Graph-ic Improvements: Adding Explicit Syntactic Graphs to Neural Machine Translation

Yuqian Dai [a,*], Serge Sharoff [b] and Marc De Kamps [c]
[a] *Centre for Translation Studies, University of Leeds, Leeds, United Kingdom*
E-mail: daiyuqian2017@outlook.com
[b] *Centre for Translation Studies, University of Leeds, Leeds, United Kingdom*
E-mail: s.sharoff@leeds.ac.uk
[c] *Artificial Intelligence, University of Leeds, Leeds, United Kingdom*
E-mail: m.dekamps@leeds.ac.uk

Abstract. Neural Language Models such as BERT or GPT operate on the basis of sequences of words. Pre-training on a large corpus endows them with implicit knowledge about the relationship between words. This study explores the extent to which the explicit incorporation of knowledge about syntactic relations, represented as a graph of dependencies, can enhance Machine Translation (MT) tasks. Specifically, it employs the Graph Attention Network (GAT), trained on a Universal Dependencies (UD) corpus, to evaluate the impact of explicit syntactic knowledge, even when derived from a smaller corpus, in comparison to the pre-training of implicit knowledge on a massive corpus. The investigation involves an experiment on integrating GAT-models into the MT framework, demonstrating robust improvement in MT quality for three language pairs, thus opening up possibilities for neurosymbolic approaches to Natural Language Processing.

Keywords: Machine Translation, Syntactic Knowledge, Graph Attention Transformers

1. Introduction

The Transformer architecture [1] has proven to be an extremely effective method for pre-training language models, from BERT [2] to GPT [3]. These models leverage the self-attention mechanism for the masked language modeling task, i.e., predicting the word masked in a context. However, this relatively simple procedure leads to rich contextual representations, which can rival human performance. Nevertheless, despite their ability to learn implicit syntactic patterns, these models often struggle with explicit syntactic structures and phenomena [4, 5]. This limitation is particularly significant in tasks like Neural Machine Translation (NMT), where syntactic accuracy is crucial for correctly interpreting and translating the structure and meaning of the source text. On the other hand, linguistic research has long focused on the detailed description and annotation of syntactic relations across languages. The Universal Dependencies (UD) [6] provides a standardized framework for annotating syntactic dependencies, creating richly annotated corpora that can be leveraged to improve NMT systems. Integrating explicit syntactic knowledge into NMT models has the potential to enhance translation quality by providing more structured and interpretable representations of language.

[*] Corresponding author. E-mail: daiyuqian2017@outlook.com.

Neurosymbolic AI aims to bridge the gap between symbolic reasoning and neural computation, thereby enabling more transparent, interpretable, and robust AI systems. Symbolic reasoning involves using explicit rules and structures to represent and manipulate knowledge, while neural networks excel at learning from large datasets and capturing complex patterns [7, 8]. Traditional sequential models, such as Recurrent Neural Networks (RNNs) and Transformers, although capable of processing and representing sentences, often fail to accurately capture complex syntactic structures and phenomena [9–11]. The advent of Graph Attention Network (GAT) [12] introduces a more explicit representation of syntactic structures and inter-word dependencies through their topology, promising better readability and interpretability in Natural Language Processing (NLP) [13, 14].

Inspired by these developments, this study introduces NMT engines improved with Syntactic knowledge via Graph attention and BERT (SGB), where GAT provides a powerful mechanism for explicitly representing syntactic structures and inter-word dependencies, complementing the implicit knowledge captured by BERT. This approach aligns with the principles of neurosymbolic AI, which seeks to combine the strengths of symbolic reasoning (explicit syntactic graphs) with the robustness and scalability of neural networks (BERT and Transformer models). By integrating syntactic data from source sentences with GATs and BERT, we aim to improve Transformer-based NMT by incorporating syntax (every sentence yields a syntactic tree structure through the parser) and leveraging the capabilities of the pre-trained BERT model. Utilizing multi-head attention mechanisms within the graph structure allows for the explicit exploitation of source-side syntactic dependencies, enhancing both the BERT embeddings on the source side and the effectiveness of the target-side decoder. The study conducts experiments on translation tasks from Chinese, German, and Russian to English to demonstrate the effectiveness of the proposed methodology, across three typologically different languages. We also examines the interpretability of the proposed NMT engines in improving translation quality, such as better identification of certain syntactic structures in the source language, and whether GAT can effectively learn syntactic knowledge. This research fills the current gap in understanding how syntactic strategies impact Machine Translation (MT) quality. The main contributions of this study are summarized as follows:

- SGB engines can demonstrate the potential and effectiveness of fusing BERT with syntactic knowledge from graph attention in MT tasks. The proposed MT engines can be fine-tuned to complete the training of the MT engine without the need for pre-training from scratch.
- This study evaluates translation quality, focusing on improvements in Quality Estimation (QE) scores. The proposed MT engines demonstrate enhanced QE scores across three MT directions. A paired t-test confirms a statistically significant difference in translation quality. Additionally, the study identifies specific syntactic structures in source sentences that the SGB engines learn optimally from, leading to better translations.
- This study reveals that while GAT possesses the capability to learn syntactic knowledge, their sensitivity in the learning process is influenced by the multi-head attention mechanism and the number of model layers. Excessive model layers can even significantly impair the GAT's ability to learn dependency relations. Furthermore, there is a correlation between the GAT's mastery of syntactic dependencies and translation quality. Better-learned syntactic structures by the GAT enable the MT engine to more accurately recognize source language sentences with those structures, resulting in smoother and more accurate translations.
- This study investigates the interpretability of translation quality improvement through the lens of syntactic knowledge. The experiments demonstrate that a syntactic structure based on GAT enables more nuanced modeling of source language sentences by the lower and middle layers within BERT, thereby enhancing translation quality. While SGB engines enhanced with graph-based syntactic knowledge exhibit improved QE score distributions, the integration of BERT plays a crucial role in forming representations of source sentences. This research underscores the importance of accurate syntactic graphs for maintaining high-quality translations and highlights the limitations of current models in interpreting jumbled sentences.
- This study assesses the versatility of the proposed approach by integrating XLM-Roberta in place of BERT. Despite this substitution, the approach consistently improve translation quality across all evaluated MT directions, underscoring its broad applicability.

2. Related Studies

2.1. Pre-trained Language Models

Pre-trained models have significantly advanced NLP, particularly with the advent of Transformer architectures, marking a paradigm shift in the field's approach to understanding language [15, 16]. Among these innovations, BERT stands out by leveraging self-supervised learning on extensive corpora through the Masked Language Model (MLM) and Next Sentence Prediction (NSP) tasks. These techniques enable BERT to capture the essence of linguistic knowledge, enriching its understanding of language context and structure [4]. The empirical analysis and applications of BERT have also helped humans understand pre-trained language models, supporting future improvements. Also, BERT has made significant contributions to MT tasks, where its contextual word embeddings and generic linguistic knowledge learned from pre-training enhance the generalization ability of MT engines, especially in cases with limited bilingual data. Most studies show that incorporating BERT improves the performance of MT engines, as demonstrated by metrics such as the BLEU score [17–19].

2.2. Syntactic Knowledge in Translation

In the realm of MT, the importance of syntactic dependency cannot be overstated. Syntactic dependency is crucial for the grammatical dissection of sentences, presenting them in easily interpretable tree diagrams. The incorporation of syntactic data into Neural Machine Translation (NMT) systems provides substantial benefits, notably in clarifying sentence structure, facilitating more accurate context interpretation, and minimizing ambiguity. In recent years, the Transformer model has garnered significant attention, and the strategy for incorporating explicit syntactic knowledge has shifted progressively from Recurrent Neural Network (RNN)-based methods to Transformer-based ones [20–22]. Within the Transformer framework, a prevalent approach involves leveraging the self-attention mechanism to capture and represent syntactic information, enabling focused analysis on particular tokens. However, the efficacy of using the Transformer's attention mechanism as an explanatory tool remains a topic of debate [23, 24]. Efforts have been made to enhance the effectiveness of downstream tasks by fusing explicit syntactic knowledge with BERT [13, 25]. However, the applications of such integration in MT have not been thoroughly explored.

2.3. Deep Learning for Graphs

In NLP tasks, representing sentences and words as linear sequences might compress or obscure crucial topological information, including tree-like syntactic structures. This loss of structure can present significant challenges for downstream tasks that depend on accurately capturing the nuanced features of source language sentences, such as speech recognition and MT. While there are many approaches for encoding graphs, see [26], Graph Neural Networks (GNNs) offer a solution through a topological graph-based approach, enabling the construction of diverse linguistic graphs. These graphs transform various textual features into a network of nodes, edges, and overall graph structures. This method allows for a more nuanced analysis and inference of linguistic patterns within input sentences, significantly benefiting downstream tasks [27, 28]. The GAT emerges as a novel solution within this space, adept at processing data in non-Euclidean domains. It utilizes attention mechanisms to dynamically assign importance to nodes, enhancing the model's capacity to learn from graph-based representations. This capability, when combined with BERT, forms a robust framework for encapsulating linguistic knowledge in downstream NLP tasks [13, 29, 30].

3. Methodology

3.1. Construction of the Proposed Engines

This section provides detailed descriptions of the individual layers within the engine. Figure 1 illustrates the comprehensive architecture of the proposed engines.

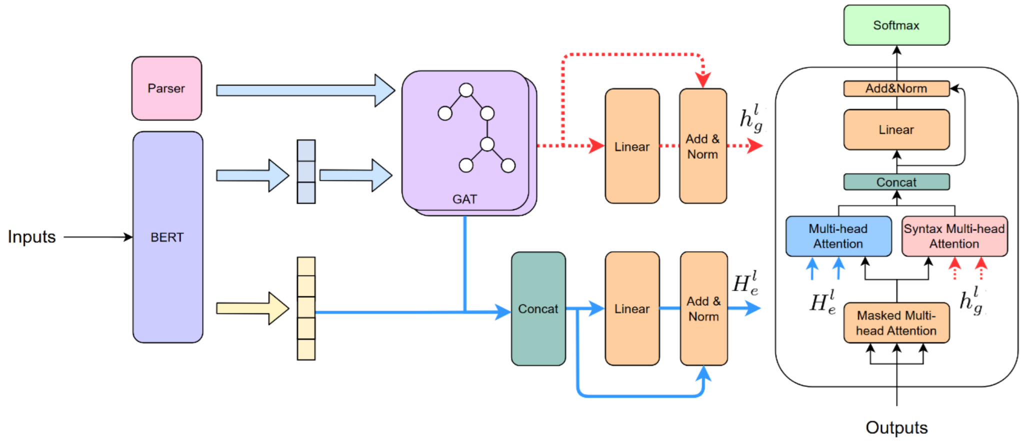


Fig. 1. The architecture of the SGB engines. The encoder with BERT and GAT on the left and the decoder on the right. Dash lines indicate the alternative connections. H_e^l and h_g^l represent the final layer output of BERT and GAT.

3.1.1. *Encoding*

Given source sentence $S = [w_1, w_2, w_3, \ldots w_i]$, where i is the number of word tokens in a sentence, S is then cut into subword tokens and fed into BERT, which become: $\tilde{S} = [[CLS], w_1^1, w_1^{1\#1}, w_2, w_3^3, w_3^{3\#3}, \ldots w_n, [SEP]]$, Where $w^{n\#n}$ represents the subwords of w_n, [CLS] and [SEP] are special tokens of BERT.

The experiments include translations from three source languages into English: Chinese to English (Zh→En), Russian to English (Ru→En), and German to English (De→En). We use three BERT variants as an encoder for each MT engine, where Chinese is chinese-bert-wwm-ext[1], Russian is rubert-base[2], and German is bert-base-german[3]. Although their model structures are the same, the approaches differ in pre-training. Chinese BERT uses Whole Word Masking, Russian BERT takes the multilingual version of BERT-base as its initialization for further pre-training, and the approach of German BERT remains the same as vanilla BERT. We aim to propose approaches that can be generalized to the BERT model structure, even their pre-training approaches are different.

