Enhancing Transformer with Sememe Knowledge

Yuhui Zhang[1*] **Chenghao Yang**[2*] **Zhengping Zhou**[1*] **Zhiyuan Liu**[3]
[1]Stanford University [2]Columbia University [3]Tsinghua University
{yuhuiz, zpzhou}@stanford.edu chenghao.yang@columbia.edu
liuzy@tsinghua.edu.cn

Abstract

While large-scale pretraining has achieved great success in many NLP tasks, it has not been fully studied whether external linguistic knowledge can improve data-driven models. In this work, we introduce sememe knowledge into Transformer and propose three sememe-enhanced Transformer models. Sememes, by linguistic definition, are the minimum semantic units of language, which can well represent implicit semantic meanings behind words. Our experiments demonstrate that introducing sememe knowledge into Transformer can consistently improve language modeling and downstream tasks. The adversarial test further demonstrates that sememe knowledge can substantially improve model robustness.[1]

1 Introduction

Self-supervised pretraining has significantly improved the performance of Transformer (Vaswani et al., 2017) on a wide range of NLP tasks (Radford et al., 2018; Devlin et al., 2019; Yang et al., 2019). While no explicit linguistic rules and concepts are introduced, models can achieve remarkable performances with extensive training signals provided by large-scale data. Nonetheless, recent works still demonstrate that external syntactic information can improve various NLP tasks, including machine translation (Sennrich and Haddow, 2016; Aharoni and Goldberg, 2017; Bastings et al., 2017) and semantic role labeling (Marcheggiani and Titov, 2017; Strubell et al., 2018).

Can external semantic information benefit the widely-adopted pretraining and fine-tuning framework as well? In response, we explore incorporating sememe knowledge into Transformer (Vaswani et al., 2017). Sememes are the minimum semantic units of meaning for natural language, as some linguists assume that a limited closed set of sememes can be composed to represent the semantic meaning of each word (Bloomfield, 1926). In this work, we adopt a high-quality sememe-based lexical knowledge base, HowNet (Dong and Dong, 2006; Qi et al., 2019), which can provide powerful support for models to understand Chinese word semantics (Gu et al., 2018; Niu et al., 2017). Some examples of sememe annotations can be found in Figure 1.

We propose to combine two simple methods to incorporate sememe knowledge into our framework: 1) based on the linguistic assumption, we add aggregated sememe embeddings to each word embedding to enhance its semantic representation; 2) we use sememe prediction as an auxiliary task to help the model gain deeper understandings of word semantics. We verify the effectiveness of our methods on several Chinese NLP tasks that are closely related to word-level and sentence-level semantics. Following general settings of pretraining and fine-tuning, our experiments show consistent improvements on all the tasks with sememe-enhanced Transformer. We also find that the sememe-enhanced model can achieve the same performance with less fine-tuning data, which is desirable as data annotation processes are always time-consuming and expensive.

We further demonstrate that, by incorporating sememe knowledge using our methods, model robustness can be significantly improved towards adversarial examples, which are generated by replacing nouns, adjectives and adverbs with their synonyms in our experiment. Our case studies further interpret why sememe knowledge can help model defend adversarial attacks.

[*] Indicates equal contribution. Work done at Tsinghua University. Y.Z. and C.Y. designed and evaluated the model architecture and performed the adversarial test. Z.Z. performed the data ablation study and case study. Z.L. supervised the work and is the corresponding author.

[1]Codes are available at https://github.com/yuhui-zh15/SememeTransformer/.

Proceedings of the 5th Workshop on Representation Learning for NLP (RepL4NLP-2020), pages 177–184

