Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 6361–6369
Marseille, 20-25 June 2022

CxLM: A Construction and Context-aware Language Model

**Yu-Hsiang Tseng, Cing-Fang Shih, Pin-Er Chen,
Hsin-Yu Chou, Mao-Chang Ku, Shu-Kai Hsieh**
Graduate Institute of Linguistics, National Taiwan University
Taipei, Taiwan
seantyh@gmail.com, r08142004@ntu.edu.tw, cckk2913@gmail.com,
r10142008@ntu.edu.tw, d08142002@ntu.edu.tw, shukaihsieh@ntu.edu.tw

Abstract

Constructions are direct form-meaning pairs with possible schematic slots. These slots are simultaneously constrained by the embedded construction itself and the sentential context. We propose that the constraint could be described by a conditional probability distribution. However, as this conditional probability is inevitably complex, we utilize language models to capture this distribution. Therefore, we build CxLM, a deep learning-based masked language model explicitly tuned to constructions' schematic slots. We first compile a construction dataset consisting of over ten thousand constructions in Taiwan Mandarin. Next, an experiment is conducted on the dataset to examine to what extent a pretrained masked language model is aware of the constructions. We then fine-tune the model specifically to perform a cloze task on the opening slots. We find that the fine-tuned model predicts masked slots more accurately than baselines and generates both structurally and semantically plausible word samples. Finally, we release CxLM and its dataset as publicly available resources and hope to serve as new quantitative tools in studying construction grammar.

Keywords: constructions, language model, slots, masked language model, construction grammar

1. Introduction

Constructions are direct form-meaning pairs without intermediate structures. Morphemes, words, idioms, and phrasal patterns are all constructions, and there are no clear-cut boundaries between lexicon and syntax (Fillmore, 1988). Following this approach, one does not have to assume a domain-specific cognitive process that is adapted to language (Croft, 2001). Constructions, just like syllables and words, are emerged from language users' everyday linguistic experiences: by sorting and matching incoming utterances by their similarities (Bybee, 2006). That is, the constructions do not have to be exactly the same; they may contain schematic slots, which could be filled by various elements. For example, the idiom construction, "*X* take *Y* for granted", has two schematic slots. They can be filled by various candidates: "*You take him for granted*", "*Many people take freedom for granted*" (Hoffmann and Trousdale, 2013).

Constructions do not leave their slots unconstrained. They have a preference and restriction on how the slots are being fulfilled in text. The measures of such constraints are extensively studied in the literature (Stefanowitsch, 2013). For example, one can compute the attraction and reliance of a word in a given construction (pattern) (Schmid, 2010). The attraction measures how a pattern prefers certain words to fill its slot; conversely, reliance indicates how a word tends to occur in particular constructions. A more detailed analysis may include the collostruction analysis, where the word attracted to a particular construction is referred to collexeme. One can study the attraction and repulsion in the slots with collexeme analysis; and the interaction among them with distinctive and covarying collexeme analyses (Stefanowitsch and Gries, 2003; Gries and Stefanowitsch, 2004; Stefanowitsch and Gries, 2005). These corpus statistics address how one can explore the construction, which inherently involves paradigmatic structure, from syntagmatic, linear, and collocational textual data.

The constraints of the slots could be further formulated as conditional probabilities, that is, what the most probable candidates (words) will be, given the construction itself and the sentential context. This conditional probability implies a (masked) language model, which could be n-gram models (Hanna et al., 2006; McMahon and Smith, 1998) or ones involving deep learning architecture. In recent years, numerous deep learning architectures have been found to be adept in such tasks. One of these models is BERT (bidirectional encoder representations from transformer) (Devlin et al., 2018), which uses a masked language model as a pretraining task and is trained with a transformer architecture (Vaswani et al., 2017). The BERT model is applied to virtually all NLP tasks and achieves good results on them. In addition to its practical value, multiple studies have attempted to argue the linguistic relevance of its internal model representations (Manning et al., 2020; Madabushi et al., 2020). These studies even show that BERT has

already learned something about constructions. If this is the case, it is fascinating and pertinent to ask whether the model could capture the conditional distributions of the construction slots, given the constructions, or, more challengingly, given the sentential context.

