Passing Parser Uncertainty to the Transformer: Labeled Dependency Distributions for Neural Machine Translation

Dongqi Liu **Khalil Sima'an**
dongqi.me@gmail.com k.simaan@uva.nl
Institute for Logic, Language and Computation
University of Amsterdam

Abstract

Existing syntax-enriched neural machine translation (NMT) models work either with the single most-likely unlabeled parse or the set of n-best unlabeled parses coming out of an external parser. Passing a single or n-best parses to the NMT model risks propagating parse errors. Furthermore, unlabeled parses represent only syntactic groupings without their linguistically relevant categories. In this paper we explore the question: Does passing both parser uncertainty and labeled syntactic knowledge to the Transformer improve its translation performance? This paper contributes a novel method for infusing the whole labeled dependency distributions (LDD) of the source sentence's dependency forest into the self-attention mechanism of the encoder of the Transformer. A range of experimental results on three language pairs demonstrate that the proposed approach outperforms both the vanilla Transformer as well as the single best-parse Transformer model across several evaluation metrics.

1 Introduction

Neural Machine Translation (NMT) models based on the seq2seq schema, e.g., Kalchbrenner and Blunsom (2013); Cho et al. (2014); Sutskever et al. (2014); Bahdanau et al. (2014), first encode the source sentence into a high-dimensional content vector before decoding it into the target sentence. Several prior studies (Shi et al., 2016; Belinkov and Bisk, 2018) have pointed out that although NMT models may induce aspects of syntactic relations, they still cannot capture the subtleties of syntactic structure that should be useful for accurate translation, particularly by bridging long distance relations.

Previous work provides support for the hypothesis that explicit incorporation of source syntactic knowledge could result in better translation performance, e.g., Eriguchi et al. (2016); Bastings et al. (2017). Most models condition translation on a single best parse **syn**:

$$\arg\max_{\mathbf{t}} P(\mathbf{t}|\mathbf{s}, \mathbf{syn}) \quad (1)$$

where **s** and **t** are the source and target sentences respectively. Other models incorporate the n-best parses or forest (without parser probabilities and labels), e.g., Neubig and Duh (2014). The idea here is that the syntactically richer input (**s**, **syn**) should be better than the bare sequential word order of **s**, leading to a more accurate and sharper translation distribution $P(\mathbf{t}|\mathbf{s}, \mathbf{syn})$.

While most syntax-enriched strategies result in performance improvements, there are two noteworthy gaps in the literature addressing source syntax. Firstly, none of the existing works conditions on the probability distributions over source syntactic relations. And secondly, none of the existing approaches conditions on the dependency labels, thereby conditioning only on the binary choice whether there is an unlabeled dependency relation between two words.

Tu et al. (2010); Ma et al. (2018); Zaremoodi and Haffari (2018) showed that the whole dependency forest provides better performance than a single best parse approach. In this paper we go

one step further and propose that *a syntactic parser is more useful if it conveys to the NMT model also its remaining uncertainty, expressed as the whole probability distributions over dependency relations rather than a mere forest.*

To the best of our knowledge, there is no published work that incorporates a parser's distributions over dependency relations into the Transformer model (Vaswani et al., 2017), let alone incorporating distributions over labeled dependency relations into NMT models at large.

This paper contributes a generic approach for infusing labeled dependency distributions into the encoder's self-attention layer of the Transformer. We represent a labeled dependency distributions as a three-dimensional tensor of parser probabilities, where the first and second dimensions concern word-positions and the third concerns the dependency labels.

The resulting tensor is infused into the computation of the multi-head self-attention, where every head is made to specialize in a specific dependency class. We contribute empirical evidence that passing uncertainty to the Transformer and passing labeled dependencies both give better performance than passing a single unlabeled parse, or an unlabeled/labeled set of dependency relations with uniform probabilities.

2 Related Work

The role of source syntactic knowledge in better reordering was appreciated early on during the Statistical Machine Translation (SMT) era. For example, Mylonakis and Sima'an (2011) propose that source language parses should play a crucial role in guiding the reordering within translation, and do so by integrating constituency labels of varying granularity into the source language. Although, NMT encoders have been claimed to have the ability to learn syntax, work on RNNs-based models shows the value of external source syntax in improving translation performance, e.g., Eriguchi et al. (2016), by refining the encoder component, leading to a combination of a tree-based encoder and a sequential encoder.

Noteworthy to recall here that the attention mechanism was originally aimed to capture all word-to-word relations, including syntactic-semantic relations. whereas, the work of Bastings et al. (2017) has shown that a single unlabeled dependency parse, encoded utilizing Graph Convolutional Networks (GCNs), can help improve MT performance. Ma et al. (2018) and Zaremoodi and Haffari (2018) attempt to incorporate parse forests into RNNs-based NMT models, mitigating parsing errors by providing more candidate options. However, these two works only rely on the binary (unlabeled) relations in all the sub-trees, ignoring the elaborate probability relations between word positions and the type of these relations.

Although the Transformer (Vaswani et al., 2017) is considered to have a better ability to implicitly learn relations between words than the RNNs-based models, existing work (Zhang et al., 2019; Currey and Heafield, 2019) shows that even incorporating a single best parse could improve the Transformer translation performance. Followup work (Bugliarello and Okazaki, 2020; Peng et al., 2021) provides similar evidence by changing the Transformer's self-attention mechanism based on the distance between the input words of dependency relations, exploiting the single best unlabeled dependency parse.