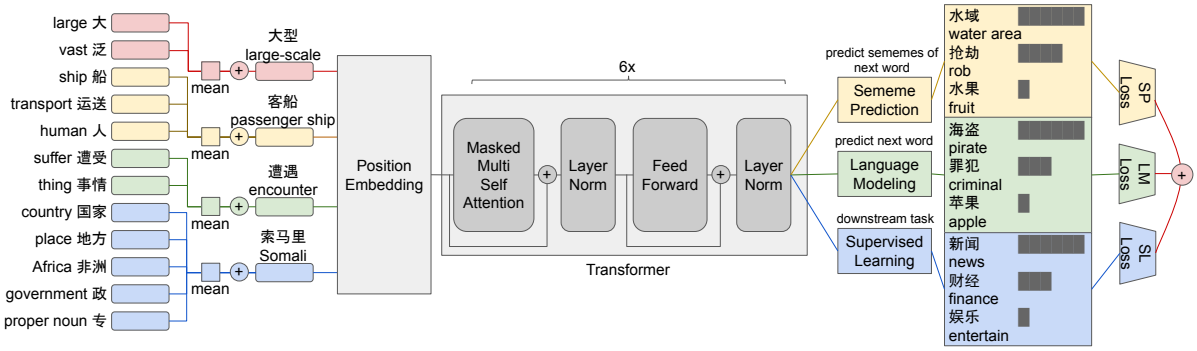


Figure 1: Our proposed model architecture. For each word, we enhance word representation by adding aggregated sememe embeddings. We use multitask learning with three tasks: **sememe prediction** (predicting sememes of next word), **language modeling** (predicting next word) and **supervised learning** (only for downstream tasks).

2 Methodology

In this section, we propose two simple methods to incorporate sememe knowledge into our framework: aggregated sememe embeddings and sememe prediction auxiliary task.

2.1 Transformer

Transformer was originally proposed by Vaswani et al. (2017) as a machine translation architecture. We use a multi-layer Transformer architecture similar to the setup in Radford et al. (2018), which has been verified effectiveness on multiple NLP tasks. At the input layer, a sequence of words $(w_1, w_2, ..., w_T)$ are embedded as $\mathbf{H}^0 = (\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_T) \in \mathbb{R}^{T \times D}$, where D indicates the hidden size of the model. A positional embedding is then added to inject position information into Transformer. After L residual multi-head self-attention layers with feed-forward connections, we obtain the contextualized sequence embedding $\mathbf{H}^L = (\mathbf{h}_1^L, \mathbf{h}_2^L, ..., \mathbf{h}_T^L) \in \mathbb{R}^{T \times D}$.

2.2 Aggregated Sememe Embeddings

Enhancing word representation is a common approach to introduce linguistic knowledge into neural networks (Sennrich and Haddow, 2016; Niu et al., 2017; Bojanowski et al., 2017). For each word w, **Transformer-SE** considers all of its sememes and enhances word representation by adding its average sememe embeddings to word embedding. Formally, we have:

$$\tilde{\mathbf{w}} = \frac{1}{n_w} \sum_{s \in S(w)} \mathbf{x}_s + \mathbf{w}$$

where $S(w)$ refers to the sememe set associated with word w with the size n_w, $\mathbf{x}_s$ refers to the embedding of the sememe s, $\mathbf{w}$ refers to the embedding of word w and $\tilde{\mathbf{w}}$ refers to the sememe-enhanced word embedding. Sememe-enhanced representation $\tilde{\mathbf{w}}$ is directly fed into Transformer.

The Transformer-SE model complies with the linguistic assumption that implicit word semantics can be composed of a limited set of sememes. Also, as sememe embeddings are shared among words, latent semantic correlations between words can be well encoded. While our method to incorporate sememe knowledge is rather straightforward, our main purpose is to verify the effectiveness of sememe knowledge. We leave more potential methods to enrich word-level semantics with sememe knowledge such as tree LSTM (Tai et al., 2015) and graph convolutional network (Bastings et al., 2017) in future work.

2.3 Sememe Prediction Auxiliary Task

Sememe prediction task aims to predict sememes for the next word and can be formulated as a multi-label classification task. Inspired by the multitask learning (Caruana, 1997; Collobert et al., 2011), we add the sememe prediction task in addition to the language modeling task for **Transformer-SP**. This task challenges the model's capability to incorporate sememe knowledge, and can be viewed as a complementary task for language modeling, as predicting the sememes of the next word is closely related to understanding semantics and it is often more learnable than directly modeling the probability of the next word. [2]

At each time step, given current contextualized

[2]For example, if a sentence starts with "How to cook", it is much easier to predict the next word is a kind of "food" than any specified word. It is worth noting that language modeling has about 20 times larger vocabulary size.

