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] emerges as a very efficient way of pre-training large language models from BERT [2] to GPT [3]. A combination of the missing word prediction task on a very large corpus with the self-attention mechanism leads to success in learning how words impact each other. Studies in the BERTology paradigm show that many syntactic relations are implicitly learned by such models without any supervision [4].

On the other hand, linguistic research contributes considerable attention to providing a careful description of the types of syntactic relations possible between across a number of languages. In particular, the framework of Universal Dependencies [5] results in corpora with unified annotations for numerous languages. An open question concerns the value of explicit linguistic annotation in graph structures and its application with transformer-based pre-trained language models in Neural Machine Translation (NMT).

This research focuses on integrating explicit syntactic knowledge, such as syntactic dependencies, to enhance translation quality. Traditional sequential models like RNNs and Transformers, despite their ability to process and represent sentences, often fall short of accurately capturing complex syntactic structures and phenomena [6–8].

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

The advent of Graph Attention Network (GAT) [9] 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) [10, 11]. This investigation assesses the potential of augmenting explicit syntactic insights via GAT with the implicit knowledge embedded in models like BERT to notably elevate translation quality. Additionally, it explores which syntactic structures in the source language most effectively contribute to the target language generation, addressing the current gap in understanding the impact of syntactic strategies on translation quality in Machine Translation (MT).

This study introduces NMT engines enhanced with syntactic knowledge via Graph Attention and BERT (SGB). Integrating syntactic data from source sentences with GAT and BERT aims to refine Transformer-based NMT by incorporating syntax and leveraging the capabilities of the pre-trained BERT model. Utilizing multi-head attention mechanisms within the graph structure enables 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 research conducts experiments across translation tasks from Chinese, German, and Russian to English, to demonstrate the effectiveness of the SGB methodology. The key contributions of this research are outlined 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 assesses translation quality improvements via syntactic knowledge, focusing on Quality Estimation (QE) score enhancements. The syntactically enhanced engines exhibit improved QE scores across three MT directions without detriment to BLEU scores, confirmed by a paired t-test. Additionally, it identifies syntactic structures in source sentences that the SGB engine optimally learns from, leading to better translations.
- This study reveals that while GAT possesses the capability to learn syntactic knowledge, its 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 its ability to learn dependency relations. Furthermore, there is a correlation between GAT's mastery of syntactic dependencies and translation quality. Better learned syntactic structures by GAT lead to the MT engine more accurately recognizing source language sentences with those structures, resulting in smoother translations.
- This study delves into the interpretability of translation quality enhancement through the prism 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 improving translation quality. The analysis also demonstrates that syntactic graph-based enhancements offer marginal improvements in machine translation quality amidst scrambled word sequences, yet fall short of substantially elevating the translation accuracy to desired levels, indicating the pivotal role of BERT in source sentence representation and the inherent limitations in interpreting jumbled semantics.
- This study assesses the versatility of the proposed approach by integrating XLM-Roberta in place of BERT. Despite this substitution, the approach consistently enhances translation quality across all evaluated MT directions, underscoring its broad applicability.

2. Related Study

2.1. Pre-trained Language Models

Pre-trained models have significantly advanced NLP, especially with the advent of Transformer architectures, marking a paradigm shift in the field's approach to understanding language [12, 13]. BERT, a standout among these innovations, leverages self-supervised learning on extensive corpora through the Masked Language Model (MLM) and Next Sentence Prediction (NSP) tasks. These techniques equip BERT to capture the essence of linguistic knowledge, enriching its understanding of language context and structure [4]. The empirical analysis and applications of BERT also help humans to understand pre-trained language models and support future improvements. BERT has made significant contributions to MT tasks, where its contextual word embeddings and generic

linguistic knowledge learned from pre-training can enhance the generalization ability of MT engines, especially in cases with limited bilingual data. Most studies show that incorporating BERT improves the performance of the MT engine, as demonstrated by metrics like the BLEU score [14–16].

2.2. Syntactic Knowledge in Translation

In the realm of MT, the importance of syntactic dependency cannot be overstated. It is crucial for the grammatical dissection of sentences, presenting them in easily interpretable tree diagrams. The incorporation of syntactic data into 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, with the strategy for incorporating explicit syntactic knowledge shifting progressively from RNN-based methods to Transformer-based ones [17–19]. 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. Yet, the efficacy of using the Transformer's attention mechanism as an explanatory tool remains a topic of debate [20, 21]. Efforts have been made to enhance the effectiveness of downstream tasks by fusing explicit syntactic knowledge with BERT [10, 22], but the applications of such integration in MT has 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. 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 [23, 24]. 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, combined with BERT, forms a robust framework for encapsulating linguistic knowledge in downstream NLP tasks [10, 25, 26].

