Constraining the Transformer NMT Model with Heuristic Grid Beam Search*

Guodong Xie guodong.xie@adaptcentre.ie
Andy Way andy.way@adaptcentre.ie
Jinhua Du jinhua.du@adaptcentre.ie
ADAPT Centre, School of Computing, Dublin City University, Dublin, Ireland
Longyue Wang* vincentwang0229@gmail.com
Tencent AI Lab

Abstract

Constrained decoding forces a certain set words or phrases to appear in the translation results and is very useful when adapting MT to a certain domain. In recent years, the Transformer model has outperformed other neural machine translation models to become the state-of-the-art paradigm. However, constrained decoding for domain adaptation remains an open problem under the Transformer model. In this paper, we first investigate how a constrained decoding method – Grid Beam Search (GBS) – performs in the Transformer model, and then propose a source-informed heuristic method that can fully take advantage of the alignment information from the multi-head attention mechanism in Transformer to speed up the decoding in the GBS method and guide the placement of constraints during the expansion of hypotheses in GBS. Experiments on English–Chinese and English–German translation domain adaptation tasks show that the proposed method significantly outperforms the basic Transformer model in terms of BLEU and METEOR score, and prunes up to 30% hypotheses to save up to 20% decoding time compared to the GBS model while maintaining comparable translation performance.

1 Introduction

With recent advances in neural machine translation (NMT), The Transformer model (Vaswani et al., 2017) has outperformed other NMT architectures, like RNN (Luong et al., 2015) and CNN (Gehring et al., 2017), to become the state-of-the-art paradigm. The Transformer model mainly consists of layers of self-attention and a feed-forward network. It is capable of being fully parallelised and is faster both in training and inference. Its multi-layer and multi-head attention mechanism enable it to capture deep syntactic and semantic relations in sentences to produce better translation results.

Constrained decoding is an approach that exerts some constraints to a decoding process (often a beam search process) and enforces the constraints to appear in the decoding results. For translation tasks, constraints are normally some target words or phrases which are acquired in advance through domain knowledge or other methods. Constrained decoding ensures constraints partially or fully appear in the translation results by means of certain algorithms. The translation results of constrained decoding are often better than the normal decoding results, as those constraints are actually some external knowledge besides the source sentences.

*This work was done while the co-author were working with us in the ADAPT Centre at Dublin City University.

Different constrained decoding approaches have been proposed. Luong and Manning (2015), Sennrich et al. (2016a) adapted NMT systems with domain-specific data by adjusting their output vocabulary to better match the target domain. Besides, Wang et al. (2017) tried early attempts to improve translation consistency for NMT models with discourse-level context. However, these methods do not strictly enforce a constraint, so constraints are not guaranteed to appear in the output.

Anderson et al. (2017) extended beam search with a finite state acceptor (FSA) whose states mark the completed subsets of the set of constraints. However, their algorithm has an exponential complexity of $\mathcal{O}(Nk2^C)$, where n is the sentence length, k is beam size, and C is the constraint count. This results in a very slow decoding speed when the number of constraints increased.

Hokamp and Liu (2017) proposed a novel grid beam search (GBS) method that can enforce any constraints to appear in the translation results. In order to ensure the constraints are placed in the right positions in the translation results, GBS assumes that all constraints may appear at each decoding step and extends a beam vertically to grid beams. This means there are several beams at each step rather than a single beam. As a result, the number of hypotheses increases linearly according to the number of constraints. GBS can adapt a general NMT model to a domain translation task and improve the translation quality (Hokamp and Liu, 2017). Even though it has a complexity of $\mathcal{O}(NkC)$, it may expand a very large search space when the number of constrains increases, which also results in a slow and computationally expensive decoding process.

Post and Vilar (2018) proposed a fast lexically constrained decoding method with dynamic beam allocation (DBA) for NMT. This method groups together hypotheses that meet the same number of constraints into banks, and dynamically divides a fixed-size beam across these banks at each time step, which results in a complexity of $\mathcal{O}(Nk)$, so the DBA is faster and can process large constraint sets easily. The disadvantage of DBA is that the translation quality strongly depends on some factors, such as the beam size k. Their experiments show that system performance experiences a significant decrease compared to the original GBS system. DBA can be regarded as a better trade-off between translation quality and decoding time when applying constrained decoding to NMT.