Task	Language Modeling	Headline Categorization	Sentiment Classification	Semantic Matching	Sememe Prediction
Metric	PPL	ACC (%)	ACC (%)	ACC (%)	MAP (%)
Transformer	49.01	71.5	52.7	81.2	40.1
Transformer-SE	47.37	72.6	53.7	82.6	52.1
Transformer-SP	49.14	72.3	53.0	81.8	40.3
Transformer-SEP	**46.53**	**72.6**	**54.9**	**83.3**	**52.8**
+ Sememe2Char	48.90	72.3	52.2	81.2	-

Table 1: Experimental results on different tasks. **Transformer**, **Transformer-SE**, **Transformer-SP** and **Transformer-SEP** refers to the vanilla Transformer model (base), Transformer with aggregated sememe embeddings, Transformer with sememe prediction auxiliary task and the hybrid model, respectively. We also compare sememe decomposition to character decomposition for our best model and demonstrate advantages of our methods.

representation $\mathbf{h}^L$ from Transformer, we estimate the probability of sememe s associated with next word w as $p(w, s) = \sigma(\mathbf{w}\mathbf{h}^L + b)$, where $\mathbf{w}$ and b are the weight and bias associated with sememe s, σ is the sigmoid activation function. We then calculate the binary cross-entropy loss of sememe prediction $\mathcal{L}_{SP}$ as:

$$\mathcal{L}_{SP} = -\frac{1}{T}\sum_{t=1}^{T}\frac{1}{n}\sum_{s\in S} g(w_t, s)\log(p(w_t, s)) + (1 - g(w_t, s))\log(1 - p(w_t, s))$$

where S refers to the overall sememe set with the size n, $g(w, s)$ is a binary variable indicating whether sememe s is associated with word w. Finally, we formulate the loss as:

$$\mathcal{L}_{PRE} = \mathcal{L}_{LM} + \mathcal{L}_{SP}$$
$$\mathcal{L} = \mathcal{L}_{SL} + \rho\mathcal{L}_{PRE}$$

where $\mathcal{L}_{LM}$ and $\mathcal{L}_{SL}$ are the conventional negative log-likelihood language modeling loss and downstream supervised learning loss. $\mathcal{L}_{PRE}$ is the loss optimized during pretraining, while $\mathcal{L}$ is the loss optimized during supervised training for downstream tasks, ρ serves as a coefficient to control the strength of $\mathcal{L}_{PRE}$ during supervised learning.

2.4 Hybrid Model

Transformer-SE and Transformer-SP are designed based on different ideas. Transformer-SE can well inform sememe knowledge to all self-attention layers, while Transformer-SP utilizes additional training signals through the back-propagation process. To combine the advantages of these models, we propose a hybrid model named **Transformer-SEP**. Transformer-SEP incorporates sememe knowledge into the input layer by adding aggregated sememe embeddings and performs the sememe prediction auxiliary task in the output layer.

3 Experiments

We experiment across a diverse set of five benchmark NLP tasks and demonstrate the effectiveness of introducing sememe knowledge.

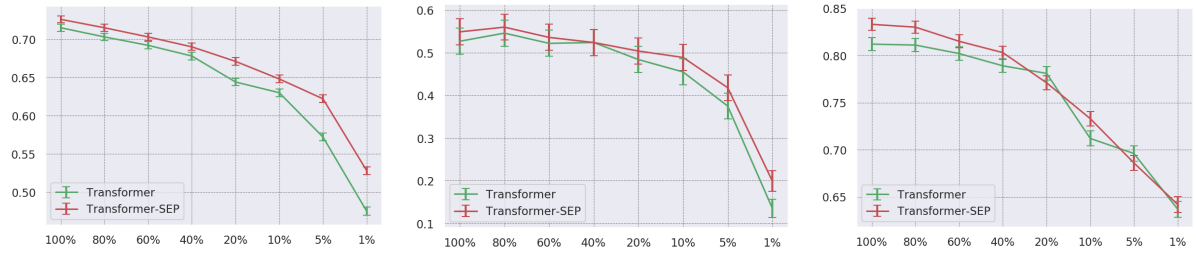
3.1 Experimental Setup

We use 6-layer 8-head Transformer with the hidden size of 768 and feedforward size of 2048. We set both word embedding and sememe embedding size as 768. We use batch size of 32 and set dropout rate as 0.2 to alleviate overfitting. The vocabulary size is 39,770 and the total number of sememes is 2,100. We truncate the sequence length to 128 for pretraining and supervised learning. When performing supervised training, we set the coefficient ρ to be 0.5. Embeddings are tied for the input layer and output layer to speed up convergence. We clip gradients less than 2 and use Adam optimizer (Kingma and Ba, 2014) with 0.001 learning rate and 8000 warmup steps. For downstream tasks, we use the best pretrained model from language modeling to initialize.

