Transformers as Graph-to-Graph Models

James Henderson [1] **Alireza Mohammadshahi** [*1,2,3] **Andrei C. Coman** [1,2]
Lesly Miculicich [*†]
[1] Idiap Research Institute [2] EPFL [3] University of Zurich
{james.henderson,andrei.coman}@idiap.ch
alireza.mohammadshahi@epfl.com
lmiculicich@google.com

Abstract

We argue that Transformers are essentially graph-to-graph models, with sequences just being a special case. Attention weights are functionally equivalent to graph edges. Our Graph-to-Graph Transformer architecture makes this ability explicit, by inputting graph edges into the attention weight computations and predicting graph edges with attention-like functions, thereby integrating explicit graphs into the latent graphs learned by pretrained Transformers. Adding iterative graph refinement provides a joint embedding of input, output, and latent graphs, allowing non-autoregressive graph prediction to optimise the complete graph without any bespoke pipeline or decoding strategy. Empirical results show that this architecture achieves state-of-the-art accuracies for modelling a variety of linguistic structures, integrating very effectively with the latent linguistic representations learned by pretraining.

1 Introduction

Computational linguists have traditionally made extensive use of structured representations to capture the regularities found in natural language. The huge success of Transformers (Vaswani et al., 2017) and their pre-trained large language models (Devlin et al., 2019; Zhang et al., 2022; Touvron et al., 2023a,b) have brought these representations into question, since these models are able to capture even subtle generalisations about language and meaning in an end-to-end sequence-to-sequence model (Wu et al., 2020; Michael et al., 2020; Hewitt et al., 2021). This raises issues for research that still needs to model structured representations, such as work on knowledge graphs, hyperlink graphs, citation graphs, or social networks.

In this paper we show that the sequence-to-sequence nature of most Transformer models is only a superficial characteristic; underlyingly they are in fact modelling complex structured representations. We survey versions of the Transformer architecture which integrate explicit structured representations with the latent structured representations of Transformers. These models can jointly embed both the explicit structures and the latent structures in a Transformer's sequence-of-vectors hidden representation, and can predict explicit structures from this embedding. In the process, we highlight evidence that the latent structures of pretrained Transformers already include much information about traditional linguistic structures. These Transformer architectures support explicit structures which are general graphs, making them applicable to a wide range of structured representations and their integration with text.

The key insight of this line of work is that attention weights and graph structure edges are effectively the same thing. Linguistic structures are fundamentally an expression of locality in the interaction between different components of a representation. As Henderson (2020) argued, incorporating this information about locality in the inductive bias of a neural network means putting connections between hidden vectors if their associated components are local in the structure. In Transformers (Vaswani et al., 2017), these connections are learned in the form of attention weights. Thus, these attention weights are effectively the induced structure of the Transformer's latent representation.

However, attention weights are not explicitly part of a Transformer's hidden representation. The output of a Transformer encoder is a sequence of vectors, and the same is true of each lower layer of self-attention. The latent attention weights are extracted from these sequence-of-vector embeddings with learned functions of pairs of vectors. Edges in explicit graphs can be predicted in the same way (from pairs of vectors), assuming that these graphs have also been embedded in the sequence of vectors.

[*]Work done while working at Idiap Research Institute.
[†]Now at Google

In recent years, the main innovation has been in how to embed explicit graphs in the hidden representations of Transformers. In our work on this topic, we follow the above insight and input the edges of the graph into the computation of attention weights. Attention weights are computed from an $n \times n$ matrix of attention scores (where n is the sequence length), so we input the label of the edge between nodes i and j into the score computation for the i,j cell of this matrix. Each edge label has a learned embedding vector, which is input to the attention score function in various ways depending on the architecture. This allows the Transformer to integrate the explicit graph into its own latent attention graph in flexible and powerful ways. This integrated attention graph can then determine the Transformer's sequence-of-vectors embedding in the same way as standard Transformers.

Researchers from the Natural Language Understanding group at Idiap Research Institute have developed this architecture for inputting and predicting graphs under the name of *Graph-to-Graph Transformer* (G2GT). G2GT allows conditioning on an observed graph and predicting a target graph. For the case where a graph is only observed at training time, we not only want to predict its edges, we also want to integrate the predicted graph into the Transformer embedding. This has a number of advantages, most notably the ability to jointly model all the edges of the graph. By iteratively refining the previous predicted graph, G2GT can jointly model the entire predicted graph even though the actual prediction is done independently for each edge. And this joint modelling can be done in conjunction with other explicit graphs, as well as with the Transformer's induced latent graph.

Our work on G2G Transformer has included a number of different explicit graph structures. The original methods were developed on syntactic parsing (Mohammadshahi and Henderson, 2021, 2020). The range of architectures was further explored for semantic role labelling (Mohammadshahi and Henderson, 2023) and collocation recognition (Espinosa Anke et al., 2022). G2GT's application to coreference resolution extended the complexity of graphs to two levels of representation (mention spans and coreference chains) over an entire document, which was all modelled with iterative refinement of a single graph (Miculicich and Henderson, 2022). Current work on knowledge extraction poses further challenges, most notably the issue of tractably modelling large graphs. The code for G2GT is open-source and available for other groups to use for other graph structures (at `https://github.com/idiap/g2g-transformer`).

In the rest of this paper, we start with a review of related work on deep learning for graph modelling (Section 2). We then present the general G2GT architecture with iterative refinement (Section 3), before discussing the specific versions we have evaluated on specific tasks (Section 4). We then discuss the broader implications of these results (Section 5), and conclude with a discussion of future work (Section 6).

2 Deep Learning for Graphs

Graph Neural Networks. Early attempts at broadening the application of neural networks to graph structures were pursued by Gori et al. (2005) and Scarselli et al. (2008), who introduced the Graph Neural Networks (GNNs) architecture as a natural expansion of Recurrent Neural Networks (RNNs) (Hopfield, 1982). This architecture regained interest in the context of deep learning, expanded through the inclusion of spectral convolution layers (Bruna et al., 2013), gated recurrent units (Li et al., 2015), spatial convolution layers (Kipf and Welling, 2017), and attention layers (Veličković et al., 2018). GNNs generally employ the iterative local message passing mechanism to aggregate information from neighbouring nodes (Gilmer et al., 2017). Recent research, analysing GNNs through the lens of Weisfeiler and Leman (1968), has highlighted two key issues: over-smoothing (Oono and Suzuki, 2020) and over-squashing (Alon and Yahav, 2021). Over-smoothing arises from repeated aggregation across layers, leading to convergence of node features and loss of discriminative information. Over-squashing, on the other hand, results from activation functions during message aggregation, causing significant information and gradient loss. These issues limit the capacity of GNNs to effectively capture long-range dependencies and nuanced graph relationships (Topping et al., 2021). The Transformer architecture (Vaswani et al., 2017) can be seen as addressing these issues, in that its stacked layers of self-attention can be seen as a fixed sequence of learned aggregation steps.