There are two common problems in the above methods. First, all are based on the RNN model. Whether term constraints are necessary and whether those constraining algorithms are effective on the Transformer model are still open problems and deserve to be verified. Second, when decoding, only the constraints' own information is exploited to guide the placement of constraints, and no other information, such as source-side words, is used. An obvious deficiency of these algorithms is that all possible constraints of a sentence have to be considered at each decoding step. Furthermore, only the decoding score is used to rank and prune hypotheses in the beams. This mechanism might place a constraint in the wrong position and generate an output with a low score due to the enforced inclusion of all constraints in the output. Intuitively, a better strategy is to guide the constrained decoder to place constraints in the correct positions with the help of more useful information, so to avoid the extensive exploration of a very large search space and obtain better decoding results. As target-side constraints are actually deduced from the source sentences, if we can utilize some source-side information, we may be able to confine the search space or guide the decoding process to locate target-side constraints more accurately.

Confronting these two problems, this paper first investigates the feasibility and effectiveness of performing constrained decoding using GBS in Transformer. We then propose a source-informed heuristic method to reduce the search space of the GBS method so to speed decoding while maintaining comparable translation performance. We propose a simple but effective

lexically constrained decoding strategy for Transformer, which makes use of the alignment information from the multi-head attention in Transformer to place probably correct constraints at time step t. In doing so, we can control the number of paths to expand in the GBS, and speed up the decoding process.

The main contributions of this paper include: (1) to the best of our knowledge, our work is the first to implement constrained decoding in the Transformer model and verify its performance; (2) we propose an effective and efficient method for the constrained Transformer model which fully uses source-side information from the multi-head attention in Transformer to guide the placement of constraints at each time step for a better balance between translation quality and decoding time; (3) we compare the proposed method with unconstrained Transformer and GBS Transformer via extensive experiments, and demonstrate that our model significantly outperforms the basic Transformer model in terms of BLEU (Papineni et al., 2002) and METEOR score (Denkowski and Lavie, 2014), and significantly saves up to 20% decoding time compared to the GBS model with no deterioration in performance.

2 Transformer and Grid Beam Search

2.1 Neural Transformer Model

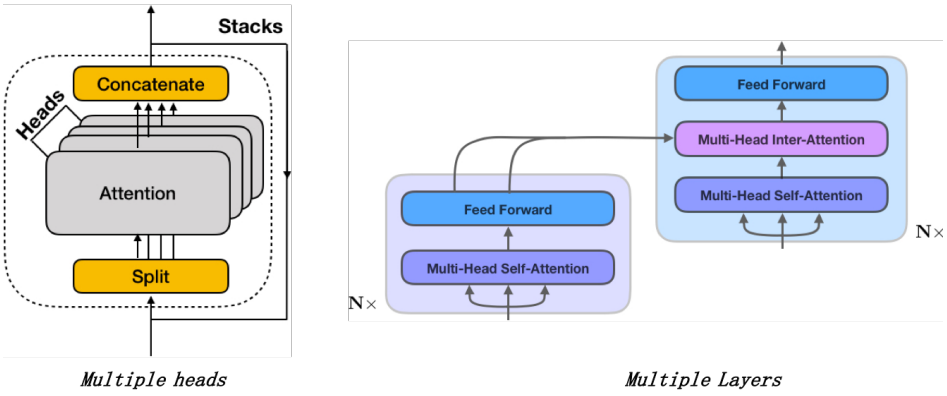

Figure 1: Transformer model

As shown in Figure 1, Transformer makes use of self-attention as the basic computational block. It uses a combination of self-attention and feed-forward layers in the encoder and additional source attention layers on the decoder side. In the standard Transformer model, the encoder is composed of a stack of $N_x = 6$ identical layers, with each layer having two sub-layers, namely a multi-head self-attention mechanism and a position-wise fully connected feed-forward network. A residual connection is employed around each of the two sub-layers, followed by layer normalisation. The decoder is also composed of a stack of $N_x = 6$ identical layers. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack. Similar to the encoder, residual connections are also employed around each of the sub-layers, followed by layer normalization. Transformer's self-attention includes attention between decoder layers and the encoder's outputs, which is similar to the attention in RNN and can be regarded as alignment between source and target. Please refer to Vaswani et al. (2017) for more details.

2.2 Grid Beam Search for Constrained NMT

In normal beam search (Lowerre, 1976; Sutskever et al., 2014), the decoder maintains a beam with a fixed size k containing a set of expanded hypotheses. As mentioned, the decoding in RNN-based and Transformer NMT models is an auto-regressive process, so at each time step t, the decoder produces a distribution over the target-language vocabulary for each of these hypotheses, depending on the previous time step $t-1$. As each beam contains k hypotheses, this produces a large matrix of dimension $k \times |V_T|$.

In order to integrate external knowledge into NMT without intervening in its learning process, constrained decoding can be adopted by NMT, which we call *constrained NMT*. By definition, constraints in constrained NMT indicate a set of pre-specified words, phrases or terms, which are acquired by automatic extraction from the corpus or via manually compilation. Constrained NMT can enforce constraints in the hypotheses and select from the set of complete hypotheses the best one that satisfies all constraints.

