Going "Deeper": Structured Sememe Prediction via Transformer with Tree Attention

Yining Ye[1], **Fanchao Qi**[1], **Zhiyuan Liu**[1,2,3,4*], **Maosong Sun**[1,2,3*]
[1]Dept. of Comp. Sci. & Tech., Institute for AI, Tsinghua University, Beijing, China
Beijing National Research Center for Information Science and Technology
[2]Institute Guo Qiang, Tsinghua University, Beijing, China
[3]International Innovation Center of Tsinghua University, Shanghai, China
[4]Beijing Academy of Artificial Intelligence
{yeyn19,qfc17}@mails.tsinghua.edu.cn,
{liuzy,sms}@tsinghua.edu.cn

Abstract

Sememe knowledge bases (SKBs), which annotate words with the smallest semantic units (i.e., *sememes*), have proven beneficial to many NLP tasks. Building an SKB is very time-consuming and labor-intensive. Therefore, some studies have tried to automate the building process by predicting sememes for the unannotated words. However, all existing sememe prediction studies ignore the hierarchical structures of sememes, which are important in the sememe-based semantic description system. In this work, we tackle the structured sememe prediction problem for the first time, which is aimed at predicting a sememe tree with hierarchical structures rather than a set of sememes. We design a sememe tree generation model based on Transformer with an adjusted attention mechanism, which shows its superiority over the baseline methods in experiments. We also conduct a series of quantitative and qualitative analyses of the effectiveness of our model. All the code and data of this paper are available at https://github.com/thunlp/STG.

1 Introduction

A word is the fundamental element of natural languages, but its meaning can be further divided. To explore semantics atomically, linguists define a *sememe* as the minimum semantic unit (Bloomfield, 1926). It is even believed that the meanings of all words in any language can be represented by a limited set of sememes, which is closely related to the idea of semantic primitives (Wierzbicka, 1996).

HowNet (Dong and Dong, 2006) is the most well-known sememe knowledge base (SKB). It comprises more than 100,000 English and Chinese words and phrases manually annotated by about 2,000 sememes that are defined by linguistic experts. Multiple senses of a polysemous word are independently annotated, and the sememes annotated to a sense are hierarchically organized as a sememe tree. Figure 1 illustrates the sememe annotations of two English words in HowNet.

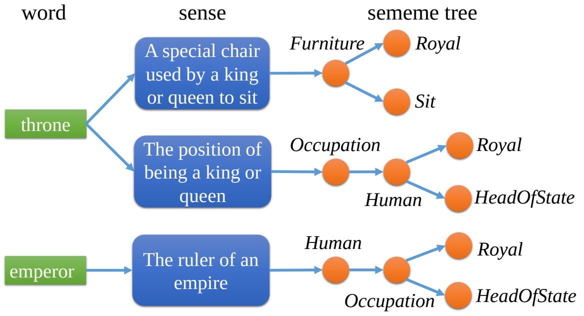


Figure 1: Sememe annotations of the words "throne" and "emperor" in HowNet.

Different from other lexical knowledge bases, SKBs like HowNet define words intensionally with a limited set of semantic units (sememes), thus have some unique strengths. For example, SKBs can be combined with neural network models smoothly by regarding sememes as the external semantic labels of words (Qi et al., 2019; Qin et al., 2020). Moreover, thanks to the limitedness of sememes, SKBs have been proven very useful in the low-data regimes, e.g., improving the representation learning of rare words by transferring knowledge from frequent words via sememes (Niu et al., 2017). As a result, SKBs have been widely utilized in many NLP tasks (Qi et al., 2021b).

However, most languages have no SKBs like HowNet, and it is too expensive to manually build an SKB for a new language from scratch.[1] In addition, even for the languages covered in HowNet (English and Chinese), new words are emerging every day and the meanings of existing words keep changing. It is also costly to expand and update

[*] Corresponding author.

[1] It took several linguistic experts more than two decades to build HowNet.

HowNet. To solve these issues, a series of studies have been conducted, trying to automatically predict sememes for monolingual or cross-lingual words (Xie et al., 2017; Jin et al., 2018; Qi et al., 2018; Du et al., 2020; Lyu et al., 2021). For simplicity, all previous sememe prediction studies ignore the hierarchical structures of sememes. They simplify sememe prediction as a multi-label classification task, and their models output a structureless set of sememes.

However, the structures of sememes are very important. For one thing, the structural information is indispensable in the sememe-based semantic description system, as it carries semantics, and branches of sememe trees stand for the relations of sememes. As shown in Figure 1, the difference in sememe structure results in the different meanings of the second sense of "throne" and "emperor", although they have four identical sememes. For another, the structures of sememes are necessary for many sememe-based applications (Liu and Li, 2002; Zhu et al., 2019; Liu et al., 2020).

In this paper, we try to tackle structured sememe prediction, which is aimed at predicting sememes together with their hierarchical structures rather than the structureless sememes only. This task is essentially a kind of tree generation task but is more challenging than other tree generation tasks. First, the size of its node type is more than 2,000 (i.e., over 2,000 sememes), which is much larger than that of most tree generation tasks, e.g., less than 100 for code generation and semantic parsing (rab). Second, the structures of sememe trees are extremely diverse — almost any sememe can be the child node of another sememe, and one sememe node can have an arbitrary number of children. Many of the existing tree prediction methods depend on the certain number of children of a node and perform strongly correlated with the number of candidates (Yin and Neubig, 2017), thus are not applicable.