3. Methodology

3.1. Construction of the SGB Engines

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

3.1.1. Encoding

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

[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

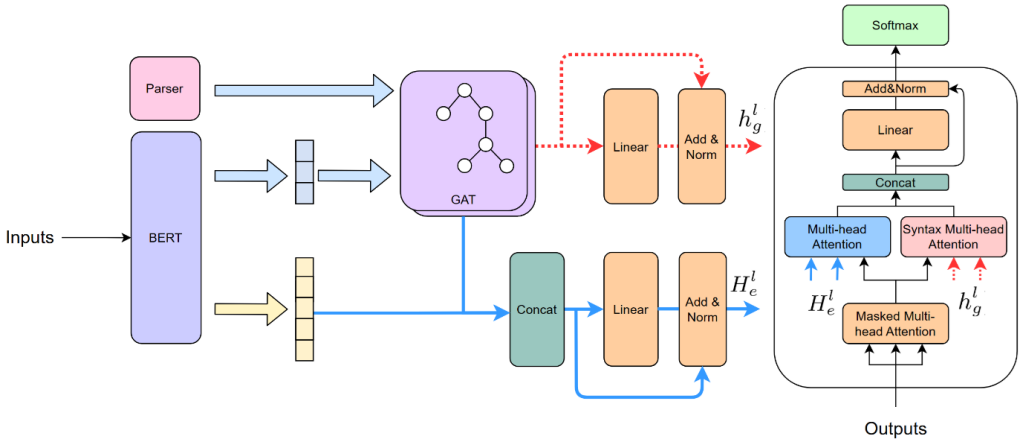


Fig. 1. The architecture of the SGB engines. The encoder with BERT and GAT on the left and the decoder on the right. Dash lines indicate the alternative connections. H_e^l and h_g^l represent the final layer output of BERT and GAT.

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, and the approach of German 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 $\tilde{S}$, we use a Universal Dependencies-based parser[4] [27] to perform tokenizing and syntactic dependency parsing on source sentences. After obtaining the parsing results, we construct the node adjacency matrix for graph representation. Each token has a corresponding node in the graph. Since word representation from BERT contains rich semantic information, nodes on the graph are encoded by BERT. Considering the subword segmentation, we merge subword token representation in an average way 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 [9] 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.

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

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

1-hop neighbors $j \in N_i$ are attended by the node i, $\Big\Vert_{k=1}^{K}$ 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.

[4] https://github.com/hankcs/HanLP

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).

$$H_e^l = concat(h_B, h_G)$$
$$h_d^l = attn_D(h_d^l, H_e^l, H_e^l)$$

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).

$$h_d^l = attn_D(h_d^l, H_e^l, H_e^l)$$
$$h_s^l = attn_S(h_d^l, h_g^l, h_g^l)$$
$$h_t^l = concat(\tilde{h}_d^l, \tilde{h}_s^l)$$

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 realm of MT, the quest for accurate and reliable evaluation metrics is perpetual. Among these metrics, BLEU [28] has emerged as a cornerstone for assessing the quality of text translated from one language to another. BLEU operates by comparing the machine-generated translations to one or more reference translations, focusing primarily on the precision of n-grams. Despite its widespread adoption, BLEU's emphasis on precision alone, without considering the fluency or the adequacy of the content, has led researchers to explore complementary evaluation strategies.

QE, a paradigm designed to assess the quality of a translation without the need for reference texts. QE shifts the focus from the comparison-based metrics to predicting the translation's quality based on the translated text and its source. This approach is not only innovative but also practical, especially in scenarios where reference translations are unavailable. QE encompasses various dimensions, including fluency, adequacy, and even the prediction of post-editing effort, offering a multifaceted view of translation quality.

Among the tools developed for QE, TransQuest [29] stands out as a notable example. Built upon cutting-edge transformer models, TransQuest introduces a novel way of performing QE by leveraging sentence-level quality estimation. It operates by predicting a quality score for each sentence pair (source and translated sentence), which correlates with human judgments on translation quality. This method has shown significant improvements over traditional QE approaches, providing more accurate and reliable assessments. TransQuest's application in QE highlights the advancements in utilizing deep learning and artificial intelligence to enhance the understanding and evaluation of machine translation quality.