Hokamp and Liu (2017) formalise the notion of lexical constraints, and propose the *grid beam search* decoding algorithm which forces constraints to appear in the output. It organises the decoding process by expanding the beam of each step to grid beams which contain more than one beam. The beam count inside a grid beam corresponds to the token number of constraints. This method puts each constraint in all potential positions during decoding and it uses a kind of traversal method to find the best result.

To be specific, each beam in the grid is indexed by time step t and constraint variable c. c indicates how many constraint tokens have been covered so far by the current active hypothesis in the current beam. At each time step, only one single constraint token is covered, i.e. the set of constraints is an array of sequences, where each token can be indexed as $constraints_{ij}$, indicating $token_j$ in the $constraint_i$. $numC$ is used to represent the total number of tokens in all constraints C.

The hypotheses in a beam can be separated into two types:

(1) **open** hypotheses: the next token can be generated either from the model, or from the available constraints;

(2) **closed** hypotheses: the next token can only be generated from a currently unfinished constraint.

At each step t of the search process, the beam at **Grid**$[t][c]$ is filled with candidates which may be created in three ways:

(1) the **open** hypotheses in the beam to the left (**Grid**$[t-1][c]$) may ***generate*** continuations from the model's distribution $p_\theta(y_i|x, y_0...y_{i-1})$;

(2) the **open** hypotheses in the beam to the left and below (**Grid**$[t-1][c-1]$) may ***start*** new constraints;

(3) the **closed** hypotheses in the beam to the left and below (**Grid**$[t-1][c-1]$) may ***continue*** constraints.

The beams at the top level of the grid (beams where $c = numC$) contain hypotheses which cover all constraints. Once a hypothesis at the top level generates the $\langle$EOS$\rangle$ token, it can be added to the set of finished hypotheses (cf. Hokamp and Liu (2017) for more detail).

3 Multi-Head Attention-Guided Source Information as Heuristics for Constrained Transformer

Our motivation to use source-side information as heuristics is that in both GBS and DBA, only the constraints' own information (often some target-side words) is used. The methods do not

decide which constraint should be placed at time step t, so all available constraints need to be considered during the decoding process. The fact is that constraints extracted from the corpus or via terminology entries are bilingual, while source-side information is simply discarded in current constrained decoding methods.

Following the work of GBS in RNN-based NMT (Hokamp and Liu, 2017), we re-implement it in the Transformer model, and then take advantage of the alignment information from the multi-head attention mechanism to obtain corresponding source-side positions at time step t, and then guide the decoder to place corresponding constraints in the beam, which we call "**Heuristic GBS** (**HGBS**)".

3.1 Algorithm

Algorithm 1 Pseudo-code for Heuristic Grid Beam Search

1: **procedure** HEURISTIC SEARCH($model, input, constraints, maxLen, numC, k$)
2: **if** $hyp.isOpen()$ **then** ▷ Start from Line 15 of GBS Algorithm
3: **for** c in $constraints$ **do**
4: **if** c not $used$ **then**
5: $p_A =$ multiHeadSearch(t, c)
6: **if** $p_A >= p_{th}$ **then**
7: $n \leftarrow n \cup$ model.start$(hyp, input, c)$ ▷ Only the constraint c is placed in the beam

When we use Pointwise Mutual Information (PMI) method (Hokamp and Liu, 2017) to extract a constraint, we actually obtain a segment pairs which contains both the source segment and target segment, which we call a "constraint pair". A Chinese–English constraint pair with source positions is shown in Table 1. There is a source sentence "传统劳动密集型产品因价格下降带来的出口值减少" whose reference target sentence is "The export value of traditional labor-intensive products decreased due to the price drop". PMI method can extract a constraint pair "劳动 密集型 产品 ||| labor intensive products" where "labor intensive products" is supposed to appear in the translation. This constraint pair is underlined in both source and target sentence in the table. The table also shows that the position of the source part of the constraint in the source sentence is 1, 2 and 3.

At time step t in decoding, before we want to take this target constraint as a candidate to start new constraints in the beam, we first retrieve the source word positions $\{1, 2, 3\}$. With this position information, we can look at the multi-head attention, and obtain the weights at time step t pointing to these three source word positions. By using these weights, we can decide whether we should start a new hypothesis of this constraint in the beam or not.

Source sentence	传统 劳动 密集型 产品 因 价格 下降 带来 的 出口 值 减少
Target sentence	The export value of traditional labor intensive products decreased due to the price drop
Target side of constraint	labor intensive products
Source side of constraint	劳动 密集型 产品
Position in source	{1, 2, 3}

Table 1: An example of a Chinese–English constraint pair

Obviously, there is a risk that if the alignment is incorrect, then we might put the constraint in the wrong place. However when hypotheses in the beam compete with each other via the

model score, the possibility of generating a hypothesis with wrongly placed constraints in the output will greatly reduce.