Graph Transformers. Transformers (Vaswani et al., 2017), initially designed for sequence tasks, represent a viable and versatile alternative to GNNs due to their intrinsic graph processing capabilities. Through their self-attention mechanism, they can seamlessly capture global wide-ranging relationships, akin to handling a fully-connected graph. Shaw et al. (2018) explicitly input relative position relations as embeddings into the attention function, thereby effectively inputting the relative position graph, instead of absolute position embeddings, to represent the sequence. Generalising this explicit input strategy to arbitrary graphs (Henderson, 2020) has led to a general class of models which we will refer to as *Graph Transformers* (GT).

GT Evolution and Applications. The history of graph input methods used in GTs started with Transformer variations that experimented with relative positions to more effectively capture distance between input elements. Rather than adopting the sinusoidal position embedding introduced by Vaswani et al. (2017) or the absolute position embedding proposed by Devlin et al. (2019), Shaw et al. (2018) added relative position embeddings to attention keys and values, capturing token distance within a defined range. Dai et al. (2019) proposed Transformer-XL, which used content-dependent positional scores and a global positional score in attention weights. Mohammadshahi and Henderson (2020) demonstrated one of the earliest successful integration of an explicit graph into Transformer's latent attention graph. They introduced the *Graph-To-Graph Transformer* (G2GT) architecture and applied it to syntactic parsing tasks by effectively leveraging pre-trained models such as BERT (Devlin et al., 2019). Huang et al. (2020) introduced new methods to enhance interaction between query, key and relative position embeddings within the self-attention mechanism. Su et al. (2021) proposed RoFormer, which utilises a rotation matrix to encode absolute positions while also integrating explicit relative position dependencies into the self-attention formulation. Liutkus et al. (2021) and Chen (2021) extended Performer (Choromanski et al., 2020) to support relative position encoding while scaling Transformers to longer sequences with a linear attention mechanism. Graphormer (Ying et al., 2021) introduced node centrality encoding as an additional input level embedding vector, node distances and edges as soft biases added at attention level, and obtained excellent results on a broad range of graph representation learning tasks. Mohammadshahi and Henderson (2021) built upon the G2GT architecture and proposed an iterative refinement procedure over previously predicted graphs, using a non-autoregressive approach. SSAN (Xu et al., 2021) leveraged the GT approach to effectively model mention dependencies for document-level relation extraction tasks. JointGT (Ke et al., 2021) exploited the GT approach for knowledge to text generation tasks via a joint graph-text encoding. Similarly, TableFormer (Yang et al., 2022) demonstrated the successful utilisation of the GT approach for combined text-table encoding in table-based question answering tasks. Espinosa Anke et al. (2022) proposed a GT architecture for simultaneous collocation extraction and lexical function typification, incorporating syntactic dependencies into the attention mechanism. Miculicich and Henderson (2022) showed that the G2GT iterative refinement procedure can be effectively applied to graphs at multiple levels of representation. Diao and Loynd (2022) further extended a GT architecture with new edge and node update methods and applied them to graph-structured problems. QAT (Park et al., 2022) substantially expanded upon GT models to jointly handle language and graph reasoning in question answering tasks. In the study conducted by Mohammadshahi and Henderson (2023), the G2GT model showed substantial improvements in the semantic role labelling tasks. The multitude of successful applications and extensions firmly establish Graph Transformers as a robust and adaptable framework for addressing complex challenges in language and graphs.

3 Graph-to-Graph Transformer Architecture

Our Graph-to-Graph Transformer (G2GT) architecture combines the idea of inputting graph edges into the self-attention function with the idea of predicting graph edges with an attention-like function. By encoding the graph relations into the self-attention mechanism of Transformers, the model has an appropriate linguistic bias, without imposing hard restrictions. Specifically, G2GT modifies the attention mechanism of Transformers (Vaswani et al., 2017) to input any graph. Given the input sequence $W = (x_1, x_2, ..., x_n)$, and graph relations $G = \{(x_i, x_j, l), 1 \leq i, j \leq n, l \in L\}$ (where L is the set of labels), the modified self-attention mechanism is calculated as[1]:

[1]Various alternative functions are possible for inputting relation embeddings into attention weight computations. Dufter

$$e_{ij} = \frac{1}{\sqrt{d}} \Big[x_i \boldsymbol{W^Q} (x_j \boldsymbol{W^K})^T + x_i \boldsymbol{W^Q} (r_{ij} \boldsymbol{W_1^R})^T + r_{ij} \boldsymbol{W_2^R} (x_j \boldsymbol{W^K})^T \Big] \quad (1)$$

where $r_{ij} \in \{0,1\}^{|L|}$ is a one-hot vector which specifies the type of the relation between x_i and x_j,[2] $\boldsymbol{W_1^R}, \boldsymbol{W_2^R} \in R^{|L| \times d}$ are matrices of graph relation embeddings which are learned during training, $|L|$ is the label size, and d is the size of hidden representations. The value equation of Transformer (Vaswani et al., 2017) is also modified to pass information about graph relations to the output of the attention function:

$$z_i = \sum_j \alpha_{ij} (x_j \boldsymbol{W^V} + r_{ij} \boldsymbol{W_3^R}) \quad (2)$$

where $\boldsymbol{W_3^R} \in R^{|L| \times d}$ is another learned relation embedding matrix.

To extract the explicit graph from the sequence of vectors output by the Transformer, a classification module is applied to pairs of vectors and maps them into the label space L. Initially, the module transforms each vector into distinct head and tail representations using dedicated projection matrices. Subsequently, a classifier (linear, bilinear or MLP) is applied, to map the vector pair onto predictions over the label space. Notably, each edge prediction can be computed in parallel (i.e. in a non-autoregressive manner), as predictions for each pair are independent of one another. Given the discrete nature of the output, various decoding methods can be employed to impose desired constraints on the complete output graph. These can range from straightforward head-tail order constraints, to more complex decoding algorithms such as the Minimum Spanning Tree (MST) algorithm.

Having an architecture which can both condition on graphs and predict graphs gives us the powerful ability to do iterative refinement of arbitrary graphs. Even when graph prediction is non-autoregressive, conditioning on the previously predicted graph allows the model to capture between-edge correlations like an autoregressive model. As illustrated in Figure 1, we propose **R**ecursive **N**on-autoregressive **G2GT** (RNGT), which predicts all edges of the graph in parallel, and is therefore non-autoregressive, but can still condition every edge prediction on all other edge predictions by conditioning on the previous version of the graph (using Equations 1 and 2).