In this paper, we present CxLM, which captures the conditional probabilities in constructions' slots and generates high-quality samples from them. This paper is organized as follows. Section 2 briefly reviews construction grammar and how it is related to deep learning models. Section 3 describes how we compile a construction dataset consisting of 11,642 construction usages in Taiwan Mandarin via social media corpus. We then conduct an experiment to demonstrate that a pretrained masked language model is aware of the constructions in the sentence (Section 4). However, the pretrained model does not capture the word distributions in variable slots. Therefore, in Section 5, we fine-tune the model with a *variable slot cloze task* and evaluate the model quantitatively and qualitatively. Finally, we release CxLM model along with its dataset as publicly available resources [1]. We hope these resources to serve as new tools to studying constructions.

2. Related Works

2.1. Construction Grammar (CxG)

Construction Grammar (CxG) is a subfield of cognitive linguistics which assumes that patterns and syntax have their own meanings. They cannot be understood solely from their components or word order. CxG suggests that a form-meaning pair, namely a construction, is the fundamental unit of human language. A construction contains immutable parts and open slots, and its productivity is achieved by replacing the element in the slots. From the usage-based point of view, humans can recognize form-meaning pairs with high frequencies. Therefore, we are able to generalize the patterns and the alternating instances which compose the constructions (Goldberg, 2006).

In Cognitive Construction Grammar (CCG), the emergence of constructions is considered to be motivated by perception, cognition, and mutual interaction among speakers. Despite the various alternatives, constructions seem to acquire instantiating words that bear similarities to each other. In addition to the internal composition of constructions, the resemblance is also noticed between different constructions representing close semantic meanings (Boas, 2013). Kaschak and Glenberg (2000) also claim that we rely on constructions to interpret the meanings of new words and that sentences in the same construction have semantic relations. With its concentration on mental activity, CxG reflects the cognitive function in the human mind, which brings about insightful viewpoints to explore the intricate language structures.

2.2. CxGBERT

To discuss to what extent deep learning models could recognize constructional information, Madabushi et al. (2020) conducted experiments to evaluate the performance of BERT (Devlin et al., 2018), when tackling constructions in textual data. With an aim to evaluate the performance, probing techniques were adopted. Madabushi et al. (2020) categorized sentences in the WikiText-103 corpus based on Dunn's (2017) construction patterns. With the assumption that sentences with the same constructions share specific knowledge, a classification model, CxGBERT, was built to analyze whether two sentences contain the same construction or not.

Results showed that instances of the same construction could provide similar linguistic information with neural network models, which is in line with humans' conceptual behavior of representing related concepts by similar lexical expressions. Moreover, the results implied that CxGBERT may have the ability to predict the similarity among 21 thousand constructions. As the finite terms formed constructions with alike combinations, their results were in accordance. In terms of constructional information, CxGBERT seemed to be capable of acquiring semantic meanings of constructions via lexical instances.

CxGBERT reached a high accuracy after being trained by merely 500 examples of certain constructions. It implies the model does have access to a significant amount of information about the constructions. However, due to the nature of the classification task used in the study, the model tends to be less sensitive to highly general constructions, which accepted a large number of variables in the slots.

2.3. Learning Probability Distribution

One way to learn a complex, non-parametric probability distribution model is through a generative adversarial model (GAN) (Goodfellow et al., 2014). Essentially, it trains two models as a pair: one *discriminator*, which is trained to differentiate real and fake samples; and one *generator*, which tries to generate fake samples indifferentiable with the real ones. Specifically, GAN's idea is that the real samples are drawn from a complex joint probability distribution, and the generator is learning from that distribution under the discriminator's supervision signal. GAN models achieve great success in computer vision, speech processing and even apply to natural language processing (Gui et al., 2021). However, compared to image generation task, learning text probability distributions with

