Chinese Couplet Generation with Syntactic Information

Yan Song
University of Science and Technology of China
clksong@gmail.com

Abstract

Chinese couplet generation aims to generate a pair of clauses (usually generating a subsequent clause given an antecedent one) with certain rules (e.g., morphological and syntactical symmetry) adhered and has long been a challenging task with cultural background. To generate high-quality couplet (antecedent) clauses, it normally requires a model to learn the correspondences between antecedent and subsequent clauses under aforementioned rules and constraint of few characters with their concise usage. To tackle this task, previous studies normally directly adopt deep neural networks without explicitly taking into account fine-grained analysis of the clauses, in this paper, we propose to enhance Chinese couplet generation by leveraging syntactic information, i.e., part-of-speech (POS) tags and word dependencies. In doing so, we identify word boundaries in the antecedent clause and then use a special attention module to encode the syntactic information over the words for better generating the subsequent clause. Experimental results on a dataset for Chinese couplet generation illustrate the validity and effectiveness of our approach, which outperforms strong baselines with respect to automatic and manual evaluation metrics.[1]

1 Introduction

A Chinese antithetical couplet is composed of a pair of two sentences (i.e., an antecedent and a subsequent clause) with particular meaning that are usually applied to festivals or certain circumstances, which can be seen as a special type of poetry with a history in China of more than one thousand years.[2] Couplets are normally concise yet have profound and expressive ability to convey a variety of meanings, thoughts and emotions with limited number of characters. Among all interesting characteristics, the most unique one of couplets is their dueling pattern that the antecedent and subsequent clauses should have an one-to-one correspondence adhere to strict rules covering multiple aspects including tone, length, word usage and even syntax.

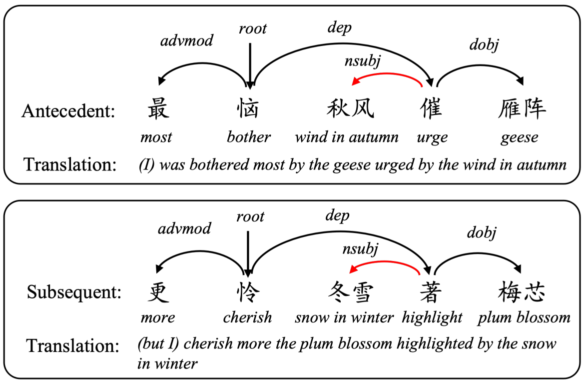

Figure 1: An example of a Chinese couplet pair marked with word dependencies and English translations.

For automatic Chinese couplet generation, recent studies follow the conventional encoder-decoder paradigm (Sutskever et al., 2014), with an general attention mechanism to capture the correspondence between the antecedent and subsequent clauses (Zhang et al., 2018; Fan et al., 2019; Gao et al., 2021). Although satisfactory results are obtained via their approaches, reasonable granularity matching and explicit structural correspondence are still worthy of improvement with special treatment. One important reason is that conventional approaches apply character-level encoding and decoding, larger granular text units are omitted in their models and thus led to the incapability of further syntactic analysis. Note that not all words in ancient Chinese are in the form of single-character, it is also applied to couplets that identifying boundaries of longer text spans is useful for their better interpretation, as shown in Figure 1 that the antecedent and subsequent clauses contains two-

[1] Related code and resources are available at https://github.com/synlp/ChiCoupletGen.

[2] People started to write paired sentences in *Pre-Qin* period over two thousand years ago, such as those ones widely used in 诗经 (*Classic of Poetry*) around 500 B.C. The first "specialized" couplet emerged as recorded in 蜀祷杌 (*Shu Tao Wu*), which was composed in the *Song* Dynasty around 964 A.D.

Proceedings of the 29th International Conference on Computational Linguistics, pages 6436–6446
October 12–17, 2022.

character words and their boundaries exactly match between clauses. Similarly, for the structural correspondence, the syntactic rule over the antecedent clause should be identical to the subsequent one, where to the best of our knowledge there is no previous study that focus on modeling couplet generation from this aspect. The example in Figure 1 also illustrates the syntactic structures of the clauses, with same word dependency patterns applied to them. For example, in the antecedent clause, the third word "秋风" (*wind in autumn*) is the nominal object (*nsubj*) of the predicate "催" (*urge*), and in the subsequent clause, the word "冬雪" (*snow in winter*) at the same position is also the nominal object of the predicate "著" (*highlight*) (marked by red arcs). As a result, enhancing Chinese couplet generation is expected to be done by identifying appropriate word boundaries and then building explicit syntactic correspondences over them[3] between the antecedent and subsequent clauses.

In this paper, we propose an approach to Chinese couplet generation with syntactic information over the identified words in the clauses, where the designed model not only provides a better understanding for couplets, but also enhances the generation process with more structural constraints. Specifically, we firstly learn word boundaries and then propose a special attention module to encode part-of-speech (POS) tags and word dependencies over the identified words in the antecedent clause and integrate such syntactic information into subsequent clause generation. Experimental results on a dataset for Chinese couplet generation demonstrate the effectiveness of our approach, which outperforms strong baselines on both automatic and manual evaluation metrics. Further analyses confirm the ability of our approach in generating couplets with neatness and artistic conception.