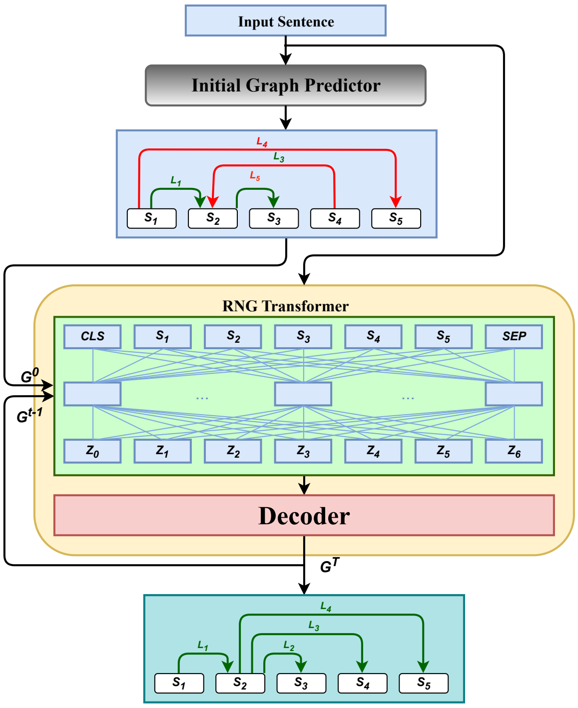

Figure 1: The Recursive Non-autoregressive Graph-to-Graph Transformer architecture.

The input to the model is the input graph W (e.g. a sequence of tokens), and the output is the final graph G^T over the same set of nodes. First, we compute an initial graph G^0 over the nodes of W, which can be done with any model. Then each recursive iteration encodes the previous graph G^{t-1} and predicts a new graph G^t. It can be formalised in terms of an encoder $\mathrm{E}^{\mathrm{RNG}}$ and a decoder $\mathrm{D}^{\mathrm{RNG}}$:

$$\begin{cases} Z^t = \mathrm{E}^{\mathrm{RNG}}(W, G^{t-1}) \\ G^t = \mathrm{D}^{\mathrm{RNG}}(Z^t) \end{cases} \quad t = 1, ..., T \quad (3)$$

where Z represents the set of vectors output by the model, and T indicates the number of refinement iterations. Note that in each step of this iterative refinement process, the G2G Transformer first computes a set of vectors which embeds the predicted graph (i.e. $\mathrm{E}^{\mathrm{RNG}}(W, G^{t-1})$), before extracting the edges of the predicted graph from this set-of-vectors embedding (i.e. $\mathrm{D}^{\mathrm{RNG}}(Z^t)$).

4 G2GT Models and Results

This section provides a more comprehensive explanation of each alternative G2GT model we have ex-

et al. (2022) provide a survey of previous proposals for relative position encoding. In ongoing work, we have found that using a relation embedding vector to reweight the dimensions in standard dot-product attention works well for some applications.

[2]This formulation can be easily extended to multi-label graphs by removing the one-hot constraint. We are investigating the most effective method for doing this.

plored, along with an outline of how we've applied these models to address various graph modelling problems. The empirical success of these models demonstrate the computational adequacy of Transformers for extracting and modelling graph structures which are central to the nature of language. The large further improvements gained by initialising with pretrained models demonstrates that Transformer pretraining encodes information about linguistic structures in its attention mechanisms.

4.1 Syntactic Parsing

Syntactic parsing is the process of analysing the grammatical structure of a sentence, including identifying the subject, verb, and object. Syntactic dependency parsing is a critical component in a variety of natural language understanding tasks, such as semantic role labelling (Henderson et al., 2013; Marcheggiani and Titov, 2017, 2020), machine translation (Chen et al., 2017), relation extraction (Zhang et al., 2018), and natural language inference (Pang et al., 2019). It is also a benchmark structured prediction task, because architectures which are not powerful enough to learn syntactic parsing cannot be computationally adequate for language understanding.

Syntactic structure is generally specified in one of two popular grammar styles, constituency parsing (i.e. phrase-structure parsing) (Manning and Schutze, 1999; Henderson, 2003, 2004; Titov and Henderson, 2007a) and dependency parsing (Nivre, 2003; Titov and Henderson, 2007b; Carreras, 2007; Nivre and McDonald, 2008; Dyer et al., 2015). There are two main approaches to compute the dependency tree: transition-based and graph-based parsers. Transition-based parsers predict the dependency graph one edge at a time through a sequence of parsing actions (Yamada and Matsumoto, 2003; Nivre and Scholz, 2004; Titov and Henderson, 2007b; Zhang and Nivre, 2011; Weiss et al., 2015; Yazdani and Henderson, 2015), and graph-based parsers compute scores for every possible dependency edge and then apply a decoding algorithm to find the highest scoring total tree (McDonald et al., 2005; Koo and Collins, 2010; Kuncoro et al., 2016; Zhou and Zhao, 2019).

In the following, we outline our proposals for using G2GT for syntactic parsing tasks.

4.1.1 Transition-based Dependency Parsing

In (Mohammadshahi and Henderson, 2020), we integrate the G2GT model with two baselines, named StateTransformer (StateTr) and SentenceTransformer (SentTr). In the former model, we directly input the parser state into the G2GT model, while the latter takes the initial sentence as the input. For better efficiency of our transition-based model, we used an alternative version of G2GT, introduced in Section 3, where the interaction of graph relations with key matrices in Equation 1 is removed. Each parser decision is conditioned on the history of previous decisions by inputting an unlabelled partially constructed dependency graph to the G2GT model. Mohammadshahi and Henderson (2020) evaluate the integrated models on the English Penn Treebank (Marcus et al., 1993), and 13 languages of Universal Dependencies Treebanks (Nivre et al., 2018).

Model	UAS	LAS
Andor et al. (2016)	94.61	92.79
StateTr	92.32	89.69
StateTr+G2GT	93.07	91.08
BERT StateTr	95.18	92.73
BERT StateTr+G2GT	**95.58**	**93.74**
BERT SentTr	95.65	93.85
BERT SentTr+G2GT	**96.06**	**94.26**

Table 1: Comparisons to the previous comparable models, including transition-based and sequence-to-sequence approaches (according to Mohammadshahi and Henderson (2020)) on English WSJ Treebank Stanford dependencies. Labelled and Unlabelled Attachment Scores (LAS,UAS) are used as evaluation metrics.