To handle this difficult task, we conduct further formalization. Different from most structureless sememe prediction studies whose input is merely a word, inspired by Du et al. (2020), we regard a sentence of definition as the input, and the task is formalized as a sequence-to-tree task. We do this for two reasons. First, sememe prediction can be conducted at the sense level (one definition corresponds to one sense of a word). Second, definitions can provide more useful information than single words for structured sememe prediction.

Further, we propose a model based on Transformer (Vaswani et al., 2017) especially designed for the task of sememe tree generation (STG). We decompose the attention in Transformer into two parts that capture the semantic similarity and topological relations between sememes, respectively, in order to better represent the characteristics of sememe trees. Experimental results show that our method outperforms baseline methods including the vanilla tree Transformer model. We also conduct quantitative and qualitative analyses of the results of our method.

2 Related Work

2.1 Sememe Knowledge Base

As a kind of special lexical knowledge base, SKBs represented by HowNet have been widely explored in various NLP applications, including word representation learning (Niu et al., 2017), word sense disambiguation (Hou et al., 2020), language modeling (Gu et al., 2018), reverse dictionary (Zhang et al., 2020b), textual adversarial and backdoor attacks (Zang et al., 2020; Qi et al., 2021c), etc.

Meanwhile, some studies focus on automating the process of expanding and constructing SKBs. They propose different methods to automatically predict sememes for words. Xie et al. (2017) present the task of lexical sememe prediction and propose two simple but effective methods that are based on collaborative filtering and matrix factorization, respectively. Jin et al. (2018) and Lyu et al. (2021) utilize the Chinese character and glyph information in lexical sememe prediction and achieve higher performance. Du et al. (2020) introduce dictionary definitions into sememe prediction and find that the abundant semantic information in definitions is very beneficial to sememe prediction. But they do not conduct sense-level sememe prediction. They simply concatenate the definitions of multiple senses of a word and predict the combined sememe set for the word.

The above studies use the sememe annotations of existing words in HowNet to predict sememes for new words, aiming to expand HowNet. Some studies try to construct SKBs for new languages automatically. Qi et al. (2018) present the task of cross-lingual lexical sememe prediction, which predicts sememes for words in a new language by bilingual word embedding alignment of a HowNet-covered language and a new language. Qi et al. (2020) propose to build a multilingual SKB based on BabelNet, a multilingual encyclopedia dictionary (Navigli and Ponzetto, 2012). BabelNet is composed of BabelNet synsets, each of which contains multilingual synonyms, e.g., *hello* (English), 你好 (Chinese) and *bonjour* (French) are included in one BabelNet synset. The multilingual synonyms in a synset convey the same meaning and should have the same sememe annotations. Therefore, they propose the task of sememe prediction for BabelNet synsets, hoping that if all synsets are annotated with sememes, all words in over 200 languages in BabelNet would obtain sememe annotations. Moreover, the sememe annotations are independently annotated to senses, because a synset corresponds to a sense. Following Qi et al. (2020), Qi et al. (2022) further utilize multilingual and multimodal information in BabelNet to improve the performance of sememe prediction for BabelNet synsets.

In addition, Qi et al. (2021a) make an attempt to construct an SKB based on a dictionary fully automatically. They regard the words in the controlled defining vocabulary of a dictionary as sememes rather than use the existing sememe set of HowNet.

Although achieving satisfactory sememe prediction results, all these studies ignore the hierarchical structures of sememes. This work is the first attempt to conduct structured sememe prediction.

2.2 Tree Generation

Structured sememe prediction is a kind of tree generation task. Some tree generation tasks have been widely explored, such as code generation (rab; Yin and Neubig, 2017; Sun et al., 2020; Nguyen et al., 2019), semantic parsing (Shiv and Quirk, 2019; Li et al., 2020) and math word problem solving (Liu et al., 2019; Zhang et al., 2020a; Wu et al., 2021). However, as explained in §1, sememe tree generation is more challenging than these tasks because of its large size of node types and a vast variety of structures.

Quite a few tree generation studies use the sequence modeling models represented by recurrent neural networks, especially LSTM (Hochreiter and Schmidhuber, 1997), and achieve great performance (Zaremba and Sutskever, 2014; Allamanis et al., 2016). Recently, with the widespread use of Transformer in sequence modeling, some studies have shown that Transformer-based models also perform well on tree generation and are more parallelizable to deal a large amount of data (Shiv and Quirk, 2019; Nguyen et al., 2019; Zugner et al., 2021). Therefore, we also design our sememe tree generation model based on Transformer.

3 Methodology

In this section, we first detail two straightforward sememe tree generation (STG) models, which will serve as the baselines. Then, we describe the modification of tree attention and introduce a novel STG model.

3.1 Neighbor-based STG (NSTG)

A sememe tree can be divided into multiple sememe paths from the root node to leaf nodes. Assuming different sememe paths are independent, the probability of generating a sememe tree can be formalized as:

$$P(T|w) = \prod_{S \in T} P(S|w), \tag{1}$$

where T refers to the sememe tree of the synset w, and S denotes a sememe path in T.

Using the multiplicative theorem of probability, the probability of each sememe path is formalized as:

$$P(S|w) = \prod_{i=1}^{N_S} P(s_i|w, S_{0:i-1}), \tag{2}$$

where N_s is the length of S, s_i is the $i-th$ sememe of S, and $S_{0:i-1}$ refers to the previous path from the beginning token `START` to the $(i-1)$-th sememe of S, where `START` is added as the root node of a sememe tree.