2 Preliminaries

Conventionally, Chinese couplet generation is regarded as a sequence-to-sequence text generation task (Fan et al., 2019; Zhang et al., 2018; Gao et al., 2021; Wang et al., 2021), whose objective is to generate character $\widehat{y}_t$ at the time step t with the given antecedent clause $\mathcal{X} = x_1, \cdots, x_t, \cdots, x_n$ (there are n characters in $\mathcal{X}$ and the t-th character is denoted as x_t) and the generated subsequent clause $\widehat{\mathcal{Y}}_{t-1} = \widehat{y}_1, \cdots, \widehat{y}_{t-1}$. The entire generation process is formalized as

$$\widehat{y}_t = \underset{y_t \in \mathcal{V}}{\arg\max}\ p(y_t|\widehat{\mathcal{Y}}_{t-1}, \mathcal{X}) \tag{1}$$

where p denotes the probability of a character y_i in the vocabulary $\mathcal{V}$ given $\widehat{\mathcal{Y}}_{t-1}$ and $\mathcal{X}$.

In general, to compute p, an encoder f_e is firstly used to encode the antecedent clause $\mathcal{X}$ through

$$\mathbf{h}_1, \cdots, \mathbf{h}_t, \cdots, \mathbf{h}_n = f_e(\mathcal{X}) \tag{2}$$

where $\mathbf{h}_t$ is the encoded hidden vector for the character x_t. Then, a decoder f_d takes all generated characters (i.e., $\widehat{\mathcal{Y}}_{t-1}$) at time step t, as well as all hidden vectors obtained from the encoder, and obtain the output vector $\mathbf{o}_t$ via

$$\mathbf{o}_t = f_d(\widehat{\mathcal{Y}}_{t-1}, \mathbf{h}_1, \cdots, \mathbf{h}_n) \tag{3}$$

Afterwards, $\mathbf{o}_t$ is mapped to the output space through a fully connected layer, where a $softmax$ function is further applied to the resulting vector to obtain the probability distribution over the vocabulary (i.e., $p(y_t|\widehat{\mathcal{Y}}_{t-1}, \mathcal{X})$ in Eq. (1)).

3 The Proposed Approach

In order to generate high-quality clauses that satisfies the the structure constrains of Chinese couplets, we propose an approach of neural text generation model enhanced by leveraging syntactic information. Figure 2 illustrates the overall architecture of our approach for Chinese couplet generation following the convention sequence-to-sequence paradigm, where the word boundary learning process and a special attention module to leverage syntactic information are presented on the top-left and top-right parts, respectively. In the following text, we firstly illustrate the word boundary learning process and then elaborate the proposed attention mechanism for syntactic information encoding.

3.1 Word Boundary Learning

Most existing approaches (Zhang et al., 2018; Fan et al., 2019; Gao et al., 2021) for Chinese couplet generation employ character-based encoding and decoding. However, consider that text in larger granularity (e.g., words) tend to deliver intact meanings in many cases, as well as the fact that not all characters in Chinese couplets should be treated as single-character words, character-level encoding has the limitation in losing particular semantic information and the risk of leading to inferior results for Chinese couplet generation.

To address this limitation, we propose to enhance

[3] Another key reason that we parse on words instead of characters is that existing parses are normally trained on words.

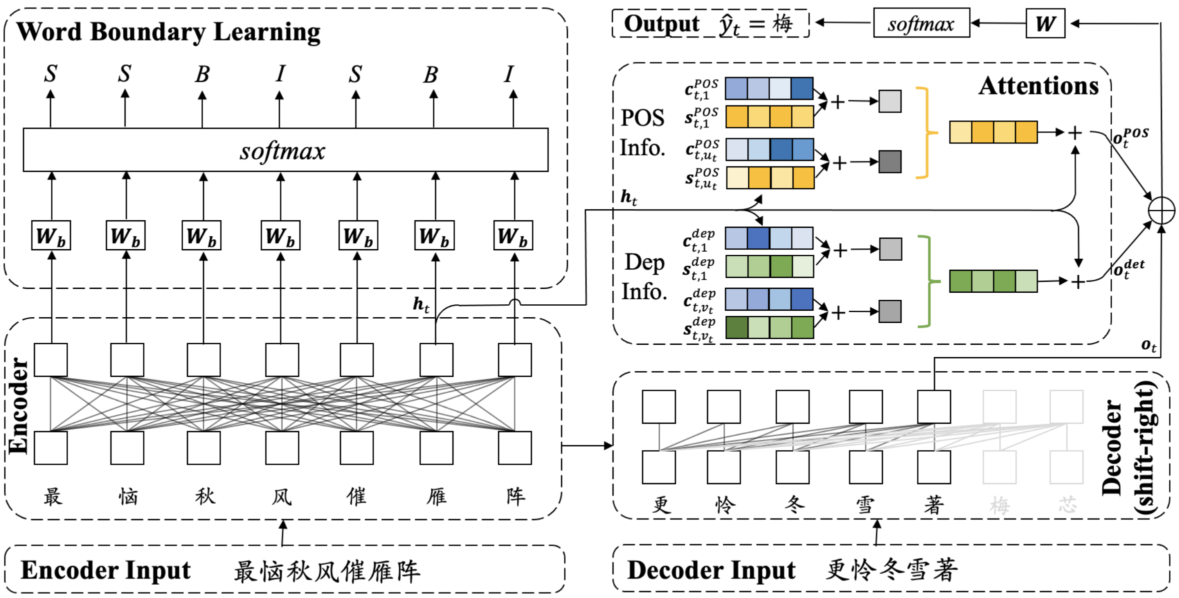

Figure 2: The overall architecture of our proposed approach for Chinese couplet generation, with an example input antecedent clause (i.e., the encoder input) and its corresponding subsequent clause (i.e., the decoder input) being generated at the sixth step on the character "梅". The process to learn word boundaries is illustrated on the top-left part; the special attention module used to encode the POS information (POS Info.) and dependency information (Dep Info.) associated with the current character (i.e., the six-th character "雁") is illustrated on the top-right part.