The work of Pham et al. (2019) suggests that the benefits of incorporating a single (possibly noisy) parse (using data manipulation, linearized or embedding-based method) can be explained as a mere regularization effect of the model, which does not help the Transformer to exploit the actual syntactic knowledge. Interestingly, Pham et al. (2019) arrive at a similar hypothesis, but they concentrate on exploring how to train one of the heads of the self-attention in the Transformer for a combined objective of parsing and translation. The parsing-translation training objective focuses the self-attention of a single head at learning the distribution of unlabeled dependencies while learning to translate as well, i.e., the distribution is not taken as source input but as a gold training objective. By training a single head with syntax, they leave all other heads without direct access to syntax.

Our work confirms the intuition of Pham et al. (2019) regarding the utility of the parser's full dependency distributions, but in our model these distributions are infused directly into the self-attention while maintaining a single training objective (translation). Furthermore, we propose that only when the full probability distribution matrices over labeled dependency relations is infused directly into the transformer's self-attention mechanism (not as training objective), syntax has a chance to teach the Transformer to better learn

syntax-informed self-attention weights.

3 Proposed Approach

A parser can be seen as an external expert system that provides linguistic knowledge to assist the NMT models in explicitly taking into account syntactic structure. For some sentences, the parser could be rather uncertain and spread its probability over multiple parses almost uniformly, but in the majority of cases the parser could have a rather sharp distribution over the alternative parses. Therefore, simply passing a dependency forest amounts merely to passing all alternative parses accompanied with zero information on parser confidence (maximum perplexity) to the Transformer NMT model, which does not help it to distinguish between the parsing information of the one input from that of another. This could increase the complexity of learning the NMT model unnecessarily.

An alternative is then to use for each sentence a dependency distribution in the form of conditional probabilities, which could be taken to represent the degree of confidence of the parser in the individual dependency relations. Furthermore, we propose that each dependency relation type (label), provides a more granular local probability distribution that could assist the Transformer model in making more accurate estimation of the context vector. This might enhance the quality of encoding the source sentence, particularly because the Transformer model relies on a weak notion or word order, which is input in the form of positional encoding outside the self-attention mechanism.

Note that the word-to-word dependency probabilities is not equivalent to using a distribution over dependency parses. This is because in some cases the word-to-word dependencies (just like word-to-word attention) could combine together into general graphs (not necessarily trees). We think that using relations between pairs of words (rather than upholding strict tree or forest structures) fits well with the self-attention mechanism.

3.1 Dependency Distributions

Denote with $|T|$ target sentence length and with $\mathrm{encode}(\cdot)$ the NMT model's encoder. We contrast different syntax-driven models:

$$P(\mathbf{t}|\mathbf{s}, \mathbf{syn}) \approx \prod_{i=1}^{|T|} P(t_i|t_{<i}, \mathrm{encode}(\mathbf{s}, \mathbf{syn})) \quad (2)$$

with **syn** $\in \{$**{L,U}DD**, **U{L,U}DD**, **{L,U}DP**$\}$, where **{L,U}DD** is the labeled/unlabeled dependency distribution[1], **U{L,U}DD** the uniform labeled/unlabeled dependency distribution[2], and **{L,U}DP** the 1-best labeled/unlabeled dependency parse. We also use **LDA** to stand for a model were the attention weights are fixed equal to **LDD** (i.e., not learned).

Our primary idea is to exert a soft influence on the self-attention in the encoder of the Transformer to allow it to fit its parameters with both syntax and translation awareness together. For infusing the labeled dependency distributions, we start with "matrixization" of labeled dependency distributions, which results in a compact tensor representation suitable for NMT models.

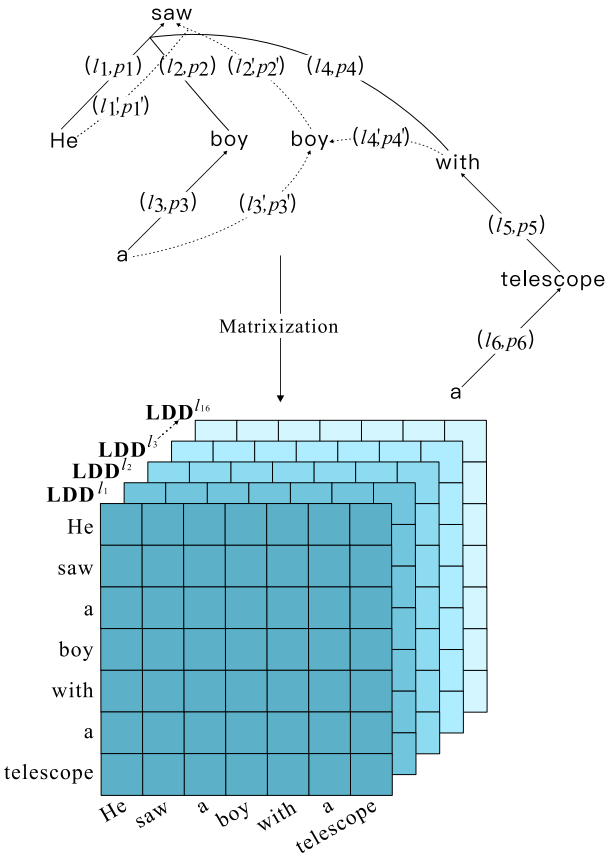

Figure 1: Labeled dependency distributions

Figure 1 illustrates by example how we convert the labeled dependency distribution (**LDD**) into a three-dimensional **LDD** tensor. The x-axis and y-

[1]Unlabeled dependency distribution is the sum of labeled dependency distributions on the z-axis, which is the same as 1-best unlabeled dependency parse.