The evaluation of MT in this study utilizes two methods. The first compares MT output against a gold standard reference using the widely recognized n-gram matching model BLEU and the advanced neural model COMET [30]. However, given that BLEU and COMET operate on the corpus level, failing to identify enhancements in specific sentences, and rely on reference translations, which overlook legitimate translation variations, an additional approach is employed. MT quality estimation techniques are adopted to gauge sentence-level improvements, utilizing TransQuest, acclaimed victor of the WMT 20 QE Shared Task.

Table 1
Despite the reduced size of the dataset, SGB engines maintain a competitive edge over Baseline engines in BLEU scores across three MT scenarios with varying training set sizes.

	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

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 is determined by the available UD corpus for a trained external syntactic parser, and the PUD corpus for evaluation of both the syntactic knowledge of BERT and GAT and the MT engine performance.

4. What Happens to Translations

4.1. BLEU scores 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 thousand (K) and 5K sentence pairs for the validation and test sets, respectively. 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 Table 1 delineates, the SGB engines consistently outperform the baseline engine in terms of BLEU scores across three distinct translation directions and multiple dataset sizes, underscoring the superior generalization capabilities of the SGB engines. Drawing inspiration from study [31], which posits that BLEU scores may not suffice in capturing the nuanced quality of translations, this study further employs the COMET QE model for a more comprehensive evaluation of engine performance. The COMET QE model, by examining the interplay between the source

[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
BLEU and QE scores for translations of three languages under 1M training set size.

Size	Zh→En	Baseline	SGBC	SGBD
1M	BLEU	47.15	**47.23**	47.17
	COMET	82.20	83.69	**84.78**
	Ru→En	Baseline	SGBC	SGBD
	BLEU	47.22	**47.36**	47.27
	COMET	80.93	81.34	**82.56**
	De→En	Baseline	SGBC	SGBD
	BLEU	37.59	**37.67**	37.63
	COMET	78.02	78.66	**79.37**

sentence, its translation, and reference translations, assigns a QE score on a scale from 0 to 100, where a higher score indicates superior translation quality. Both the SGB engines and the Baseline engine are subject to evaluation under the metrics of BLEU scores and QE scores.

As illustrated in Table 2, the SGB engines demonstrate enhanced performance on both the BLEU and QE metrics, with the SGBD engine, in particular, exhibiting a notably higher disparity in QE scores relative to the baseline model. In practical translation scenarios, where reference translations may not be available and translations can vary while preserving semantic integrity, the QE score proves to be a more effective measure of an engine's capacity to handle source sentences flexibly. It reflects the engines' adeptness at leveraging syntactic knowledge on the graphs and fully utilizing the latent linguistic capabilities of BERT, thereby enabling the generation of more accurate translations not limited by the singular reference answers present in the dataset. The QE model, in comparison to BLEU scores, offers a more nuanced advantage in accommodating reasonable variations in translations.

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

Building on the insights from the above tables presented, this study delve deeper into the improvement of translation quality by integrating syntactic knowledge, acknowledging the inadequacies of BLEU scores in reflecting linguistic subtleties and aligning with human evaluative criteria [32, 33]. a gold-standard syntactic annotation corpus and a QE model are employed. To address these shortcomings, a gold-standard syntactic annotation corpus and a QE model are employed. This approach prioritizes the fidelity of source sentence semantics, the coherence of translated content, and the propriety of word sequencing.

The PUD corpuses (PUD Chinese, PUD Russian, PUD German) are translated using the Baseline and SGB engines for the three MT directions. Since PUD corpus encompasses sentences from out-of-domain sources, such as news and wikis, thereby increasing the demands on the MT engine's ability to effectively summarize and clarify sentence structures. The state-of-the-art QE model called TransQuest[10], which evaluates the source language sentences and translations, is employed to score these translations from 0 to 1, with a higher score indicating better translation quality. A paired t-test is utilized to analyze the changes and distribution in translation quality before and after the implementation of these approaches, setting the significance level at 0.05 in the paired t-test.

From Table 3, when comparing the Zh Baseline and SGBC engines, $\bar{x}_d$ of them is 0.024, S_d is 0.109 and the test statistic (t) is 7.18, corresponding to a p-value < 0.001. The t and p-value in SGBD also reveal the statistical significance of the QE scores before and after proposed approach. Both reject H_0 at the significant level of 0.05 (H_0 is that proposed approaches do not significantly differ in QE scores compared to the baselines.). Instead, H_1 is accepted that the differences between baseline and SGB engines in QE scores are large enough to be statistically significant.