Chinese couplet generation with a pre-processing of learning couplet clauses in different granularities, i.e., both characters and words, which is performed by identifying word boundaries in the antecedent. Specifically, we firstly segment $\mathcal{X}$ into several words[4], where each character x_t is assigned a word boundary label y_t^b based on the position of the character in the word[5]. Then, for each character x_t, we take its hidden vector $\mathbf{h}_t$ obtained from the encoder and feed it into a fully connected layer with $softmax$ activation function through

$$\mathbf{p}_t^b = softmax(\mathbf{W_b} \cdot \mathbf{h}_t + \mathbf{b}_b) \quad (4)$$

where $\mathbf{W_b}$ and $\mathbf{b}_b$ are trainable matrix and bias vector in the fully connected layer and $\mathbf{p}_b$ is the probability distribution vector over the the word boundary label set with the value at each dimension illustrating the probability of character x_t having the corresponding word boundary label. Afterwards, the model predicts the word boundary label $\widehat{y}_t^b$ of x_t with the highest probability, computes the negative log-likelihood loss by comparing $\widehat{y}_t^b$ with y_t^b,, and updates model parameters accordingly.

Through this process, the model learns the word boundary information from the antecedent clause (so as the subsequent clause when training)[6] and preserve it in an implicit manner for later processes.

3.2 Attentive Syntactic Information Encoding

Consider that the effectiveness of using syntactic information to improve neural models on many natural language understanding (NLU) tasks has been demonstrated by previous studies (Strubell et al., 2018; Zhang et al., 2019; Guo et al., 2019; Tian et al., 2020a; Chen et al., 2020; Wang et al., 2020), it is straightforward to consider its usefulness for couplet generation in modeling the syntactic correspondence between antecedent and subsequent clauses. Therefore in this work, we propose a special attention module to leverage it in doing so.

Specifically, in our approach, we focus on two types of syntactic information, namely, POS tags and word dependencies, which (silver standard annotations) can be obtained from off-the-shelf natural language processing (NLP) toolkits for the input antecedent clause $\mathcal{X}$ (where the preserved word boundaries are input to the toolkit in obtaining the syntactic information). For each character

[4]In practice, one can either use an existing Chinese word segmenter or use human annotations for this step. The advantage of using human annotation allows one to train a model learning high-quality word boundary information and thus could lead to better understanding of the couplets.

[5]For example, one can use the conventional *BIES* schema for the word boundary label, where the label of a character is "B", "I", "E" if the character is at the initial, inside, final position of a word, respectively, and the label "S" stands for the case if the character is a single-character word.

[6]In our practice, when use the model for inference, we do not predict the word boundary label in subsequent clauses.

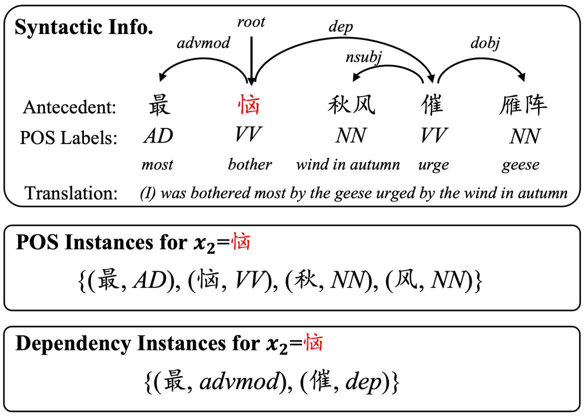

Figure 3: An example antecedent with its POS tags and word dependencies obtained from an off-the-shelf toolkit, where the POS and dependency instances associated with the second character "恼" (*bother*) are presented for better illustrating syntactic instance extraction. English translation is also provided for reference.

x_t, we firstly extracts a set of POS instances (tags) $POS_i = \{(c^{POS}_{t,i}, s^{POS}_{t,i})|1 \leq i \leq u_t\}$ and a set of dependency instances $Dep_i = \{(c^{dep}_{t,i}, s^{dep}_{t,i})|1 \leq i \leq v_t\}$ associated with x_t, where each POS and dependency instance is a pair of context feature $c^{type}_{t,i}$ and syntactic feature $s^{type}_{t,i}$ ($type \in \{POS, dep\}$).[7]

Specifically, for POS instances, we employ a five-character window[8] to extract the context characters (i.e., $x_{t-2} \cdots x_{t+2}$) and regard each of them as the context feature $c^{POS}_{t,i}$ in a POS instance. For the corresponding syntactic feature $s^{POS}_{t,i}$, we use the POS label associated with the word that contains $c^{POS}_{t,i}$. For example, in Figure 3, for x_2 = "恼" (*bother*), the associated POS instances are ("最", *AD*), ("恼", *VV*), ("秋", *NN*), ("风", *NN*). For dependency instances, the extraction process is elaborated as follows. First, we find the word (denote it as w) that contains x_t and extract all dependents (denote them as $w'_1 \cdots w'_j \cdots w'_l$) of w from the parsed dependency tree. Next, for each w'_j, we regard each character in w'_j as a context feature $c^{dep}_{t,i}$ and use the dependency connection type between w and w'_j as the corresponding syntactic feature $s^{dep}_{t,i}$ for each $c^{dep}_{t,i}$. As a special case, if w has no dependent (outbound connection to other words), we regard all characters in w as context features and use *null* as their corresponding syntactic features. For the example clause in Figure 3, x_2="恼" (*bother*) is associated with dependency instances ("最", *advmod*), ("催", *dep*), while x_1="最" (*most*) corresponds to the dependency instance ("最", *null*).