By capturing the representation of each subword token through BERT, the final embedded sequence is accessible via the last layer of BERT, $h_B = BERT(\tilde{S})$. To obtain the syntactic dependency information of the source sentence S, we use a Universal Dependencies-based parser[4] [31] to perform tokenizing and syntactic dependency parsing on source sentences, as shown in Table 1. After obtaining the parsing results, we aim to represent the syntactic connections between words in the sentence using a graph. We construct the node adjacency matrix for graph representation, where each token corresponds to a node in the graph as shown in Figure 2. Since word representations from BERT contain rich semantic information, nodes on the graph are encoded by BERT embeddings. Considering the subword segmentation, we average subword token representations to obtain the node embeddings on the graph.

3.1.2. *Graph Attention*

Words and adjacency relations in a sentence can be represented as a graph structure, where the words (known as tokens in the model) on the graph are as nodes, and the relationships called syntactic dependencies between words are regarded as edges connecting nodes. We use GAT [12] as our critical component to fuse the graph-structured information and node features. The node features given to a GAT layer are $\tilde{G} = [x_1, x_2, \ldots x_i, \ldots x_n], x_i \in \mathbb{R}^F$, where n is the total number of nodes, F is the feature size of each node, the same with BERT embedding. The Equation (1) and (2) summarise the working mechanism of the GAT.

---

[1]https://huggingface.co/hfl/chinese-bert-wwm-ext
[2]https://huggingface.co/DeepPavlov/rubert-base-cased
[3]https://huggingface.co/bert-base-german-cased
[4]https://github.com/hankcs/HanLP

Table 1

To illustrate the working principle, consider the input sentence: "The new spending is fueled by Clinton's large bank account.". This sentence is subsequently parsed to provide detailed linguistic information, such as part-of-speech (POS) tags, head node IDs, and syntactic dependency labels (DepRel). Source language sentences in Chinese, Russian, and German also follow the same parsing steps.

index	Word	POS	Head	DepRel
1	The	DET	3	det
2	new	ADJ	3	amod
3	spending	NOUN	5	nsubj:pass
4	is	AUX	5	aux:pass
5	fueled	VERB	0	root
6	by	ADP	11	case
7	Clinton	PROPN	11	nmod:poss
8	's	PART	7	case
9	large	ADJ	11	amod
10	bank	NOUN	11	compound
11	account	NOUN	5	obl:agent
12	.	PUNCT	5	punct

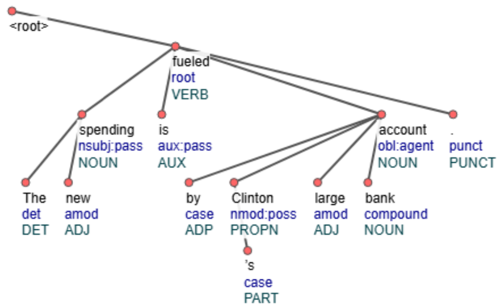


Fig. 2. The input sentence is parsed, and it is then expected to be converted into a graph structure based on the connections between parent nodes in the syntactic dependencies.

$$h_i^{out} = \overset{K}{\underset{k=1}{\Big\|}} \sigma \left(\sum_{j \in N_i} \alpha_{ij}^k W^k x_j \right) \tag{1}$$

$$\alpha_{ij}^k = \frac{exp(LeakyReLU(a^T[Wx_i \parallel Wx_j]))}{\sum_{v \in N_i} exp(LeakyReLU(a^T[Wx_i \parallel Wx_v]))} \tag{2}$$

1-hop neighbors $j \in N_i$ are attended by the node i, $\overset{K}{\underset{k=1}{\|}}$ represents K multi-head attention output concatenation. h_i^{out} is the representation of node i at the given layer. α_{ij}^k means attention between node i and j. W^k is linear transformation, a is the weight vector for attention computation, $LeakyReLU$ is activation function. Simplistically, the feature calculation of one-layer GAT can be concluded as $h_G = GAT(X, A; \Theta^l)$. The input is $X \in \mathbb{R}^{n \times F}$, and the final output is $h_G \in \mathbb{R}^{n \times F'}$ where n is the number of nodes, F is the feature size for each node, F' is the hidden state for GAT, $A \in \mathbb{R}^{n \times n}$ is the graph adjacency matrix indicating node connection, Θ^l is the parameters during training. During training, the GAT faithfully represents the syntactic information provided by the parser in the adjacency matrix. It then obtains the representations of the vertices and passes them to subsequent model modules. However, we cannot guarantee that all information from the parser is correct. Therefore, we treat incorrect information as noise, allowing the model to learn and enhance its robustness against such noise.

3.1.3. Fusion and Output

Two methodologies for integrating syntactic knowledge into machine translation (MT) engines are introduced. The initial approach, termed Syntactic Knowledge via Graph Attention with BERT Concatenation (SGBC), involves merging syntactic information from graphs with BERT for the encoder's operation, as detailed in Equations (3) and (4).

$$H_e^l = concat(h_B, h_G) \tag{3}$$

$$\tilde{h}_d^l = attn_D(h_d^l, H_e^l, H_e^l) \tag{4}$$

where $attn_D$ stands for encoder-decoder attention in MT engines. l is the output of the l-th layer, d is the representation of the tokens in decoder-side. H_e^l contains the features of BERT (h_B) and GAT (h_G) fed into the encoder-decoder attention module in the decoder. The feed-forward network subsequently processes the attention features alone with residual connection, as in the case of the vanilla Transformer model.

The second one, called Syntactic knowledge via Graph attention with BERT and Decoder (SGBD), is that the syntactic knowledge on the graph is not only applied to the encoder but also guides the decoder through the syntax-decoder attention, as shown in Equations (5), (6) and (7).

$$\tilde{h}_d^l = attn_D(h_d^l, H_e^l, H_e^l) \tag{5}$$

$$\tilde{h}_s^l = attn_S(h_d^l, h_g^l, h_g^l) \tag{6}$$

$$\tilde{h}_t^l = concat(\tilde{h}_d^l, \tilde{h}_s^l) \tag{7}$$

where $attn_D$ and $attn_S$ represent encoder-decoder attention and syntax-decoder attention respectively. h_g^l is the output of GAT containing syntactic dependency features of sentences via another feed-forward network. $\tilde{h}_t^l$ is the final attention features obtained by concatenating $attn_D$ and $attn_S$. As with the vanilla Transformer, the predicted word is generated by a feed-forward network with residual connection and softmax function.

3.2. Metrics for Machine Translation Evaluation

In the domain of MT, there is an active search for accurate and reliable evaluation metrics. Among these metrics, BLEU [32] has become a fundamental tool for evaluating the quality of text translated from one language to another. BLEU functions by comparing machine-generated translations to one or more reference translations, primarily focusing on the precision of n-grams. Despite its widespread use, BLEU's sole emphasis on precise matching the reference translations, without considering fluency or content adequacy, has led researchers to seek supplementary evaluation strategies.

QE offers an innovative approach to translation assessment that does not require reference texts, by building models that directly predict whether the suggested translation is an accurate and fluent translation of the source text. This method is not only innovative but also practical, especially in contexts where reference translations are unavailable. QE engines can be trained to evaluate various aspects including fluency, adequacy, and even the predicted post-editing effort, providing a comprehensive view of translation quality.

In this study, the evaluation of MT primarily employs two methods: the widely recognized n-gram matching model, BLEU, and advanced neural network-based QE models, specifically COMET QE [33] and TransQuest QE [34]. However, both BLEU and COMET QE operate at the corpus level, failing to identify improvements in specific sentences and relying on reference translations, which can overlook legitimate translation variants. In contrast, TransQuest QE employs MT quality assessment techniques to measure sentence-level improvements without relying on reference translations. Additionally, TransQuest QE leverages state-of-the-art transformer models, introducing a novel quality assessment method through sentence-level quality estimation. It predicts a quality score for each sentence pair (source and translated sentence), which correlates with human judgments on translation quality. This approach represents significant advancements over traditional QE methods, providing more accurate and reliable assessments. TransQuest is also the winner of the WMT 20 QE shared task. Therefore, in the subsequent experiments, the QE scores are derived from the TransQuest QE methodology unless otherwise specified.

3.3. Datasets

The Parallel Universal Dependencies (PUD) corpus is a collection of multilingual datasets designed to facilitate cross-linguistic analysis and the development of MT engines. Comprising texts translated into 20 languages, each dataset within the PUD corpus contains 1,000 sentences that are syntactically annotated, ensuring a high level of linguistic consistency and quality across different languages. These sentences are selected from a wide range of sources, including news articles and Wikipedia, providing a diverse mix of genres and topics.

The experiments utilize three typologically different languages to be translated into English: PUD Chinese[5], PUD Russian[6], and PUD German[7]. The choice of these languages is determined by the availability of the UD corpus for a trained external syntactic parser and the PUD corpus for evaluating both the syntactic knowledge of BERT and GAT and the performance of the MT engine.

4. What Happens to Translations

4.1. Translation Performance with BLEU and Quality Estimation

The effectiveness of the proposed approach is evaluated by BLEU score on the UNPC[8] (Zh→En, Ru→En) and Europarl[9] (De→En) datasets. 1 million (M) sentence pairs are selected as the training set for each language, with 6,000 and 5,000 sentence pairs for the validation and test sets, respectively. The dataset is randomly divided to ensure that each subset is representative of the overall distribution, thereby reducing bias and ensuring a fair evaluation of the model's performance. The validation set is used to monitor the model's performance during training and to implement early stopping to prevent overfitting, while the test set is used for final evaluation to assess the model's generalization capabilities. The baseline involves an encoder based on fine-tuned BERT, compared fairly with the proposed SGB engines using the same training setup. Decoders from the vanilla Transformer model are used, featuring BERT variants for each source language with 6 layers and 8 attention heads, while maintaining consistency in other parameters. The GAT within SGB engines includes 2 layers and 6 attention heads for Zh, and 4 attention heads for Ru and De, optimizing model performance. Training utilizes the Adam optimizer with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.98$, a learning rate of 2e-5, word embedding of 768, and cross entropy as the loss function. All experiments are performed on RTX 3080 and 3090 GPUs.

As shown in Table 2, the proposed two engines achieve higher BLEU scores than the baseline engines across all three translation directions, regardless of the changes in the training set size. This demonstrates the effectiveness and generalization capability of the proposed approach. SGBC consistently outperforms both the baseline and SGBD. This can be attributed to the fact that the output of SGBC more closely aligns with the criteria used in the BLEU score calculation. It is likely to generate translations that have a higher degree of n-gram overlap with the reference translations, thus achieving higher BLEU scores. In contrast, the more complex SGBD produces translations that are more varied or nuanced, which may not always align as closely with the reference translations in terms of n-gram precision. Inspired by the study revealing BLEU reliability [35], BLEU scores may not be sufficient to capture the nuanced quality of translations. Therefore, two QE models, COMET and TransQuest, are introduced to further evaluate the translation quality of the proposed models. The key difference between these models is that COMET

[5]https://github.com/UniversalDependencies/UD_Chinese-PUD
[6]https://github.com/UniversalDependencies/UD_Russian-PUD
[7]https://github.com/UniversalDependencies/UD_German-PUD
[8]https://opus.nlpl.eu/UNPC.php
[9]https://opus.nlpl.eu/Europarl.php

Table 2

The performance of SGB engines compared to baseline engines in BLEU scores across three MT directions with varying training set sizes. Despite the reduced dataset Size, SGB engines maintain competitive BLEU scores.

Language	Training Size	Baseline	SGBC	SGBD
Zh→En	0.1M	*24.26*	**24.89**	24.72
	0.5M	*38.48*	**38.71**	38.53
	1M	*47.15*	**47.23**	47.17
Ru→En	0.1M	*21.12*	**21.45**	21.33
	0.5M	*37.69*	**37.74**	37.68
	1M	*47.22*	**47.36**	47.27
De→En	0.1M	*15.41*	**15.79**	15.50
	0.5M	*26.89*	**27.13**	26.92
	1M	*37.59*	**37.67**	37.63

Table 3

Performance comparison of BLEU, COMET, and TransQuest scores for three translation directions (Zh→En, Ru→En, De→En) with a training set size of 1 million. The table shows the scores for the Baseline, SGBC, and SGBD models, highlighting the best performance in each metric with bold text.