Algorithm 1 shows the core parts of our proposed HGBS method, which is modified based on *Line 15 and 16* of the GBS Algorithm (Hokamp and Liu, 2017) by adding alignment information to guide the grid search.

3.2 Multi-Head Attention for Source-Informed Constraints

An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key (Vaswani et al., 2017).

From Figure 1 we can see that there are multiple layers for the encoder and decoder, respectively. The multi-head self-attention from the last layer of the encoder is fed to each layer of the decoder to construct the soft attention alignment between the target and source positions. We found that the alignment in the last layer of the decoder works best, so we only use the alignment information from the last multi-head attention layer of the decoder to guide our constrained decoding approach.

By applying the multi-head attentions to the HGBS algorithm, we follow the steps below:

- **S1**: at each decoding step, we retrieve the attention weight distribution at the current time step t;
- **S2**: multiple weight distributions from the multi-head attention are averaged to obtain one single attention distribution;
- **S3**: taking a target constraint c as a candidate to **start** new constraints in the beam, we derive the source-word positions of its corresponding source constraint;
- **S4**: we sum up the probabilities of all links to the source word positions in **S3**. The sum is denoted as p_A;
- **S5**: if $p_A > p_{th}$, where p_{th} is a pre-defined threshold, then put c in the beam to **start** a new constraint.
- **S6**: loop from **S3** to **S6** until all available target constraints are traversed.

From Algorithm 1 and the above steps, the source-side information derived from the multi-head attention mechanism acts as a filter to remove those constraints that are not necessary to expand the hypothesis at the current time step t. The source-side alignment information helps place constraints in more reasonable positions. In this way, a number of hypotheses in GBS are pruned, and the search space decreases significantly.

4 Experiments

4.1 Translation Tasks

In our view, the most interesting finding in Hokamp and Liu (2017) is that GBS-based constrained decoding has a significant role to play in domain adaptation via terminology, which is a very important issue in application scenarios in which the translation process has to comply with specific terminology and/or style guides (Chatterjee et al., 2017).

Therefore, in order to compare the proposed **HGBS** method with **GBS**, we focus in our experiments on the domain adaptation task for constrained decoding via terminology. We use WMT English–German (EN-DE) and Chinese–English (ZH-EN) translation tasks to perform the comparison experiments.

4.2 Data

We use the same data settings for the domain adaptation experiment as in Hokamp and Liu (2017) in terms of the EN-DE task:

- the training corpus consists of 4.4 Million segments from Europarl (Koehn, 2005) and CommonCrawl (Smith et al., 2013);
- for the target domain data, the Autodesk Post-Editing corpus (Zhechev, 2012) from the domain of software localisation is used, which is quite different from the WMT data. The corpus is divided into 100,000 training sentences and 1,000 test sentences. Constraints are extracted automatically using PMI between source and target n-grams. The maximum length of a constraint or terminology is set to 5-gram as in Hokamp and Liu (2017).

For the ZH-EN translation task, in terms of the training data and testing data,

- we use LDC corpora to train the general domain Transformer, which consists of 1.25 Million segments;* Most sentences in this corpus come from the News domain.
- for the target domain data, we also use the Autodesk Post-Editing corpus. 159,816 sentences are extracted as the training set for PMI and constraint extraction. An additional 1,000 sentences are extracted as the test set for our constrained Transformer experiment. The maximum length for PMI constraint extraction is set to 5-grams.

All English and German sentences are preprocessed using tools from Moses (Koehn et al., 2007). Chinese sentences are segmented into words using *Jieba*,† a popular Python toolkit for Chinese word segmentation. Finally, the parallel pre-processed data are segmented to subwords by applying *Byte Pair Encoding* (Sennrich et al., 2016b), which is capable of encoding open vocabularies with a compact symbol vocabulary of variable-length subword units.

4.3 Systems

We use the Transformer model in the open source toolkit **THUMT** as our baseline system (Zhang et al., 2017).‡ For the constrained Transformer model, we first reimplement the GBS method under Transformer model, which we call **GBS-T**, and we then apply our HGBS algorithm to **GBS-T** to improve its decoding speed, which we call **HGBS-T**. The evaluation metrics are case-insensitive BLEU and METEOR.