Once the POS and dependency instances are extracted, we use two separate attention parts following the same encoding procedure to model them, respectively, where the syntactic instances in each type (i.e., either POS or dependency) are dynamically weighed within its own type and distinguished based on their contribution to couplet generation, so as to address the noise in the syntactic instances. Using the encoding of POS instances as an example, for each instance $(c^{POS}_{t,i}, s^{POS}_{t,i})$, we firstly map the context feature $c^{POS}_{t,i}$ and the syntactic feature $s^{POS}_{t,i}$ to their embeddings, i.e., $\mathbf{c}^{POS}_{t,i}$ and $\mathbf{s}^{POS}_{t,i}$, respectively (we use boldface to represent the embedding of the features). Next, we add $\mathbf{c}^{POS}_{t,i}$ and $\mathbf{s}^{POS}_{t,i}$ to obtain the instance representation $\mathbf{e}^{POS}_{t,i} = \mathbf{c}^{POS}_{t,i} + \mathbf{s}^{POS}_{t,i}$. Then, we compute the attention $a^{POS}_{t,i}$ assigned to the instance by

$$a^{POS}_{t,i} = \frac{exp(\mathbf{h}_t \cdot \mathbf{e}^{POS}_{t,i})}{\sum_{i=1}^{u_t} exp(\mathbf{h}_t \cdot \mathbf{e}^{POS}_{t,i})} \quad (5)$$

Afterwards, we apply $a^{POS}_{t,i}$ to $\mathbf{e}^{POS}_{t,i}$ and computes the weighted sum of different POS instances, where the hidden vector $\mathbf{h}_t$ of x_t is further added to the resulting vector to obtain the encoded output vector $\mathbf{o}^{POS}_t$ with POS information through

$$\mathbf{o}^{POS}_t = \sum_{i=1}^{u_t} a^{POS}_{t,i} \cdot \mathbf{e}^{POS}_{t,i} + \mathbf{h}_t \quad (6)$$

Similarly, we obtain the output vector $\mathbf{o}^{dep}_t$ for dependency information following the same procedure. At last, we concatenate $\mathbf{o}^{POS}_t$, $\mathbf{o}^{dep}_t$, and $\mathbf{o}_t$ (obtained from the decoder in Eq. (3)) and feed the resulting vector $\mathbf{o}'_t = \mathbf{o}^{POS}_t \oplus \mathbf{o}^{dep}_t \oplus \mathbf{o}_t$ to the last fully connected layer for final prediction.

4 Experiments

4.1 Dataset

We follow Zhang et al. (2018) and use their experimental dataset[9] with 785K pairs of couplets crawled from Internet. Since there is no official train/dev/test split for this dataset, we make our own for each part and report the statistics of them in Table 1. In addition, we randomly sampled 1,000 couplets from the training set and manually annotated word segmentation for them,[10] where the re-

[7] Herein, u_t and v_t denote the numbers of POS and dependency instances associated with x_t, respectively.

[8] We use five as the window size because it is a hyper-parameter setting used in many previous studies to leverage POS tags and five achieves the optimal results in experiments.

[9] https://gitlab.com/feng-7/VV-couplet

[10] Annotating such a small amount of data does not require heavy manual work while later experiments confirm

Dataset	Couplet #	Char #
Train	755K	7,071K
Dev	10K	94K
Test	20K	186K

Table 1: The statistics of the Chinese couplet dataset with respect to the number of couplet pairs and characters in the training, development, and test sets.

Hyper-parameters	Values
Learning Rate	$\mathbf{1e^{-5}}, 5e^{-5}, 1e^{-4}, 1e^{-3}$
Dropout Rate	0.1, **0.2**, 0.3, 0.4
Batch Size	8, 16, **32**

Table 2: The hyper-parameters tested in tuning our models. Bold values illustrate the best hyper-parameter configuration that is applied in our experiments.

sulted dataset is used for our alternative training process with labeled word boundary information.

4.2 Implementation Details

Since our approach requires syntactic information as extra input features, for each input antecedent, we obtain the POS tags via TwASP[11] (Tian et al., 2020b) and the word dependencies via DMPar[12] (Tian et al., 2022).[13] For the encoder, considering that pre-trained language models have demonstrated their effectiveness in obtaining high-quality text representations for many NLP tasks (Yang et al., 2019; Diao et al., 2020; Raffel et al., 2019; Sun et al., 2020; Song et al., 2021), in the experiments, we use the pre-trained Chinese BERT-base (Devlin et al., 2019) following the default settings (i.e., 12 layers of self-attention with 768 dimensional hidden vectors). For the decoder, we use the standard Transformer setting with 6 layers of self-attention and 768 dimensional hidden vectors.

In training, we design an alternatively training strategy, where we firstly train the model with both losses on word boundary information and couplet generation on the 1,000 human annotated instances, and then train the model on the rest entire training set without considering word boundary loss since word segmentation for such data is provided by automatic tools. So that in each iteration the model can be enhanced by gold-standard word boundary information from few training instances. For other hyper-parameters, we try the combinations illustrated in Table 2 and use the ones (which are highlighted by boldface) that achieve the best performance on the development set. Following previous studies, we use BLEU (Papineni et al., 2002) and format accuracy for model evaluation).

it is enough to help our model in better generating couplets.