With the Markov assumption, we further decompose a sememe path into parent-child sememe pairs. Generating a child sememe based on a father sememe is the atomic step of generating a sememe path:

$$P(S) = \prod_{i=1}^{N_S} P(s_i|w, s_{i-1}), \tag{3}$$

Inspired by Xie et al. (2017), we assume that similar words should share similar sememe tree structures and we can apply collaborative filtering (Xie et al., 2017) to the STG task and propose the Neighbor-based STG (NSTG) model.

Specifically, for each sememe pair $e_i = (s_{i-1}, s_i)$, the non-normalized generation proba-

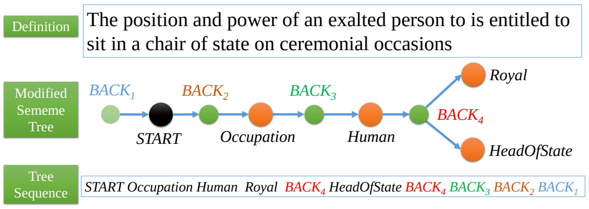

Definition
The position and power of an exalted person to is entitled to sit in a chair of state on ceremonial occasions
Modified Sememe Tree
BACK1
BACK2
BACK3
Royal
BACK4
START
Occupation
Human
HeadOfState
Tree Sequence
START Occupation Human Royal BACK4 HeadOfState BACK4 BACK3 BACK2 BACK1

Figure 2: Definition and sememe tree sequence of "bn:00077087n" in BabalNet

bility can be approximated as:

$$\hat{P}(s_i|w, s_{i-1}) = \sum_{w_j} sim(w_j, w) \times M_{j,e_i} \times d^{r_i}, \tag{4}$$

where $sim(w_j, w)$ measures the similarity between two words (senses), based on the embeddings of the two words' definitions from BERT (Devlin et al., 2019). M_{j,e_i} indicates whether the synset w_j possesses the sememe pair e_i. r_j is the descending rank of the similarity. $d \in (0, 1)$ is a hyper-parameter, which can be viewed as the declined confidence factor that helps the model concentrate on the most similar words. We use sigmoid as the normalization strategy.

We also adopt the beam search algorithm to generate sememe paths. The key point in beam search is to design a well-performed generation function at each search step.

3.2 Transformer-based STG (TSTG)

NSTG model is simple and efficient because it does not require extra training. Nevertheless, the generalization ability of NSTG is limited to the representative ability of sentence encoding. And it fails to utilize the sequential information in the generated sememe paths, which are of critical importance in the STG task. To address this issue, we can follow previous tree generation studies and use a Transformer model to learn and decode hierarchical sememe structures. This method is named Transformer-based STG (TSTG).

The normal Transformer architecture accepts sequential inputs. Therefore, we need to convert trees into sequences. We linearize sememe trees by the pre-order depth-first traversal. However, the count of branches of a node is not certain in STG, so we use a special `BACK` token to represent the back and eventually get a one-to-one mapping from sememe tree to sememe tree sequence. An example of the sememe tree sequence is shown in Figure 2.

We decompose the step of STG into repeatedly sememe generation and `BACK` token generation, ending with the depth going back to 0.

3.3 Tree-attention Transformer Model (TaSTG)

The above method enables transformer architecture to generate trees. However, it suffers some problems.

Problems of Attention Computation

Normal attention in Transformer is formalized as:

$$\alpha_{ij} = \frac{\left((\boldsymbol{w}_i + \boldsymbol{p}_i)W^Q\right)\left((\boldsymbol{w}_j + \boldsymbol{p}_j)W^K\right)^T}{\sqrt{d}}, \tag{5}$$

where $\boldsymbol{w}_i, \boldsymbol{w}_j$ refer to node embeddings, and bmp_i, bmp_j refer to positional embeddings. α_{ij} is the attention score of the i-th and j-th nodes.

Absolute positional embedding is tied with node embedding in the normal transformer. However, for the exact position i and j, there is little evidence that the node and where it appears in a sequence has a strong correlation. This randomness may cause noise in attention computation, especially for tree-structured data. One position has two neighbors in a sequence, but it is not true in a tree. As in the example in Figure 2, the topological relations of nodes (Human, Royal) and nodes (Human, HeadOfState) are considered to be the same. However, the distance in the sequence representation of them is 3 and 1, which differs a lot.

To better capture the structure of tree data, we think attention should satisfy the following three requirements: (1) topologically neighbored nodes' attention should be high; (2) semantically similar nodes' attention should be high; (3) some sub-trees of brother nodes can convert symmetrically in STG tasks.

Our Modification

Inspired by Ke et al. (2020), we untie the correlations between positions and words. We divide the computation of attention into two parts: semantic attention and positional attention. (1) **Semantic attention** captures the semantic similarity of twos nodes in the tree, and the computation is the same as normal attention. (2) **Positional attention** is specially designed to capture the topological relations of nodes in the tree.

Correspondingly, we design a new self attention computation method for tree structure as follows:

$$\alpha_{ij} = \frac{\boldsymbol{s}_i \cdot \boldsymbol{s}_j + \boldsymbol{p}_i \cdot \boldsymbol{p}_j}{\sqrt{2d}} + \boldsymbol{b_{i,j}}, \tag{6}$$

where $\boldsymbol{s}_i, \boldsymbol{s}_j$ refers to the node encoding, $\boldsymbol{p}_i, \boldsymbol{p}_j$ refers to the positional encoding of i and j, and $\boldsymbol{b}_{i,j}$ refers to the distance encoding of i and j. $\frac{1}{\sqrt{2d}}$ is used to retain the scale of attention score.