Training Size	Language	Zh→En			Ru→En			De→En		
	Metric	Baseline	SGBC	SGBD	Baseline	SGBC	SGBD	Baseline	SGBC	SGBD
1M	BLEU	47.15	**47.23**	47.17	47.22	**47.36**	47.27	37.59	**37.67**	37.63
	COMET	82.20	83.69	**84.78**	80.93	81.34	**82.56**	78.02	78.66	**79.37**
	TransQuest	70.08	72.66	**73.01**	81.65	83.31	**83.95**	75.49	77.00	**77.94**

assesses the translation quality by examining the interplay between the source sentence, its translation, and reference translations, whereas TransQuest only requires the source sentence and its translation. All performance metrics are scored on a scale from 0 to 100, with higher scores indicating better translation quality.

Table 3 demonstrates that when the training set size reaches 1 million, both SGB series engines exhibit higher scores on the BLEU and COMET QE performance metrics. However, SGBC and SGBD exhibit notable differences in their performance across these metrics: SGBC achieves the highest BLEU scores in all three translation directions, while SGBD obtains the highest COMET and TransQuest QE scores. SGBD's scores are generally at least 2 points higher than those of the baseline engines. These performance metrics reflect the engines' proficiency in leveraging syntactic knowledge from graphs and fully utilizing BERT's potential language capabilities, enabling them to generate more accurate translations. It is important to note that BLEU is a paired metric, which can be unreliable, and both BLEU and COMET QE depend on reference translations. In real-world translation scenarios, reference translations may not always be available, and the semantic diversity of output sentences cannot be reliably verified. Therefore, compared to BLEU and COMET QE scores, the TransQuest QE score offers a more nuanced advantage in adapting to reasonable variations in translation. This is because it does not require reference translations, making it a more robust and practical metric for evaluating translation quality in diverse and realistic settings.

4.2. Translation of In-domain and Out-of-domain Sentences

Based on the results of the above experiments, BLEU scores still fail to reflect linguistic subtleties and align with human evaluative criteria [36, 37]. To address these shortcomings, we employ a gold-standard syntactically annotated corpus, the PUD corpus, and the TransQuest QE model to further investigate changes in translation quality. The PUD corpus, with its diverse range of sources, including out-of-domain content, ensures a comprehensive evaluation of the MT engines' ability to handle various linguistic structures and contexts. Additionally, the syntactic annotations in the PUD corpus provide a gold-standard reference, allowing for a detailed analysis of the engines' performance in capturing and translating syntactic dependencies. We utilize the PUD corpus (PUD Chinese, PUD

Table 4

The baseline and the SGB engines compare the translations of the PUD corpus, scored by the QE model and subjected to paired t-tests to demonstrate the differences in translation quality scores.

Source Language	Sample Size	Models		$\bar{x}_d$	S_d	t	P-value
Zh	1000	Baseline	SGBC	0.024	0.109	7.18	p < 0.001
			SGBD	0.032	0.111	9.12	p < 0.001
Ru	1000	Baseline	SGBC	0.024	0.042	18.38	p < 0.001
			SGBD	0.034	0.045	23.67	p < 0.001
De	1000	Baseline	SGBC	0.007	0.113	2.16	p = 0.030
			SGBD	0.012	0.110	3.61	p < 0.001

Russian, and PUD German) to evaluate the translation quality of the Baseline and SGB engines across three translation directions. The PUD corpus includes sentences from various out-of-domain sources, not limited to news and Wikipedia content, thus placing higher demands on the MT engines' ability to effectively summarize and clarify the structure of input sentences. The QE model is used to estimate the quality of the source language sentences and their translations, rating the translations on a scale from 0 to 1, where higher scores indicate better translation quality. Paired t-tests are used to analyze the changes and distribution of translation quality before and after implementing the proposed strategies, with a significance level of 0.05.

From Table 4, when comparing the Zh Baseline and SGBC engines, average of differences ($\bar{x}_d$) of them is 0.024, standard deviation of the difference (S_d) is 0.109 and the test statistic (t) is 7.18, corresponding to a p-value < 0.001. Similarly, the t and p-values for the SGBD engine also reveal the statistical significance of the QE scores before and after the proposed approach. Both comparisons reject the null hypothesis H_0 at the significance level of 0.05, where H_0 states that the proposed approaches do not significantly differ in QE scores compared to the baselines. Instead, the alternative hypothesis H_1 is accepted, which states that the differences between the baseline and SGB engines in QE scores are large enough to be statistically significant. Specifically, H_1 asserts that the QE scores of the SGB engines are significantly higher than those of the baseline engines.

Comparable outcomes are evident for Ru and De, wherein the quality of translations, upon the implementation of proposed methodologies, manifests a significant divergence from the prior state, as gauged by QE scores. The incorporation of syntactic knowledge via graph representations alongside the employment of BERT substantially enhances the translation efficacy of MT engines. It is noteworthy that the SGBD engines consistently achieve elevated QE scores, indicating a robust improvement in translation quality. Contrarily, while the SGBC engines are favored by BLEU scores, achieving higher metrics under that evaluation, the QE scores highlight a different aspect of translation quality, underscoring the nuanced and comprehensive analysis provided by QE metrics over BLEU. This divergence underscores the complexity of translation quality evaluation, revealing how different evaluation metrics may prioritize various aspects of translation performance.

4.3. *Identifying Syntactic Relations in Source Language Sentences*

Multiple dependency relations signify the structural attributes of a given sentence. To identify which dependency relation in the source language sentence from the PUD corpus contributes most to the enhancement of translation quality through translation engines, we retain and categorize sentences based on their dependency relations. Specifically, both the baseline engine and the two proposed SGB engines translate their own source language sentences from the PUD corpus. The translations are then ranked according to their TransQuest QE scores. The bottom 30% of translations, based on TransQuest QE scores, are considered low-quality translations. Source language sentences corresponding to these low-quality translations and containing the same dependency relation are grouped together. For example, for a given dependency relation d, any source language sentence with a low-quality translation containing such dependency d is grouped together. The average TransQuest QE score for each group, characterized by specific dependency relations, is calculated both before and after the application of the proposed methodologies.

This approach allows us to conduct a detailed examination of the impact of distinct syntactic structures on the efficacy of translation quality improvements facilitated by the engines. By analyzing these groups, we can determine which dependency relations are most influential in improving translation quality, thereby providing insights into the syntactic features that benefit most from the proposed improvements.

Table 5 details how SGB engines outperform the baseline engines in accurately identifying syntactic relations within source language sentences, thereby markedly improving translation quality. It particularly emphasizes the top five syntactic relations that contribute to this improvement. Although both SGBC and SGBD engines incorporate graph-based syntactic knowledge, their approaches to learning dependency relations diverge. For instance, the "*flat*" (flat structure) in Zh is markedly significant in the SGBC engine yet receives less emphasis in the SGBD engine. Despite SGBD's decoders being similarly guided by syntactic knowledge derived from graph representations, it does not uniformly excel across all syntactic relations in achieving a higher QE score compared to the SGBC engine. Specifically, in languages such as Zh, Ru, and De, the SGBC model outperforms SGBD in handling certain syntactic relations, including "*discourse:sp*" (discourse marker: speech), "*orphan*" (orphan), and "*csubj*" (clausal subject). This discrepancy may suggest that an overly focused reliance on syntactic knowledge could lead to knowledge redundancy, detrimentally affecting translation quality in the SGBD engine. Conversely, the importance of some syntactic relations remains consistent across both SGBC and SGBD engines, underscoring that the integration of syntactic knowledge via graph attention alongside BERT enables the MT engine to more precisely address specific common relations. This consistency, irrespective of the methodological differences between the two engines, indicates that leveraging graph-based syntactic knowledge in conjunction with BERT enhances the MT engine's ability to explicitly navigate certain syntactic structures, thus contributing to the refinement of translation quality.

5. What Happens to Graphs

5.1. Syntactic Knowledge in GAT

Graph Attention Networks (GATs) have the capability to represent syntactic structures in sentences using graph-based models. However, whether this capability signifies their ability to effectively learn syntactic knowledge remains an open question. To address this, we design a syntactic dependency prediction experiment where GATs are tasked with predicting the relevant syntactic labels in the syntactic structure. For this experiment, we utilize the PUD corpus, which provides gold-standard syntactic annotations, as our foundational dataset. The experimental process involves converting the syntactic annotations and sentence words into syntactic trees, which are subsequently transformed into graph structures for GAT analysis. In these graph structures, each word is represented as a node, and the edges represent the syntactic dependency connections as defined by the PUD corpus. The primary objective of the GAT is to infer the dependency relations for each word by integrating information from both nodes and edges. Unlike traditional syntactic dependency models, which often follow a unidirectional flow from parent to child nodes, this approach treats dependencies as bidirectional graphs. This bidirectional model acknowledges the mutual influence between parent and child nodes, which is crucial for GATs to understand the varying implications of node connections. By considering these bidirectional relationships, GATs can enhance their ability to accurately identify dependency relations among nodes, thereby improving their syntactic learning capabilities.

Similar to the Transformer model, GAT utilizes multi-head attention and layers stacked upon each other. The study initially explores how the number of multi-head attention heads and layers influences GATs' acquisition of syntactic knowledge, examining the advantages these configurations offer for learning syntactic dependencies. In the experiments, the attention head counts (Heads) tested for GATs are 2, 4, 6, and 8, while the layer counts (L) explored are 2, 3, 4, 5, and 6. For each language, datasets are divided into training, validation, and test sets with 800, 100, and 100 sentences, respectively, to tune hyperparameters, monitor model performance during training to prevent overfitting, and evaluate the model on unseen data. The model parameters are set with a learning rate of 2e-5, a dropout rate of 0.2, Adam as the optimizer, and a hidden size of 768. The F1-score is used as the evaluation metric.

Table 6 emphasizes the critical importance of judiciously configuring the number of attention heads and layers in GAT, as this configuration significantly influences the model's sensitivity to accurately learn syntactic knowledge.

Table 5

The top-5 dependency relations identified by the SGB engines are those that show the greatest improvement in QE scores. These relations highlight which syntactic dependencies are most effectively detected and contribute most significantly to the enhancement of translation quality in each translation direction. "Qual" denotes the percentage increase in QE scores for sentences containing such a syntactic structure.

	Baseline	SGBC	Qual		Baseline	SGBD	Qual
Zh							
obl:agent	0.379	0.576	+51.978%	obl:agent	0.379	0.597	+57.519%
discourse:sp	0.388	0.502	+29.381%	iobj	0.387	0.511	+32.041%
flat	0.387	0.494	+27.648%	nsubj:pass	0.423	0.545	+28.841%
flat:name	0.415	0.518	+24.819%	appos	0.404	0.518	+28.217%
mark:prt	0.435	0.532	+22.298%	discourse:sp	0.388	0.501	+29.123%
Ru							
	Baseline	SGBC	Qual		Baseline	SGBD	Qual
orphan	0.608	0.768	+26.315%	orphan	0.608	0.719	+18.256%
aux	0.700	0.764	+9.142%	aux	0.700	0.777	+11.000%
ccomp	0.681	0.745	+9.397%	ccomp	0.681	0.747	+9.691%
flat:name	0.703	0.761	+8.250%	discourse	0.614	0.676	+10.097%
fixed	0.688	0.742	+7.848%	fixed	0.688	0.750	+9.011%
De							
	Baseline	SGBC	Qual		Baseline	SGBD	Qual
csubj	0.449	0.566	+26.057%	flat	0.442	0.625	+41.402%
flat	0.442	0.553	+25.113%	csubj	0.449	0.554	+23.385%
expl	0.486	0.573	+17.901%	expl	0.486	0.589	+21.193%
compound:prt	0.493	0.579	+17.444%	compound:prt	0.493	0.595	+20.689%
compound	0.495	0.577	+16.565%	cop	0.502	0.586	+16.733%

For example, the Russian language experiment reveals that a GAT setup with 2 layers and 4 attention heads outperforms a configuration with 8 attention heads in terms of overall prediction efficacy. As the model is expanded to 4 layers, a higher number of attention heads enhances performance, with the F1-score increasing from 0.44 to 0.57. Conversely, increasing the number of layers tends to degrade the model's ability to accurately predict dependency relations. Specifically, a configuration with 2 layers outperforms one with 6 layers, regardless of the number of attention heads. This decline suggests that an increase in GAT layers might lead to performance degradation, potentially due to nodes losing their specific attributes or incorporating irrelevant information during the aggregation process.