3.2 Tasks and Datasets

Language Modeling Language modeling on a large corpus provides additional training signals for supervised downstream tasks. We use perplexity (PPL) to measure the performance of the language model. Lower PPL indicates better performance. We pretrain the language model on the People's Daily corpus, which contains $\sim$ 15M words.

Headline Categorization Automatic and accurate news categorization is essential for recommen-



Headline Categorization | Sentiment Classification | Semantic Matching

Figure 2: Performance of Transformer and Transformer-SEP with different amounts of training data. More significant improvements can be achieved on tasks that depend more on word-level semantics. X-axis: Percent of supervised training data. Y-axis: Accuracy. The error bars indicate the 95% confidence interval.

dation systems. We use NLPCC 2017 news headline categorization dataset (Qiu et al., 2017), which contains 156,000 news for training and 36,000 news for validation, divided into 18 categories including finance, society, game, etc. We use accuracy (ACC) to measure the performance.

Sentiment Classification Sentiment classification is a useful task for emoticon recommendation, depression detection, etc. We use NLPCC 2013 Weibo sentiment detection dataset and conduct experiments on sentence-level sentiment classification. The dataset includes 7 different sentiment genres. We remove sentences without any sentiment and resplit the data to 8,225 / 997 / 1,020 for training, validation, test, respectively.

Semantic Matching Semantic matching is fundamental for question answering, which aims to match the input question to similar questions in an existing database. We use LCQMC (Liu et al., 2018) dataset for this task, which contains 238,766 / 8,802 / 12,500 training, validation, test data, respectively. For each pair of questions, we concatenate them with a special token for classification.

Sememe Prediction Predicting sememes for given words by its definitions is important for the HowNet extension (Xie et al., 2017). The definitions are extracted from the Contemporary Chinese Dictionary and the sememes of target words are masked for fair comparison. We create a dataset containing 41,081 / 5,135 / 5,136 word-definition pairs for training, validation and test.

3.3 Overall Performance

From Table 1, we observe that simply adding sememe embedding (i.e., Transformer-SE) can lead to significant improvements over all tasks. These tasks challenge models on the capability of modeling word-level semantics and sentence-level semantics, which demonstrates that sememe knowledge can provide beneficial semantic information for Transformer. The improvement of Transformer-SP is rather less, which may due to the difficulty of predicting new knowledge without previous knowledge. Transformer-SEP achieves further improvements over Transformer-SE. The additional improvement can be interpreted as combining the advantages of these two models.

As characters provide strong semantics for Chinese (Chen et al., 2015), we also compare sememe decomposition with character decomposition (Sememe2Char) for our best model (i.e., with aggregated character embedding and character prediction auxiliary task). From Table 1, we observe clear performance drops over all tasks, which demonstrates that decomposing word into sememes are much more effective.

3.4 Data Ablation Study

We further perform data ablation study and observe overall consistent improvements for downstream tasks over different amounts of training data, indicating that incorporating external sememe knowledge could benefit model robustness when faced with limited training data (Figure 2). It is also worth noting that, when training data is limited, the more a task depends on **word-level semantics** (e.g., headline categorization > sentiment classification > semantic matching[3]), the larger improvement can be achieved by incorporating sememe knowledge. We hypothesize this is due to the increased unseen words in the test set when faced

[3]For instance, the word *football* strongly indicates *sport* for headline categorization, while *what's football?* $\neq$ *is it a football?* for semantic matching.