For tree position, we define Depth embedding as learnable parameters to capture features of tree input. Simultaneously, we define Distance embedding as learnable parameters as the bias in position attention. Attention is considered to be higher when depths are closer and distance is smaller.

For the multi-head version, Depth embedding and Distance embedding are different in all the heads. And for efficiency, we share the Depth embedding and Distance embedding in all the layers, so we only need to compute position attention in the first layer and reuse it in other layers. The function can be quickly computed by:

$$attn = (\frac{A_Q * A_K^T}{\sqrt{2d}} + \frac{P_Q * P_K^T}{\sqrt{2d}} + B)A_V, \tag{7}$$

where B is formalized as the distance metric of all the nodes in the tree. With the help of BACK, we can compute the B in $O(n^2)$ times with a stack-based algorithm.

BACK token is special in tree sequence because it has the same number as other nodes and distributes randomly in all the depths. To overcome the imbalance of nodes, we specially add the BACK token in odd depth between two sememe nodes, while sememe nodes are in even depth. We will further discuss the efficiency of Tree-attention in §5.1.

The transformer decoder layer is composed of three sub-layers. We adopt Tree-attention in the self-attention sub-layer. For the sub-layer to perform multi-head attention over the output of the encoder stack, we use normal attention because it is hard to capture the attention between tree nodes and sequence reasonably, we leave it for future work.

4 Experiments

4.1 Dataset

HowNet provides no definitions for words, and using an external dictionary requires special efforts to conduct a sense-level alignment with HowNet. In this paper, we resort to the BabelSememe dataset, which is built by Qi et al. (2020). A BabelNet synset corresponds to a sense of a word and includes definitions from other sources like WordNet (Miller, 1998), and some BabelNet synsets are manually aligned with senses of words in HowNet. One example is Figure 2.

Since there is no other attempt aligned with sense-level definitions and sememe trees, we finally use BabelNet as the only dataset. In other words, we try to predict sememes for BabelNet synsets given their definitions. There are 34,964/3,228/3,228 synsets with definitions in the training/validation/test sets.

4.2 Experimental and Parameter Settings

For NSTG, we use sentence-BERT (Reimers et al., 2019) to encode definitions and compute similarity. The embedding dimension is 768. For hyperparameters, we set the beam size in beam search to 50 and select the top 10 candidates for merging. We set the declined confidence factor base d to 0.9 empirically.

For TSTG and TaSTG, we use the base version of BERT as the encoder, and the dimension of word embeddings is 768. We use sememe embedding pre-trained by SPSE (Xie et al., 2017), and the dimension is 200. We train an 8-layer, 8-head transformer decoder, and the learning rate is set to 10^{-5}. To avoid duplicate prediction, we only choose the valid sememes that have not been predicted. We also use beam search during the prediction.

4.3 Baselines

We use NSTG and TSTG as the baseline. We ablate our TsSTG to understand the efficiency of the modification of the decoder. First, we remove bias and build up the TaSTB-B model, which has almost the same parameters as TaSTG. To understand the compute of depth encoding, we also convert the relative position of tree node i from the depth of i to the traversal order of i, and build up the TaSTB-D model.

4.4 Evaluation Protocol

We use the following metrics for STG:

BLEU Since the generated tree sequence is short, and higher order n-grams may not overlap, we use smoothed BLEU-4 score (Lin and Och, 2004), following Feng et al. (2020).

Strict-F1 To measure the structural similarity of the sememe tree T and predicted tree T', we define the Strict-F1 metric as follows:

Method	BLEU	Strict	Edge	Vertex
NSTG	10.7	25.6	27.5	33.9
TSTG	15.5	35.6	37.2	45.0
TaSTG	**17.0**	**39.7**	**41.2**	**48.2**
TaSTG-D	14.9	37.5	39.0	45.9
TaSTG-B	15.1	39.1	40.5	47.4

Table 1: Result of different models.

1. Start from the roots of T and T' and put them into the current node sets O and O'. The intersection list U is empty.

2. Get the intersections U_i for the children of both O and O' in layer i. Add U_i to U, and then update both O and O' with their children until reaching the deepest leaves.

3. For precision (P), recall (R) and F1 score $F1$, we define $P = \frac{\text{Size}(U)}{\text{Size}(T')}, R = \frac{\text{Size}(U)}{\text{Size}(T)}, F1 = \frac{2 \times P \times R}{P+R}$.

The Strict-F1 metric is challenging because it supposes that if the predicted parent sememe node is incorrect, all its corresponding children sememes are not considered.

Edge, Vertex Inspired by the classical evaluation metrics in structure learning tasks such as taxonomy induction (Bordea et al., 2016), we also use the Edge and Vertex metrics. The former evaluates the precision, recall, and F1-score after breaking down trees into edges, while the latter computes the non-hierarchical prediction result after breaking down trees into nodes.

4.5 Main Results

The experimental results for all the models are shown in Table 1, from which we observe that:

(1) TaSTG model reaches the highest F1 score, which indicates that Tree-attention works more conservatively than the other models. All transformer-based models significantly outperform the NSTG model, which is mainly because NSTG merely makes predictions based on similar synsets and existing sememe pairs, while 11.5% synsets in the test set have unseen sememe pairs, which are hard for NSTG to predict.

