

4. Kenton Murray, David Chiang. Department of Computer Science and Engineering, University of Notre Dame. Correcting Length Bias in Neural Machine Translation. WMT 2018.

5. Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi. Google's Neural Machine Translation System: Bridging the Gapbetween Human and Machine Translation. ArXiv 2016

6. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser. Attention Is All You Need. NIPS 2017.

7. Philipp Koehn, Rebecca Knowles. Six Challenges for Neural Machine Translation. NMT@ACL 2017.

8. Aviral Kumar, Sunita Sarawagi. Calibration of Encoder Decoder Models for Neural Machine Translation. ArXiv 2019.

9. Myle Ott, Michael Auli, David Grangier, Marc'Aurelio Ranzato. Analyzing Uncertainty in Neural Machine Translation. PMLR 80, 2018.

10. Andrey Malinin, Mark Gales. Predictive Uncertainty Estimation via Prior Networks. NeurIPS 2018.