[2]It is used for the purpose of ablation experiments, that is, the value of each point in the 3-dimensional tensor is identical.

axis of the tensor are the words in the source sentence, and the z-axis represents the type of dependency relation. Each point representing a conditional probability $p(i, j, l) = p(s_j, l|s_i) \in [0, 1] \subseteq \mathbb{R}$ of source word s_i modifying another source word s_j with relation l.

LDD Matrix for a specific label l: The matrix $\mathbf{LDD}^l$ extracted from the **LDD** tensor for a dependency label l is defined as the matrix in which every entry (i, j) contains the probability of a word s_i to modify word s_j with dependency relation l.

3.2 Parser-Infused Self-attention

Inspired by Bugliarello and Okazaki (2020), we propose a novel Transformer NMT model that incorporates the **LDD** into the first layer of the encoder side. Figure 2 shows our LDD sub-layer.

The standard self-attention layer employs a multi-head attention mechanism of h heads. For an input sentence of length T, the input of self-attention head h_i in the LDD layer is the word embedding matrix $\mathbf{X} \in \mathbb{R}^{T \times d_{\text{model}}}$ and the dependency distribution matrix $\mathbf{LDD}^{l_i} \in \mathbb{R}^{T \times T}$ for label l_i assigned to head h_i uniquely[3]. Hence, when we refer to head h_i, we refer also to its uniquely assigned dependency label l_i, but we omit l_i to avoid complicating the notation.

As usual in multi-head self-attention (h being the number of heads) for head h_i, first it linearly maps three input vectors, $\mathbf{q}, \mathbf{k}, \mathbf{v} \in \mathbb{R}^{1 \times d_{\text{model}}}$ for each token, resulting in three matrices $\mathbf{Q}^{h_i} \in \mathbb{R}^{T \times d}$, $\mathbf{K}^{h_i} \in \mathbb{R}^{T \times d}$, and $\mathbf{V}^{h_i} \in \mathbb{R}^{T \times d}$, where d_{model} is the dimension of input vectors, and $d = d_{\text{model}}/h$. Subsequently, an attention weight for each position is obtained by:

$$\mathbf{S}^{h_i} = \frac{\mathbf{Q}^{h_i} \cdot {\mathbf{K}^{h_i}}^\top}{\sqrt{d}} \tag{3}$$

At this point we infuse the resulting self-attention weight matrix $\mathbf{S}^{h_i}$ for head h_i with the specific **LDD** matrix $\mathbf{LDD}^{l_i}$ for label l_i using element-wise multiplication. Assuming that $d^{l_i}_{p,q} \in \mathbf{LDD}^{l_i}$, this is to say:

$$n^{h_i}_{p,q} = s^{h_i}_{p,q} \times d^{l_i}_{p,q}, \text{ for } p, q = 1, ..., T \tag{4}$$

The purpose of element-wise multiplication is to nudge the attention mechanism to "dynamically" learn weights that optimize the translation objective but also diverge the least from the parser probabilities in the dependency distribution matrix.

Next, the resulting weights are softmaxed to obtain the final syntax-infused distribution matrix for head h_i and the label attached to this head l_i:

$$\mathbf{N}^{h_i} = \text{softmax}(\mathbf{S}^{h_i} \odot \mathbf{LDD}^{l_i}) \tag{5}$$

We stress that every attention head is infused with a different dependency relation matrix $\mathbf{LDD}^{l_i}$ for a particular dependency relation l_i. By focusing every head on a different label we hope to "soft label", or specialize, it for that label.

Now that we have syntax-infused weights $\mathbf{N}^{h_i}$ we multiply them with the value matrix $\mathbf{V}^{h_i}$ to get the attention weight matrix of the attention head h_i for the relation l_i.

$$\mathbf{M}^{h_i} = \mathbf{N}^{h_i} \cdot \mathbf{V}^{h_i} \tag{6}$$

Subsequently, the multi-head attention linearly maps the concatenation of all the heads with a parameter matrix $\mathbf{W}^o \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$, and sends this hidden representation to the standard Transformer encoder layers for further computations.

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\mathbf{M}^{h_i}, ..., \mathbf{M}^{h_m})\mathbf{W}^o \tag{7}$$

Finally, the objective function for training our model with syntax knowledge is identical to that of the vanilla Transformer (Vaswani et al., 2017):

$$\text{Loss} = -\sum_{t=1}^{T} [y_t \ln(o_t) + (y_t - 1) \ln(1 - o_t)] \tag{8}$$

Where y_t and o_t are, respectively, the true and the model-predicted value at state t, and T represents the number of states. The syntactic distribution matrices are not the object of optimization in the model, so it is incorporated into the model in the form of a parameter-free matrix.

4 Experiments and Analysis

Experimental Setup We establish seven distinct sets of experiments, refer to Table 1. To be specific, we will conduct particular experiments to validate the empirical performance under both medium size and small size training parallel corpora. Apart from the different network structures used in the models, the number of network layers are identical in the same language pair translation experiments for all models. Additionally,

[3]We group the original dependency labels into 16 alternative group labels. The grouping is provided in Appendix A.