In all our experiments, we employ the base Transformer configuration with embedding size and hidden size both 512, 6 encoder and decoder layers, 8 attention heads, the standard ReLu activation function and sinusoidal positional embedding, maximum sentence length 80, batch size 4096 tokens, beam size 10. The vocabulary sizes in EN-DE are 80,711 for English and 88,990 for German. The vocabulary sizes in ZH-EN are 30,568 for Chinese and 24,585 for English. The maximum number of constraints in a sentence is 6, and the alignment threshold p_{th} is set to 0.1.§

4.4 Results

Table 2 shows the results of three Transformer systems in terms BLEU and METEOR score on two translation tasks. From Table 2 we can see that:

*The segments are extracted from LDC2003E07, LDC2003E14, LDC2004T07, LDC2005E83, LDC2005T06, LDC2006E24, LDC2006E34, LDC2006E85, LDC2006E92, LDC2007E87, LDC2007E101, LDC2007T09, LDC2008E40, LDC2008E56, LDC2009E16 and LDC2009E95.

†`https://github.com/fxsjy/jieba`

‡`https://github.com/thumt/THUMT`

§Our implementation of GBS-T and HGBS-T is available at `https://github.com/gdxie1/THUMT-GBS.git`

System	EN-DE		ZH-EN	
	BLEU	METEOR	BLEU	METEOR
Baseline	34.82	0.29	5.94	0.12
GBS-T	37.92*	0.33*	11.38*	0.23*
HGBS-T	37.13*	0.33*	11.43*	0.22*

Table 2: Comparison of three Transformer systems. $*$ indicates a significantly better result compared to the Baseline.

- GBS-T significantly outperforms the baseline on the EN-DE task by absolute 3.10 (8.9%) points and 0.04 (13.8%) points in terms of BLEU and METEOR score, respectively, and on the ZH-EN task by absolute 5.44 points and 0.11 points in terms of BLEU and METEOR score. The low Baseline score on ZH-EN confirms that the domain of the Autodesk data is significantly different from that of the LDC data. This big improvement also shows that using constrained decoding for domain adaptation via constraints is a feasible solution for the scenario of low-resource domain translation.

- HGBS underperforms GBS-T by absolute 0.79 points on EN-DE in terms of BLEU score. However, we can see that it has the same METEOR score as GBS-T. HGBS-T significantly outperforms the Baseline on EN-DE in terms of BLEU and METEOR as well, and it has a comparable performance with GBS-T.

- For ZH-EN, HGBS-T is slightly better than GBS-T in terms of BLEU, and has almost the same result as GBS-T in terms of METEOR.

From the above observations, we can conclude that (1) our HGBS-T model has a comparable translation performance to GBS-T in terms of BLEU and METEOR score; (2) our constrained Transformer model is effective for domain adaptation, especially when the domains of the training data and testing data are significantly different.

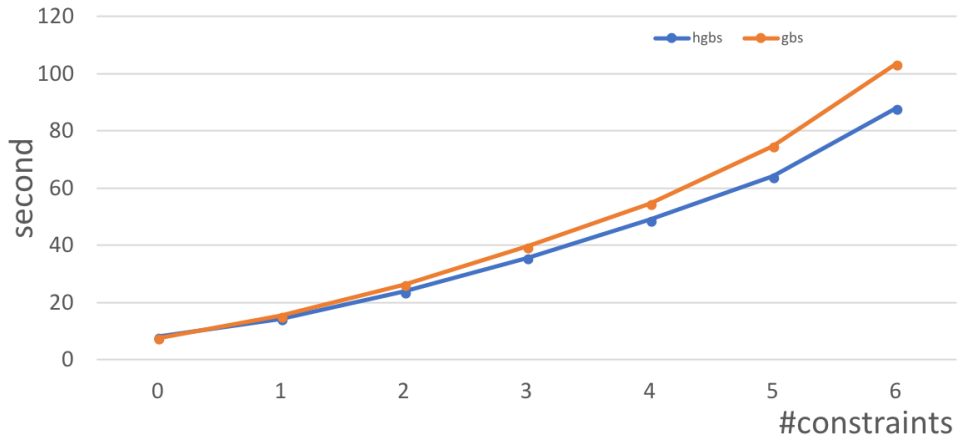

Figure 2: Comparison of decoding time consumption (seconds/constraints)

Figure 2 compares speed as a function of the number of constraints for the EN-DE task. We divide the sentences into different groups where each sentence in the same group contains the same number of constraints, and then we average decoding time over all sentences in the

same group. The numbers on the vertical axis represent the average decoding time, i.e. seconds per sentence.

We can see that by using source information to prune hypotheses in the GBS, our HGBS has significantly decreased decoding time, especially when a sentence contains more than 3 constraints. In our experiments, we observed that the averaged hypotheses of each sentence in GBS-T is 4,887, while our HGBS-T has 3560, so about 30% paths are removed, as shown in Figure 3, where we can see that there is a significant decrease in the number of hypotheses in the beam that need to be expanded. As a result, in this figure, when constraints are up to 6, the average saving in decoding time can up to 20% compared to GBS.