When examining the prediction scores for individual dependency relations across the three languages, the results further validate this observation. As shown in Table 7, when the number of layers exceeds 3, the F1-scores for some syntactic relations tend to decrease and even drop to 0 as the number of layers increases. Increasing the number of attention heads does little to mitigate this degradation. However, certain syntactic tags remain unaffected by this trend. Regardless of the number of layers, GAT consistently learns and maintains high F1-scores for tags such as "*advmod*" (adverbial modifier), "*case*" (case marking), "*cc*" (coordinating conjunction), "*mark*" (marker), "*nsubj*" (nominal subject) and "*punct*" (punctuation). This indicates that GAT exhibits a high sensitivity and reliable capture of these specific syntactic features.

We continue to compare the F1 scores of GAT's dependency relation predictions with the QE scores of the SGB engines when processing prior low-quality translations containing these specific dependency relations (from Sec 4.3), as shown in Table 8. It highlights the top-10 dependency relations with the highest prediction scores

Table 6

GAT performance in syntactic dependency prediction for three languages with different numbers of attention heads and layers. The number of attention heads increases incrementally from 2 to 6, and the number of model layers increases from 2 to 8.

Layers	Zh			
	2 Heads	4 Heads	6 Heads	8 Heads
2	0.63	0.62	**0.64**	**0.64**
3	0.64	0.61	0.62	0.63
4	0.56	0.58	0.64	0.49
5	0.49	0.50	0.51	0.50
6	0.37	0.40	0.33	0.33
Layers	**Ru**			
	2 Heads	4 Heads	6 Heads	8 Heads
2	0.58	**0.61**	0.47	0.56
3	0.45	0.55	0.54	0.53
4	0.44	0.47	0.56	0.57
5	0.42	0.52	0.46	0.49
6	0.41	0.36	0.31	0.33
Layers	**De**			
	2 Heads	4 Heads	6 Heads	8 Heads
2	0.64	**0.67**	0.64	0.56
3	0.60	0.56	0.56	0.57
4	0.56	0.50	0.53	0.53
5	0.58	0.61	0.50	0.47
6	0.48	0.49	0.48	0.42

by GAT across various source language sentences, along with the corresponding changes in translation quality facilitated by different MT engines. The results demonstrate a clear positive correlation between GAT's syntactic dependency prediction scores and the improvement in translation quality, especially when using the SGBC and SGBD engines. For Zh, dependency relations such as "*mark*" (marker), "*cc*" (coordinating conjunction), and "*conj*" (conjunct) have very high prediction scores by GAT (0.986, 0.984, and 0.970, respectively). These high scores correlate with significant improvements in translation quality, as evidenced by the higher QE scores of the SGBC and SGBD models compared to the baseline. Similarly, for Ru, dependency relations like "*det*" (determiner), "*root*" (root), and "*amod*" (adjectival modifier) have high prediction scores (0.990, 0.987, and 0.982, respectively), leading to notable improvements in translation quality. For De, dependency relations such as "*case*" (case marking), "*cc*" (coordinating conjunction), and "*det*" (determiner) also exhibit high prediction scores (0.992, 0.987, and 0.987, respectively), resulting in improved translation quality. The positive correlation between GAT's prediction scores and translation quality is consistent across the three languages, suggesting that GAT's ability to accurately predict syntactic dependencies is a robust indicator of its potential to enhance translation quality. This underscores the importance of integrating syntactic information into MT systems to achieve more accurate and reliable translations. Also, The consistent improvement in translation quality across different languages and MT engines demonstrates the robustness of GAT in learning and applying graph-based syntactic structures.

Table 7

The prediction of syntactic dependencies for three languages is conducted using different numbers of attention heads and layers. As the number of layers increases, the performance of the GAT in predicting dependency labels declines, and it gradually loses the ability to learn certain dependency labels, resulting in the F1 scores dropping to zero. However, some dependency relations remain unaffected and continue to achieve relatively high prediction scores.

GAT		**Zh**			**Ru**			**De**		
Layers	Heads	advmod	clf	dep	case	flat	mark	acl:relcl	cc	nsubj
2	2	0.90	0.87	0.64	0.99	0.85	0.97	0.71	0.97	0.75
	4	0.90	0.82	0.63	0.99	0.86	0.94	0.75	0.99	0.72
	6	0.91	0.89	0.66	0.98	0.87	0.96	0.75	0.96	0.72
	8	0.90	0.83	0.62	0.98	0.86	0.90	0.41	0.97	0.69
3	2	0.90	0.88	0.64	0.98	**0.00**	0.93	0.60	0.96	0.78
	4	0.91	0.86	0.64	0.98	0.86	0.94	0.45	0.96	0.71
	6	0.90	0.88	0.66	0.98	0.77	0.93	0.41	0.96	0.72
	8	0.91	0.90	0.66	0.99	0.86	0.93	0.46	0.96	0.74
4	2	0.89	0.68	0.64	0.97	**0.00**	0.94	0.52	0.84	0.74
	4	0.90	0.66	0.65	0.99	0.77	0.94	0.45	0.85	0.73
	6	0.91	0.69	0.68	0.99	0.67	0.97	0.40	0.85	0.77
	8	0.90	**0.00**	0.64	0.99	0.80	0.94	0.45	0.96	0.74
5	2	0.90	**0.00**	**0.00**	0.97	0.55	0.93	0.42	0.85	0.78
	4	0.90	**0.00**	**0.00**	0.98	0.77	0.96	0.68	0.82	0.79
	6	0.90	**0.00**	**0.00**	0.97	0.67	0.93	0.44	0.81	0.72
	8	0.89	**0.00**	**0.00**	0.99	0.48	0.96	0.43	0.86	0.73
6	2	0.83	**0.00**	**0.00**	0.94	**0.00**	0.91	**0.00**	0.83	0.65
	4	0.86	**0.00**	**0.00**	0.95	**0.00**	0.97	**0.00**	0.78	0.65
	6	0.84	**0.00**	**0.00**	0.94	**0.00**	0.93	**0.00**	0.79	0.67
	8	0.86	**0.00**	**0.00**	0.96	**0.00**	0.93	0.37	0.85	0.63

6. What Happens to Syntactic Features

6.1. Representational Similarity Analysis

Representational Similarity Analysis (RSA) is a technique used to analyze the similarity between different representation spaces of neural networks. Inspired by the work of Merchant et al. [38], RSA uses *n* examples to build two sets of comparable representations between neural networks. The representations are then transformed into a similarity matrix, and the Pearson correlation between the upper triangles of the similarity matrix is used to obtain the final similarity score between the representation spaces. We select the source sentences corresponding to the prior 300 low-quality translations and use them as the input stimulus for our analysis. The stimulus consists of groups of sentences, where each group is defined by a specific type of dependency relation. For example, if the current dependency relation is *x*, all source sentences of low-quality translations containing *x* are grouped together to form one stimulus group. To provide an example, consider the dependency relation "*obl:agent*" (oblique agent); all source sentences from the 300 low-quality translations that contain the "*obl:agent*" (oblique agent) relation are grouped together. Similarly, for the dependency relation "*nsubj:pass*" (nominal subject in a passive construction), all source sentences containing this relation are grouped together. BERT representations are extracted from both the baseline model and the SGB engines (e.g., baseline vs. SGBC) for each stimulus group, allowing us to compare the representation spaces of the different models. Cosine similarity is used as the kernel to compute the similar-

Table 8

Top-10 dependency relations with the highest GAT F1-score across various source language sentences, alongside corresponding changes in translation quality as measured by QE scores from different MT engines.

Dependency Relation	GAT F1-score	Baseline QE score	SGBC QE score	SGBD QE score
Zh				
mark	0.986	0.424	0.510	0.529
cc	0.984	0.436	0.513	0.512
conj	0.970	0.435	0.521	0.518
nummod	0.965	0.429	0.514	0.522
root	0.955	0.426	0.514	0.523
cop	0.945	0.426	0.520	0.511
det	0.935	0.438	0.530	0.528
case	0.934	0.428	0.511	0.526
nmod	0.933	0.429	0.509	0.523
amod	0.927	0.435	0.528	0.520
Ru				
det	0.990	0.697	0.747	0.746
root	0.987	0.700	0.748	0.750
amod	0.982	0.707	0.753	0.752
case	0.978	0.702	0.748	0.760
aux:pass	0.974	0.718	0.749	0.760
cop	0.971	0.720	0.774	0.781
advmod	0.934	0.704	0.750	0.747
cc	0.930	0.698	0.751	0.748
flat:foreign	0.921	0.678	0.701	0.727
obl	0.900	0.701	0.749	0.749
De				
case	0.992	0.504	0.568	0.574
cc	0.987	0.509	0.565	0.561
det	0.987	0.504	0.565	0.571
mark	0.981	0.511	0.561	0.570
advmod	0.932	0.506	0.573	0.582
root	0.931	0.503	0.570	0.574
aux:pass	0.927	0.498	0.576	0.556
amod	0.913	0.507	0.567	0.571
flat:name	0.876	0.505	0.551	0.565
aux	0.868	0.520	0.586	0.597

ity between the BERT representations of the stimulus groups, helping us understand how the addition of syntactic knowledge affects the representation space of BERT.

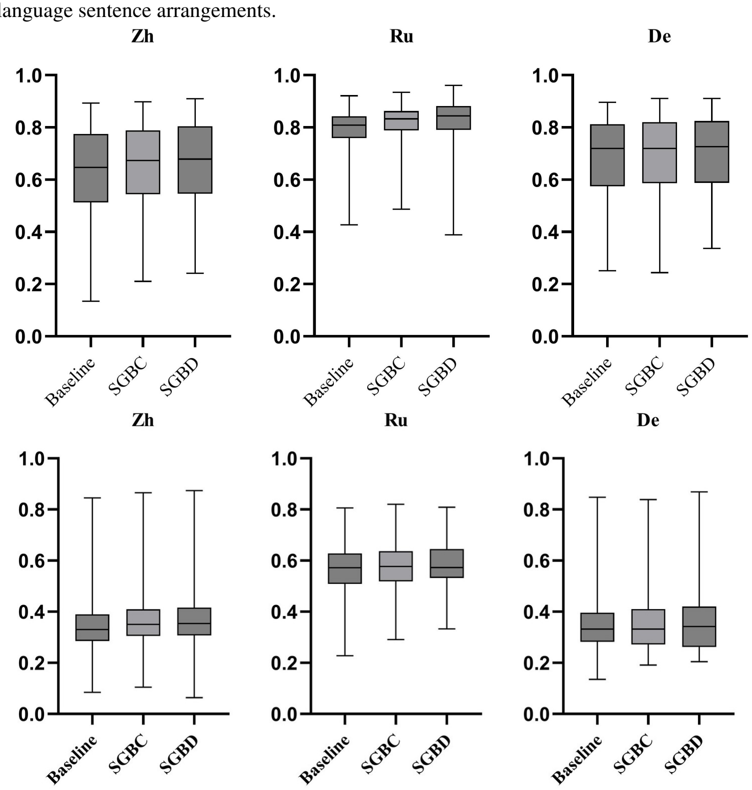
Table 9
Top-5 syntactic labels with the highest F1 scores for GAT predictions for each language, along with the BERT layers where the lowest RSA scores are observed.

Zh					
Relation	GAT F1-Score	RSA Score	BERT Layer	RSA Score*	BERT Layer
mark (marker)	0.986	0.418	5	0.407	3
cc (coordinating conjunction)	0.984	0.274	4	0.354	5
conj (conjunct)	0.970	0.380	5	0.340	4
nummod (numeric modifier)	0.965	0.274	4	0.237	3
root (root)	0.955	0.216	4	0.390	4
Ru					
Relation	GAT F1-Score	RSA Score	BERT Layer	RSA Score*	BERT Layer
det (determiner)	0.990	0.426	4	0.408	3
root(root)	0.987	0.466	3	0.504	3
amod (adjectival modifier)	0.982	0.444	3	0.391	4
case (case marking)	0.978	0.462	4	0.413	4
aux:pass (passive auxiliary)	0.974	0.357	3	0.327	3
De					
Relation	GAT F1-Score	RSA Score	BERT Layer	RSA Score*	BERT Layer
case (case marking)	0.992	0.686	5	0.759	2
cc (coordinating conjunction)	0.987	0.591	6	0.741	6
det (determiner)	0.987	0.584	8	0.817	6
mark (marker)	0.981	0.676	6	0.769	6
advmod (adverbial modifier)	0.932	0.733	6	0.774	8

Table 9 lists partial results from an RSA analysis comparing Baseline BERT and SGB models based on syntactic prediction scores by GAT (full results are provided in Appendix A). The analysis shows that the lowest RSA scores mainly occur in the lower and middle layers of BERT, regardless of whether the model is used in the SGBC or SGBD engine. Specifically, when GAT achieves high F1 scores for a particular dependency relation, the representations of sentences containing this relation typically undergo significant changes in the lower and middle layers of BERT. These changes are most pronounced in layers 3-5 for Chinese and Russian, and in layers 5-8 for German. This suggests that the syntactic structure represented through graphs influences BERT's reanalysis of input sentences, leading to a syntactic reconstruction of the input sentence. Also, the lower and middle layers of BERT are particularly sensitive to modifications in modeling both shallow and deep syntactic structures. In contrast, layers 9-12 are primarily involved in processing abstract semantic information and are task-oriented. However, the RSA scores in these layers do not consistently reach 0.8 or higher (see detailed results in Appendix A), indicating that changes in the syntactic representation in the lower layers can also affect the processing of deep linguistic information in the upper layers. These findings further explain why integrating syntactic structures represented through graphs can help BERT reconstruct the structure of input sentences, leading to a more accurate representation of source language sentences and, consequently, improved translation quality.