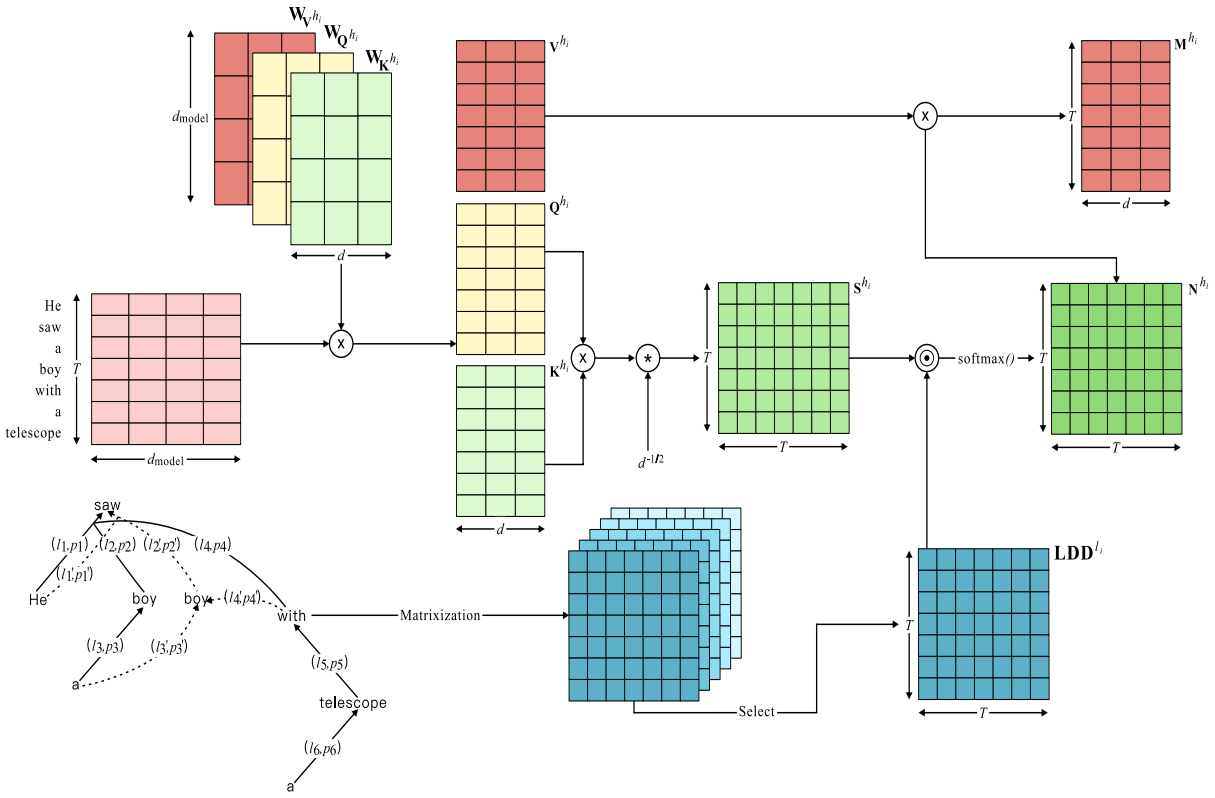

Figure 2: Labeled dependency distribution sub-layer ($\mathbf{LDD}^{l_i}$ for head h_i)

the seven models in each experiment will use the same parameter settings, loss function, and optimizer algorithm. Experiments will employ BLEU-{1,4} score (Papineni et al., 2002), RIBES score (Isozaki et al., 2010), TER score (Snover et al., 2006), and BEER score (Stanojevic and Sima'an,, 2014) as criteria for evaluating the model's effectiveness.

Parser: We employ an external dependency parser *SuPar* (Zhang et al., 2020) to automatically parse the source sentences. Since this parser was trained using the biaffine method (Dozat and Manning, 2016), we can extract dependency distributions by changing its source code.

Data: We evaluate the translation tasks for three language pairs from three different language families: English-Chinese (En→Zh), English-Italian (En→It), and English-German (En→De). We chose $dev2010$ and $test2010$ as our validation and test datasets from IWSLT2017 En→De and En→It tasks. In En→Zh, we randomly selected a 110K subset from the IWSLT2015 dataset as training set and used $dev2010$ as validation set, $tst2010$ as test set. Table 2 exhibits the division and statistics of the datasets.

For training only, we first filtered out the source sentences that SuPar cannot parse and sentences that exceed 256 tokens in length. And then, we used SuPar[4] to parse each source language sentence to obtain the labeled dependency distributions and applied Spacy[5] to tokenize the source and target languages, respectively. Finally, we replaced words in the corpus with "<unk>" for words with frequency less than two counts, and for each mini-batch sentences, added "<bos>","<eos>" tokens at the beginning and end, and for sentences with inconsistent lengths per mini-batch, added a corresponding number of "<pad>" tokens at the end of the sentences to keep the batch length consistent.