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

[10]https://github.com/TharinduDR/TransQuest

Table 3
Paired t-test for PUD corpus translations of three languages between Baseline and SGB engines.

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	$p < 0.001$

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 this 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 assessment, 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 gains the most in terms of translation quality enhancement through translation engines, sentences associated with lower-quality translations are retained and categorized based on their dependency relations. Given a dependency relation d, any source language sentence corresponding to low-quality translations and containing dependency d is grouped. The average QE score for each group, characterized by dependency relations, is calculated both before and after the application of enhancement methodologies. This enables a detailed examination of the impact of distinct syntactic structures on the efficacy of translation quality improvements facilitated by the engines.

Table 4 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 enhancement. Although both SGBC and SGBD engines incorporate graph-based syntactic knowledge, their approaches to learning dependency relations diverge. For instance, the "flat" dependency 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," "orphan," and "csubj." 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 models, 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

Incorporating syntactic structures via GAT has proven beneficial for enhancing translation quality. However, whether GAT's syntactic learning capability is on par with BERT's remains an open question. Additionally, the degree to which GAT's proficiency in syntactic knowledge can be considered tangible proof of translation quality

Table 4
The top-5 syntactic improvements in source language sentences identified by SGB engines, where the QE scores differ the most.

			Zh				
	Baseline	SGBC	Qual		Baseline	SGBD	Qual
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	0.75	+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%

improvement requires further exploration. Therefore, this chapter focuses on a detailed examination of the relationship between GAT's acquisition of syntactic knowledge and its effect on translation quality, seeking to clarify the importance of graph-based syntactic knowledge in the field of MT.

5.1. Syntactic Knowledge in GAT

A syntactic dependency prediction model is introduced to investigate the types of syntactic knowledge GAT are capable of learning. Utilizing the PUD corpus, which provides gold standard syntactic annotations, as the foundational dataset, this model converts syntactic annotations and sentence words into syntactic trees, which are then adapted into graph structures for GAT analysis. In this model, each word is represented as a node within a graph, with edges illustrating syntactic dependencies as defined by the PUD corpus. GAT's objective is to deduce dependency relations by assimilating information from both nodes and edges. Unlike traditional syntactic dependencies that typically follow a unidirectional flow from parent to child nodes, this approach treats dependencies as bidirectional graphs, acknowledging the reciprocal influence between parent and child nodes. This bidirectional consideration is crucial for GAT to comprehend the varying implications of node connections, thereby enhancing its ability to accurately identify the dependency relations among nodes.

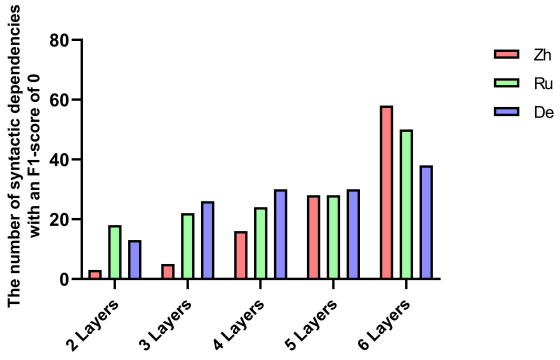
Like the Transformer model, GAT utilize multi-head attention and layers stacked upon each other. The study initially explores how the number of multi-head attention heads and layers influences GAT's acquisition of syntactic knowledge, examining the advantages these configurations offer for learning syntactic dependencies. In the experiments, the attention head counts (Heads) tested for GAT 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. 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 5 underscores the criticality of judiciously configuring the number of attention heads in GAT, as it influences the model's sensitivity to accurately learn syntactic knowledge. For example, the Russian language experiment

Table 5
Overall GAT predictions of syntactic relationships for three languages with different numbers of attention heads and layers.

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

Fig. 2. The number of F1-score dropped to 0 made by the GAT in different layers with a different number of attention heads.



reveals that a GAT setup with 2 layers and 4 attention heads outshines a configuration with 8 attention heads in terms of overall prediction efficacy. Notably, with the model expanded to 4 layers, a higher count of attention heads enhances performance. Conversely, augmenting the model with additional layers tends to degrade its ability to accurately predict dependency relations, with a configuration of two layers surpassing the performance of one with six layers. 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.

The analysis extends to the prediction scores for individual dependency relations across the three languages. As the number of GAT layers increases, the challenge of accurately predicting certain dependency relations becomes apparent, with F1-scores diminishing and in some cases plummeting to 0, as detailed in Table 6. This study doc-

Table 6

The predictions of some syntactic dependencies in three different languages are shown. As the number of layers increases, GAT gradually loses the learning of syntactic dependencies, and even F1-score drops to 0. Some dependencies are unaffected and continue to have relatively high prediction scores.