Results of our models on the Penn Treebank are shown in Table 1 (see (Mohammadshahi and Henderson, 2020) for further results on UD Treebanks). Integrating the G2GT model with the StateTr baseline achieves 9.97% LAS Relative Error Reduction (RER) improvement, which confirms the effectiveness of modelling the graph information in the attention mechanism. Furthermore, initialising our model weights with the BERT model (Devlin et al., 2019), provides significant improvement (27.65% LAS RER), which shows the compatibility of our modified attention mechanism with the latent representations learned by BERT pretraining. Integrating the G2GT model with the SentTr baseline results in a similar significant improvement (4.62% LAS RER).

4.1.2 Graph-based Dependency Parsing

The StateTr and SentTr models generate the dependency graph in an autoregressive manner, predicting each parser action conditioned on the history of parser actions. Many previous models have achieved better results with graph-based parsing methods, which use non-autoregressive computation of scores for all individual candidate dependency relations and then use a decoding method to reach the maximum scoring structure (McDonald et al., 2005; Koo and Collins, 2010; Ballesteros et al., 2016; Wang and Chang, 2016; Kuncoro et al., 2016; Zhou and Zhao, 2019). However, these models usually ignore correlations between edges while predicting the complete graph. In (Mohammadshahi and Henderson, 2021), we propose the **R**ecursive **N**on-autoregressive **G**raph-to-Graph **Tr**ansformer (RNGT) architecture, as discussed in Section 3. The RNGT architecture can be applied to any task with a sequence or graph as input and a graph over the same set of nodes as output. Here, we apply it for the syntactic dependency parsing task, and preliminary experiments showed that removing the interaction of graph relations with key vectors, in Equation 1, results in better performance and a more efficient attention mechanism. Mohammadshahi and Henderson (2021) evaluate this RNGT model on Universal Dependency (UD) Treebanks (Nivre et al., 2018), Penn Treebanks (Marcus et al., 1993), and the German CoNLL 2009 Treebank (Hajič et al., 2009) for the syntactic dependency parsing task.

Table 2 shows the results on 13 languages of UD Treebanks. First, we use UDify (Kondratyuk and Straka, 2019), the previous state-of-the-art multilingual dependency parser, as the initial parser for the RNGT model. The integrated model achieves significantly better LAS performance than the UDify model in all languages, which demonstrates the effectiveness of the RNGT model at refining a dependency graph. Then, we combine RNGT with Syntactic Transformer (SynTr), a stronger monolingual dependency parser, which has the same architecture as the RNGT model except without the graph input mechanism. The SynTr+RNGT model reaches further improvement over the strong SynTr baseline (four languages are significant), which is stronger evidence for the effectiveness of the graph refinement method. Interestingly, there is little difference between the performance with different initial parsers, implying that the RNGT model is effective enough to refine any initial graphs. In fact, even when we initialise with an empty parse, the Empty+RNGT model achieves competitive results with the other RNGT models, again confirming our powerful method of graph refinement.

4.2 Semantic Role Labelling

The semantic role labelling (SRL) task provides a shallow semantic representation of a sentence and builds event properties and relations among relevant words, and is defined in both dependency-based (Surdeanu et al., 2008) and span-based (Carreras and Màrquez, 2005; Pradhan et al., 2012) styles. Previous work (Marcheggiani and Titov, 2017; Strubell et al., 2018; Cai and Lapata, 2019; Fei et al., 2021; Zhou et al., 2020) showed that the syntactic graph helps SRL models to predict better output graphs, but finding the most effective way to incorporate the auxiliary syntactic information into SRL models was still an open question. In (Mohammadshahi and Henderson, 2023), we introduce the Syntax-aware Graph-to-Graph Transformer (SynG2G-Tr) architecture. The model conditions on the sentence's dependency structure and jointly predicts both span-based (Carreras and Màrquez, 2005) and dependency-based (Hajič et al., 2009) SRL structures. Regarding the self-attention mechanism, we remove the interaction of graph embeddings with value vectors in Equation 2, as it reaches better performance in this particular task (Mohammadshahi and Henderson, 2023).

Results for span-based SRL are shown in Table 3. Without initialising the models with BERT (Devlin et al., 2019), the SynG2G-Tr model outperforms a previous comparable state-of-the-art model (Strubell et al., 2018) in both *end-to-end* and *given-predicate* scenarios. The improvement indicates the benefit of encoding the graph information in the self-attention mechanism of Transformer with a soft bias, instead of hard-coding the graph structure into deep learning models (Marcheggiani and Titov, 2017; Strubell et al., 2018; Xia et al., 2019), as the model can still learn other attention patterns in combination with this graph knowledge. BERT (Devlin et al., 2019) initialisation results in further significant improvement in both settings, which again shows the compatibility of the G2GT modified self-attention mechanism with the latent structures learned by BERT pretraining.

4.3 Coreference Resolution

Coreference resolution (CR) is an important and complex task which is necessary for higher-level semantic representations. We show that it benefits from a graph-based global optimisation of all the coreference chains in a document.

Language	Multi UDify	Multi+Mono UDify+RNGT	Mono SynTr	Mono SynTr+RNGT	Mono Empty+RNGT
Arabic	82.88	**85.93** (+17.81%)	**86.23**	**86.31** (+0.58%)	**86.05**
Basque	80.97	87.55 (+34.57%)	87.49	**88.2** (+5.68%)	**87.96**
Chinese	83.75	89.05 (+32.62%)	89.53	**90.48** (+9.08%)	89.82
English	88.5	91.23 (+23.74%)	**91.41**	**91.52** (+1.28%)	91.23
Finnish	82.03	**91.87** (+54.76%)	**91.80**	**91.92** (+1.46%)	**91.78**
Hebrew	88.11	90.80 (+22.62%)	**91.07**	**91.32** (+2.79%)	90.56
Hindi	91.46	93.94 (+29.04%)	93.95	**94.21** (+4.3%)	93.97
Italian	93.69	94.65 (+15.21%)	**95.08**	**95.16** (+1.62%)	**94.96**
Japanese	92.08	**95.41** (+42.06%)	**95.66**	**95.71** (+1.16%)	**95.56**
Korean	74.26	**89.12** (+57.73%)	**89.29**	**89.45** (+1.5%)	**89.1**
Russian	93.13	**94.51** (+20.09%)	**94.60**	**94.47** (-2.4%)	94.31
Swedish	89.03	92.02 (+27.26%)	92.03	**92.46** (+5.4%)	**92.40**
Turkish	67.44	72.07 (+14.22%)	**72.52**	**73.08** (+2.04%)	71.99
Average	85.18	89.86	90.05	90.33	89.98

Table 2: Labelled attachment scores of monolingual (SynTr) and multilingual (UDify (Kondratyuk and Straka, 2019)) baselines, and the refined models (+RNGT) pre-trained with BERT (Devlin et al., 2019) on 13 languages of UD Treebanks. The relative error reduction after the integration is illustrated in parentheses. Bold scores are not significantly different from the best score in that row (with $\alpha = 0.01$).