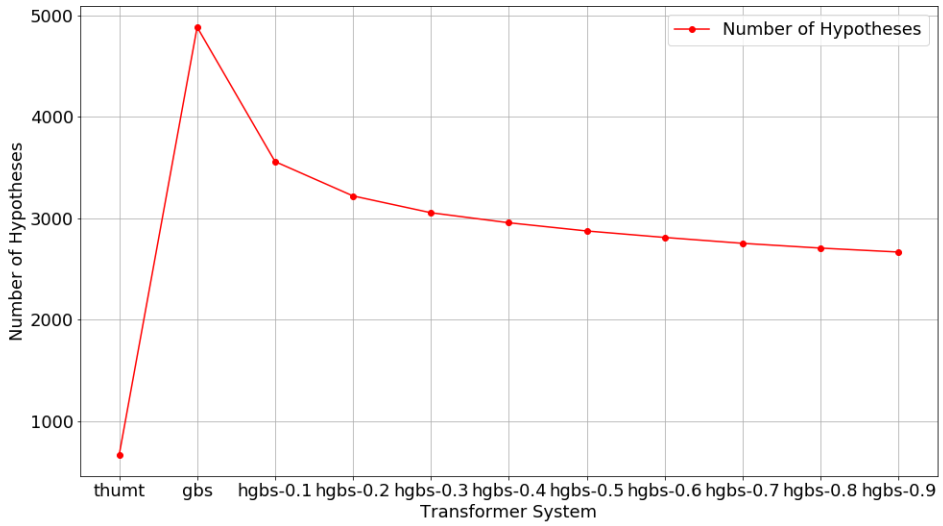

Figure 3: Comparison of number of hypotheses expanded in the decoding

4.5 Analysis

From the above experimental results, we can see that it is feasible to use the multi-head attention-guided source information for GBS in Transformer to save decoding time while maintaining comparable translation performance. To obtain this result, we carried out some experiments to look into the multi-head attention mechanism and layers of Transformer to optimise and determine some key hyper-parameters, such as the threshold p_{th}, averaging weights of the multi-head attention of the last layer to provide alignment information. In this section, we describe these experiments and provide an analysis of the results obtained.

4.5.1 Effects of Attention on Different Thresholds

Table 3 shows how the performance of HGBS changes with different settings for the alignment threshold p_{th} on the EN-DE task. The BLEU scores for HGBS-T are based on applying different thresholds p_{th} on the same test set in our experimental setting.

From Table 3, we can see that:

- our **HGBS-T** model achieves similar performance to **GBS-T** when p_{th} is set to small values. It can be seen that with the increase in threshold, translation quality decreases

Baseline	34.82								
GBS-T	**37.92**								
HGBS-T	**37.13**	36.40	36.16	35.64	35.11	34.65	34.92	35.11	35.45
Threshold p_{th}	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9

Table 3: Performance changes with different thresholds at Layer 5

significantly. When $p_{th} = 0.6$, **HGBS-T** drops to almost the same performance as the baseline.

- based on the above observations, we set $p_{th} = 0.1$ in **HGBS-T** for all our experiments.

4.5.2 How Word Alignment Quality Affects BLEU Score

In our HGBS method, the placement of a constraint is guided by the multi-head attention information. Therefore, we infer that the quality of word alignment between the target and source is closely correlated with translation quality, i.e. a better quality word alignment will produce a higher quality of translations. In this section, we look into this issue by evaluating the word alignment of multi-head attention mechanism and measuring their correlations.