GAT		Zh			Ru			De		
Layers	Heads	advmod	clf	dep	case	flat	mark	acl:relcl	cc	naubj
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	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	0.9	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	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	0.64	0.99	0.8	0.94	0.45	0.96	0.74
5	2	0.90	0	0	0.97	0.55	0.93	0.42	0.85	0.78
	4	0.90	0	0	0.98	0.77	0.96	0.68	0.82	0.79
	6	0.90	0	0	0.97	0.67	0.93	0.44	0.81	0.72
	8	0.89	0	0	0.99	0.48	0.96	0.43	0.86	0.73
6	2	0.83	0	0	0.94	0	0.91	0	0.83	0.65
	4	0.86	0	0	0.95	0	0.97	0	0.78	0.65
	6	0.84	0	0	0.94	0	0.93	0	0.79	0.67
	8	0.86	0	0	0.96	0	0.93	0.37	0.85	0.63

uments the instances of dependency relations that achieve an F1-score of 0, varying with the number of attention heads across each layer for each language, depicted in Figure 2. A pattern emerges where, beyond three layers, occurrences of F1-scores at 0 increase, and the addition of more attention heads does little to mitigate this deterioration. Remarkably, GAT demonstrates a consistent ability to effectively learn specific dependency relations (where the F1-score never drops to 0), irrespective of layer adjustments. These resilient dependency relations, identified in the evaluations of all three languages, include "advmod", "case", "cc", "mark", "nsubj", and "punct", suggesting GAT's heightened sensitivity and reliable capture of these syntactic knowledge features.

5.2. Graph Features and Translation Quality

Enhancing translation quality may hinge on the GAT capability for syntactic comprehension. This inquiry delves into the types of syntactic knowledge readily assimilated by GAT. To explore the linkage between graph-based syntactic knowledge and translation efficacy, a syntactic dependency prediction framework is established for GAT, utilizing the PUD corpus across three languages. This framework models words as nodes and dependency connections between words as edges within a graph, challenging GAT to deduce dependency relations from the lexical information. Such dependencies are conceptualized as undirected graphs, compelling nodes to integrate insights from all adjacent nodes. The dataset for each language is segmented into sets of 800 training, 100 validation, and 100 test sentences. GAT's architecture is delineated with two layers and attention heads tailored to each language where 4 for Zh and 6 for Ru and De aligning with the specifications of the SGB engines. The model parameters include a hidden size of 768, a dropout rate of 0.2, the Adam optimizer with a learning rate of 5e-5, and evaluation based on the F1-score metric.

Table 7

The top-10 dependency relations with the highest prediction scores by GAT across various source language sentences, alongside the corresponding changes in translation quality as facilitated by different MT engines for these sentences.

		Zh		
	GAT	Baseline	SGBC	SGBD
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		
	GAT	Baseline	SGBC	SGBD
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		
	GAT	Baseline	SGBC	SGBD
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

Table 7 reveals GAT's efficiency in learning dependency relations, achieving proficiency with merely two layers and limited data samples. The data illustrates a direct correlation between GAT's predictive capabilities regarding syntactic dependencies and the enhancement of translation quality. Notably, GAT's precise identification of "make", "cc", and "conj" relations in Chinese significantly enhances the SGBC and SGBD engines' ability to accurately recognize sentences bearing these syntactic markers, leading to improved translation outcomes. This improvement in recognizing and translating syntactic features extends similarly to Russian and German, indicating the broad applicability of GAT's learning efficiency across languages.

GAT struggles with specific syntactic relations like "iobj" and "nsubj:pass". Yet, these challenges in prediction accuracy, such as with "obl:tmod" in Chinese and German, do not necessarily translate to diminished translation quality. In fact, the SGB engines successfully leverage these syntactic insights to enhance translation, indicating that GAT's robust learning of dependencies significantly contributes to translation improvements. One of the factors contributing to the improvement of translation quality can be the robust dependency relation learning by GAT. But it is not absolute since GAT may not effectively learn such features with fewer samples in the test, or the encoder or decoder needs more explicit sentence structure information provided by GAT rather than whether the syntactic annotation from the parser is correct.