Model	in-domain	out-of-domain
end-to-end		
Strubell et al. (2018)	84.99	74.66
SynG2G-Tr (w/o BERT)	**85.45**	**75.26**
+pre-training		
Strubell et al. (2018)	86.9	78.25
SynG2G-Tr	**87.57**	**80.53**
given predicate		
Strubell et al. (2018)	86.04	76.54
SynG2G-Tr (w/o BERT)	**86.50**	**77.45**
+pre-training		
Jia et al. (2022)	88.25	81.90
SynG2G-Tr	**88.93**	**83.21**

Table 3: Comparing our SynG2G-Tr with previous comparable SoTA model on CoNLL 2005 test sets for both in-domain (WSJ), and out-of-domain (Brown) sets. Scores being boldfaced means that they are significantly better.

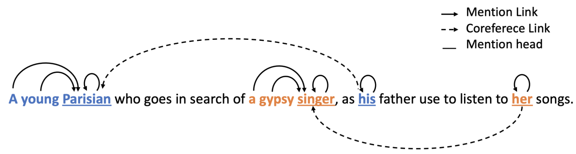

Figure 2: Example of a graph structure for coreference. Mention spans are shown in bold, and colours represent entity clusters. The mention heads are underlined.

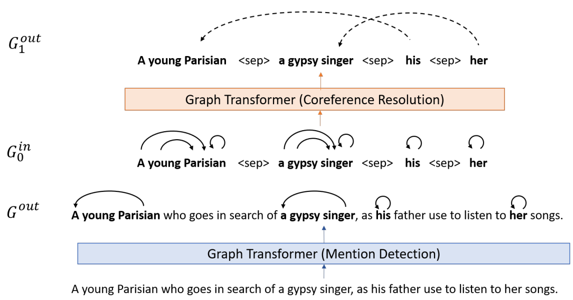

Figure 3: Example of iterations with G2GT in two stages.

4.3.1 CR Task Definition and Background

Coreference resolution is the task of linking all linguistic expressions in a text that refer to the same entity. Solutions for this task involve three parts: mention-detection (Yu et al., 2020; Miculicich and Henderson, 2020), classification or ranking of mentions, and finally reconciling the decisions to create entity chains. These approaches fall within three principal categories: mention-pair models which perform binary decisions (McCarthy and Lehnert, 1995; Aone and William, 1995; Soon et al., 2001), entity-based models which focus on maintaining single underlying entity representation, contrasting the independent pair-wise decisions of mention-pair approaches (Clark and Manning, 2015, 2016), and ranking models which aim at ranking the possible antecedents of each mention instead of making binary decisions (Wiseman et al., 2016). A limitation of these methods lies in their bottom-up construction, resulting in an underutilisation of comprehensive global information regarding coreference links among all mentions in individual decisions. Furthermore, these methods tend to exhibit significant complexity. Modelling of coreference resolution as a graph-based approach offer an alternative to deal with these limitations.

4.3.2 Iterative Graph-based CR

Miculicich and Henderson (2022) proposed a novel approach to modelling coreference resolution, treating it as a graph problem. In this framework, the tokens within the text serve as nodes, and the connections between them signify coreference links (see Figure 2). Given a document $D=[x_1,...,x_N]$ with length N, the coreference graph is formally defined as the matrix $G \subset \mathbb{N}^{N\times N}$, which represents the relationships between tokens. Specifically, the relationship type between any two tokens, x_i and x_j, is labelled as $g_{i,j} \in \{0,1,2\}$ for the three distinct relation types: (0) no link, (1) mention link, and (2) coreference link.

The primary objective of this approach is to learn the conditional probability distribution $p(G|D)$. To achieve this, an iterative refinement strategy is employed, which captures interdependencies among relations. The model iterates over the same document D for a total of T iterations. In each iteration t, the predicted coreference graph G_t is conditioned on the previous prediction, denoted as G_{t-1}. Thus, the conditional probability distribution of the model is defined as follows:

$$p(G^t|D,G^{t-1}) = \prod_{i=1}^{N}\prod_{j=1}^{i} p(g_{i,j}|D,G^{t-1}) \quad (4)$$

The proposed model operates on two levels of representation. In each iteration, it predicts the entire graph. However, during the first iteration, the model focuses on predicting edges that pinpoint mention spans, given that coreferent links only have relevance when mentions are detected. From the second iteration, both mention links, and coreference links are refined. This iterative strategy permits the model to enhance mention-related decisions based on coreference resolutions, and vice versa. This framework utilises iterative graph refinement as a substitute for conventional pipeline architectures in multi-level deep learning models. The iterative process concludes either when the graph no longer undergoes changes or when a predetermined maximum iteration count is attained (see Figure 3).

Ideally, encoding the entirety of the document in a single pass would be optimal. However, in practical scenarios, a constraint on maximum length arises due to limitations in hardware memory capacity. To address this challenge, Miculicich and Henderson (2022) introduce two strategies: overlapping windows and reduced document approach. In the latter strategy, mentions are identified during an initial iteration with a focus on optimising recall, as previously suggested in (Miculicich and Henderson, 2020). Only the representations of these identified spans are subsequently used as inputs for the following iterations.

Miculicich and Henderson (2022) conducted experiments on the CoNLL 2012 corpus (Pradhan et al., 2012) and showed improvements over relevant baselines and previous state-of-the-art methods, summarised in Table 4. We compare our model with three baselines: Lee et al. (2017) proposed the first end-to-end model for coreference resolution; Lee et al. (2018) extended the previous model by introducing higher order inference; and Xu and Choi (2020) used the span based pretrained model SpanBERT (Joshi et al., 2020). The 'Baseline' of Lee et al. (2018) uses ELMo (Peters et al., 2018) to obtain token representations, so versions of this Baseline which use 'BERT-large' (Joshi et al., 2019) and 'SpanBERT-large' (Joshi et al., 2020) as their pretrained models, are directly comparable to our 'G2GT BERT-large' and 'G2GT SpanBERT-large' models, respectively.

These results show that coreference resolution benefits from making global coreference decisions using document-level information, as supported by the G2GT architecture. Our model achieves its optimal solution within a maximum of three iterations. Notably, due to the model's ability to predict the entire graph in a single iteration, its computational complexity is lower compared to that of the baseline approaches.