[1] https://github.com/lopentu/CxLM

GAN has a unique challenge. Texts are composed of discrete tokens rather than pixels of continuous real values; therefore, the backpropagation signals could not travel back to update the generator's parameters. One way to address the issue is to formulate the problem as a reinforcement learning task and generate the samples through policy gradient (Guo et al., 2018). However, Clark et al. (2020) found that if the task is just to generate samples from the masked sites, as in the MLM task, using maximum likelihood estimates (MLE) is good enough, if not better, than the policy gradient. Consistently, Alvarez-Melis et al. (2020) also showed that optimizing MLE is as efficient as training an adversarial model in NLP settings. Taken together, to learn the probability distribution underlying a particular site in the text, for example, the constructions' variable slots, training a masked language model may be a viable option.

Therefore, we build a masked language model called CxLM specifically tuned for constructions' variable slots. First, we compile a construction dataset. Secondly, we examine to what extent a pretrained model is already aware of constructions. Finally, we build CxLM and evaluate the model both quantitatively and qualitatively.

3. Construction Dataset

3.1. Data Pre-processing

To collect construction instances, we utilized *Xiàndài Hànyǔ Gòushì Shùjùkù* (CCGD)[2], a knowledge database of Mandarin Chinese constructions (Zhan, 2017). Being designed as a language resource based on CxG, it contains 1,110 simplified Chinese constructions. Each construction is a phrasal expression that conveys an idiomatic meaning that cannot be inferred from its components. For instance, the construction *a*+ 到 + 爆 (*a+dào+bào*) literally means *explosion*, but its genuine meaning is to exaggerate the intensity of the adjective *a*. Following Zhan (2017), the `a` is the schematic slot, or the variable in the construction; the fixed elements, i.e., 到 (*dào*), 爆 (*bào*) are constants.

This paper focuses on constructions that contain repetitive variables or constants. These constructions are productive in usage and less susceptible to *false negative* in pattern detection algorithm. Examples of these constructions are 走一走 (*zǒu yī zǒu* 'take a walk'), which has a repetitive variable 走 (*zǒu*), and 敢愛敢恨 (*gǎn ài gǎn hèn* 'dare to love and hate'), which has a repetitive constant 敢 (*gǎn*). Other constructions without repetitive elements are excluded from the dataset (e.g. '神 +verb' *shén*+verb 'extremely (stative verb)') because the algorithm cannot readily detect the intended construction usages (e.g. '神好吃' *shén hǎo chī* 'extremely delicious'), from those that incidentally follow the surface form (e.g. 神愛 (世人) *shén ài (shì rén)* 'God loved (the world)'). All construction candidates were automatically converted to traditional Chinese before proceeding to the next processing stage.

3.2. Construction Selection

To collect example sentences for the constructions, we first verified that the construction candidates (originally in simplified Chinese) were also commonly used in Taiwan Mandarin. Some constructions, such as 'X+ 黨' *X+dǎng* 'X+club' (e.g. 熬夜党 *áo yè dǎng* nightowl club), are less used this way in Taiwan Mandarin. Identifying these constructions also help us find the intended construction usages. Therefore, we devised a frequent-of-use annotation task. Two annotators, both of whom are Taiwan Mandarin native speakers with linguistics-related majors, were recruited to rate "how commonly use each construction candidate" is on the 5-point Likert scale. The rating scale from 1 to 5 was respectively very uncommon, uncommon, neutral, common, and very common. Constructions were removed from the candidate lists if rated as no more than 2 points by both annotators.

After determining commonly used constructions, their example sentences were collected from the PTT [3] text data. PTT is the largest bulletin board system (BBS) in Taiwan with more than 1.5 million registered users. There are more than ten thousand boards in PTT, and each board serves for a specific discussion topic. With its immediate and interactive nature, posts on PTT excellently reflect present-day language use. We gathered posts in 2020 from sixteen different boards, including popular boards such as `Gossiping` and `WomenTalk`, regional ones such as `Tainan` and `Kaohsiung`, and other boards with different content to balance the topics. All posts content was first segmented with Jseg (Liu, 2014) and POS-tagged with CkipTagger[4], an open-source Chinese NLP tool developed by Academia Sinica.