[11]https://github.com/SVAIGBA/TwASP

[12]https://github.com/synlp/DMPar

[13]TwASP is a joint model for Chinese word segmentation and POS tagging. So we use it to obtain the word segmentation result for each couplet and feed it to dependency parsing.

4.3 Overall Performance

Table 3 reports the experimental results from several BERT-based baselines and our proposed model enhanced by word boundaries and syntactic information (i.e., POS tags and word dependencies), respectively. Specifically, "BERT" denotes the vanilla BERT baseline model, "+WB", "+POS", and "+Dep" refer to the enhancement of the baseline with word boundaries, POS information, and dependency information, respectively, with "+Full" denoting the full model combines all the aforementioned information. There are several observations that are explained in the following paragraphs.

First, overall, although the baselines with different settings have already achieved outstanding performance, it is promising to observe that our model with all enhancements (i.e., "+Full") is able to outperform them with respect to all evaluation metrics, which confirms the usefulness of leveraging different information for couplet generation.

Second, comparing "+WB" and "BERT", we observe that "+WB" presents higher performance than "BERT" on all evaluation metrics (especially on BLEU-3 and BLEU-4 scores), which demonstrates the effectiveness of our word boundary learning process to leverage word boundary information in improving couplet generation. The interpretation is as follows. Although word boundary information is leveraged in an implicit way, it is encoded and the loss back-propagated to the encoder helps the model understand the semantic units in different granularities in the antecedent clause, then the generation process is enhanced accordingly with such information.

Third, models with a single type of syntactic information (i.e., "+POS" and "+Dep") outperform the baseline model with respect to all evaluation metrics, showing their effectiveness in guiding the model to generate better couplets. Furthermore, it is observed that the model with word dependencies ("+Dep") outperforms the the one with POS tags ("+POS") on BLEU-2, BLEU-3, and BLEU-4. This observation indicates that word de-

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	FA	Para. #	Speed
BERT	20.53	13.09	9.88	7.62	98.64	156M	49.6
+ WB	20.99	13.21	10.01	7.87	98.70	156M	48.7
+ POS	21.34	13.63	10.23	8.50	98.98	189M	45.9
+ Dep	21.39	13.82	10.36	8.77	99.01	189M	46.4
+ Full	**21.62**	**13.95**	**10.68**	**8.94**	**99.10**	221M	42.1

Table 3: The BLEU scores and format accuracy (FA) of different models on the test set. "BERT" denotes the baseline model with BERT encoder; "+WB", "+POS", and "+Dep" refer to the baseline model with enhancement of word boundaries, POS tags, and dependency information, respectively; "+Full" is our proposed model with enhancement of all aforementioned information. "Para. #" presents the number of parameters in different models; "Speed" refers to the number of generated couplets per second in inference.

Model	BL-1	BL-2	BL-3	BL-4	FA
BERT	20.53	13.09	9.88	7.62	98.64
+WB	**20.99**	**13.21**	**10.01**	**7.87**	**98.70**
+Word	17.98	9.63	6.05	4.21	90.01

Table 4: Comparison between two different BERT-based approach to leverage word boundary information without using syntactic information. "+WB" denotes the approach with the proposed word boundary learning process; "+Word" refer to the approach that encodes and decodes the couplets on word level. The results of the BERT baseline is also reported for reference. "BL" is the abbreviation of "BLEU".

Model	Syntax Intgrt.	BL-1	BL-2	BL-3	BL-4	FA
+POS	Ours	**21.34**	**13.63**	**10.23**	**8.50**	**98.98**
	Alter.	21.26	13.01	9.89	8.29	98.60
+Dep	Ours	**21.39**	**13.82**	**10.36**	**8.77**	**99.01**
	Alter.	21.24	13.57	10.14	8.01	98.91
+Full	Ours	**21.62**	**13.95**	**10.68**	**8.94**	**99.10**
	Alter.	21.42	13.67	10.20	8.15	98.82

Table 5: Comparison of model performance between different methods to integrate syntactic information in different settings. "Ours" denotes our approach to integrate POS or dependency information, where the hidden vector $\mathbf{h}_t$ is added to the weighted sum of syntactic instances (see Eq. (6)), whereas "Alter." refers to the method where $\mathbf{h}_t$ is not used in the integration.

pendencies provide stronger enhancement to couplet generation than that of POS tags, which could be explained by that the dependency information contains long distance constraints over the entire couplet clause while a POS tag only presents the function of the corresponding local words.

5 Analysis

5.1 The Effect of Word Boundaries

To analyze the effect of our model design by leveraging word boundary information in an implicit way, we run experiments with another straightforward approach (which is denoted as "+Word"), where the couplets are segmented into words and the couplet generation process is performed on the word-level prediction. Table 4 reports the performance of "+Word" model, as well as our "+WB" model and the BERT baseline. It is observed that "+Word" shows inferior results compared with our "+WB" model and the BERT baseline, which confirms the superiority of the designed learning process.[14] This observation can be explained by that the "+Word" approach generates word-by-word subsequent clauses, the vocabulary size for it is relatively larger than that of "+WB", which performs the generation in a character-by-character manner. The generation process of "+Word" is thus inevitably bothered by the out-of-vocabulary problem where at many decoding steps the model cannot find appropriate word candidates.

5.2 The Effect of Syntactic Information

Similar to the analysis for word boundaries, there are also alternatives to integrate syntactic heuristics into our model, where our approach also use an implicit way by adding the hidden vector $\mathbf{h}_t$ obtained from the encoder to the weighted sum of syntactic information (see Eq. (6)). In this investigation, we run another model without using $\mathbf{h}_t$, which means only the output of the attention module is applied for syntactic information integration. This comparison illustrates how syntactic information affects model performance, where the results are presented in Table 5. Clearly, the alternative method shows inferior results compared to our approach with respect to all metrics. The reason for this observation complies with our hypothesis that although directly

[14] We still use character-based BLEU in this experiment to evaluate the generated clause for all models.