5 Discussion

The empirical success of Graph-to-Graph Transformers on modelling these various graph structures helps us understand how Transformers model language. This success demonstrates that Transformers are computationally adequate for modelling linguistic structures, which are central to the nature of language. The reliance of these G2GT models on using self-attention mechanisms to extract and encode these graph relations shows that self-attention is crucial to how Transformers can do this modelling. The large improvements gained by initialising with pretrained models indicates that pretrained Transformers are in fact using the same mechanisms to learn about this linguistic structure, but in an unsupervised fashion.

These insights into pretrained Transformers give us a better understanding of the current generation of Large Language Models (LLMs). It is not that these models do not need linguistic structure (since their attention mechanisms do learn it); it is that these models do not need supervised learning of linguistic structure. But perhaps in a

Model	MUC			B^3			CEAF$_{\phi_4}$			
	P	R	F1	P	R	F1	P	R	F1	Avg. F1
Lee et al. (2017)	78.4	73.4	75.8	68.6	61.8	65.0	62.7	59.0	60.8	67.2
Xu and Choi (2020)	**85.9**	85.5	85.7	79.0	78.9	79.0	**76.7**	75.2	75.9	80.2
Baseline (Lee et al., 2018)	81.4	79.5	80.4	72.2	69.5	70.8	68.2	67.1	67.6	73.0
+ BERT-large (Joshi et al., 2019)	84.7	82.4	83.5	76.5	74.0	75.3	74.1	69.8	71.9	76.9
+ SpanBERT-large (Joshi et al., 2020)	85.8	84.8	85.3	78.3	77.9	78.1	76.4	74.2	75.3	79.6
G2GT BERT-large *reduced*	84.7	83.1	83.9	76.8	74.0	75.4	75.3	70.1	72.6	77.3
G2GT SpanBERT-large *reduced*	**85.9**	**86.0***†	**85.9***	**79.3***	**79.4***†	**79.3***	76.4	**75.9***	**76.1***	**80.5***

Table 4: Evaluation of CR on the test set (CoNLL 2012) in terms of precision (P), recall (R) and F1 score for three metrics, as well as the average F1 over metrics. $*$ significant at $p < 0.01$ compared to (Joshi et al., 2020), † significant at $p < 0.05$ compared to (Xu and Choi, 2020).

low-resource scenario LLMs would benefit from the inductive bias provided by supervised learning of linguistic structures, such as for many of the world's languages other than English. And these insights are potentially relevant to the issues of interpretability and controllability of LLMs.

These insights are also relevant for any applications which could benefit from integrating text with structured representations. Our current work investigates jointly embedding text and parts of a knowledge base in a single G2GT model, providing a way to integrate interpretable structured knowledge with knowledge in text. Such representations would be useful for information extraction, question answering and information retrieval, amongst many other applications. Other graphs we might want to model with a Transformer and integrate with text include hyperlink graphs, citation graphs, and social networks. An important open problem with such models is the scale of the resulting Transformer embedding.

6 Conclusion and Future Work

The Graph-to-Graph Transformer architecture makes explicit the implicit graph processing abilities of Transformers, but further research is needed to fully leverage the potential of G2GT.

6.1 Conclusions

The success of the above models of a variety linguistic structures shows that Transformers are underlyingly graph-to-graph models, not limited to sequence-to-sequence tasks. The G2GT architecture with its RNGT method provides an effective way to exploit this underlying ability when modelling explicit graphs, effectively integrating them with the implicit graphs learned by pre-trained Transformers. Inputting graph relations as features to the self-attention mechanism enables the information input to the model to be steered by domain-specific knowledge or desired outcomes but still learned by the Transformer, opening up the possibility for a more tailored and customised encoding process. Predicting graph relations with attention-like functions and then re-inputting them for iterative refinement, encodes the input, predicted and latent graphs in a single joint Transformer embedding which is effective for making global decisions about structure in a text.

6.2 Future Work

One topic of research where explicit graphs are indispensable is knowledge graphs. Knowledge needs to be interpretable, so that it can be audited, edited, and learned by people. And it needs to be integrated with existing knowledge graphs. Our current work uses G2GT to integrate knowledge graphs with knowledge conveyed by text.

One of the limitations of the models discussed in this paper is that the set of nodes in the output graph needs to be (a subset of) the nodes in the input graph. General purpose graph-to-graph mappings would require also predicting a set of new nodes in the output graph. One natural solution would be autoregressive prediction of one node at a time, as is done for text generation, but an exciting alternative would be to use methods from non-autoregressive text generation in combination with our iterative refinement method RNGT.

The excellent performance of the models presented in this paper suggest that many more problems can be successfully formulated as graph-to-graph problems and modelled with G2GT, in NLP and beyond. The code for G2GT and RNGT is open-source and publicly available at `https://github.com/idiap/g2g-transformer`.

Acknowledgement

We would like to especially thank the Swiss National Science Foundation for funding this work, under grants 200021E_189458, CRSII5_180320, and 200021_178862. We would also like to thank other members of the the Natural Language Understanding group at Idiap Research Institute for useful discussion and feedback, including Florian Mai, Rabeeh Karimi, Andreas Marfurt, Melika Behjati, and Fabio Fehr.

References

Uri Alon and Eran Yahav. 2021. On the bottleneck of graph neural networks and its practical implications. In *International Conference on Learning Representations*.

Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally normalized transition-based neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2442–2452, Berlin, Germany. Association for Computational Linguistics.

Chinatsu Aone and Scott William. 1995. Evaluating automated and manual acquisition of anaphora resolution strategies. In *33rd Annual Meeting of the Association for Computational Linguistics*, pages 122–129, Cambridge, Massachusetts, USA. Association for Computational Linguistics.

Miguel Ballesteros, Yoav Goldberg, Chris Dyer, and Noah A. Smith. 2016. Training with exploration improves a greedy stack LSTM parser. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2005–2010, Austin, Texas. Association for Computational Linguistics.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2013. Spectral networks and locally connected networks on graphs. *CoRR*, abs/1312.6203.

Rui Cai and Mirella Lapata. 2019. Semi-supervised semantic role labeling with cross-view training. 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)*, pages 1018–1027, Hong Kong, China. Association for Computational Linguistics.

Xavier Carreras. 2007. Experiments with a higher-order projective dependency parser. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 957–961, Prague, Czech Republic. Association for Computational Linguistics.

Xavier Carreras and Lluís Màrquez. 2005. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 152–164, Ann Arbor, Michigan. Association for Computational Linguistics.

Huadong Chen, Shujian Huang, David Chiang, and Jiajun Chen. 2017. Improved neural machine translation with a syntax-aware encoder and decoder. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Peng Chen. 2021. PermuteFormer: Efficient relative position encoding for long sequences. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10606–10618, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy J. Colwell, and Adrian Weller. 2020. Rethinking attention with performers. *ArXiv*, abs/2009.14794.