Hyperparameters: In the low-resource experiments, the batch size was 256, the number of layers for the encoder and decoder was 4, and the number of warm-up steps was 400. In the medium-resource experiments, their values were 512, 6, 4000, respectively. For the rest, we use the base configuration of the Transformer (Vaswani et al., 2017): All experiments were optimized using Adam (Kingma and Ba, 2015) (where β_1 was 0.9, β_2 was 0.98, ϵ was 10-9) and the initial learning rate was set to 0.0001, gradually reduced during training as follows:

[4]https://github.com/yzhangcs/parser
[5]https://spacy.io/

Table 1: Five sets of experimental group description

Experimental group	Description
Baseline (BL)	The original Transformer model.
+Labeled dependency attention only (LDA)	Replace **S** matrix directly with the labeled dependency distributions.
+1-best labeled dependency parse (LDP)	Incorporate 1-best dependency tree with specific (e.g. l_1) label.
+1-best unlabeled dependency parse (UDP)	Incorporate 1-best (regardless the type of dependency relations) dependency tree.
+Uniform labeled dependency distributions (ULDD)	Incorporate uniform labeled dependency distributions.
+Uniform unlabeled dependency distributions (UUDD)	Incorporate uniform unlabeled dependency distributions.
+Labeled dependency distributions (LDD)	Incorporate labeled dependency distributions with standard Transformer self-attention.

Table 2: Datasets statistics

Task	Corpus	Training set	Validation set	Test set
English → German	Multi30k	29000	1014	1000
	IWSLT 2017	206112	888	1568
English → Italian	IWSLT 2017	231619	929	1566
English → Chinese	IWSLT 2015	107860	802	1408

$$\mathrm{lr} = d_{\mathrm{model}}^{-0.5} \cdot \min(\mathrm{step\_num}^{-0.5}, \mathrm{step\_num} \cdot \mathrm{warmup\_steps}^{-1.5}) \quad (9)$$

The number of heads in multi-head attention was set to 8 (16 in LDD layer), the dimension of the model was 512, the dimension of inner fully-connected layers was set to 2048, and the loss function was the cross-entropy loss function. The checkpoint with the highest BLEU-4 score on the validation set was saved for model testing during training. The number of epochs was set to 50 (one epoch represents a complete training produce). In order to prevent over-fitting, we set the dropout rate (also in our LDD layer) to 0.1.

4.1 Experimental Results

The experimental results for each model under low- and medium-resource scenarios are shown in Tables 3 to 6. The first group represents the baseline model, while the remaining groups represent the control models. It is necessary to note that the last group is the model proposed in this paper.

As compared to the baseline model, either form of modeling the syntactic knowledge of the source language could be beneficial to the NMT models. Whether it was in the choice of lexical (BLEU-1) or in the order of word (RIBES), there was a certain degree of improvement, which also supports the validity and rationality of incorporating syntactic knowledge. The proposed model (LDD) achieved the best score in at least three of the five different evaluation metrics, regardless of the language translation tasks. The proposed model consistently reached the highest results on BLEU-4,

Table 3: Multi30k evaluation results (En → De)

Model	BLEU-1	RIBES	BLEU-4	TER	BEER
BL	**58.13**	78.86	30.14	62.95	0.59
+LDA	54.10	**80.10**	30.49	63.47	**0.61**
+LDP	54.26	79.58	30.71	79.58	0.61
+UDP	55.84	78.96	31.05	63.38	0.60
+ULDD	52.20	79.50	27.80	63.02	0.59
+UUDD	53.38	79.75	29.09	63.34	0.60
+LDD	55.65	79.97†‡	**31.29**†‡	**62.66**†‡	**0.61**
LDD compared to BL	−Δ2.48	+Δ1.11	+Δ1.15	+Δ0.29	+Δ0.02
LDD compared to UDP	−Φ0.19	+Φ1.01	+Φ0.24	+Φ0.72	+Φ0.01

[1] The black bold in the table represents the best experimental results under the same test set.
[2] Δ and Φ represent the improvement of our model compared to baseline and 1-best unlabeled parse system respectively.
[3] † and ‡ indicate statistical significance ($p<0.05$) against baseline and 1-best unlabeled parse system via T-test and Kolmogorov-Smirnov test respectively.

Table 4: IWSLT2017 evaluation results (En → De)

Model	BLEU-1	RIBES	BLEU-4	TER	BEER
BL	51.63	68.64	26.13	83.34	0.53
+LDA	49.89	69.04	26.16	83.53	0.53
+LDP	51.12	68.91	26.38	83.93	0.53
+UDP	50.90	69.20	26.39	84.65	0.53
+ULDD	50.80	**69.56**	25.10	82.76	0.53
+UUDD	48.85	68.90	25.41	86.19	0.53
+LDD	**54.98**†‡	68.83†	**27.78**†‡	**81.85**†‡	**0.54**
LDD compared to BL	+Δ3.35	+Δ0.19	+Δ1.65	+Δ1.49	+Δ0.01
LDD compared to UDP	+Φ4.08	−Φ0.37	+Φ1.39	+Φ2.80	+Φ0.01

[1] The black bold in the table represents the best experimental results under the same test set.
[2] Δ and Φ represent the improvement of our model compared to baseline and 1-best unlabeled parse system respectively.
[3] † and ‡ indicate statistical significance ($p<0.05$) against baseline and 1-best unlabeled parse system via T-test and Kolmogorov-Smirnov test respectively.

which increased by at least one point when compared to the baseline model, with an average increase rate of more than 5%. Furthermore, in most translation experiments, incorporating labeled dependency distributions provided better outcomes than the 1-best unlabeled dependency parse system (UDP)[6]. This indicates the efficacy of providing more parsing information, particularly the dependency probabilities. In the low resource scenarios, the models of incorporating syntactic knowledge

[6] All previous work uses only 1-best unlabeled parse, which is also our main comparison object. We will refer to it as 1-best parse or 1-best tree below.