*RSA scores for representations from the baseline and SGBD models for comparison.

Fig. 3. The box plot distribution of QE scores for translations in three MT directions, contrasting translations from ordered (above) versus disordered (below) source language sentence arrangements.



6.2. Randomization of Word Order and Disruption of Syntactic Graphs

The impact of BERT and graph-based syntactic knowledge on enhancing translation quality presents an area for further investigation, particularly concerning the robustness of syntactic knowledge. This raises questions about the relative contributions of BERT versus graph-based syntactic knowledge to translation quality and the potential limitations of the proposed MT engines. To address these questions, the study involves altering the word order in source language sentences from each language in the PUD corpus. For example, the sentence "A B C D E F" is transformed into a randomized sequence like "C B A D F E". Both the baseline and SGB engines are then tasked with translating these modified sentences. The translations are subsequently reassessed by Transquest QE model, which compares the translations of the shuffled sentences against those of the original, orderly sentences. This comparison provides insights into the adaptability and efficacy of syntactic knowledge in translation.

To further validate the importance of accurate syntactic knowledge in enhancing the performance of the proposed MT engines, we conduct an additional experiment where we intentionally introduce incorrect syntactic graphs. In this experiment, we replace the parsers for Chinese, Russian, and German with an English parser to extract the syntactic structures of these three source languages. This deliberate introduction of incorrect syntactic graphs is then applied to the SGBC and SGBD engines. The goal is to observe how the performance of these models is affected when provided with inaccurate syntatic information.

As shown in Figure 3, scrambled word sequences in source sentences cause a significant decrease in translation quality for both baseline and SGB engines across all MT directions. Integrating GAT into the encoder or providing explicit syntactic knowledge to the decoder does not guarantee a substantial improvement in translation quality. It is unrealistic to expect the median QE scores in the box plots to increase from below 0.4 to 0.7. This finding suggests that BERT plays a more crucial role in forming representations of source sentences and influencing translation quality in this hybrid approach. The scrambling of input sentence order, which leads to a loss of syntactic information, indicates that while SGB engines, enhanced by graph-based syntactic knowledge, can mitigate some of the negative

Table 10
Comparison of QE scores with correct and incorrect syntactic graphs for SGBC and SGBD engines and the percentage decrease in QE scores.

	SGBC Model			SGBD Model		
	Correct Graph	**Incorrect Graph**	**%**	**Correct Graph**	**Incorrect Graph**	**%**
Zh→En	0.682	0.510	-25.21%	0.726	0.558	-23.14%
Ru→En	0.757	0.621	-17.96%	0.770	0.618	-19.74%
De→En	0.669	0.545	-18.53 %	0.720	0.599	-16.80%

effects, they are still unable to interpret and comprehend the correct semantics of jumbled sentences as effectively as humans.

The table 10 provides a detailed comparison of QE scores for the SGBC and SGBD models when using correct versus incorrect syntactic graphs. In all translation directions, the introduction of incorrect syntactic graphs results in a significant decrease in QE scores for both the SGBC and SGBD models, with reductions exceeding 15% in all cases. The largest decrease in QE scores is observed for the Zh→En direction, where both the SGBC and SGBD engines experience a decline of over 20%. Conversely, the smallest decrease is noted for the De→En direction, with reductions of 18.53% and 16.80% for the SGBC and SGBD models, respectively. This difference may be attributed to the closer linguistic proximity between German and English, which results in fewer detrimental effects from the parser's incorrect syntactic structures. In contrast, the lower similarity between Chinese and English means that incorrect syntactic structures have a more significant adverse impact on the SGBC and SGBD engines. Despite the use of incorrect syntactic graphs, the SGBD engine still demonstrates a greater likelihood of maintaining higher translation performance, indicating that the SGBD model benefits more from syntactic graphs, even when they are incorrect.

These findings highlight that accurate syntactic graphs are not only beneficial but essential for maintaining high-quality translations, as inaccuracies in these graphs significantly affect the performance of MT systems. However, the performance degradation is not as severe as when input sentences are randomized. This further suggests that in the SGB models, BERT plays a dominant role, and while incorrect syntactic graphs do harm performance, the impact is more severe when the input errors are so significant that even BERT cannot effectively process them.

7. What Happens when using Another Pre-trained Model

The central focus of this investigation is to determine whether the proposed use of syntactic knowledge on graphs continues to benefit alternative pre-trained models, thereby further improving translation quality. XLM-Roberta-large [39] replaces BERT in all three MT scenarios. To distinguish from earlier versions, MT engines incorporating XLM-Roberta-large are labeled Baseline-X, SGBC-X, and SGBD-X. The Chinese and Russian (Zh→En and Ru→En) MT engines utilize the UNPC corpus, whereas the German (De→En) engines employ Europarl. Each training set comprises 0.1M sentence pairs, with validation and test sets featuring 6K parallel sentence pairs each. Specifications include word embeddings of 1024, a learning rate (excluding GAT) of 2e-5, a GAT learning rate of 5e-5, a GAT dropout rate of 0.1, a batch size of 8, and the Adam optimizer. Training is conducted on an RTX 3090 GPU.

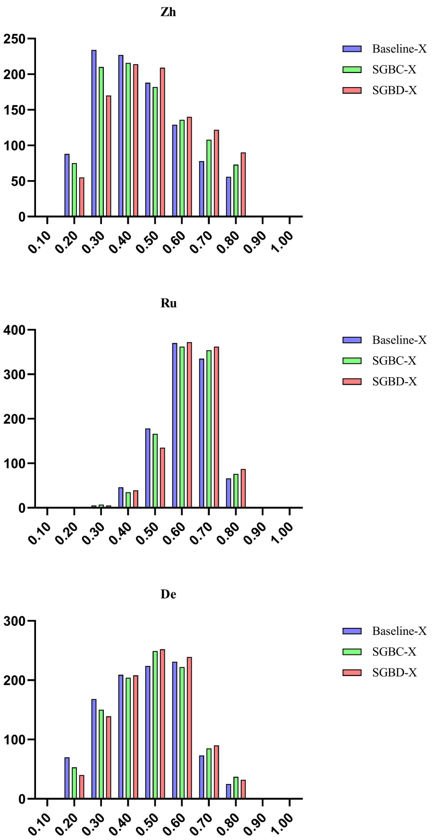
Table 11 demonstrates that both SGB engines consistently achieve higher BLEU scores than Baseline-X across various MT directions, with the SGBD-X engine surpassing the SGBC-X engine in every scenario through superior BLEU scores. Furthermore, Figure 4 illustrates the QE scores for translations within the PUD corpus for each engine. Baseline-X yields the highest number of translations with QE scores in the 0.2, 0.3, and 0.4 intervals along the X-axis for both Zh and De, a pattern also observed in Ru at the 0.4 and 0.5 intervals. A notable shift in the distribution of translations for Zh and De occurs at the 0.5 mark on the X-axis, where SGBC-X and SGBD-X engines begin to outperform Baseline-X, a trend that persists up to the 0.8 interval. In Ru, the SGB engines similarly exhibit a higher count of translations with elevated QE scores than the Baseline engine at the 0.7 and 0.8 intervals on the X-axis.

Table 11

BLEU scores in different MT directions for the MT engines that replaced BERT with XLM-Roberta-large.

	Baseline-X	SGBC-X	SGBD-X
Zh→En	26.28	26.59	**27.13**
Ru→En	23.62	23.86	**24.01**
De→En	22.93	23.28	**24.46**

Fig. 4. Distribution of QE scores for the MT engines after replacing BERT. The Y-axis shows the number of sentences, while the X-axis shows the range of scores for the QE scores of the translations.



The demonstrated efficacy of our method with XLM-Roberta indicates its applicability beyond a single pre-trained model, extending to encoder-based pre-trained models in general. This suggests that our approach is not confined to a specific architecture. However, adapting our method to other pre-trained models, such as GPT or T5, presents distinct challenges. These models are primarily decoder-based and sequence-to-sequence models, respectively, which differ significantly from the encoder-based architecture of XLM-Roberta. Integrating syntactic knowledge into these models may necessitate alternative strategies, such as modifying the input format or adjusting the attention mechanisms. Despite these challenges, the potential benefits of incorporating syntactic knowledge into a broader range of pre-trained models are substantial, as it can lead to more accurate and contextually appropriate translations. Future research will explore these adaptations to further enhance the robustness and applicability of our method.

8. Conclusions

This study explores the integration of syntactic knowledge into MT, particularly focusing on the evaluation of BERT and GAT. Two SGB engines are introduced for translating from Chinese to English (Zh→En), Russian to English (Ru→En), and German to English (De→En), and by leveraging GAT, the representation capabilities of the BERT encoder are enhanced, and the decoder's understanding of source language sentence structures is improved. The results demonstrate that the proposed SGB engines outperform baseline models in terms of BLEU scores, COMET QE scores, and TransQuest QE scores, indicating significant improvements in translation accuracy and robustness. When translating the PUD corpus, paired t-tests confirm a statistically significant difference in TransQuest QE scores, further validating the substantial improvement in translation quality. We find that the SGB engines, which incorporate graph-structured knowledge, are more adept at recognizing the structural nuances of source language sentences, thereby enhancing translation quality, for instance, the SGB engines achieve notably higher QE scores for Chinese sentences with the "*obl:agent*" (oblique agent) structure compared to baseline engines. The study also evaluate the syntactic dependency learning performance of GAT using the PUD corpus, and the results show that the learning efficiency improves with an increase in attention heads, though the optimal configuration varies across languages, however, excessive model complexity, beyond two layers, tends to degrade prediction performance, highlighting the importance of balancing complexity and predictive effectiveness. Additionally, the study investigate the impact of GAT's dependency prediction on translation quality, and the findings indicate that accurate predictions by GAT for certain dependency relations can lead to better translations of source sentences containing those dependencies. RSA experiments further reveal that although GAT is not initially part of BERT, its integration allows specific BERT layers to re-evaluate the syntactic structure of source sentences through fine-tuning, and this effect is particularly pronounced in the early and mid-layers of BERT across different languages. Experiments on word order randomization and parser replacement emphasize the critical role of syntactic information embedded in graph structures in enhancing translation quality. We also show that our approach is not limited to BERT; similar performance improvements have been achieved with XLM-Roberta as an alternative model. In summary, this study underscores the significant potential of combining syntactic knowledge embedded in graph structures with language models like BERT and XLM-Roberta to enhance MT, and the findings support further research into these synergies to improve translation accuracy and interpretability with better knowledge about syntax.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser and I. Polosukhin, Attention is all you need, in: *Proc Advances in Neural Information Processing Systems*, 2017.

[2] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, Minnesota, 2019, pp. 4171–4186. doi:10.18653/v1/N19-1423. https://aclanthology.org/N19-1423.

[3] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever and D. Amodei, Language models are few-shot learners, *Advances in neural information processing systems* **33** (2020), 1877–1901.

[4] A. Rogers, O. Kovaleva and A. Rumshisky, A Primer in BERTology: What We Know About How BERT Works, *Transactions of the Association for Computational Linguistics* **8** (2020), 842–866.