Kevin Clark and Christopher D. Manning. 2015. Entity-centric coreference resolution with model stacking. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1405–1415, Beijing, China. Association for Computational Linguistics.

Kevin Clark and Christopher D. Manning. 2016. Improving coreference resolution by learning entity-level distributed representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653, Berlin, Germany. Association for Computational Linguistics.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Cameron Diao and Ricky Loynd. 2022. Relational attention: Generalizing transformers for graph-structured tasks. *ArXiv*, abs/2210.05062.

Philipp Dufter, Martin Schmitt, and Hinrich Schütze. 2022. Position information in transformers: An overview. *Computational Linguistics*, 48(3):733–763.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 334–343, Beijing, China. Association for Computational Linguistics.

Luis Espinosa Anke, Alexander Shvets, Alireza Mohammadshahi, James Henderson, and Leo Wanner. 2022. Multilingual extraction and categorization of lexical collocations with graph-aware transformers. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 89–100, Seattle, Washington. Association for Computational Linguistics.

Hao Fei, Fei Li, Bobo Li, and Donghong Ji. 2021. Encoder-decoder based unified semantic role labeling with label-aware syntax. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12794–12802.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR.

Marco Gori, Gabriele Monfardini, and Franco Scarselli. 2005. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734. IEEE.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 1–18, Boulder, Colorado. Association for Computational Linguistics.

James Henderson. 2003. Inducing history representations for broad coverage statistical parsing. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 103–110.

James Henderson. 2004. Discriminative training of a neural network statistical parser. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 95–102, Barcelona, Spain.

James Henderson. 2020. The unstoppable rise of computational linguistics in deep learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6294–6306, Online. Association for Computational Linguistics.

James Henderson, Paola Merlo, Ivan Titov, and Gabriele Musillo. 2013. Multilingual joint parsing of syntactic and semantic dependencies with a latent variable model. *Computational Linguistics*, 39(4):949–998.

John Hewitt, Kawin Ethayarajh, Percy Liang, and Christopher Manning. 2021. Conditional probing: measuring usable information beyond a baseline. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1626–1639, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

John J Hopfield. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558.

Zhiheng Huang, Davis Liang, Peng Xu, and Bing Xiang. 2020. Improve transformer models with better relative position embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3327–3335, Online. Association for Computational Linguistics.

Zixia Jia, Zhaohui Yan, Haoyi Wu, and Kewei Tu. 2022. Span-based semantic role labeling with argument pruning and second-order inference. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):10822–10830.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld. 2019. BERT for coreference resolution: Baselines and analysis. 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)*, pages 5803–5808, Hong Kong, China. Association for Computational Linguistics.

Pei Ke, Haozhe Ji, Yu Ran, Xin Cui, Liwei Wang, Linfeng Song, Xiaoyan Zhu, and Minlie Huang. 2021. JointGT: Graph-text joint representation learning for text generation from knowledge graphs. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2526–2538, Online. Association for Computational Linguistics.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing Universal Dependencies universally. 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)*, pages 2779–2795, Hong Kong, China. Association for Computational Linguistics.

Terry Koo and Michael Collins. 2010. Efficient third-order dependency parsers. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1–11.

Adhiguna Kuncoro, Miguel Ballesteros, Lingpeng Kong, Chris Dyer, and Noah A. Smith. 2016. Distilling an ensemble of greedy dependency parsers into one MST parser. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1744–1753, Austin, Texas. Association for Computational Linguistics.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692, New Orleans, Louisiana. Association for Computational Linguistics.

Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard S. Zemel. 2015. Gated graph sequence neural networks. *CoRR*, abs/1511.05493.

Antoine Liutkus, Ondřej Cífka, Shih-Lun Wu, Umut Simsekli, Yi-Hsuan Yang, and Gaël Richard. 2021. Relative positional encoding for transformers with linear complexity. In *International Conference on Machine Learning*.

Christopher Manning and Hinrich Schutze. 1999. *Foundations of statistical natural language processing*. MIT press.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, Copenhagen, Denmark. Association for Computational Linguistics.

Diego Marcheggiani and Ivan Titov. 2020. Graph convolutions over constituent trees for syntax-aware semantic role labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3915–3928, Online. Association for Computational Linguistics.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Joseph F. McCarthy and Wendy G. Lehnert. 1995. Using decision trees for coreference resolution. In *International Joint Conference on Artificial Intelligence*.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 91–98, Ann Arbor, Michigan. Association for Computational Linguistics.

Julian Michael, Jan A. Botha, and Ian Tenney. 2020. Asking without telling: Exploring latent ontologies in contextual representations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6792–6812, Online. Association for Computational Linguistics.

Lesly Miculicich and James Henderson. 2020. Partially-supervised mention detection. In *Proceedings of the Third Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 91–98, Barcelona, Spain (online). Association for Computational Linguistics.

Lesly Miculicich and James Henderson. 2022. Graph refinement for coreference resolution. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2732–2742, Dublin, Ireland. Association for Computational Linguistics.

Alireza Mohammadshahi and James Henderson. 2020. Graph-to-graph transformer for transition-based dependency parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3278–3289, Online. Association for Computational Linguistics.

Alireza Mohammadshahi and James Henderson. 2021. Recursive Non-Autoregressive Graph-to-Graph Transformer for Dependency Parsing with Iterative Refinement. *Transactions of the Association for Computational Linguistics*, 9:120–138.

Alireza Mohammadshahi and James Henderson. 2023. Syntax-aware graph-to-graph transformer for semantic role labelling. In *Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023)*, pages 174–186, Toronto, Canada. Association for Computational Linguistics.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the Eighth International Conference on Parsing Technologies*, pages 149–160, Nancy, France.