We extracted the construction instances with regular expressions for each selected constructions. To minimize the false positives, repetitive elements and part-of-speech tags were checked to better ensure the intended use of such construction patterns. However, since the extraction algorithm considered only forms, some of the instances were incorrect due to word segmentation errors or polysemous words. Therefore, we randomly selected 20 instances of each construction from PTT. We then devised another annotation task to verify whether the matched instance was the intended use of those construction patterns. Annotators should mark

[2]http://ccl.pku.edu.cn/ccgd/

[3]https://www.ptt.cc/bbs/index.html

[4]https://github.com/ckiplab/ckiptagger

```
{
    "board": "Kaohsiung",
    "cnstr_form": ["v", " 一", "v"],
    "cnstr_example":
            [" 動", " 一", " 動"],
    "text": ['運動', '強度',
            '沒有', '太', '高',
            '圖', '個',
            '動', '一', '動']
    "cnstr": ['O', 'O', 'O',
            'O', 'O', 'O', 'O',
            'BX', 'IX', 'IX'],
    "slot": ['O', 'O', 'O',
            'O', 'O', 'O', 'O',
            'BV', 'BC', 'BV']
}
```

Listing 1: An example of a construction instance

each instance as `1` (correct) or `0` (incorrect). Finally, 38 construction forms with 100% correct instances made it to our final construction list. The list included 11,642 constructions, and was stored in a `JSON` file. An example is demonstrated in Listing 1.

3.3. Slot Tags

After all data that comprise appropriate construction types were collected, we create inside–outside–beginning (IOB) tags for each instance. Three data fields are created for each instance: `text`, `cnstr`, and `slot`. `text` contains each of the raw tokens in the given sentence. `cnstr` and `slot`, on the other hand, symbolize the IOB tags given to their corresponding tokens in `text`. `cnstr` marks whether a character is inside, outside, or at the beginning of a construction. A `BX` tag that is given to a character stands for the beginning of a construction, while a character with an `IX` tag is located inside a construction. An `O` tag indicates that the token is outside of a construction. `slot` concerns the constant and the variable slots in a construction. The `BC` tag and the `BV` tag are attached to a token if it occupies the starting position of the constant and the variable slots, respectively, in a construction. The `IC` tag and the `IV` tag are given to the tokens inside the constant and variable slots, respectively, in a construction. And all the other characters that are outside of a construction also carry an `O` tag in `slot`. The IOB tags together with the textual data will serve as the input of the following experiments and model training. An example is demonstrated in Listing 1.

4. Constructions and MLM

To build a construction-tuned language model, we first examine whether, or to what extent, the model has already learned about constructions. If the model behavior is undifferentiated between constructions' sites and random masking, we should first improve the language model. On the other hand, if the model already learned something about constructions, we can furthermore fine-tune the model to learn the specifics, for example, the variable sites.

To examine the extent to which the pretrained model captures the constructions, we compare the conditional probabilities in a series of conditions. The underlying rationale is that if the model could already capture the occurrence of the constructions, along with its constants and variables, the conditional probability should be high. Conversely, if the model does not learn the usage of constructions, especially their constituents, the predictive probability should be low.

We systematically test the hypothesis with a series of cloze tasks. We compare three different masking sites. For each construction, we mask (1) the whole construction, (2) the constant sites, or (3) the variable sites. The hypothesis is that if the model already captures the usage of constructions, and given that the constructions are rather independent form-meaning pairs, the prediction of such usage will be harder for the model. The comparison is based on two different controls. The first control is *shifted* condition, where the masked sites have the same length as the constructions, but we shift the masks to the left or the right with a random offset. This condition helps us establish a baseline when a random number of consecutive tokens are masked. The second control is *random* condition, where the masked sites are randomly chosen and they are not necessarily following the original masked pattern. This condition establishes a baseline of masked token number. Specifically, if the masked constructions are harder to predict, but not in the shifted and the random conditions, the prediction difficulties cannot be attributed to the number of masked tokens or the consecutive masks alone.