(2) Removing the Distance embedding and converting the Depth embedding to Forward embedding both result in a negative impact on the model's performance. This suggests that in tree-structured input, it's more important to focus on topologically similar nodes. And the gain of Depth embedding is much more than that of Distance embedding, which might be because the number of learnable parameters for tree structure in Depth embedding is much more.

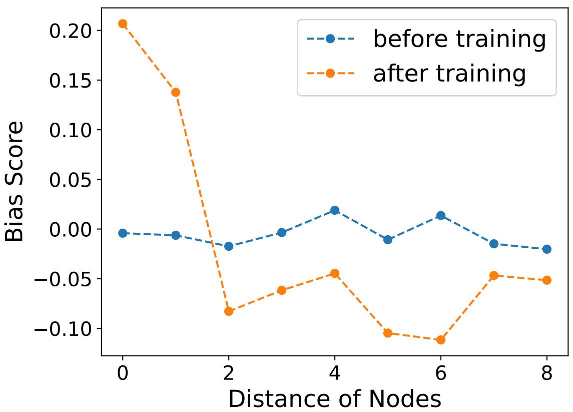
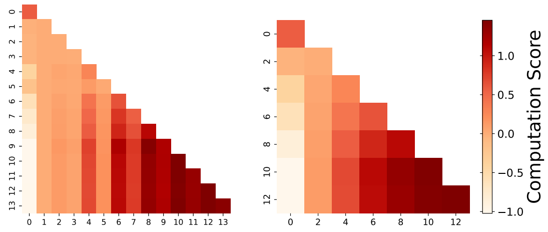


Figure 3: The average score of Distance Embedding of different heads.



(a) heat with BACK token (b) heat without BACK token

Figure 4: Visualization of computation results of different Depth embedding.

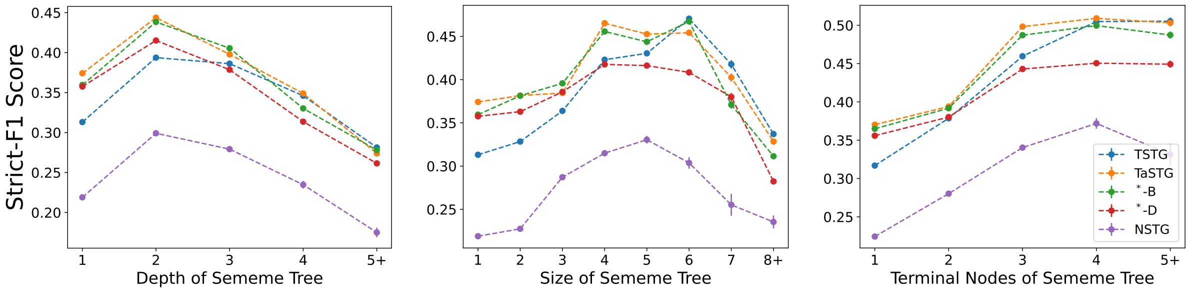
(3) Differences between the BLEU are smaller than those of F1, which indicates that the BLEU score may not capture the hierarchical similarity between the output tree and the answer.

5 Analysis

In this part, we further discussed the efficiency of positional attention and analyzed the performance of our model in different tree complexity, and make a case analysis of our models.

5.1 Hierarchical Feature Capture

In this section, we study whether Tree-attention learns hierarchical structures. And we analyze the performance of Positional attention in structure reconstructing.



(a) change over depth

(b) change over size

(c) change over the number of terminal nodes

Figure 5: Test results over the complexity of trees. Confidence intervals shown in the figures are estimated with a confidence level of 95%.

Visualization of Positional Attention

Considering that the most straightforward way of interpreting the hierarchical features is to visualize the attention scores, we plot the heat-map of our Positional attention result.

The average scores of Distance embedding of different heads are shown in Figure 3. We can explicitly see that when the distance is small, the Positional attention bias is high, which indicates that our Distance embedding mainly focuses on topologically similar nodes. The bias is lower when distance is 2, we guess this is used to eliminate the influence of brother nodes, in which depths are the same and the Depth encoding score is high.

Then we visualize the Depth embedding computation result of different depths, the result is shown in Figure 4. Knowing that we define `BACK` token in odd depth and sememe node in even depth, we plot the score with and without `BACK` token. From the result, we can see that:

(1) The result in the deeper layer tends to be high, which means when generating atomic steps, models focus more on longer tree paths, this may be because different sememe paths indicate different dimensions of senses, and during generating a new path, models need to avoid the existing paths.

(2) In Figure 4(a), scores of depth-0 are much lower than others. It is because depth-0 represents `START`, which is noise when generating other nodes. Likely, `BACK` token in even columns retains a lower score. Our model captures this feature and focuses more on meaningful nodes.

(3) For a row in Figure 4(b) $(i < j)$, the score is higher when the depth is closer; and for a column$(i > j)$, the score is higher when the depth is far, which indicates that during generation, our models focus more on succeeds, and focus more on closer ancestors.

Method	BLEU	Strict	Edge	Vertex
NSTG	23.9	44.3	46.9	65.4
TSTG	**38.2**	69.7	71.9	**82.6**
TaSTG	35.9	**70.3**	**72.5**	82.1
TaSTG-D	32.1	67.6	70.3	81.0
TaSTG-B	33.8	69.5	72.4	82.0

Table 2: The Restricted evaluation result of different models.