6. What Happens to Syntactic Features

The impact of explicit syntactic knowledge via graphs on translation quality, and its interaction with BERT through GAT, is a focal point of interest. To understand the interpretability of these approaches in relation to syntax, this study conducts representation similarity analysis with BERT, and evaluates the effects of syntax disruption by artificially randomizing word order. These methods aim to uncover how syntactic information on graphs influences BERT's decision-making and the overall translation accuracy, providing insights into the integration of syntactic knowledge within NMT engines.

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 work [34], RSA uses n examples for building 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. The addition of syntactic knowledge on the graph also impacts the representation space of BERT and thus improves the modeling of source sentences. The source sentences corresponding to the 300 low-quality translations are divided according to the type of dependency relations as the stimulus. Given the current dependency relation is x, the source sentences of low-quality translations containing x are all composed into one group stimulus. BERT representations from both models are extracted for comparison (e.g., Baseline vs SGBC), and cosine similarity is used as the kernel for all experiments.

Table 8 presents selected results from an RSA analysis, comparing Baseline BERT with SGB engines based on syntactic prediction scores by GAT (full results are in Appendix A). Across all three languages, the lowest RSA scores typically occur in the lower and middle layers of BERT. Specifically, layers 3-5 for Chinese and Russian, and layers 5-8 for German, exhibit the lowest RSA scores. The integration of syntactic knowledge from graphs into these layers results in decreased representation similarity, indicating a reconfiguration of syntactic knowledge. This observation suggests that the lower and middle layers of BERT are particularly susceptible to modifications in modeling both shallow and deeper syntactic knowledge. Layers 9-12 are primarily associated with abstract semantic information processing, variations in their representational similarity imply that alterations in the lower layers can impact the processing of deep linguistic information in the upper layers, despite the task-oriented nature of the final layer. It reveals that incorporating linguistic knowledge into the fine-tuned representation of BERT can lead to reconsidering such knowledge to obtain a more accurate representation of the source language sentences and thus improve translation quality.

6.2. Disruption of Word Order

The impact of BERT and graph-based syntactic knowledge on enhancing translation quality presents an area for further investigation, particularly regarding the robustness of syntactic knowledge. This leads to questions about the relative contributions of BERT versus graph syntactic knowledge to translation quality and the potential limitations

*Representations from Baseline and SGBD for comparison.

Table 8
Top-5 highest F1-score of syntactic knowledge learning on the graph and its BERT layer with the lowest similarity in RSA analysis for each language.

	GAT	RSA	Layer	RSA*	Layer
Zh					
mark	0.986	0.418	5	0.407	3
cc	0.984	0.274	4	0.354	5
conj	0.970	0.380	5	0.340	4
nummod	0.965	0.274	4	0.237	3
root	0.955	0.216	4	0.390	4
Ru					
det	0.990	0.426	4	0.408	3
root	0.987	0.466	3	0.504	3
amod	0.982	0.444	3	0.391	4
case	0.978	0.462	4	0.413	4
aux:pass	0.974	0.357	3	0.327	3
De					
case	0.992	0.686	5	0.759	2
cc	0.987	0.591	6	0.741	6
det	0.987	0.584	8	0.817	6
mark	0.981	0.676	6	0.769	6
advmod	0.932	0.733	6	0.774	8