Table 5: IWSLT2017 evaluation results (En → It)

Model	BLEU-1	RIBES	BLEU-4	TER	BEER
BL	54.14	68.58	27.11	77.52	0.56
+LDA	51.25	**69.90**	26.13	81.23	0.56
+LDP	51.72	68.26	25.65	80.03	0.55
+UDP	53.17	**69.90**	28.13	**76.18**	0.56
+ULDD	51.30	67.83	25.23	80.62	0.54
+UUDD	54.00	66.83	25.23	78.41	0.55
+LDD	**56.73**†‡	69.69†	**29.34**†‡	76.34†	**0.57**
LDD compared to BL	+Δ2.59	+Δ1.11	+Δ2.23	+Δ1.18	+Δ0.01
LDD compared to UDP	+Φ3.56	−Φ0.21	+Φ1.21	−Φ0.16	+Φ0.01

[1] The black bold in the table represents the best experimental results under the same test set.
[2] Δ and Φ represent the improvement of our model compared to baseline and 1-best unlabeled parse system respectively.
[3] † and ‡ indicate statistical significance ($p<0.05$) against baseline and 1-best unlabeled parse system via T-test and Kolmogorov-Smirnov test respectively.

Table 6: IWSLT2015 evaluation results (En → Zh)

Model	BLEU-1	BLEU-4	TER	BEER
BL	46.53	18.31	67.96	0.20
+LDA	44.91	18.25	70.96	0.20
+LDP	47.34	18.85	70.02	0.20
+UDP	46.92	19.71	**67.29**	0.20
+ULDD	40.67	17.89	77.04	0.19
+UUDD	34.14	18.05	79.27	0.18
+LDD	**47.62**†‡	**20.25**†‡	67.38†	0.20
LDD compared to BL	+Δ1.09	+Δ1.94	+Δ0.58	+Δ0.00
LDD compared to UDP	+Φ0.70	+Φ0.54	−Φ0.09	+Φ0.00

[1] The black bold in the table represents the best experimental results under the same test set.
[2] Δ and Φ represent the improvement of our model compared to baseline and 1-best unlabeled parse system respectively.
[3] † and ‡ indicate statistical significance ($p<0.05$) against baseline and 1-best unlabeled parse system via T-test and Kolmogorov-Smirnov test respectively.

paid less attention to the neighboring words in the corpus sentence because syntactic knowledge may assist models in focusing on distant words with syntactic relations, which was reflected in the decrease of BLEU-1 scores. This problem was alleviated in the richer-resource scenarios, which also showed that the robustness of the models improved.

For ablation experiments, passing the uniform dependency distributions verifies our hypothesis. A uniform probability tensor cannot provide valuable information to the Transformer model and risks misleading the model, resulting in the worst performance. Another notable finding is that simply incorporating labeled dependency distributions (replacing the **K** and **Q** matrices in the attention matrices) as dependency attention outperformed the baseline model on average. The benefit of this strategy is that by replacing **K** and **Q** matrices and their associated calculation process can drastically decrease the number of parameters and computing requirements.

4.2 Qualitative Analysis

BLEU-4 Scores Comparison: We also attempted to visualize the results to understand the performance of the proposed model better. In Figure 3, although the 1-best parse model performs better than the baseline model, the model we propose has higher scores than the baseline model and the 1-best parse model in all the median, upper and lower quartile scores. From the original scatter diagram, we can observe the scatter distribution of the proposed model at the upper position in general, indicating that, our model can earn higher scores for translated results than the baseline model and 1-best parse model.

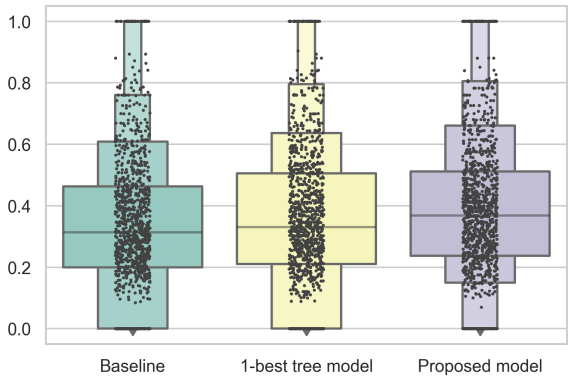

Figure 3: Box plot of baseline model, 1-best tree model and proposed model results

Impact of Sentence Length: We investigated translation performance for different target sentence lengths, by grouping the target sentences in the IWSLT datasets by sentence length intervals. We choose to group the target sentence lengths rather than source sentence lengths because, cf. Moore (2002), the source sentence and target sentence lengths are proportional. Second, since the target languages are different, and the source language is English, we are particularly concerned about the change in the length of sentences across different target languages.