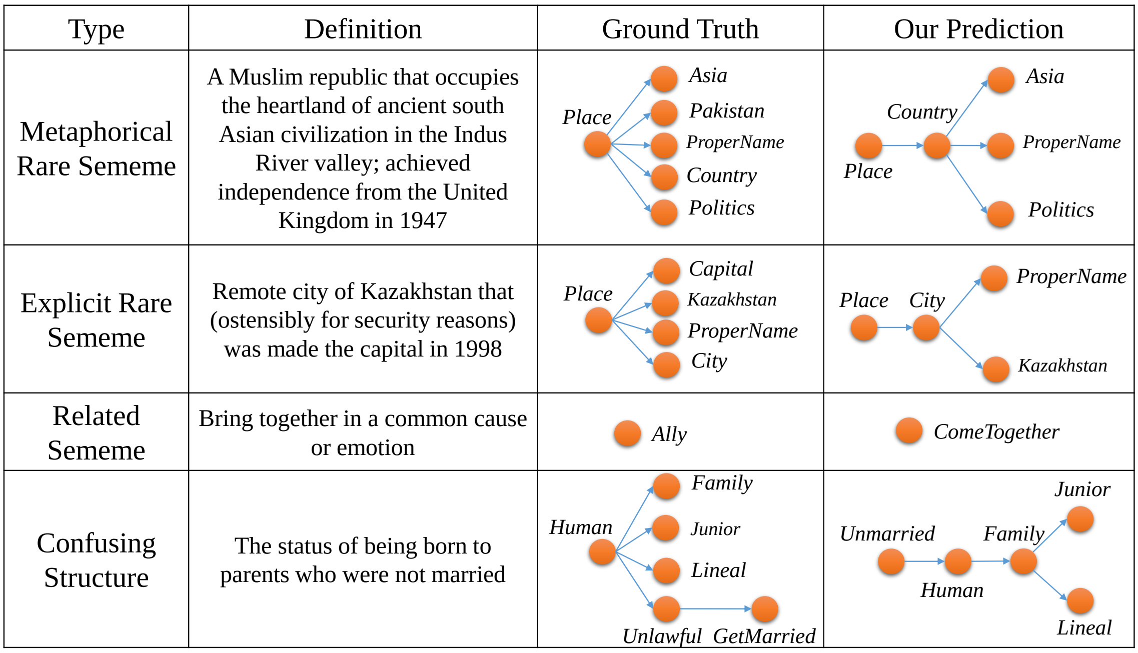
From the visualize, we can directly see that our model successfully captures the hierarchical feature of tree-structured input by using Depth embedding and Distance embedding.

Structure Reconstruction Ability

To better measure the ability of models to capture hierarchical information, we design a Restricted evaluation, during which we provide correct sememes without structures for our models and ask the models to predict structures for the input sememes. This evaluation focuses on evaluating the structure organization ability of our models in STG. Especially, We ignore the synset who have sememe tree of size 1 in Restricted evaluation, because this has no structural information. Results are demonstrated in Table 2, from which we can observe that:

(1) All the models achieve significant improvement over the results in Table 1. It indicates that the major challenge for the STG task comes from selecting appropriate candidate sememes at each level.

(2) NSTG model shares the lowest gain because it tends to give a relatively conservative prediction, resulting in the lowest Recall score (36.6 in the restricted test compared with 53.3 of TaSTG). In



Type	Definition	Ground Truth	Our Prediction
Metaphorical Rare Sememe	A Muslim republic that occupies the heartland of ancient south Asian civilization in the Indus River valley; achieved independence from the United Kingdom in 1947	Place Asia Pakistan ProperName Country Politics	Asia Country ProperName Place Politics
Explicit Rare Sememe	Remote city of Kazakhstan that (ostensibly for security reasons) was made the capital in 1998	Capital Place Kazakhstan ProperName City	ProperName Place City Kazakhstan
Related Sememe	Bring together in a common cause or emotion	Ally	ComeTogether
Confusing Structure	The status of being born to parents who were not married	Family Human Junior Lineal Unlawful GetMarried	Junior Unmarried Family Human Lineal

Figure 6: Some representative cases of STG.

the contrast, the Base transformer model generates big trees (18% larger than TaSTG) and gets a higher Recall score in the restricted test, gaining most improvements in the restricted test, performing similarly with TaSTG.

However, our STG models' performances are far from perfect, which implies that understanding sememe tree structures is still challenging.

5.2 Sememe Tree Complexity (STC) Analysis

In order to further investigate our models under different scenarios and get a deeper understanding of STG tasks, we further conduct three auxiliary experiments over different levels of sememe tree complexity (STC). Here we define the STC as the annotated sememe number of target words, depth of target tree, and number of terminal nodes in a tree. We conduct these experiments with Strict-F1 on Open evaluation due to limited space. We combine results of words that have more than 8 sememes, which is deeper than 6, or which have more than 5 terminal nodes since there are less than 1%. From the result, we can see that:

In Figure 5(a),Figure 5(b), we can see that prediction performance first increases and then drops with the growth of tree size and depth, which indicates that the STG task is difficult both when there are too few or many sememes in synset. This is in compliance with previous work Qi et al. (2020).