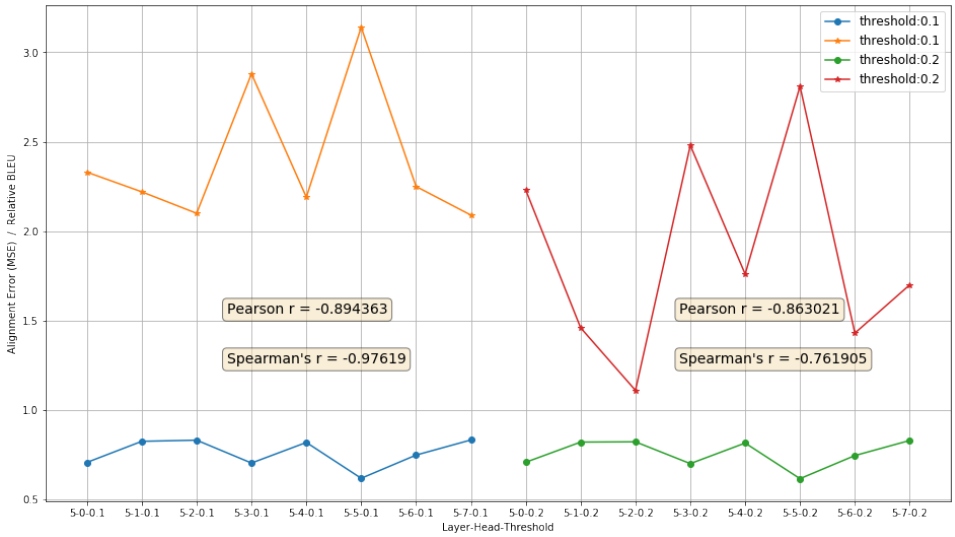

Figure 4: Correlations of the alignment error and BLEU score

To evaluate the quality of the word alignment from the multi-head attention, we use the word alignment links generated from FastAlign (Dyer et al., 2013) as the "Ground Truth". Since the word alignment from the multi-head attention is a probability distribution of the time step t in the decoder against all source words, we use Mean Square Error (MSE) as the metric to evaluate the alignment quality as in Equation (1):

$$E_{mse}(A, \alpha) = \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{I_t} (A_{ti} - \alpha_{ti})^2 \quad (1)$$

where A is the alignment from FastAlign, α is the alignment from the multi-head attention. T is the total time steps for the target sequence, and I_t is the number of alignment links of time step t against the source words. The alignment model of FastAlign is trained with the same EN-DE training corpus as in Section 4.2. Table 4 shows the relationship between the alignment error E_{mse} and the BLEU score with different thresholds for heads at Layer 5. "Relative BLEU" indicates that we scale the values of the column "BLEU" by subtracting an offset of 34 so that we can plot all curves in one figure in order to compare their trends. In Figure 4, the top-left curve shows the changes of relative BLEU against the changes of alignment error (bottom-left) at each head of Layer 5 when the threshold is set to 0.1. The top-right curve shows the changes of relative BLEU against the changes of alignment error (bottom-right) at each head of Layer 5 when the threshold is set to 0.2.

Layer	Head	Threshold	Alignment Error E_{mse}	BLEU	Relative BLEU
5	0	0.1	0.707	36.33	2.33
5	1	0.1	0.825	36.22	2.22
5	2	0.1	0.832	36.10	2.10
5	3	0.1	0.704	36.88	2.88
5	4	0.1	0.819	36.19	2.19
5	5	0.1	0.618	37.14	3.14
5	6	0.1	0.748	36.25	2.25
5	7	0.1	0.833	36.09	2.09
5	0	0.2	0.708	36.23	2.23
5	1	0.2	0.821	35.46	1.46
5	2	0.2	0.822	35.11	1.11
5	3	0.2	0.700	36.48	2.48
5	4	0.2	0.815	35.76	1.76
5	5	0.2	0.616	36.81	2.81
5	6	0.2	0.745	35.43	1.43
5	7	0.2	0.830	35.70	1.70

Table 4: Alignment Error and BLEU score of different heads at Layer 5 of Transformer model

We can see that for "threshold:0.1" and "threshold:0.2", the *Pearson* coefficients are -0.89 and -0.86, respectively, and the *Spearman's* coefficients are -0.98 and -0.76, respectively, which show that the BLEU score of the translations has high negative linear and monotonic correlations with the alignment errors, i.e. if the quality of word alignment is better, the translation quality is better. From this observation and analysis, regarding the proposed HGBS method, the hypothesis will be that if we can improve the quality of word alignment of multi-head attention mechanism, we would further improve translation quality and better guide the placement of constraints during the decoding to further improve translation quality.

5 Refining the model with alignmental guiding training

In order to verify the effect of alignment (or attention) on our HGBS method, we refined the Transformer model using Guided Alignment Training (Chen et al., 2016). Currently the general Transformer model normally uses 6 layers and 8 heads in each layer. We average all the attention of the 6 layers and 8 heads as a whole attention value, as the A_{ti} in Equation (1). Similar to Chen et al. (2016), we combine decoder cost and alignment cost to build the new loss function

$H(y, x, A, \alpha)$ in Equation (2):

$$H(y, x, A, \alpha) = H_D(y, x) + \omega E_{mse}(A, \alpha) \quad (2)$$

Here $H_D(y, x)$ is the normal decoder cost of the Transformer model, and ω is the weights for E_{mse}. In our experiments, we set ω as 0.05 and obtain the best performance. Our refining process is as follows: first we train a normal Transformer NMT model. Then we use $H(y, x, A, \alpha)$ as our model loss function to continue to train the model beginning from the best checkpoint. During training, all the parameters will be updated so the model will gradually output better multi-head alignment attentions. After about 200,000 more iterations, we obtain a new model refined from the alignment information. We perform this experiments in the previous section again and obtain the results shown in Table 5. We insert the previous result in the the table for convenience. From that table, we can see that the refined baseline system's performance is even

System	BLEU of EN-DE	
	not refined	refined
Baseline	34.82	34.85
GBS-T	37.92	38.16*
HGBS-T	37.13	38.12*

Table 5: Comparison of three Transformer systems after refining training. $*$ indicates a better result compared to the unrefined system.

better than the original baseline system. When we apply the GBS method and HGBS method on the refined baseline model, both produce higher BLEU scores. The GBS method obtains 0.24 increment and HGBS obtains 0.99 increment. Comparing with the refined baseline, the GBS obtains 3.31 increment and HGBS almost achieves the same performance as GBS. However on the unrefined system, HGBS obtains a lower score of 37.13 than the GBS's 37.92.