Overall, our model outperformed the baseline system and 1-best parse system, as shown in Figure 4. Among them, the increase in the length range (20,30], (30,40] and (40,50] were more pronounced over the baseline system and 1-best parse system. The BLEU-4 scores of both our model and 1-best parse model were in danger of slipping

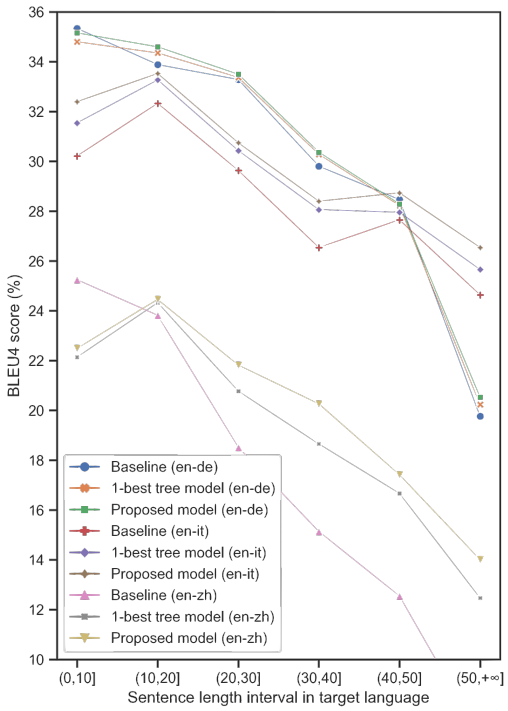

Figure 4: BLEU-4 comparison in sentences length

below the baseline model in the sentence length interval (0,10]. Corpus analysis shows that this length interval contains many fragments, remaining after slicing long sentences. Because the syntactic structures of these fragments were incomplete, they may negatively impact on the model's translation performance. As sentence length increased further, all models saw substantial declines in BLEU-4 scores, following similar downward patterns. When the sentence length exceeds 50, the BLEU-4 scores of our method remained significantly different from both the baseline model and the 1-best parse model. These showed that our proposed model has better translation performance in lengthy sentences, but BLEU-4 scores were still relatively low, indicating that the NMT models have much room for improvement.

Attention Weights Visualization: The final layer's attention weights of the 1-best parse model and the model we proposed are depicted in Figures 5 and 6, respectively. Judging from the comparison of the figures, we find that there are certain consistencies; for example, each word has higher attention weights to the words around it. However, the distinction is also discernible.

Specifically, for the word "A", the word "A" and the word "man" have a syntactic relation, which was represented in both figures. However, the 1-best parse model also provided "staring" a higher

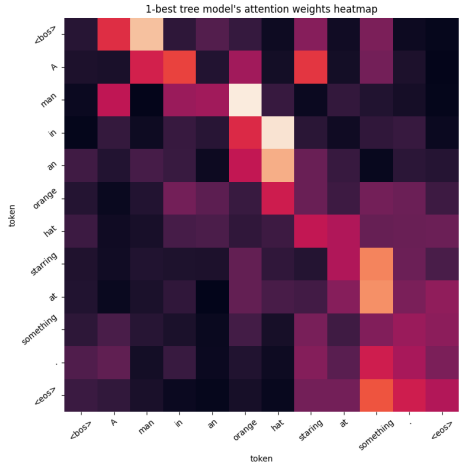

Figure 5: An example of 1-best parse model's attention weights

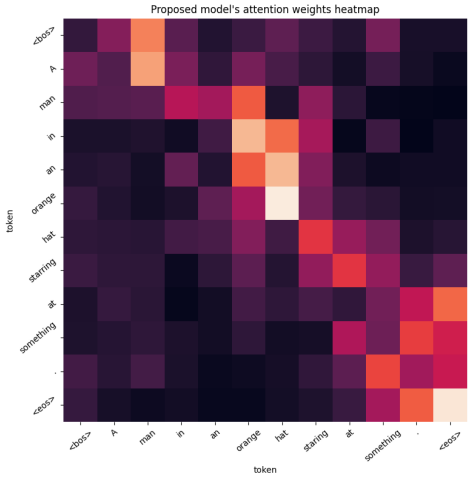

Figure 6: An example of proposed model's attention weights

attention weight, which is contrary to the syntactic structures, and the model we proposed resolved this problem. For the word "man", the 1-best parse model did not pay proper attention to distance but with syntactic relation word "staring", on the contrary, in the proposed model, "staring" was paid attention with a very high value. In a nutshell, both the 1-best parse model and the proposed model are better than the baseline model in terms of attention alignment which demonstrates that the syntactic knowledge contained in dependency distributions can guide the weight computation of the attention mechanism, directing it to pay more attention to words with syntactic relations, thereby improving the alignment quality to a certain extent.

5 Conclusion

This paper presented a novel supervised conditional labeled dependency distributions Trans-

former network (LDD-Seq). This method primarily improves the self-attention mechanism in the Transformer model by converting the dependency forest to conditional probability distributions; each self-attention head in the Transformer learns a dependency relation distribution, allowing the Transformer to learn source language's dependency constraints, and generates attention weights that are more in line with the syntactic structures. The experimental outcomes demonstrated that the proposed method was straightforward, and it could effectively leverage the source language dependency syntactic structures to improve the Transformer's translation performance without increasing the complexity of the Transformer network or interfering with the highly parallelized characteristic of the Transformer model.

References

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bastings, Jasmijn and Ivan Titov and Wilker Aziz and Diego Marcheggiani and Khalil Sima'an. 2017. Graph Convolutional Encoders for Syntax-aware Neural Machine Translation. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 1957–1967.

Belinkov, Yonatan and Yonatan Bisk. 2018. Synthetic and Natural Noise Both Break Neural Machine Translation. *International Conference on Learning Representations*.