[5] J. Bai, Y. Wang, Y. Chen, Y. Yang, J. Bai, J. Yu and Y. Tong, Syntax-BERT: Improving Pre-trained Transformers with Syntax Trees, in: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, P. Merlo, J. Tiedemann and R. Tsarfaty, eds, Association for Computational Linguistics, Online, 2021, pp. 3011–3020. doi:10.18653/v1/2021.eacl-main.262. https://aclanthology.org/2021.eacl-main.262/.

[6] J. Nivre, M.-C. de Marneffe, F. Ginter, Y. Goldberg, J. Hajič, C.D. Manning, R. McDonald, S. Petrov, S. Pyysalo, N. Silveira, R. Tsarfaty and D. Zeman, Universal Dependencies v1: A Multilingual Treebank Collection, in: *Proc LREC 2016*, Portorož, Slovenia, 2016.

[7] D. Tilwani, R. Venkataramanan and A.P. Sheth, Neurosymbolic AI Approach to Attribution in Large Language Models, *IEEE Intelligent Systems* **39**(6) (2024), 10–17. doi:10.1109/MIS.2024.3477108.

[8] T.R. Besold, A. d'Avila Garcez, S. Bader, H. Bowman, P. Domingos, P. Hitzler, K.-U. Kühnberger, L.C. Lamb, P.M.V. Lima, L. de Penning et al., Neural-symbolic learning and reasoning: A survey and interpretation 1, in: *Neuro-Symbolic Artificial Intelligence: The State of the Art*, IOS press, 2021, pp. 1–51.

[9] A. Conneau, G. Kruszewski, G. Lample, L. Barrault and M. Baroni, What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties, in: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 2126–2136. doi:10.18653/v1/P18-1198. https://www.aclweb.org/anthology/P18-1198.

[10] S. Egea Gómez, E. McGill and H. Saggion, Syntax-aware Transformers for Neural Machine Translation: The Case of Text to Sign Gloss Translation, in: *Proceedings of the 14th Workshop on Building and Using Comparable Corpora (BUCC 2021)*, INCOMA Ltd., Online (Virtual Mode), 2021, pp. 18–27. https://aclanthology.org/2021.bucc-1.4.

[11] R. Peng, T. Hao and Y. Fang, Syntax-aware neural machine translation directed by syntactic dependency degree, *Neural Computing and Applications* **33**(23) (2021), 16609–16625.

[12] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò and Y. Bengio, Graph Attention Networks, *arXiv preprint arXiv:1710.10903* (2017).

[13] L. Huang, X. Sun, S. Li, L. Zhang and H. Wang, Syntax-aware graph attention network for aspect-level sentiment classification, in: *Proceedings of the 28th international conference on computational linguistics*, 2020, pp. 799–810.

[14] G. Li, C. Zheng, M. Li and H. Wang, Automatic Requirements Classification Based on Graph Attention Network, *IEEE Access* **10** (2022), 30080–30090.

[15] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: *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)*, Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. doi:10.18653/v1/N19-1423. https://aclanthology.org/N19-1423.

[16] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer and V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, *arXiv preprint arXiv:1907.11692* (2019).

[17] K. Imamura and E. Sumita, Recycling a pre-trained BERT encoder for neural machine translation, in: *Proceedings of the 3rd Workshop on Neural Generation and Translation*, 2019, pp. 23–31.

[18] J. Yang, M. Wang, H. Zhou, C. Zhao, W. Zhang, Y. Yu and L. Li, Towards making the most of bert in neural machine translation, in: *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34, 2020, pp. 9378–9385.

[19] J. Zhu, Y. Xia, L. Wu, D. He, T. Qin, W. Zhou, H. Li and T.-Y. Liu, Incorporating bert into neural machine translation, *arXiv preprint arXiv:2002.06823* (2020).

[20] A. Currey and K. Heafield, Incorporating Source Syntax into Transformer-Based Neural Machine Translation, in: *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, Association for Computational Linguistics, Florence, Italy, 2019, pp. 24–33. doi:10.18653/v1/W19-5203. https://aclanthology.org/W19-5203.

[21] Z. Zhang, Y. Wu, J. Zhou, S. Duan, H. Zhao and R. Wang, SG-Net: Syntax guided transformer for language representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[22] C. McDonald and D. Chiang, Syntax-Based Attention Masking for Neural Machine Translation, in: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, Association for Computational Linguistics, Online, 2021, pp. 47–52. doi:10.18653/v1/2021.naacl-srw.7. https://aclanthology.org/2021.naacl-srw.7.

[23] S. Jain and B.C. Wallace, Attention is not Explanation, in: *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)*, Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 3543–3556. doi:10.18653/v1/N19-1357. https://aclanthology.org/N19-1357.

[24] S. Wiegreffe and Y. Pinter, Attention is not not Explanation, in: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 11–20. doi:10.18653/v1/D19-1002. https://aclanthology.org/D19-1002.

[25] K. Wang, W. Shen, Y. Yang, X. Quan and R. Wang, Relational Graph Attention Network for Aspect-based Sentiment Analysis, in: *Annual Meeting of the Association for Computational Linguistics*, 2020. https://api.semanticscholar.org/CorpusID:216553425.

[26] X. Chen, Y. Wang, J. He, Y. Du, S. Hassoun, X. Xu and L.-P. Liu, Graph Generative Pre-trained Transformer, 2025. https://arxiv.org/abs/2501.01073.

[27] L. Song, D. Gildea, Y. Zhang, Z. Wang and J. Su, Semantic neural machine translation using AMR, *Transactions of the Association for Computational Linguistics* **7** (2019), 19–31.

[28] Y. Yin., F. Meng., J. Su., C. Zhou., Z. Yang., J. Zhou. and J. Luo, A Novel Graph-based Multi-modal Fusion Encoder for Neural Machine Translation, in: *Annual Meeting of the Association for Computational Linguistics*, 2020. https://api.semanticscholar.org/CorpusID:220045417.

[29] M. Chen, W. Wu, Y. Zhang and Z. Zhou, Combining Adversarial Training and Relational Graph Attention Network for Aspect-Based Sentiment Analysis with BERT, in: *2021 14th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, IEEE, 2021, pp. 1–6.

[30] X. Zhou, T. Zhang, C. Cheng and S. Song, Dynamic multichannel fusion mechanism based on a graph attention network and BERT for aspect-based sentiment classification, *Applied Intelligence* (2022), 1–14.

[31] H. He and J.D. Choi, The Stem Cell Hypothesis: Dilemma behind Multi-Task Learning with Transformer Encoders, in: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 5555–5577. https://aclanthology.org/2021.emnlp-main.451.

[32] K. Papineni, S. Roukos, T. Ward and W.-J. Zhu, BLEU: a method for automatic evaluation of machine translation, Technical Report, RC22176 (W0109-022), IBM Thomas J. Watson Research Center, 2001.

[33] R. Rei, C. Stewart, A.C. Farinha and A. Lavie, COMET: A Neural Framework for MT Evaluation, in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Online, 2020, pp. 2685–2702. doi:10.18653/v1/2020.emnlp-main.213. https://aclanthology.org/2020.emnlp-main.213.

[34] T. Ranasinghe, C. Orasan and R. Mitkov, TransQuest at WMT2020: Sentence-Level Direct Assessment, in: *Proc WMT*, Association for Computational Linguistics, online, 2020, pp. 1127–1136.

[35] T. Kocmi, C. Federmann, R. Grundkiewicz, M. Junczys-Dowmunt, H. Matsushita and A. Menezes, To Ship or Not to Ship: An Extensive Evaluation of Automatic Metrics for Machine Translation, in: *Proceedings of the Sixth Conference on Machine Translation*, Association for Computational Linguistics, Online, 2021, pp. 478–494. https://aclanthology.org/2021.wmt-1.57.

[36] C. Callison-Burch, M. Osborne and P. Koehn, Re-evaluating the Role of Bleu in Machine Translation Research, in: *11th Conference of the European Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, Trento, Italy, 2006, pp. 249–256. https://aclanthology.org/E06-1032.

[37] J. Novikova, O. Dušek, A.C. Curry and V. Rieser, Why we need new evaluation metrics for NLG, *arXiv preprint arXiv:1707.06875* (2017).

[38] A. Merchant, E. Rahimtoroghi, E. Pavlick and I. Tenney, What happens to bert embeddings during fine-tuning?, *arXiv preprint arXiv:2004.14448* (2020).

[39] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer and V. Stoyanov, Unsupervised Cross-lingual Representation Learning at Scale, in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Online, 2020, pp. 8440–8451. doi:10.18653/v1/2020.acl-main.747. https://aclanthology.org/2020.acl-main.747.

[40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser and I. Polosukhin, Attention is all you need, *Advances in neural information processing systems* **30** (2017).

[41] R. Yan, J. Li, X. Su, X. Wang and G. Gao, Boosting the Transformer with the BERT Supervision in Low-Resource Machine Translation, *Applied Sciences* **12**(14) (2022), 7195.

Appendix A. Appendix A. Representational Similarity Analysis

Table 12 to Table 17 show the RSA tests of the dependency relations in the given groups of BERT in the Baseline, SGBC and SGBD models for different languages in 12 layers (L).

Table 12
Comparison of the representation from BERT in the baseline and SGBC model when tested on Chinese sentences containing target dependency.

	Baseline vs SGBC											
Zh Relations	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12
acl:relcl	0.891	0.733	0.877	0.239	0.452	0.656	0.506	0.712	0.587	0.623	0.442	0.424
advcl	0.875	0.734	0.203	0.479	0.378	0.685	0.462	0.693	0.664	0.668	0.517	0.522
advmod	0.856	0.794	0.878	0.292	0.528	0.781	0.576	0.733	0.638	0.697	0.514	0.512
amod	0.818	0.705	0.962	0.632	0.483	0.662	0.379	0.580	0.398	0.587	0.341	0.335
appos	0.908	0.770	0.901	0.411	0.429	0.694	0.519	0.653	0.599	0.677	0.483	0.485
aux	0.873	0.803	0.954	0.449	0.600	0.760	0.551	0.718	0.614	0.683	0.476	0.441
aux:pass	0.872	0.637	0.972	0.666	0.663	0.504	0.468	0.672	0.540	0.748	0.394	0.300
case	0.880	0.743	0.893	0.576	0.529	0.677	0.514	0.699	0.588	0.649	0.550	0.599
case:loc	0.898	0.744	0.216	0.322	0.477	0.752	0.553	0.762	0.684	0.669	0.509	0.587
cc	0.915	0.782	0.498	0.274	0.442	0.702	0.620	0.660	0.667	0.710	0.588	0.557
ccomp	0.847	0.767	0.808	0.403	0.442	0.783	0.572	0.757	0.684	0.752	0.503	0.570
clf	0.857	0.753	0.840	0.219	0.560	0.673	0.543	0.698	0.606	0.662	0.420	0.501
compound	0.877	0.748	0.871	0.402	0.483	0.727	0.545	0.692	0.615	0.650	0.506	0.684
conj	0.910	0.770	0.479	0.396	0.380	0.706	0.604	0.651	0.664	0.701	0.571	0.566
cop	0.898	0.785	0.480	0.238	0.484	0.743	0.578	0.722	0.720	0.738	0.634	0.613
csubj	0.889	0.895	0.283	0.467	0.623	0.751	0.563	0.761	0.814	0.799	0.557	0.567
dep	0.868	0.798	0.599	0.386	0.584	0.777	0.552	0.703	0.708	0.751	0.447	0.428
det	0.860	0.753	0.937	0.386	0.414	0.721	0.535	0.707	0.573	0.677	0.572	0.511
discourse:sp	0.898	0.810	0.961	0.855	0.784	0.804	0.635	0.802	0.638	0.747	0.627	0.615
flat	0.884	0.858	0.277	0.220	0.408	0.776	0.364	0.607	0.511	0.731	0.542	0.644
flat:name	0.868	0.769	0.330	0.285	0.579	0.594	0.644	0.689	0.594	0.643	0.374	0.409
iobj	0.674	0.478	0.427	0.798	0.382	0.679	0.635	0.701	0.719	0.414	0.289	0.391
mark	0.880	0.705	0.596	0.478	0.418	0.749	0.598	0.722	0.682	0.683	0.467	0.432
mark:adv	0.992	0.936	0.961	0.993	0.698	0.999	0.993	0.984	0.973	0.833	0.999	0.994
mark:prt	0.847	0.741	0.249	0.639	0.354	0.703	0.560	0.697	0.601	0.697	0.644	0.727
mark:relcl	0.889	0.771	0.859	0.545	0.418	0.674	0.484	0.686	0.607	0.655	0.484	0.494
nmod	0.882	0.751	0.870	0.584	0.566	0.668	0.485	0.675	0.579	0.620	0.569	0.593
nsubj	0.863	0.788	0.874	0.437	0.555	0.751	0.538	0.725	0.619	0.691	0.532	0.515
nsubj:pass	0.869	0.729	0.979	0.664	0.690	0.480	0.649	0.728	0.589	0.754	0.531	0.505
nummod	0.870	0.785	0.380	0.274	0.560	0.691	0.519	0.696	0.649	0.697	0.459	0.512
obj	0.873	0.792	0.881	0.469	0.507	0.720	0.577	0.713	0.639	0.683	0.507	0.493
obl	0.881	0.747	0.898	0.491	0.514	0.670	0.498	0.698	0.619	0.602	0.514	0.504
obl:agent	0.956	0.922	0.675	0.753	0.633	0.782	0.900	0.904	0.812	0.764	0.657	0.456
obl:patient	0.840	0.767	0.688	0.580	0.770	0.633	0.737	0.730	0.408	0.560	0.416	0.559
obl:tmod	0.867	0.763	0.391	0.200	0.357	0.817	0.587	0.739	0.697	0.697	0.294	0.403
xcomp	0.831	0.790	0.776	0.519	0.474	0.769	0.682	0.769	0.564	0.400	0.577	0.322
root	0.863	0.791	0.893	0.216	0.541	0.757	0.561	0.741	0.638	0.704	0.503	0.494

Table 13
Comparison of the representation from BERT in the baseline and SGBD model when tested on Chinese sentences containing target dependency.