Replace	Semantic Matching			Sentiment Classification			Headline Categorization		
	#Count	Base	Ours	#Count	Base	Ours	#Count	Base	Ours
-	0	0.0	0.0	0	0.0	0.0	0	0.0	0.0
Noun.	30,858	18.0	**15.4**(-14%)	2,313	14.1	**11.8**(-16%)	168,516	14.8	**13.4**(-10%)
Adj.	6,498	16.7	**14.8**(-11%)	1,143	20.4	**16.9**(-17%)	54,054	9.4	9.5(+1%)
Adv.	3,306	16.1	**14.1**(-12%)	1,803	14.0	**12.3**(-12%)	65,136	8.5	**8.0**(-6%)
ALL	40,662	17.6	**15.2**(-14%)	5,259	15.4	**13.1**(-15%)	287,706	12.4	**11.4**(-8%)

Table 2: Adversarial test for the base model and our best model (i.e., Transformer v.s. Transformer-SEP). We generate adversarial examples by replacing nouns, adjectives, and adverbs for cases that both models can predict correctly. We report **error rate** (lower the better) categorized by part-of-speech and the number of generated adversarial examples.

with less training data. As semantically similar words would share similar sememes, the sememe-informed model would better understand semantics and outperform the baseline by a large margin.

3.5 Adversarial Test and Case Study

Recent research has demonstrated that neural networks are vulnerable to adversarial examples (Goodfellow et al., 2015; Jia and Liang, 2017; Alzantot et al., 2018). To evaluate the robustness of our models, we generate adversarial examples by replacing similar nouns, adjectives and adverbs for the cases that both Transformer and Transformer-SEP can predict correctly. Intuitively, these words are generally more informative for prediction and models are more likely to overfit such words.

Specifically, we compute the word similarity based on the novel Cilin metric (Tian and Zhao, 2010) and we use THULAC (Sun et al., 2016) for part-of-speech (POS) tagging. For the semantic matching task, we only replace words that occur in both sentences to ensure semantic consistency.

奸商（骗子）如何有工作牌在行李大厅里明目张胆行骗?

How do the **profiteers (cheaters)** have staff cards and blatantly cheat in the baggage hall?

有罪 **guilty** 人 **human** 欺骗 **deceive** 商业 **commerce**

有罪 **guilty** 人 **human** 骗 **cheat**

Table 3: Case study for the adversarial test. The **original word** with its sememes is colored in blue, while the **replaced word** with its sememes is colored in red.

We report the adversarial test error rate categorized by POS in Table 2. Sememe-enhanced Transformer-SEP achieves consistent improvement over the vanilla Transformer. An interesting finding is that, in headline categorization and semantic matching, the largest performance drops are observed by replacing nouns while intuitively sentiment classification should be more sensitive to adjectives.

We further perform the case study to get a better interpretation of why sememe knowledge can improve model robustness to adversarial attacks. We show an example that Transformer-SEP can predict correctly but get wrong for Transformer in Table 3. As word "cheater" and "profiteer" share the same sememes "guilty" and "human" and similar sememes "deceive" and "cheat", this sememe knowledge can propagate through all self-attention layers, thus it is easy to interpret why sememe knowledge can enhance word representation and defend such word-replacement attack. More examples can be found in the Appendix.

4 Conclusion

In this work, we introduce sememe knowledge into Transformer and verify the effectiveness of external semantic knowledge for data-driven models. We further demonstrate the robustness of our methods via data ablation study and adversarial test. For future work, we would like to explore more ways to leverage semantic knowledge and generate different adversarial examples for evaluation.

Acknowledgments

The authors would like to thank the anonymous reviewers for their many insightful comments. This work is (jointly or partly) funded by the Natural Science Foundation of China (NSFC) and the German Research Foundation (DFG) in Project Crossmodal Learning, NSFC 61621136008 / DFG TRR-169.

References

Roee Aharoni and Yoav Goldberg. 2017. Towards string-to-tree neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 132–140.

Moustafa Alzantot, Yash Sharma Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Joost Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. 2017. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1957–1967.

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

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

Xinxiong Chen, Lei Xu, Zhiyuan Liu, Maosong Sun, and Huanbo Luan. 2015. Joint learning of character and word embeddings. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2461–2505.

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.

Zhendong Dong and Qiang Dong. 2006. *Hownet and the computation of meaning (with Cd-rom)*. World Scientific.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*.

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 the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4642–4651.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Xin Liu, Qingcai Chen, Chong Deng, Huajun Zeng, Jing Chen, Dongfang Li, and Buzhou Tang. 2018. Lcqmc: A large-scale chinese question matching corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1952–1962.