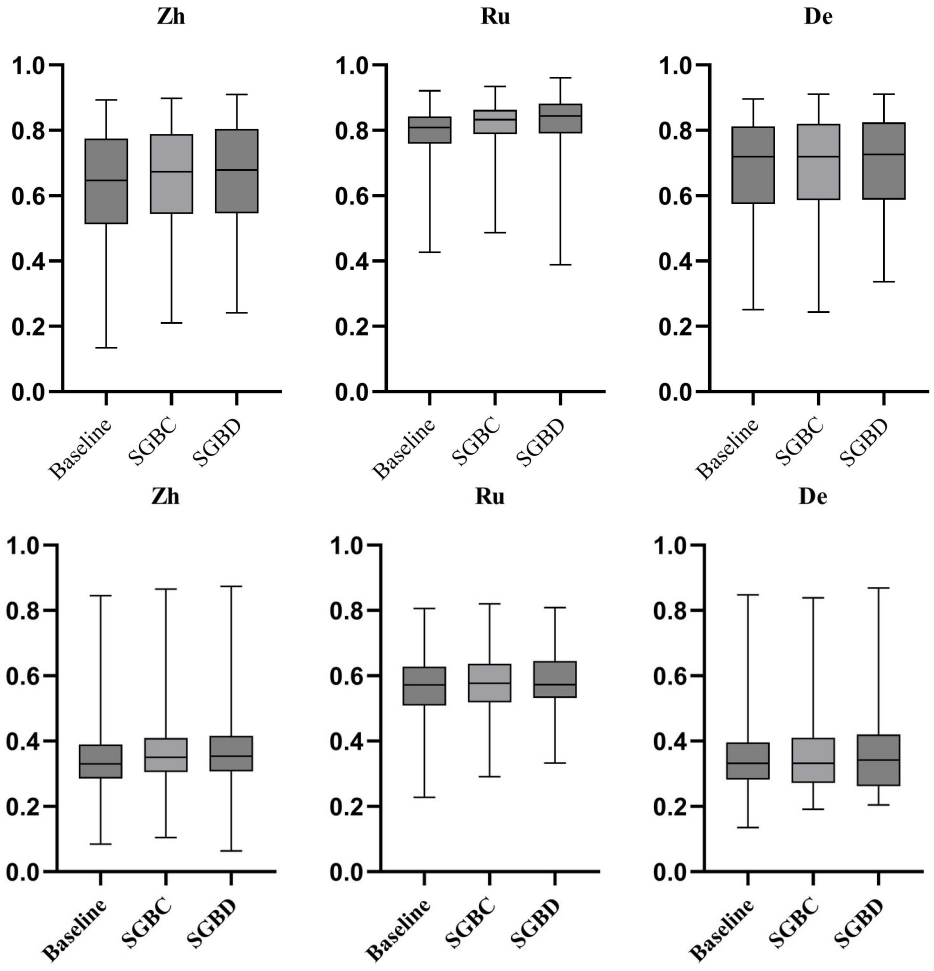
of the proposed MT engines. To address these inquiries, the study involves altering the word order in source language sentences from each language in the PUD corpus, for instance, transforming "A B C D E F" into a randomized sequence like "C B A D F E". Following this, both Baseline and SGB engines are tasked with translating these modified sentences. The translations are then reassessed by a QE model, which contrasts the translations of the shuffled sentences against those of the original, orderly sentences, thereby providing insights into the adaptability and efficacy of the syntactic knowledge in translation.

Figure 3 demonstrates, it is observed that scrambled word sequences in source sentences cause a significant decrease in translation quality for both Baseline and SGB engines across all MT directions. The SGB engines demonstrate a slightly improved distribution of QE scores relative to the Baseline engines, highlighting the effectiveness of the applied syntactic strategy. Nevertheless, it is crucial to recognize that the integration of GAT into the encoder or the provision of explicit syntactic knowledge to the decoder does not ensure a marked improvement in translation quality. Expecting the median QE scores in the box plots to surge from below 0.4 to 0.7 is not feasible. This finding indicates that BERT is more instrumental in forming representations of source sentences and affecting translation quality in this hybrid approach. The jumbling of sentence order, leading to syntactic information loss, indicates that while SGB engines, bolstered by graph-based syntactic knowledge, can alleviate some effects of this disarray, they are still unable to interpret and comprehend the correct semantics of jumbled sentences in the manner humans do.

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 [35] 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

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



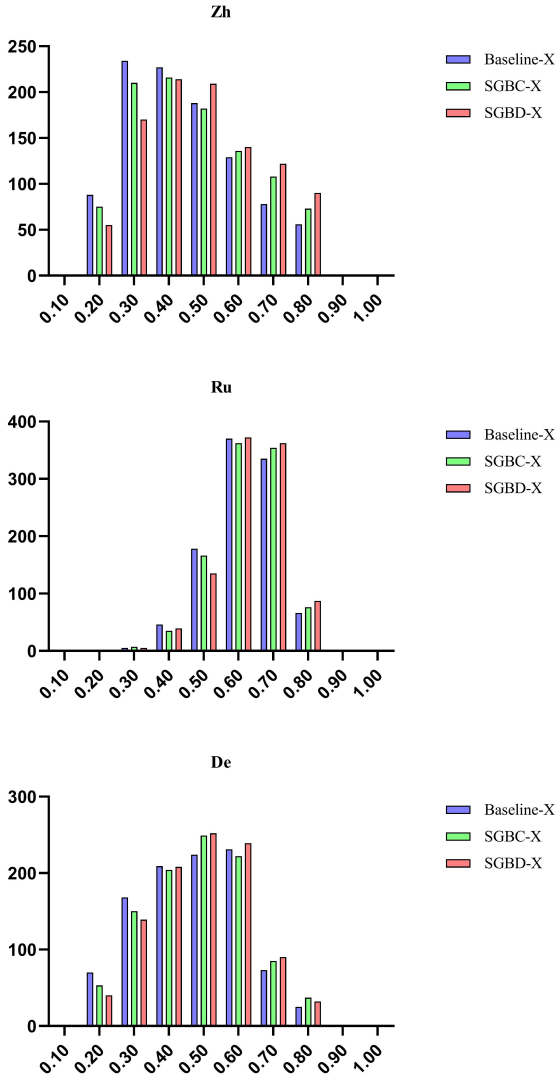
Ru→En) MT engines utilize the UNPC corpus, whereas the German (De→En) engines employ Europarl. Each training set comprises 0.1M sentence pairs, with validation and test sets featuring 6K parallel sentence pairs each. Specifications include word embeddings of 1024, a learning rate (excluding GAT) of 2e-5, 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 9
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**