Likewise, the constant sites and the variable sites have the three corresponding conditions as the whole constructions do. The contrasts of different conditions of constant sites help us clarify the nature of the fixed elements in the constructions. Similarly, the comparisons in variable sites will show how the model captures the word selection distributions on the open slots. The overall experiment scheme is shown in Figure 1

The experiment is carried out with a BERT-based model pretrained on traditional Chinese text, by CKIP, Academia Sinica [5] and an off-the-shelf pretrained `bert-base-chinese` model (Wolf et al., 2020). We randomly sampled 10% (1,165) of the items from our construction dataset (Sec. 3). For

[5] https://github.com/ckiplab/ckip-transformers

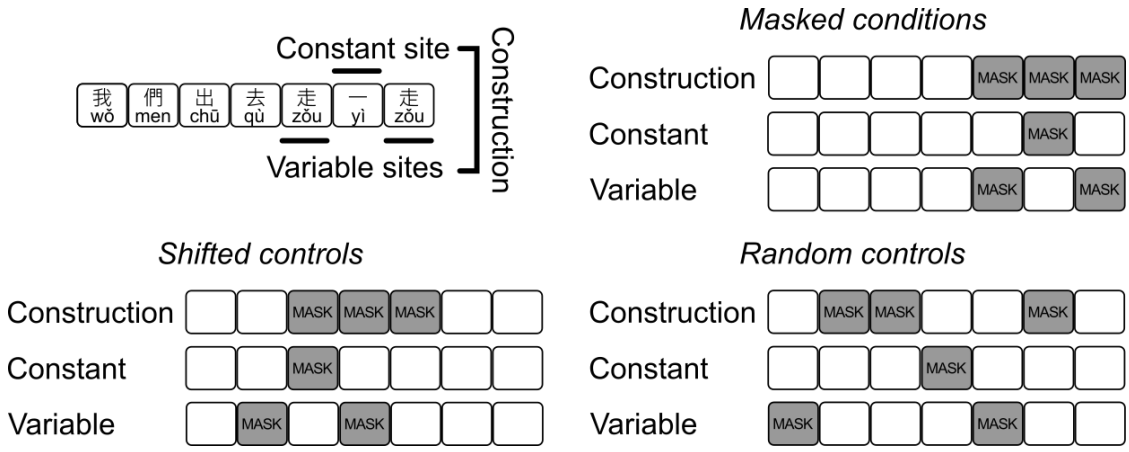

Figure 1: Masking conditions in the MLM experiment.

each sample, we mask the target sites in the condition respectively and calculate the average of the probabilities (the negative log-likelihood) of each token in masked sites. The condition score is the average of all the masked token probabilities. We repeat the sampling and computation process 10 times to estimate the standard deviations in each condition. The results are shown in Table 2.

The results show that the models, both `ckip-bert` and `bert-base-chinese`, find predicting constructions harder than the ones in control baselines. Specifically, the negative log-likelihood (NLL, the lower the numbers mean the higher the probabilities) is higher in the masked constructions (8.51 and 8.09 for `ckip-bert` and `bert-base-chinese`, respectively) than the ones in shifted (7.35 and 6.94) and random controls (5.96 and 5.57). That is, consistent with our hypotheses, even though the pretrained model is not directly tuned to the constructions, the models do pick up the usage pattern in constructions. The difference between the masked conditions and the controls also demonstrates that the difficulties of predicting constructions cannot be accounted for by the consecutive patterns or the number of masked tokens alone. More interestingly, the difference between different masking sites shows that variable sites are the most difficult prediction targets. The score of variable sites (9.95 and 8.90) is both higher than the ones in constants (8.09 and 7.36), and the full constructions (8.51 and 8.09). The same patterns are also observed when the second order difference is compared: the difference between control baselines and the target mask condition in variable sites are the largest among the ones in constructions and constants. The general patterns suggest that the pretrained model do capture the usage of constructions, and the variable sites, or the open slots, in the constructions are the least predictable for the models.