Bugliarello, Emanuele and Naoaki Okazaki. 2020. Enhancing Machine Translation with Dependency-Aware Self-Attention. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online. 1618–1627.

Chen, Kehai and Rui Wang and Masao Utiyama and Eiichiro Sumita and Tiejun Zhao. 2018. Syntax-directed attention for neural machine translation. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Cho, Kyunghyun and Bart van Merriénboer and Caglar Gulcehre and Dzmitry Bahdanau and Fethi Bougares and Holger Schwenk and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1724–1734.

Currey, Anna and Kenneth Heafield. 2019. Incorporating Source Syntax into Transformer-Based Neural Machine Translation. *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers*. 24–33.

Deguchi, Hiroyuki and Akihiro Tamura and Takashi Ninomiya. 2019. Dependency-based self-attention for transformer NMT. *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*. 239–246.

Dozat, Timothy and Christopher D Manning. 2016. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734*.

Duan, Sufeng and Hai Zhao and Junru Zhou and Rui Wang. 2019. Syntax-aware transformer encoder for neural machine translation. *2019 International Conference on Asian Language Processing (IALP)*. IEEE. 396–401.

Eriguchi, Akiko and Kazuma Hashimoto and Yoshimasa Tsuruoka. 2016. Tree-to-Sequence Attentional Neural Machine Translation. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany 823–833.

Isozaki, Hideki and Tsutomu Hirao and Kevin Duh and Katsuhito Sudoh and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language pairs. *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. 944–952.

Kalchbrenner, Nal and Phil Blunsom. 2013. Recurrent Continuous Translation Models. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 1700–1709.

Kingma, Diederik P and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. *ICLR (Poster)*.

Ma, Chunpeng and Akihiro Tamura and Masao Utiyama and Tiejun Zhao and Eiichiro Sumita. 2018. Forest-Based Neural Machine Translation. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia. 1253–1263.

Moore, Robert C. 2002. Fast and accurate sentence alignment of bilingual corpora. *Conference of the Association for Machine Translation in the Americas*. Springer. 135–144.

Omote, Yutaro and Akihiro Tamura and Takashi Ninomiya. 2019. Dependency-based relative positional encoding for transformer NMT. *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*. 854–861.

Mylonakis, Markos and Khalil Sima'an. 2011. Learning hierarchical translation structure with linguistic annotations. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. 642–652.

Neubig, Graham and Kevin Duh. 2014. On the elements of an accurate tree-to-string machine translation system. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 143–149.

Papineni, Kishore and Salim Roukos and Todd Ward and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 311–318.

Peng, Ru and Tianyong Hao and Yi Fang. 2021. Syntax-aware neural machine translation directed by syntactic dependency degree. *Neural Computing and Applications*. 16609–16625.

Pham, Thuong Hai and Dominik Macháček and Ondřej Bojar. 2019. Promoting the Knowledge of Source Syntax in Transformer NMT Is Not Needed. *Computación y Sistemas*. 923–934.

Shi, Xing and Inkit Padhi and Kevin Knight. 2016. Does String-Based Neural MT Learn Source Syntax? *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas. 1526–1534.

Snover, Matthew and Bonnie Dorr and Richard Schwartz and Linnea Micciulla and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*. 223–231.

Stanojević, Miloš and Khalil Sima'an. 2014. Fitting Sentence Level Translation Evaluation with Many Dense Features. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar. 202–206.

Sutskever, Ilya and Oriol Vinyals and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*.

Tu, Zhaopeng and Yang Liu and Young-Sook Hwang and Qun Liu and Shouxun Lin. 2010. Dependency forest for statistical machine translation. *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. 1092–1100.

Vaswani, Ashish and Noam Shazeer and Niki Parmar and Jakob Uszkoreit and Llion Jones and Aidan N Gomez and Łukasz Kaiser and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*. 5998–6008.

Zaremoodi, Poorya and Gholamreza Haffari. 2018. Incorporating Syntactic Uncertainty in Neural Machine Translation with a Forest-to-Sequence Model. *Proceedings of the 27th International Conference on Computational Linguistics*. 1421–1429.

Zhang, Tianfu and Heyan Huang and Chong Feng and Longbing Cao. 2021. Self-supervised bilingual syntactic alignment for neural machine translation. *Proceedings of the AAAI Conference on Artificial Intelligence*. 14454–14462.

Zhang, Meishan and Zhenghua Li and Guohong Fu and Min Zhang. 2019. Syntax-Enhanced Neural Machine Translation with Syntax-Aware Word Representations. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers*, Minneapolis, Minnesota. 1151–1161.

Zhang, Yu and Zhenghua Li and Min Zhang. 2020. Efficient Second-Order TreeCRF for Neural Dependency Parsing. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online. 3295–3305.

A Appendix: Dependency group labels

Table A: 16 alternative dependency group labels

Dependency group labels	Original dependency labels
l_1	root
l_2	aux, auxpass, cop
l_3	acomp, ccomp, pcomp, xcomp
l_4	dobj, iobj, pobj
l_5	csubj, csubjpass
l_6	nsubj, nsubjpass
l_7	cc
l_8	conj, preconj
l_9	advcl
l_{10}	amod
l_{11}	advmod
l_{12}	npadvmod, tmod
l_{13}	det, predet
l_{14}	num, number, quantmod
l_{15}	appos
l_{16}	punct