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.

Yilin Niu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2017. Improved word representation learning with sememes. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2049–2058.

Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Qiang Dong, Maosong Sun, and Zhendong Dong. 2019. Openhownet: An open sememe-based lexical knowledge base. *arXiv preprint arXiv:1901.09957*.

Xipeng Qiu, Jingjing Gong, and Xuanjing Huang. 2017. Overview of the nlpcc 2017 shared task: Chinese news headline categorization. In *National CCF Conference on Natural Language Processing and Chinese Computing*, pages 948–953. Springer.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf*.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 83–91.

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.

Maosong Sun, Xinxiong Chen, Kaixu Zhang, Zhipeng Guo, and Zhiyuan Liu. 2016. Thulac: An efficient lexical analyzer for chinese. Technical report, Technical Report. Technical Report.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. 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 1556–1566.

Jiu-le Tian and Wei Zhao. 2010. Words similarity algorithm based on tongyici cilin in semantic web adaptive learning system. *Journal of Jilin University(Information Science Edition)*, 28(6):602–608.

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 *Advances in Neural Information Processing Systems*, pages 5998–6008.

Ruobing Xie, Xingchi Yuan, Zhiyuan Liu, and Maosong Sun. 2017. Lexical sememe prediction via word embeddings and matrix factorization. In *IJCAI*, pages 4200–4206.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Task	Input	Ours	Base
Sentiment Classification	奸商 （骗子） 如何有工作牌在行李大厅里明目张胆行骗? How do the **profiteers (cheaters)** have staff cards and blatantly cheat in the baggage hall? **有罪 guilty** **人 human** **欺骗 deceive** **商业 commerce** **有罪 guilty** **人 human** **骗 cheat**	disgust	surprise
	吓人 （可怕） ，中药比西药更不安全。 **Frightful (Fearful)**, Chinese medicine is less safe than Western medicine. **能 able** **促使 urge** **害怕 fear** **能 able** **促使 urge** **害怕 fear**	fear	disgust
Headline Categorization	转载一个成方 （秘方） ，主治一切骨折，据说一剂见效 We republish a **set prescription (secret prescription)** , which mainly treats all kinds of fractures, and is said to be effective with only one dose. **医 medical** **药物 medicine** **准备 prepare** **文书 document** **命令 order** **医 medical** **药物 medicine** **有效 effective** **医治 doctor** **全 all** **方法 method** **疾病 disease**	regimen	essay
	他是三征高句丽的强将 （猛将） ，最后死于一群无赖之手 He was a **good general (valiant general)** that attacked Goguryeo for three times, yet was killed by a group of rogues. **人 human** **军 military** **官 official** **人 human** **军 military** **官 official** **军队 army** **勇 brave** **争斗 fight**	history	story
Semantic Matching	A. 如何选择大哥大 （手机） ? A. How to choose **hand phone (mobile phone)**? B. 怎么选择大哥大 （手机） ? B. What is the way to choose **hand phone (cell phone)**? **携带 bring** **能 able** **用具 tool** **交流 communicate** **样式值 PatternValue** **携带 bring** **能 able** **用具 tool** **交流 communicate** **样式值 PatternValue**	same	different
	A. 初中生 （男生） 暗恋女生会有什么表现? A. What performance will **junior high school students (boy students)** have if they secretly love a girl? B. 初中生 （男生） 暗恋女生表现是什么? B. What is the performance of **junior high school students (boy students)** if they secretly love a girl? **学习 study** **教 teach** **场所 InstitutePlace** **人 human** **教育 education** **中等 intermediate** **学习 study** **教 teach** **场所 InstitutePlace** **人 human** **教育 education** **初等 elementary** **男 male**	same	different

Case Study for adversarial test. The **original words** are shown in parenthesis and colored in blue, while the **replaced words** (similar words calculated by Cilin (Tian and Zhao, 2010)) are colored in red. Both the base model and our model (i.e. Transformer v.s. Transformer-SEP) predict correctly on sentences with the original words, yet only ours succeed in the sentences with the replaced words. We show **sememes for original words** and **sememes for replaced words** in blue and red color boxes respectively. *Best viewed in color.*