Table 9 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

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 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. These findings indicate that integrating syntactic knowledge from graphs enhances translation quality across three MT directions with a different pre-trained language model, extending beyond just BERT. This enhancement is especially pronounced for translations with lower QE scores, effectively reducing their occurrence while boosting the proportion of translations achieving higher QE scores.

8. Conclusions

This research explores the integration of syntactic knowledge into MT, specifically assessing the efficacy of BERT and GAT. It introduces two SGB engines for translations from Zh to En, Ru to En, and De to En. By leveraging GAT,

the BERT encoder's representation is enhanced, and the decoder's understanding of the source language's sentence structure is improved. The results indicate that SGB engines outperform the Baseline engines in both BLEU scores, across various dataset sizes, and COMET scores, indicating increased translation accuracy and robustness. Analysis of PUD datasets shows that SGB engines produce more coherent translations, with significant improvements confirmed by Paired t-tests. The study also finds that SGB engines are better at recognizing specific source language sentence structures, defined by dependency relations, than Baseline engines, leading to improved translation quality. For example, SGB engines achieve notably higher QE scores for sentences with the "obl:agent" structure in Chinese compared to the Baseline.

The study also examines GAT's syntactic dependency learning through the PUD corpus, adjusting attention heads and model layers for optimal dependency prediction. Results suggest an improvement in learning efficiency with increased attention heads, though optimal configurations vary by language. However, model complexity beyond two layers tends to reduce the prediction performance, indicating a balance between complexity and prediction effectiveness. This study further explores the impact of GAT's prediction of dependency relations on translation quality, demonstrating a direct correlation between GAT's prediction accuracy for certain dependency relations and improved QE scores across various engines and languages.

Furthermore, RSA indicates that incorporating GAT, although not originally part of BERT, allows specific BERT layers to reassess source sentences' syntactic structures through fine-tuning, benefiting translation quality. This effect is notably present in the early to mid layers of BERT across different languages. Experiments on word order disruption underscore the crucial role of BERT's accurate modeling of source sentences for the effectiveness of syntactic knowledge enhancement via GAT.

The study not only underscores the value of syntactic knowledge in machine translation improvements but also considers XLM-R as a viable BERT alternative. Findings highlight the consistent benefits of syntactic knowledge across different translation approaches and models, emphasizing its crucial role in refining translation quality.

In conclusion, this study underscores the significant potential of melding syntactic knowledge embedded in graph structures with cutting-edge language models such as BERT and XLM-Roberta to refine MT. The results support further investigation into these synergies to boost translation precision and interpretability, aiming to establish a new standard of excellence in MT.

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. 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.

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.
[11] G. Li, C. Zheng, M. Li and H. Wang, Automatic Requirements Classification Based on Graph Attention Network, *IEEE Access* **10** (2022), 30080–30090.
[12] 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.
[13] 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).
[14] 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.
[15] 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.
[16] 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).
[17] 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.
[18] 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).
[19] 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.
[20] 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.
[21] 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.
[22] 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.
[23] 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.
[24] 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.
[25] 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.
[26] 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.
[27] 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.
[28] 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.
[29] 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.
[30] 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.
[31] 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.
[32] 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.
[33] J. Novikova, O. Dušek, A.C. Curry and V. Rieser, Why we need new evaluation metrics for NLG, *arXiv preprint arXiv:1707.06875* (2017).
[34] A. Merchant, E. Rahimtoroghi, E. Pavlick and I. Tenney, What happens to bert embeddings during fine-tuning?, *arXiv preprint arXiv:2004.14448* (2020).

[35] 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.

[36] 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).

[37] 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 10 to Table 15 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 10
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 11
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 12
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 13
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 14
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 15
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