Model	Syn.	Sem.	All
BERT	0.784	0.636	1.420
+WB	0.805	0.659	1.464
+POS	0.818	0.679	1.497
+Dep	0.809	0.665	1.474
+Full	0.830	0.728	1.558

Table 6: Human evaluation for different models. "Syn." and "Sem." are syntactic and semantic scores, respectively, and "All" refers to the sum of them.

using the syntactic information shows its advantage over the BERT baseline (see Table 3), the decoding process still relies mainly on the contextual information from the encoder. Therefore, only using the syntactic information from the attention module in a standalone way may not be enough to drive a better couplet generation process.

5.3 Human Evaluation

Following the convention in previous studies (Jiang and Zhou, 2008; He et al., 2012; Zhang et al., 2018; Fan et al., 2019; Gao et al., 2021; Wang et al., 2021), we perform human evaluation for different models in addition to BLEU scores and format accuracy. Particularly, we use "syntactic" and "semantic" satisfaction to assess how antecedent and subsequent clauses are matched in terms of their patterns[15] and meanings[16], respectively. In doing so, human annotators are given an antecedent clause and a subsequent clause generated by a model and they are asked to assign 0-1 score to the generated subsequent for both criteria ("0" for *not satisfying*, "1" for *satisfying*).

We randomly select 50 couplets from the test set for human evaluation and employ five human experts who are familiar with Chinese couplets to conduct human evaluation with a blind review manner. In detail, they are provided with the subsequent clauses generated by different models (i.e., all baselines and the "+Full" model) and they do not know the model that generates the given subsequent clause. The results from five experts on the aforementioned two aspects (i.e., syntactic satisfaction and semantic satisfaction) are averaged and presented in Table 6. Similar to the observations from Table 3, word boundaries and syntactic information show their advantages in helping couplet generation with more satisfying results from humans. Particularly, compared with automatic evaluations (i.e., BLEU scores and format accuracy), human judgements for different models shows a more significant trend that the proposed components are useful especially when they are integrated together in our proposed approach.

[15] There is a special assessment about the ending tone of the two clauses, where the tones of the last character in antecedent and subsequent clauses should be opposed to each other.

[16] Ideally, the meaning of the subsequent clause should be relevant to the antecedent clause but not the same.

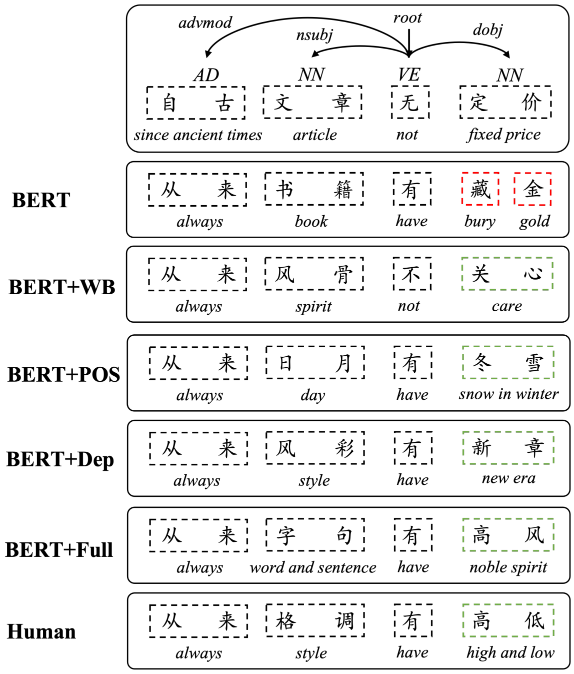

Figure 4: Comparison of subsequent clause generated from different models for an example input antecedent clause with its POS tags and word dependencies makred (presented at the top). Original subsequent clause (Human) written by human is also presented at the bottom. Translations for every clause in the verbatim manner are provided for reference. Characters belonging to the same word are bounded by dash boxes, where mismatched patterns is marked in red boxes.

5.4 Case Study

To further investigate the effect of our approach, we conduct a case study to qualitatively illustrate the performance of different models with their generated subsequent clauses for a given input. Figure 4 illustrates the results. For the given antecedent, there are five generated clauses from all baselines and our proposed model (i.e., "BERT+Full"), as well as the original subsequent clause composed by human for reference, where all clauses are segmented (in dashed boxes) in order to explicitly show word boundaries. It is clearly presented that, the BERT baseline (i.e., "BERT") fails to produce a subsequent clause with correct word mappings to the antecedent one, e.g., the word boundary of "藏/金" (*bury the gold*) does not match that

Antecedent:	几笔人生诠世味
Translation:	*A few strokes of life interpret the taste of the world*
Generation:	一轮明月鉴冰心
Translation:	*A bright moon reflects my pure heart*

Antecedent:	踏遍青山长骨气
Translation:	*Walking all over the green mountains to develop the integrity character*
Generation:	邀来明月好心情
Translation:	*Inviting the bright moon to change your mood*

Antecedent:	寒苦并非梅本色
Translation:	*Cold and bitterness is not the true nature of plum*
Generation:	温馨总是菊真情
Translation:	*Warmth is always the true character of chrysanthemum*