6 Conclusions and Future Work

In this paper, we first reimplement and investigate the grid beam search (GBS) method based on the Transformer model, and then propose heuristic GBS – a source-informed heuristic method guided by the multi-head attention mechanism – to speed up decoding while maintaining comparable translation performance. We compare our proposed method with unconstrained Transformer and GBS Transformer via a range of experiments on domain adaptation translation tasks, and demonstrate that our model significantly outperforms the basic Transformer model in terms of BLEU and METEOR, and at the same time significantly prunes up to 30% hypotheses and saves up to 20% decoding time with comparable results. Experimental results also show that our method is more practical for application to the scenario of low-resource domain adaptation translation compared with GBS.

In future work, we will further optimise the proposed HGBS method in terms of translation quality, decoding time and reducing the complexity by better exploiting the multi-head alignment information.

Acknowledgments

We wish to thank our erstwhile colleagues Qun Liu and Liangyou Li of Huawei Noah's Ark Lab for discussions which impacted the design of the algorithm presented in Section 3.1. We would also like to thank the three anonymous reviewers for their useful comments. This work is supported by the ADAPT Centre for Digital Content Technology which is funded under the

Science Foundation Ireland (SFI) Research Centres Programme (Grant No. 13/RC/2106) and is co-funded under the European Regional Development Fund.

References

Anderson, P., Fernando, B., Johnson, M., and Gould, S. (2017). Guided open vocabulary image captioning with constrained beam search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 936–945, Copenhagen, Denmark.

Chatterjee, R., Negri, M., Turchi, M., Federico, M., Specia, L., and Blain, F. (2017). Guiding neural machine translation decoding with external knowledge. In *Proceedings of the Second Conference on Machine Translation*, pages 157–168, Copenhagen, Denmark.

Chen, W., Matusov, E., Khadivi, S., and Peter, J.-T. (2016). Guided alignment training for topic-aware neural machine translation. *arXiv preprint: 1607.01628*.

Denkowski, M. and Lavie, A. (2014). Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*, page 376–380, Baltimore, Maryland, USA.

Dyer, C., Chahuneau, V., and Smith, N. A. (2013). A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia.

Gehring, J., Auli, M., Grangier, D., and Dauphin, Y. (2017). A convolutional encoder model for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 123–135, Vancouver, Canada.

Hokamp, C. and Liu, Q. (2017). Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada.

Koehn, P. (2005). Europarl: a parallel corpus for statistical machine translation. In *MT Summit X, Conference Proceedings: the Tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Prague, Czech Republic.

Lowerre, B. T. (1976). *The Harpy Speech Recognition System.* PhD thesis, Carnegie-Mellon University, Pittsburgh PA.

Luong, M.-T. and Manning, C. D. (2015). Stanford neural machine translation systems for spoken language domains. In *Proceedings of the 12th International Workshop on Spoken Language Translation)*, pages 76–79, Da Nang, Vietnam.

Luong, M.-T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA.

Post, M. and Vilar, D. (2018). Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1314–1324, New Orleans, Louisiana.

Sennrich, R., Haddow, B., and Birch, A. (2016a). Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany.

Sennrich, R., Haddow, B., and Birch, A. (2016b). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany.

Smith, J. R., Saint-Amand, H., Plamada, M., Koehn, P., Callison-Burch, C., and Lopez, A. (2013). Dirt cheap web-scale parallel text from the common crawl. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1374–1383, Sofia, Bulgaria.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*, pages 3104–3112, Montreal, Canada.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008, Long Beach, CA, USA.

Wang, L., Tu, Z., Way, A., and Liu, Q. (2017). Exploiting cross-sentence context for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2826–2831, Copenhagen, Denmark.

Zhang, J., Ding, Y., Shen, S., Cheng, Y., Sun, M., Luan, H., and Liu, Y. (2017). Thumt: An open source toolkit for neural machine translation. *arXiv preprint: 1706.06415*.

Zhechev, V. (2012). Machine translation infrastructure and post-editing performance at Autodesk. In *AMTA 2012 Workshop on Post-Editing Technology and Practice (WPTP 2012)*, pages 87–96, San Diego, USA.