ckip-bert			
	Masked	Shifted	Random
Cnstr.	8.51 (0.04)	7.35 (0.06)	5.96 (0.05)
Cst.	8.07 (0.05)	7.83 (0.05)	7.66 (0.06)
Var.	9.95 (0.04)	6.63 (0.08)	6.33 (0.07)
bert-base-chinese			
	Masked	Shifted	Random
Cnstr.	8.09 (0.04)	6.94 (0.05)	5.57 (0.07)
Cst.	7.36 (0.06)	7.16 (0.05)	7.00 (0.07)
Var.	8.90 (0.05)	6.15 (0.09)	5.88 (0.08)

Table 1: Model predictions' negative log-likelihood in different conditions. The lower the numbers the *easier* the model to predict the masked tokens. *Cnstr*, Construction. Number in parentheses are standard deviations. *Cst.*, Constant sites. *Var.* Variable sites.

However, the word distributions of the variable sites are one of the most valuable information in the constructions. On the one hand, the word selection in the variable site determines the concrete meaning realization conveyed by the constructions. On the other, the word selections themselves reflect how the sentential context interact with the structural constrained exerted by the constructions. Therefore, given the results of the experiments, we specifically fine-tuned the masked language model on the variable sites in constructions, and build CxLM.

5. Training and Evaluating CxLM

To better capture the constraints on the constructions' variable slots, we fine-tuned the masked language model to be construction aware. The training scheme follows a standard procedure of masked language modeling: tokens in the sequence are randomly masked out, and the model is to predict the

masked token. However, here, we change the masking scheme to only mask those tokens in variable slots. That is, the model is forced to learn to predict words in variable sites.

We use the dataset compiled in Section 3 as training data. There are 11,642 constructions in the dataset, and we used random 90-10 splits for training and testing data, with proportional stratified random sample on each construction types. The training data comprise 10,477 sequences, which are composed of 461,745 tokens under a character-based tokenization, among which 92% are Chinese characters. We masked every occurrence of variable sites in the constructions, which are 23,417 (5%) masked tokens. The model training is based on CKIP-pretrained BERT model, and a masked-language-model readout head. The parameters are updated by AdamW, the base learning rate is 1e-4, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\lambda = 0.1$, and the batch size is 16. The learning rate is further adjusted by a linear decreasing scheduler with a warmup period of 50 iterations throughout the training process.

We use the top-k prediction accuracies to evaluate the model performance. In addition, we include three models for comparison: the pre-trained CKIP base-bert, the unigram model, and a random guessing baseline. The unigram model serves as a *the-most-frequent baseline*. It first calculates the frequency distributions of all the masked characters (there are 891 unique characters that occurred in the masked sites), and it invariably predicts the most frequent ten characters. For example, the top 3 characters in the unigram model are 死 (*sǐ* 'pass away'), 想 (*xiǎng* 'think'), 看 (*kàn* 'see'). Conversely, the random guessing baseline randomly selects a set of characters as its predictions.

Table 2 shows that the fine-tuned CxLM achieved the highest accuracies among the four models, no matter top-1, top-5, or top-10 accuracies. It shows that the fine-tuned model becomes better at predicting the constructions' variable slots. However, an interesting comparison is the one between the bert-base and unigram models. It is worth noting that the bert-base model has never *seen* the training data, while the unigram model is directly trained on the dataset (on which it computes the frequency distribution). Despite their difference in the experiences, the two models are very similar in this task-specific performance. It implies that the bert-base model, although not specifically tuned on constructions, nevertheless captures, at least functionally similar to the unigram model, the possible candidates on the variable sites. This observation is consistent with the experiment results in Section 4, where we found the pretrained model already has an idea of construction usage. Therefore, after fine-tuning, the model could better capture the candidates in variable slots with higher accuracies.