Since the big size and high depth of a tree may not absolutely represent high complexity, we also implement the performance of models with the number of terminal nodes, et tree paths, the result are shown in Figure 5(c).

(1) With the help of Depth embedding and Distance embedding, TaSTG reaches the highest score in all the cases. And base transformer model performs worse when there are fewer tree paths.

(2) Due to the number of learnable parameters of structure capture, TaSTG-B performs much better than TaSTG-D. And the gain of Distance embedding and Depth embedding is huge when there are more tree paths. This is because Distance embedding distinguishes nodes from different tree paths.

5.3 Case Study

To show the insights and challenges intuitively, we give some representative cases in Table 6 and make a qualitative case analysis of our model.

(1) Rare Sememe: Some predictions include very rare sememes. This kind of case challenges our model to get the meaning of sememe from a few train data. Our model successfully captures rare information from definition when it appears. For Example, our model learns the connection with sememe "Kazakhstan" and the word "Kazakhstan", because it appears a few times in the train set. However, in some predictions, definitions don't directly imply the meanings of some sememes, and it's difficult for our model to make such predictions without extra training data. For example, our model cannot predict “Pakistan” from the definition. This kind

of case challenges models on learning sememe definitions, but it is not contained in our train set.

(2) Related Sememe: The most common error type is Related Sememes (e.g., predicting "Come-Together" while the correct sememe is "Ally"). It implies that learning BabelNet's annotation preferences to distinguish related sememes that only have minor differences is still challenging for current STG models.

(3) Confusing Structure: Some definitions of synsets have rich meaning. For example, our model predicts correct sememes for "premarital pregnancy" but the incorrect structure, which shows the challenge of predicting correct structures. However, tackling the confusing structure of sememes is a difficult problem even for human experts.

6 Conclusion and Future Work

In this paper, we handle the structured sememe prediction task for the first time. We propose a Transformer-based tree generation model by adapting the attention mechanism to trees. Experimental results show that our model outperforms baselines including the general tree Transformer. We also conduct extensive experiments and detailed analyses to demonstrate the different properties of our models and the challenges of the task.

We will explore the following research directions in the future: (1) We will better measure the semantic similarity of tree nodes. In this paper, the Strict-F1 score only focuses on the structure and ignores the semantic similarity of generated sememe pairs with the answer. (2) We will further explore to import the tree-attention mechanism in all sublayers of the decoder and figure out the influence. (3) We will try to combine our method with other sememe-based applications and further analyze the influence of the structure information of sememes.

7 Acknowledgements

This work is supported by the National Key RD Program of China (No. 2020AAA0106502), Institute Guo Qiang at Tsinghua University, Beijing Academy of Artificial Intelligence (BAAI), and International Innovation Center of Tsinghua University, Shanghai, China. We also thank all the anonymous reviewers for their valuable comments and suggestions.

References

Miltiadis Allamanis, Hao Peng, and Charles Sutton. 2016. A convolutional attention network for extreme summarization of source code. In *Proceedings of ICML*. PMLR.

Leonard Bloomfield. 1926. A set of postulates for the science of language. *Language*, 2(3):153–164.

Georgeta Bordea, Els Lefever, and Paul Buitelaar. 2016. Semeval-2016 task 13: Taxonomy extraction evaluation (texeval-2). In *Proceedings of SemEval*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*.

Zhendong Dong and Qiang Dong. 2006. *HowNet and the Computation of Meaning (With CD-Rom)*. World Scientific.

Jiaju Du, Fanchao Qi, Maosong Sun, and Zhiyuan Liu. 2020. Lexical sememe prediction by dictionary definitions and local semantic correspondence. *Journal of Chinese Information Processing*, 34(5):1–9.

Zhangyin Feng, Daya Guo, Duyu Tang, Nan Duan, Xiaocheng Feng, Ming Gong, Linjun Shou, Bing Qin, Ting Liu, Daxin Jiang, et al. 2020. Codebert: A pre-trained model for programming and natural languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Yihong Gu, Jun Yan, Hao Zhu, Zhiyuan Liu, Ruobing Xie, Maosong Sun, Fen Lin, and Leyu Lin. 2018. Language modeling with sparse product of sememe experts. In *Proceedings of EMNLP*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Bairu Hou, Fanchao Qi, Yuan Zang, Xurui Zhang, Zhiyuan Liu, and Maosong Sun. 2020. Try to substitute: An unsupervised chinese word sense disambiguation method based on hownet. In *Proceedings of COLING*.

Huiming Jin, Hao Zhu, Zhiyuan Liu, Ruobing Xie, Maosong Sun, Fen Lin, and Leyu Lin. 2018. Incorporating chinese characters of words for lexical sememe prediction. In *Proceedings of ACL*.

Guolin Ke, Di He, and Tie-Yan Liu. 2020. Rethinking positional encoding in language pre-training. In *Proceedings of ICLR*.

Shucheng Li, Lingfei Wu, Shiwei Feng, Fangli Xu, Fengyuan Xu, and Sheng Zhong. 2020. Graph-to-tree neural networks for learning structured input-output translation with applications to semantic parsing and math word problem. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Chin-Yew Lin and Franz Josef Och. 2004. Orange: a method for evaluating automatic evaluation metrics for machine translation. In *Proceedings of COLING*.

Qianying Liu, Wenyv Guan, Sujian Li, and Daisuke Kawahara. 2019. Tree-structured decoding for solving math word problems. In *Proceedings of EMNLP-IJCNLP*.