Antecedent:	花好朝朝携酒醉
Translation:	*Drink every day in a place with a great view*
Generation:	月圆夜夜伴愁眠
Translation:	*Sleep sadly under the company of the full moon*

Antecedent:	闲倚纱窗观秋雨
Translation:	*Leaning on the screen window to watch the autumn rain*
Generation:	醉眠石榻听松涛
Translation:	*Drunk sleeping on the stone couch listening to the pines*

Antecedent:	一场风雨匆匆过
Translation:	*A storm passed quickly*
Generation:	几缕春情渐渐生
Translation:	*Some love developed gradually*

Figure 5: Six example pairs of Chinese couplets with their subsequent clauses generated by our proposed model (i.e., "BERT+Full") corresponding to the input antecedent clauses. The English translations are also provided for each clause for all couplets based on the semantics of each entire sentence instead of the verbatim manner.

of "定价" (*a fixed price*). All rest clauses generated from the models enhanced by word boundaries or syntactic information do not suffer from such mis-matching problem, where the one from "BERT+Full" is indisputably better than the others and the reason can be explained in three aspects.

First, the pattern in the subsequent clause is identical to that of the antecedent clause, including that the tones of the last character in both clauses satisfy the rule mentioned in the previous section (see footnote 15), such as "风" (flat tone) v.s. "价" (oblique tone). Second, each unit in the subsequent clause shows a direct correspondence with the unit at the same position in the antecedent clause, e.g., "字句" (*word and sentence*) v.s. "文章" (*article*), "无" (*not*) v.s. "有" (*has*), "高风" (*noble spirit*) v.s. "定价" (*fixed price*), etc. Third, the meaning of the entire subsequent clause makes a counterpoint to the antecedent one with semantic relatedness, i.e., the subsequent clause refers to that *the words and sentences always have noble spirit* while the antecedent clause states that *the articles are priceless since ancient times*. As comparisons, although other generated clauses have their advantages in delivering either more interesting meaning or better artistic conceptions, they are not well performed on pattern matching or semantic relatedness.

To have more intuitive understanding about the performance of our proposed model, we also show in Figure 5 six randomly selected pairs of couplets with generated subsequent clauses. It is clearly to observe from all couplet pairs that the generated clauses match the antecedent inputs well on their patterns (on all aspects) and also keep the balance of expressing good artistic conception while being strictly corresponded to the meaning of the antecedent clauses, such as "温馨" (*warmth*) v.s. "寒苦" (*cold and bitterness*), "明月" (*the bright moon*) v.s. "青山" (*the green mountains*), which are all neat correspondences and promote the overall effect (including both meaning and artistic conception) of the entire couplet pairs to a higher level.

6 Related Work

Chinese couplet generation is an intriguing natural language generation task, which is relevant to poem generation (Greene et al., 2010; Zhang and Lapata, 2014; Wang et al., 2016; Yang et al., 2018b; Zhang et al., 2017; Yi et al., 2017; Ghazvininejad et al., 2017; Yi et al., 2018; Yang et al., 2018a; Li et al., 2018; Liu et al., 2018, 2019; Liao et al., 2019; Bena and Kalita, 2020). Yet, couplet generation differs from that for poem in the way that it requires the generation process sticking to more strict patterns and semantic requirements, while they are similar that they all conventionally performed as a sequence-to-sequence text generation task, including statistic-based approaches (Jiang and Zhou, 2008; Zhang and Sun, 2009; He et al., 2012) and neural approaches following the encoding-decoding paradigm (Zhang et al., 2018; Fan et al., 2019; Gao et al., 2021). To improve model performance, previous neural approaches try to model the character-character correspondence between antecedent and subsequent clauses, where some advanced approaches such as attention mechanism (Zhang et al., 2018) and character embedding pre-training (Gao et al., 2021) are applied. Compared with existing studies, our approach offers an alternative and explicit way to model the correspondence between couplet clauses through syntactic information, which provides useful knowledge to control patternized generation and is integrated into our approach via a carefully designed module.

7 Conclusion

In this paper, we proposed a neural approach following the encoder-decoder paradigm for Chinese couplet generation enhanced by of syntactic information (i.e., POS tags and word dependencies). Specifically, our approach models word boundaries to facilitate the learning of syntactic information, where POS tags and word dependencies are leveraged to provide pattern guidance for couplet generation through a special attention module. Experimental results on a prevously used dataset for Chinese couplet generation illustrate the effectiveness of our approach, which outperforms strong baselines on both automatic and human evaluations. Further analyses also confirm the ability of our approach to generating high quality couplets.

Acknowledgements

We thank Mr. Dongming Li for his early preliminary help to this study.

References

Brendan Bena and Jugal Kalita. 2020. Introducing Aspects of Creativity in Automatic Poetry Generation. *arXiv preprint arXiv:2002.02511*.

Guimin Chen, Yuanhe Tian, and Yan Song. 2020. Joint Aspect Extraction and Sentiment Analysis with Directional Graph Convolutional Networks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 272–279.

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.

Shizhe Diao, Jiaxin Bai, Yan Song, Tong Zhang, and Yonggang Wang. 2020. ZEN: Pre-training Chinese Text Encoder Enhanced by N-gram Representations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4729–4740.

Haoshen Fan, Jie Wang, Bojin Zhuang, Shaojun Wang, and Jing Xiao. 2019. Automatic Acrostic Couplet Generation with Three-Stage Neural Network Pipelines. In *Pacific Rim International Conference on Artificial Intelligence*, pages 314–324. Springer.