Joakim Nivre, Mitchell Abrams, Željko Agić, Lars Ahrenberg, Lene Antonsen, Katya Aplonova, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Victoria Basmov, John Bauer, Sandra Bellato, Kepa Bengoetxea, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Rogier Blokland, Victoria Bobicev, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Savas Cetin, Fabricio Chalub, Jinho Choi, Yongseok Cho, Jayeol Chun, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Arantza Diaz de Ilarraza, Carly Dickerson, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Tomaž Erjavec, Aline Etienne, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta Gonzáles Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Nizar Habash, Jan Hajič, Jan Hajič jr., Linh Hà Mỹ, Na-Rae Han, Kim Harris, Dag Haug, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Jena Hwang, Radu Ion, Elena Irimia, Ọlájídé Ishola, Tomáš Jelínek, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşıkara, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Kamil Kopacewicz, Natalia Kotsyba, Simon Krek, Sookyoung Kwak, Veronika Laippala, Lorenzo Lambertino, Lucia Lam, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phương Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Keying Li, KyungTae Lim, Nikola Ljubešić, Olga Loginova, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Gustavo Mendonça, Niko Miekka, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Keiko Sophie Mori, Shinsuke Mori, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Lương Nguyễn Thị, Huyền Nguyễn Thị Minh, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Adédayọ̀ Olúòkun, Mai Omura, Petya Osenova, Robert Östling, Lilja Øvrelid, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Siyao Peng, Cenel-Augusto Perez, Guy Perrier, Slav Petrov, Jussi Piitulainen, Emily Pitler, Barbara Plank, Thierry Poibeau, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Andriela Rääbis, Alexandre Rademaker, Loganathan Ramasamy, Taraka Rama, Carlos Ramisch, Vinit Ravishankar, Livy Real, Siva Reddy, Georg Rehm, Michael Rießler, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Mykhailo Romanenko, Rudolf Rosa, Davide Rovati, Valentin Roșca, Olga Rudina, Jack Rueter, Shoval Sadde, Benoît Sagot, Shadi Saleh, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Baiba Saulīte, Yanin Sawanakunanon, Nathan Schneider, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Muh Shohibussirri, Dmitry Sichinava, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Isabela Soares-Bastos, Carolyn Spadine, Antonio Stella, Milan Straka, Jana Strnadová, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Dima Taji, Yuta Takahashi, Takaaki Tanaka, Isabelle Tellier, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Sowmya Vajjala, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Lars Wallin, Jing Xian Wang, Jonathan North Washington, Seyi Williams, Mats Wirén, Tsegay Woldemariam, Taksum Wong, Chunxiao Yan, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, Manying Zhang, and Hanzhi Zhu. 2018. Universal dependencies 2.3. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Joakim Nivre and Ryan McDonald. 2008. Integrating graph-based and transition-based dependency parsers. In *Proceedings of ACL-08: HLT*, pages 950–958, Columbus, Ohio. Association for Computational Linguistics.

Joakim Nivre and Mario Scholz. 2004. Deterministic dependency parsing of English text. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 64–70, Geneva, Switzerland. COLING.

Kenta Oono and Taiji Suzuki. 2020. Graph neural networks exponentially lose expressive power for node classification. In *International Conference on Learning Representations*.

Deric Pang, Lucy H. Lin, and Noah A. Smith. 2019. Improving natural language inference with a pretrained parser.

Jinyoung Park, Hyeong Kyu Choi, Juyeon Ko, Hyeon ju Park, Ji-Hoon Kim, Jisu Jeong, Kyungmin Kim, and Hyunwoo J. Kim. 2022. Relation-aware language-graph transformer for question answering. *ArXiv*, abs/2212.00975.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018*

Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038, Brussels, Belgium. Association for Computational Linguistics.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. *ArXiv*, abs/2104.09864.

Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008. The CoNLL 2008 shared task on joint parsing of syntactic and semantic dependencies. In *CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 159–177, Manchester, England. Coling 2008 Organizing Committee.

Ivan Titov and James Henderson. 2007a. Constituent parsing with incremental sigmoid belief networks. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 632–639, Prague, Czech Republic. Association for Computational Linguistics.

Ivan Titov and James Henderson. 2007b. A latent variable model for generative dependency parsing. In *Proceedings of the Tenth International Conference on Parsing Technologies*, pages 144–155, Prague, Czech Republic. Association for Computational Linguistics.

Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M. Bronstein. 2021. Understanding over-squashing and bottlenecks on graphs via curvature. *ArXiv*, abs/2111.14522.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *International Conference on Learning Representations*. Accepted as poster.

Wenhui Wang and Baobao Chang. 2016. Graph-based dependency parsing with bidirectional LSTM. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2306–2315, Berlin, Germany. Association for Computational Linguistics.

Boris Weisfeiler and Andrei Leman. 1968. The reduction of a graph to canonical form and the algebra which appears therein. *nti, Series*, 2(9):12–16.

David Weiss, Chris Alberti, Michael Collins, and Slav Petrov. 2015. Structured training for neural network transition-based parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference* on Natural Language Processing (Volume 1: Long Papers)*, pages 323–333, Beijing, China. Association for Computational Linguistics.

Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. 2016. Learning global features for coreference resolution. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 994–1004, San Diego, California. Association for Computational Linguistics.

Zhiyong Wu, Yun Chen, Ben Kao, and Qun Liu. 2020. Perturbed masking: Parameter-free probing for analyzing and interpreting BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4166–4176, Online. Association for Computational Linguistics.

Qingrong Xia, Zhenghua Li, Min Zhang, Meishan Zhang, Guohong Fu, Rui Wang, and Luo Si. 2019. Syntax-aware neural semantic role labeling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7305–7313.

Benfeng Xu, Quan Wang, Yajuan Lyu, Yong Zhu, and Zhendong Mao. 2021. Entity structure within and throughout: Modeling mention dependencies for document-level relation extraction. In *AAAI Conference on Artificial Intelligence*.

Liyan Xu and Jinho D. Choi. 2020. Revealing the myth of higher-order inference in coreference resolution. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8527–8533, Online. Association for Computational Linguistics.

Hiroyasu Yamada and Yuji Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of the Eighth International Conference on Parsing Technologies*, pages 195–206, Nancy, France.

Jingfeng Yang, Aditya Gupta, Shyam Upadhyay, Luheng He, Rahul Goel, and Shachi Paul. 2022. TableFormer: Robust transformer modeling for table-text encoding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 528–537, Dublin, Ireland. Association for Computational Linguistics.

Majid Yazdani and James Henderson. 2015. Incremental recurrent neural network dependency parser with search-based discriminative training. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 142–152, Beijing, China. Association for Computational Linguistics.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. 2021. Do transformers really perform bad for graph representation? In *Neural Information Processing Systems*.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Neural mention detection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1–10, Marseille, France. European Language Resources Association.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models.

Yue Zhang and Joakim Nivre. 2011. Transition-based dependency parsing with rich non-local features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 188–193, Portland, Oregon, USA. Association for Computational Linguistics.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215, Brussels, Belgium. Association for Computational Linguistics.

Junru Zhou, Zuchao Li, and Hai Zhao. 2020. Parsing all: Syntax and semantics, dependencies and spans. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4438–4449, Online. Association for Computational Linguistics.

Junru Zhou and Hai Zhao. 2019. Head-Driven Phrase Structure Grammar parsing on Penn Treebank. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2396–2408, Florence, Italy. Association for Computational Linguistics.