Qun Liu and Sujian Li. 2002. Word similarity computing based on HowNet. *International Journal of Computational Linguistics & Chinese Language Processing*, 7(2):59–76.

Yijiang Liu, Meishan Zhang, and Donghong Ji. 2020. End to end chinese lexical fusion recognition with sememe knowledge. In *Proceedings of COLING*.

Boer Lyu, Lu Chen, and Kai Yu. 2021. Glyph enhanced Chinese character pre-training for lexical sememe prediction. In *Findings of the Association for Computational Linguistics: EMNLP 2021*.

George Miller. 1998. *WordNet: An electronic lexical database*. MIT press.

Roberto Navigli and Simone Paolo Ponzetto. 2012. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Xuan-Phi Nguyen, Shafiq Joty, Steven Hoi, and Richard Socher. 2019. Tree-structured attention with hierarchical accumulation. In *Proceedings of ICLR*.

Yilin Niu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2017. Improved word representation learning with sememes. In *Proceedings of ACL*.

Fanchao Qi, Liang Chang, Maosong Sun, Sicong Ouyang, and Zhiyuan Liu. 2020. Towards building a multilingual sememe knowledge base: Predicting sememes for babelnet synsets. In *Proceedings of AAAI*.

Fanchao Qi, Yangyi Chen, Fengyu Wang, Zhiyuan Liu, Xiao Chen, and Maosong Sun. 2021a. Automatic construction of sememe knowledge bases via dictionaries. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*.

Fanchao Qi, Junjie Huang, Chenghao Yang, Zhiyuan Liu, Xiao Chen, Qun Liu, and Maosong Sun. 2019. Modeling semantic compositionality with sememe knowledge. In *Proceedings of ACL*.

Fanchao Qi, Yankai Lin, Maosong Sun, Hao Zhu, Ruobing Xie, and Zhiyuan Liu. 2018. Cross-lingual lexical sememe prediction. In *Proceedings of EMNLP*.

Fanchao Qi, Chuancheng Lv, Zhiyuan Liu, Xiaojun Meng, Maosong Sun, and Hai-Tao Zheng. 2022. Sememe prediction for babelnet synsets using multilingual and multimodal information. In *Findings of the Association for Computational Linguistics: ACL 2022*.

Fanchao Qi, Ruobing Xie, Yuan Zang, Zhiyuan Liu, and Maosong Sun. 2021b. Sememe knowledge computation: a review of recent advances in application and expansion of sememe knowledge bases. *Frontiers of Computer Science*, 15(5):1–11.

Fanchao Qi, Yuan Yao, Haoji Xu, Zhiyuan Liu, and Maosong Sun. 2021c. Turn the combination lock: Learnable textual backdoor attacks via word substitution. In *Proceedings of ACL*.

Yujia Qin, Fanchao Qi, Sicong Ouyang, Zhiyuan Liu, Cheng Yang, Yasheng Wang, Qun Liu, and Maosong Sun. 2020. Improving sequence modeling ability of recurrent neural networks via sememes. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Nils Reimers, Iryna Gurevych, Nils Reimers, Iryna Gurevych, Nandan Thakur, Nils Reimers, Johannes Daxenberger, Iryna Gurevych, Nils Reimers, Iryna Gurevych, et al. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of EMNLP*.

Vighnesh Shiv and Chris Quirk. 2019. Novel positional encodings to enable tree-based transformers. *Proceedings of NeurIPS*.

Zeyu Sun, Qihao Zhu, Yingfei Xiong, Yican Sun, Lili Mou, and Lu Zhang. 2020. Treegen: A tree-based transformer architecture for code generation. In *Proceedings of the AAAI*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NeurIPS*.

Anna Wierzbicka. 1996. *Semantics: Primes and universals: Primes and universals*. Oxford University Press, UK.

Qinzhuo Wu, Qi Zhang, and Zhongyu Wei. 2021. An edge-enhanced hierarchical graph-to-tree network for math word problem solving. In *Findings of the Association for Computational Linguistics: EMNLP 2021*.

Ruobing Xie, Xingchi Yuan, Zhiyuan Liu, and Maosong Sun. 2017. Lexical sememe prediction via word embeddings and matrix factorization. In *Proceedings of IJCAI*.

Pengcheng Yin and Graham Neubig. 2017. A syntactic neural model for general-purpose code generation. In *Proceedings of ACL*.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of ACL*.

Wojciech Zaremba and Ilya Sutskever. 2014. Learning to execute. *arXiv preprint arXiv:1410.4615*.

Jipeng Zhang, Roy Ka-Wei Lee, Ee-Peng Lim, Wei Qin, Lei Wang, Jie Shao, and Qianru Sun. 2020a. Teacher-student networks with multiple decoders for solving math word problem. In *Proceedings of IJCAI*.

Lei Zhang, Fanchao Qi, Zhiyuan Liu, Yasheng Wang, Qun Liu, and Maosong Sun. 2020b. Multi-channel reverse dictionary model. In *Proceedings of AAAI*.

Jingwen Zhu, Yuji Yang, Bin Xu, and Juezi Li. 2019. Semantic representation learning based on hownet. *Journal of Chinese Information Processing*, 33(03):33–41.

Daniel Zugner, Tobias Kirschstein, Michele Catasta, Jure Leskovec, and Stephan Gunnemann. 2021. Language-agnostic representation learning of source code from structure and context. In *Proceedings of ICLR*.