Rui Gao, Yuanyuan Zhu, Mingye Li, Shoufeng Li, and Xiaohu Shi. 2021. Encoder–Decoder Couplet Generation Model Based on 'Trapezoidal Context'Character Vector. *The Computer Journal*, 64(3):286–295.

Marjan Ghazvininejad, Xing Shi, Jay Priyadarshi, and Kevin Knight. 2017. Hafez: an Interactive Poetry Generation System. In *Proceedings of ACL 2017, System Demonstrations*, pages 43–48.

Erica Greene, Tugba Bodrumlu, and Kevin Knight. 2010. Automatic Analysis of Rhythmic Poetry with Applications to Generation and Translation. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 524–533.

Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. Attention Guided Graph Convolutional Networks for Relation Extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 241–251.

Jing He, Ming Zhou, and Long Jiang. 2012. Generating Chinese Classical Poems with Statistical Machine Translation Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26.

Long Jiang and Ming Zhou. 2008. Generating Chinese Couplets Using a Statistical MT Approach. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 377–384.

Juntao Li, Yan Song, Haisong Zhang, Dongmin Chen, Shuming Shi, Dongyan Zhao, and Rui Yan. 2018. Generating Classical Chinese Poems via Conditional Variational Autoencoder and Adversarial Training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3890–3900, Brussels, Belgium.

Yi Liao, Yasheng Wang, Qun Liu, and Xin Jiang. 2019. GPT-based Generation for Classical Chinese Poetry. *arXiv preprint arXiv:1907.00151*.

Dayiheng Liu, Quan Guo, Wubo Li, and Jiancheng Lv. 2018. A Multi-modal Chinese Poetry Generation Model. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Zhiqiang Liu, Zuohui Fu, Jie Cao, Gerard de Melo, Yik-Cheung Tam, Cheng Niu, and Jie Zhou. 2019. Rhetorically Controlled Encoder-decoder for Modern Chinese Poetry Generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1992–2001.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-text Transformer. *arXiv preprint arXiv:1910.10683*.

Yan Song, Tong Zhang, Yonggang Wang, and Kai-Fu Lee. 2021. ZEN 2.0: Continue Training and Adaption for N-gram Enhanced Text Encoders. *arXiv preprint arXiv:2105.01279*.

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.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. ERNIE 2.0: A Continual Pre-Training Framework for Language Understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8968–8975.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. *Advances in Neural Information Processing Systems*, 27:3104–3112.

Yuanhe Tian, Wang Shen, Yan Song, Fei Xia, Min He, and Kenli Li. 2020a. Improving Biomedical Named Entity Recognition with Syntactic Information. *BMC Bioinformatics*, 21:1471–2105.

Yuanhe Tian, Yan Song, Xiang Ao, Fei Xia, Xiaojun Quan, Tong Zhang, and Yonggang Wang. 2020b. Joint Chinese Word Segmentation and Part-of-speech Tagging via Two-way Attentions of Auto-analyzed Knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8286–8296, Online.

Yuanhe Tian, Yan Song, and Fei Xia. 2022. Enhancing Structure-aware Encoder with Extremely Limited Data for Graph-based Dependency Parsing. In *Proceedings of the 29th International Conference on Computational Linguistics*.

Daisy Zhe Wang, Wei He, Hua Wu, Haiyang Wu, Wei Li, Haifeng Wang, and Enhong Chen. 2016. Chinese Poetry Generation with Planning Based Neural Network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1051–1060.

Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020. Relational Graph Attention Network for Aspect-based Sentiment Analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3229–3238, Online.

Yufeng Wang, Jiang Zhang, Bo Zhang, and Qun Jin. 2021. Research and Implementation of Chinese Couplet Generation System with Attention Based Transformer Mechanism. *IEEE Transactions on Computational Social Systems*.

Cheng Yang, Maosong Sun, Xiaoyuan Yi, and Wenhao Li. 2018a. Stylistic Chinese Poetry Generation via Unsupervised Style Disentanglement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3960–3969.

Xiaopeng Yang, Xiaowen Lin, Shunda Suo, and Ming Li. 2018b. Generating Thematic Chinese Poetry Using Conditional Variational Autoencoders with Hybrid Decoders. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4539–4545.

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 32*, pages 5753–5763.

Xiaoyuan Yi, Ruoyu Li, and Maosong Sun. 2017. Generating Chinese Classical Poems with RNN Encoder-decoder. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 211–223.

Xiaoyuan Yi, Maosong Sun, Ruoyu Li, and Wenhao Li. 2018. Automatic Poetry Generation with Mutual Reinforcement Learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3143–3153.

Hongming Zhang, Yan Song, Yangqiu Song, and Dong Yu. 2019. Knowledge-aware Pronoun Coreference Resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 867–876, Florence, Italy.

Jiyuan Zhang, Yang Feng, Dong Wang, Yang Wang, Andrew Abel, Shiyue Zhang, and Andi Zhang. 2017. Flexible and Creative Chinese Poetry Generation Using Neural Memory. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1364–1373.

Jiyuan Zhang, Zheling Zhang, Shiyue Zhang, and Dong Wang. 2018. VV-Couplet: An Open Source Chinese Couplet Generation System. In *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1756–1760. IEEE.

Kai-Xu Zhang and Mao-Song Sun. 2009. An Chinese Couplet Generation Model Based on Statistics and Rules. *Journal of Chinese Information Processing*, 1:017.

Xingxing Zhang and Mirella Lapata. 2014. Chinese Poetry Generation with Recurrent Neural Networks. In *Proceedings of the 2014 Conference on Empirical*

Methods in Natural Language Processing (EMNLP), pages 670–680.