	Baseline vs SGBD											
Zh Relations	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12
acl:relcl	0.902	0.726	0.267	0.237	0.231	0.574	0.339	0.554	0.477	0.554	0.425	0.444
advcl	0.893	0.747	0.278	0.425	0.251	0.554	0.425	0.522	0.454	0.507	0.386	0.411
advmod	0.874	0.768	0.262	0.260	0.401	0.664	0.409	0.492	0.493	0.581	0.448	0.548
amod	0.813	0.688	0.569	0.476	0.411	0.498	0.217	0.364	0.289	0.411	0.260	0.416
appos	0.905	0.779	0.463	0.454	0.357	0.657	0.432	0.455	0.457	0.610	0.466	0.449
aux	0.884	0.770	0.369	0.291	0.400	0.622	0.443	0.512	0.483	0.559	0.458	0.489
aux:pass	0.915	0.692	0.463	0.591	0.728	0.656	0.473	0.355	0.360	0.676	0.386	0.531
case	0.884	0.736	0.678	0.456	0.272	0.573	0.363	0.444	0.435	0.526	0.424	0.492
case:loc	0.909	0.771	0.372	0.297	0.299	0.627	0.391	0.491	0.477	0.489	0.379	0.475
cc	0.885	0.780	0.607	0.362	0.354	0.540	0.360	0.410	0.535	0.660	0.496	0.448
ccomp	0.886	0.725	0.355	0.249	0.400	0.666	0.381	0.459	0.400	0.482	0.401	0.449
clf	0.881	0.725	0.635	0.421	0.378	0.597	0.392	0.500	0.490	0.540	0.371	0.425
compound	0.888	0.750	0.484	0.398	0.308	0.639	0.388	0.447	0.438	0.550	0.443	0.434
conj	0.887	0.777	0.599	0.340	0.452	0.552	0.346	0.405	0.515	0.654	0.494	0.555
cop	0.894	0.772	0.431	0.434	0.272	0.638	0.455	0.524	0.510	0.498	0.480	0.393
csubj	0.913	0.820	0.748	0.591	0.483	0.831	0.347	0.655	0.563	0.643	0.608	0.689
dep	0.881	0.819	0.523	0.491	0.420	0.627	0.436	0.470	0.513	0.566	0.395	0.419
det	0.855	0.713	0.269	0.217	0.285	0.581	0.355	0.517	0.507	0.578	0.384	0.406
discourse:sp	0.922	0.747	0.234	0.603	0.614	0.705	0.409	0.577	0.640	0.760	0.578	0.434
flat	0.891	0.857	0.342	0.445	0.257	0.585	0.342	0.457	0.400	0.682	0.442	0.486
flat:name	0.897	0.776	0.282	0.419	0.274	0.481	0.385	0.362	0.395	0.482	0.309	0.455
iobj	0.699	0.917	0.556	0.470	0.357	0.669	0.695	0.560	0.598	0.467	0.386	0.558
mark	0.901	0.723	0.407	0.408	0.434	0.641	0.684	0.469	0.452	0.428	0.482	0.417
mark:adv	0.970	0.994	0.883	0.992	0.975	0.999	0.993	0.988	0.657	0.716	0.984	0.958
mark:prt	0.883	0.800	0.759	0.527	0.240	0.584	0.346	0.544	0.451	0.482	0.377	0.446
mark:relcl	0.892	0.754	0.459	0.226	0.239	0.575	0.352	0.520	0.478	0.551	0.452	0.520
nmod	0.874	0.737	0.552	0.424	0.298	0.595	0.353	0.422	0.439	0.510	0.395	0.495
nsubj	0.879	0.777	0.508	0.427	0.436	0.662	0.412	0.501	0.492	0.560	0.462	0.554
nsubj:pass	0.909	0.755	0.508	0.601	0.765	0.553	0.552	0.504	0.488	0.678	0.389	0.524
nummod	0.886	0.790	0.237	0.371	0.384	0.606	0.375	0.467	0.490	0.575	0.434	0.533
obj	0.880	0.779	0.424	0.272	0.388	0.626	0.413	0.496	0.509	0.554	0.451	0.435
obl	0.907	0.717	0.585	0.430	0.218	0.575	0.366	0.503	0.515	0.570	0.480	0.430
obl:agent	0.953	0.864	0.920	0.860	0.374	0.635	0.496	0.706	0.687	0.768	0.653	0.639
obl:patient	0.822	0.789	0.654	0.720	0.604	0.673	0.502	0.540	0.345	0.586	0.480	0.530
obl:tmod	0.872	0.781	0.442	0.229	0.375	0.589	0.377	0.536	0.571	0.647	0.544	0.605
xcomp	0.900	0.747	0.220	0.330	0.347	0.692	0.468	0.497	0.505	0.576	0.465	0.433
root	0.878	0.781	0.413	0.390	0.431	0.669	0.433	0.525	0.511	0.583	0.480	0.460

Table 14
Comparison of the representation from BERT in the baseline and SGBC model when tested on Russian sentences containing target dependency.

	Baseline vs SGBC											
Ru Relations	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12
acl	0.824	0.424	0.392	0.625	0.555	0.738	0.646	0.618	0.571	0.644	0.641	0.559
acl:relcl	0.617	0.309	0.310	0.454	0.412	0.640	0.519	0.635	0.576	0.553	0.506	0.475
advcl	0.710	0.613	0.556	0.609	0.409	0.631	0.623	0.734	0.756	0.748	0.685	0.587
advmod	0.877	0.608	0.428	0.651	0.618	0.764	0.711	0.723	0.721	0.746	0.734	0.618
amod	0.855	0.572	0.444	0.635	0.576	0.731	0.668	0.694	0.693	0.731	0.722	0.597
appos	0.679	0.617	0.286	0.700	0.606	0.707	0.591	0.700	0.769	0.774	0.787	0.569
aux	0.627	0.590	0.504	0.445	0.556	0.527	0.303	0.690	0.768	0.571	0.431	0.396
aux:pass	0.699	0.528	0.357	0.706	0.644	0.730	0.586	0.632	0.605	0.691	0.742	0.560
case	0.856	0.574	0.572	0.462	0.591	0.756	0.694	0.725	0.721	0.740	0.733	0.624
cc	0.872	0.679	0.365	0.654	0.584	0.740	0.726	0.731	0.746	0.766	0.743	0.594
ccomp	0.600	0.566	0.320	0.568	0.561	0.714	0.716	0.806	0.835	0.792	0.778	0.700
compound	0.636	0.587	0.603	0.477	0.474	0.996	0.975	0.988	0.940	0.614	0.942	0.994
conj	0.821	0.663	0.355	0.641	0.595	0.744	0.738	0.739	0.751	0.753	0.743	0.585
cop	0.803	0.548	0.317	0.629	0.547	0.797	0.593	0.633	0.723	0.757	0.768	0.612
csubj	0.525	0.463	0.480	0.368	0.426	0.432	0.517	0.750	0.707	0.621	0.475	0.332
det	0.851	0.670	0.626	0.426	0.537	0.721	0.642	0.678	0.707	0.744	0.713	0.607
fixed	0.759	0.579	0.578	0.633	0.641	0.659	0.615	0.689	0.685	0.699	0.671	0.578
flat	0.665	0.404	0.514	0.572	0.565	0.608	0.484	0.666	0.677	0.627	0.593	0.424
flat:foreign	0.704	0.435	0.548	0.588	0.604	0.704	0.554	0.729	0.758	0.700	0.604	0.419
flat:name	0.703	0.533	0.442	0.596	0.636	0.748	0.629	0.658	0.639	0.599	0.596	0.555
iobj	0.629	0.474	0.553	0.685	0.606	0.659	0.603	0.719	0.697	0.655	0.673	0.556
mark	0.668	0.528	0.231	0.500	0.516	0.629	0.603	0.699	0.723	0.691	0.642	0.498
nmod	0.860	0.478	0.453	0.648	0.544	0.740	0.658	0.696	0.699	0.730	0.726	0.610
nsubj	0.820	0.584	0.466	0.687	0.567	0.732	0.685	0.718	0.719	0.738	0.729	0.596
nsubj:pass	0.711	0.580	0.336	0.723	0.561	0.711	0.575	0.614	0.618	0.708	0.732	0.610
nummod	0.575	0.624	0.270	0.515	0.610	0.689	0.526	0.669	0.618	0.591	0.562	0.445
nummod:gov	0.640	0.401	0.443	0.579	0.759	0.783	0.531	0.612	0.589	0.644	0.640	0.543
obj	0.756	0.542	0.483	0.661	0.506	0.691	0.641	0.683	0.645	0.675	0.674	0.535
obl	0.764	0.592	0.479	0.657	0.568	0.746	0.684	0.711	0.702	0.709	0.704	0.591
obl:agent	0.638	0.394	0.509	0.825	0.837	0.891	0.851	0.340	0.582	0.717	0.770	0.607
orphan	0.733	0.661	0.241	0.620	0.937	0.800	0.519	0.330	0.651	0.424	0.558	0.638
parataxis	0.825	0.629	0.391	0.598	0.659	0.786	0.714	0.723	0.683	0.670	0.621	0.680
xcomp	0.756	0.658	0.486	0.683	0.575	0.762	0.712	0.731	0.748	0.761	0.754	0.620
root	0.855	0.587	0.466	0.704	0.597	0.751	0.701	0.729	0.722	0.744	0.739	0.623

Table 15
Comparison of the representation from BERT in the baseline and SGBD model when tested on Russian sentences containing target dependency.

	Baseline vs SGBD											
Ru Relations	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12
acl	0.918	0.416	0.296	0.617	0.501	0.541	0.611	0.562	0.573	0.752	0.779	0.627
acl:relcl	0.505	0.299	0.292	0.484	0.402	0.474	0.606	0.643	0.628	0.739	0.744	0.651
advcl	0.585	0.541	0.489	0.508	0.505	0.562	0.665	0.676	0.692	0.747	0.804	0.686
advmod	0.931	0.442	0.509	0.608	0.613	0.646	0.731	0.720	0.710	0.830	0.846	0.666
amod	0.910	0.548	0.488	0.391	0.573	0.605	0.679	0.683	0.674	0.796	0.777	0.594
appos	0.574	0.331	0.303	0.614	0.432	0.517	0.618	0.715	0.740	0.787	0.731	0.581
aux	0.344	0.563	0.494	0.457	0.433	0.288	0.321	0.208	0.321	0.496	0.458	0.401
aux:pass	0.491	0.385	0.327	0.537	0.633	0.528	0.618	0.708	0.723	0.779	0.669	0.588
case	0.903	0.502	0.504	0.413	0.602	0.634	0.721	0.722	0.721	0.808	0.808	0.639
cc	0.943	0.392	0.417	0.624	0.590	0.626	0.705	0.719	0.724	0.822	0.826	0.646
ccomp	0.517	0.432	0.341	0.521	0.540	0.615	0.722	0.741	0.763	0.864	0.885	0.667
compound	0.699	0.777	0.474	0.902	0.365	0.902	0.991	0.764	0.996	0.988	0.954	0.955
conj	0.887	0.442	0.452	0.600	0.402	0.594	0.687	0.698	0.707	0.799	0.797	0.634
cop	0.651	0.415	0.545	0.536	0.586	0.722	0.729	0.668	0.761	0.833	0.758	0.583
csubj	0.450	0.488	0.473	0.417	0.496	0.229	0.480	0.603	0.676	0.544	0.468	0.393
det	0.895	0.446	0.408	0.675	0.616	0.673	0.759	0.755	0.774	0.848	0.854	0.742
fixed	0.666	0.415	0.516	0.673	0.605	0.599	0.698	0.644	0.683	0.800	0.748	0.603
flat	0.643	0.511	0.452	0.519	0.430	0.512	0.627	0.690	0.711	0.749	0.764	0.620
flat:foreign	0.638	0.520	0.387	0.542	0.523	0.545	0.621	0.683	0.728	0.772	0.786	0.677
flat:name	0.657	0.357	0.472	0.587	0.546	0.531	0.647	0.664	0.678	0.786	0.772	0.641
iobj	0.519	0.287	0.599	0.663	0.552	0.563	0.675	0.690	0.671	0.787	0.821	0.699
mark	0.537	0.367	0.274	0.288	0.515	0.591	0.711	0.714	0.724	0.817	0.842	0.704
nmod	0.911	0.379	0.462	0.596	0.573	0.611	0.686	0.682	0.677	0.787	0.771	0.594
nsubj	0.884	0.528	0.508	0.623	0.576	0.621	0.706	0.720	0.711	0.803	0.785	0.598
nsubj:pass	0.504	0.314	0.292	0.538	0.585	0.574	0.634	0.667	0.695	0.791	0.703	0.551
nummod	0.467	0.588	0.389	0.525	0.426	0.460	0.555	0.648	0.647	0.786	0.827	0.703
nummod:gov	0.570	0.536	0.331	0.686	0.523	0.595	0.682	0.689	0.726	0.825	0.815	0.639
obj	0.826	0.578	0.487	0.598	0.508	0.609	0.703	0.717	0.717	0.793	0.775	0.613
obl	0.797	0.520	0.507	0.619	0.572	0.618	0.715	0.720	0.721	0.780	0.756	0.579
obl:agent	0.806	0.454	0.250	0.742	0.744	0.607	0.472	0.633	0.640	0.694	0.479	0.299
orphan	0.301	0.240	0.524	0.420	0.750	0.709	0.579	0.427	0.419	0.322	0.228	0.243
parataxis	0.935	0.444	0.472	0.657	0.574	0.618	0.704	0.733	0.711	0.828	0.833	0.643
xcomp	0.611	0.587	0.593	0.565	0.569	0.665	0.729	0.765	0.754	0.830	0.808	0.648
root	0.901	0.506	0.504	0.637	0.612	0.649	0.720	0.724	0.716	0.806	0.787	0.612

Table 16

Comparison of the representation from BERT in the baseline and SGBC model when tested on German sentences containing target dependency.

	Baseline vs SGBC											
De Relations	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12
acl:relcl	0.696	0.776	0.763	0.690	0.601	0.604	0.670	0.627	0.621	0.629	0.613	0.629
advcl	0.640	0.776	0.781	0.716	0.645	0.506	0.632	0.602	0.572	0.514	0.527	0.575
advmod	0.775	0.819	0.841	0.800	0.737	0.733	0.750	0.793	0.747	0.790	0.750	0.748
amod	0.651	0.739	0.774	0.721	0.631	0.662	0.708	0.645	0.670	0.641	0.644	0.663
appos	0.695	0.766	0.814	0.751	0.682	0.664	0.702	0.667	0.671	0.678	0.680	0.674
aux	0.649	0.796	0.795	0.716	0.657	0.649	0.638	0.669	0.690	0.700	0.670	0.648
aux:pass	0.644	0.735	0.766	0.723	0.627	0.721	0.684	0.661	0.700	0.641	0.629	0.661
case	0.734	0.773	0.781	0.747	0.686	0.694	0.708	0.691	0.689	0.699	0.765	0.716
cc	0.613	0.721	0.719	0.675	0.602	0.591	0.631	0.592	0.606	0.595	0.595	0.598
ccomp	0.686	0.768	0.824	0.767	0.757	0.695	0.661	0.698	0.702	0.706	0.729	0.664
compound	0.687	0.780	0.785	0.733	0.661	0.649	0.721	0.700	0.688	0.653	0.654	0.691
compound:prt	0.671	0.760	0.763	0.662	0.703	0.694	0.730	0.680	0.717	0.735	0.681	0.790
conj	0.586	0.716	0.712	0.661	0.588	0.583	0.620	0.588	0.588	0.592	0.595	0.611
cop	0.679	0.794	0.808	0.772	0.649	0.690	0.753	0.735	0.730	0.670	0.695	0.726
csubj	0.686	0.730	0.860	0.809	0.770	0.853	0.798	0.660	0.824	0.860	0.714	0.737
cc:preconj	0.633	0.443	0.411	0.823	0.647	0.557	0.563	0.471	0.424	0.471	0.462	0.415
csubj:pass	0.868	0.742	0.886	0.904	0.492	0.937	0.977	0.731	0.760	0.806	0.785	0.638
det	0.628	0.757	0.773	0.724	0.654	0.694	0.702	0.584	0.597	0.596	0.587	0.597
expl	0.568	0.803	0.658	0.669	0.607	0.438	0.653	0.442	0.566	0.600	0.452	0.443
flat	0.609	0.770	0.921	0.721	0.761	0.554	0.923	0.455	0.577	0.520	0.786	0.649
flat:name	0.686	0.719	0.729	0.698	0.678	0.633	0.706	0.677	0.662	0.641	0.649	0.672
iobj	0.692	0.826	0.792	0.706	0.681	0.784	0.735	0.692	0.698	0.728	0.781	0.803
mark	0.693	0.787	0.799	0.752	0.701	0.676	0.684	0.696	0.708	0.681	0.682	0.693
nmod	0.725	0.767	0.776	0.750	0.677	0.711	0.695	0.586	0.649	0.649	0.617	0.657
nmod:poss	0.694	0.758	0.758	0.731	0.667	0.719	0.681	0.689	0.671	0.694	0.671	0.681
nsubj	0.655	0.794	0.806	0.768	0.695	0.705	0.725	0.610	0.780	0.788	0.793	0.760
nsubj:pass	0.694	0.758	0.758	0.731	0.667	0.719	0.681	0.689	0.671	0.694	0.671	0.681
nummod	0.716	0.858	0.839	0.728	0.714	0.705	0.730	0.777	0.790	0.714	0.741	0.729
obj	0.625	0.773	0.785	0.729	0.654	0.672	0.682	0.528	0.534	0.646	0.640	0.671
obl	0.659	0.767	0.776	0.753	0.684	0.685	0.703	0.656	0.678	0.663	0.667	0.706
obl:tmod	0.683	0.741	0.791	0.716	0.660	0.740	0.696	0.696	0.732	0.686	0.681	0.815
parataxis	0.652	0.798	0.792	0.756	0.775	0.674	0.645	0.658	0.700	0.667	0.674	0.689
xcomp	0.841	0.884	0.885	0.806	0.802	0.822	0.818	0.852	0.884	0.863	0.816	0.827
root	0.782	0.843	0.841	0.834	0.765	0.726	0.736	0.758	0.783	0.763	0.754	0.739

Table 17
Comparison of the representation from BERT in the baseline and SGBD model when tested on German sentences containing target dependency.

	Baseline vs SGBD											
De Relations	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12
acl:relcl	0.793	0.740	0.860	0.831	0.845	0.883	0.850	0.828	0.863	0.801	0.778	0.689
advcl	0.773	0.720	0.843	0.815	0.820	0.894	0.867	0.856	0.894	0.842	0.840	0.747
advmod	0.782	0.796	0.849	0.832	0.856	0.859	0.827	0.774	0.787	0.783	0.785	0.794
amod	0.773	0.732	0.802	0.816	0.844	0.808	0.801	0.800	0.812	0.768	0.766	0.780
appos	0.762	0.778	0.806	0.830	0.855	0.729	0.812	0.820	0.817	0.767	0.735	0.788
aux	0.747	0.735	0.833	0.810	0.836	0.796	0.717	0.777	0.781	0.742	0.746	0.734
aux:pass	0.766	0.728	0.799	0.839	0.867	0.825	0.825	0.806	0.815	0.746	0.798	0.748
case	0.774	0.759	0.819	0.812	0.849	0.830	0.825	0.820	0.826	0.790	0.797	0.797
cc	0.777	0.780	0.764	0.789	0.816	0.741	0.775	0.766	0.779	0.759	0.749	0.742
ccomp	0.792	0.794	0.822	0.831	0.877	0.841	0.829	0.829	0.818	0.775	0.788	0.798
compound	0.790	0.788	0.845	0.847	0.849	0.797	0.778	0.789	0.790	0.798	0.790	0.780
compound:prt	0.795	0.795	0.808	0.791	0.827	0.811	0.831	0.850	0.879	0.865	0.835	0.804
conj	0.797	0.787	0.795	0.784	0.814	0.773	0.784	0.778	0.787	0.786	0.780	0.783
cop	0.792	0.779	0.839	0.831	0.874	0.855	0.840	0.830	0.839	0.801	0.797	0.790
csubj	0.679	0.767	0.939	0.901	0.922	0.651	0.668	0.664	0.710	0.792	0.733	0.692
cc:preconj	0.634	0.557	0.642	0.684	0.818	0.459	0.411	0.595	0.678	0.673	0.644	0.520
csubj:pass	0.843	0.805	0.799	0.770	0.786	0.850	0.897	0.839	0.773	0.774	0.781	0.800
det	0.872	0.889	0.837	0.819	0.836	0.817	0.851	0.849	0.831	0.820	0.866	0.827
expl	0.753	0.770	0.719	0.850	0.884	0.840	0.822	0.829	0.860	0.843	0.824	0.786
flat	0.679	0.610	0.913	0.958	0.933	0.956	0.977	0.958	0.953	0.835	0.747	0.779
flat:name	0.682	0.643	0.817	0.831	0.869	0.833	0.829	0.833	0.832	0.811	0.777	0.655
iobj	0.769	0.797	0.791	0.746	0.793	0.832	0.889	0.871	0.881	0.869	0.843	0.789
mark	0.804	0.812	0.798	0.804	0.848	0.796	0.813	0.814	0.802	0.802	0.799	0.801
nmod	0.759	0.716	0.834	0.825	0.835	0.824	0.814	0.744	0.735	0.762	0.748	0.744
nmod:poss	0.796	0.795	0.793	0.809	0.841	0.768	0.815	0.786	0.785	0.792	0.782	0.797
nsubj	0.794	0.795	0.835	0.820	0.854	0.851	0.834	0.717	0.735	0.731	0.771	0.723
nsubj:pass	0.888	0.875	0.821	0.853	0.878	0.829	0.828	0.808	0.819	0.855	0.819	0.882
nummod	0.844	0.879	0.847	0.842	0.841	0.856	0.854	0.854	0.859	0.892	0.871	0.849
obj	0.775	0.784	0.801	0.799	0.824	0.812	0.797	0.732	0.793	0.760	0.791	0.799
obl	0.787	0.793	0.814	0.812	0.850	0.828	0.820	0.814	0.818	0.780	0.746	0.782
obl:tmod	0.794	0.805	0.829	0.816	0.870	0.805	0.752	0.815	0.849	0.844	0.858	0.851
parataxis	0.792	0.792	0.811	0.877	0.866	0.776	0.726	0.729	0.754	0.753	0.739	0.767
xcomp	0.877	0.889	0.861	0.847	0.868	0.858	0.858	0.855	0.856	0.888	0.893	0.875
root	0.797	0.795	0.828	0.819	0.854	0.846	0.829	0.717	0.791	0.728	0.772	0.743