	Top1	Top5	Top10
CxLM	30.05	50.70	59.98
Bert-Base	6.11	13.26	18.28
Unigram	4.74	15.67	21.66
Random	0.04	0.05	1.12

Table 2: Top-k accuracies of different models

The prediction accuracy is an important metric, but not the whole picture as to characterize the constraints on variable slots. Similar to the challenge in generative adversarial models, it is not straightforward to evaluate a model that aims to capture underlying representations (Salimans et al., 2016). One of the most useful evaluations is to inspect samples the model generates. Table 3 shows three examples, each of which contains the model input and the model predictions. Each example exemplifies interesting model behaviors which help us delineate the underlying learned distributions.

The first example is when the model is well-tuned and *has a clear idea* what the sentence is about. It can be seen by the model predictions that the model unequivocally predicts the correct words (i.e. 拌 *bàn* 'mix'). It is also interesting to note that the other predicted candidates, while not being "correct", are also plausible in the given sentence. There are two types of predictions shown in Table 3. The *separated* predictions are computed with the respective logits from each of the masked sites, while the *merged* predictions are the pooled predictions by summing the logits of all the masked sites. The merged predictions essentially assume that both masked slots should be the same characters, which is usually the case in this dataset.

The second example shows the case when the model misses one of the slots. In this example, the model only correctly predicts the second slot but missed the first one. However, if we constrain the two slots from being the same character (the merged predictions), the prediction is nevertheless accurate (i.e. 收 *shōu* 'take'). It is noteworthy that while the second and third candidates in merged predictions (i.e. 打 *dǎ* 'do, get'; 吃 *chī* 'eat') are not interchangeable with the correct one, they are still structurally and semantically acceptable in the sentence.

The third example is when the model is slightly *confused*. The model completely missed the correct ones, at least in the top-3 predictions shown here. However, even in this scenario, the model still provides plausible samples that are acceptable in the construction. One highlight in the listed samples is that all of them are structurally correct candidates given the samples, and most of them are context-appropriate.

	Samples
	Input: 裡面的馬鈴薯也很入味泡麵這樣[MASK] 一 [MASK] 味道很棒耶 ([MASK]= 拌) the potatoes inside are also delicious; the instant noodle tastes good when (properly) *mixed*.
1	CxLM (sep): **拌 (mix)**, 煮 (cook), 吃 (eat); **拌 (mix)**, 吃 (eat), 煮 (cook) CxLM (mrg): **拌 (mix)**, 吃 (eat), 煮 (cook)
	Input: 不 [MASK] 白不 [MASK] 囉大方一點收禮無妨 [MASK]([MASK]= 收) Just *take* it; it's alright to receive a gift.
2	CxLM (sep): 打 (get), 踩 (stamp), 炸 (fry); **收 (take)**, 送 (give), 吃 (eat) CxLM (mrg): **收 (take)**, 打 (get), 吃 (eat)
	Input: 運動強度沒有太高圖個 [MASK] 一 [MASK] ([MASK]= 動) The exercise intensity is not very high, (I) just want to *work* out a bit.
3	CxLM (sep): 升 (rise), 緩 (relax), 加 (add); 緩 (relax), 忍 (endure), 升 (rise) CxLM (mrg): 升 (rise), 緩 (relax), 忍 (endure)

Table 3: Three samples generated by CxLM. The Input are masked at the variable sites, and the full constructions are indicated with underlines. The top three words generated by CxLM are listed, where the bold texts are the correct words. *sep*, separated, the respective samples at the two variable sites. *mrg*, merged, the samples from the joint distributions of the variable sites.

Other examples illustrate how CxLM captures higher-order dependencies on the constructions' realization and its context. For example, when the input is a standalone construction, [MASK] 一 [MASK], the model predicts 想 (*xiǎng* 'think'). On the contrary, when the input has more context, such as 買本書 [MASK] 一 [MASK], the model predicts 看 (*kàn* 'look'). Both of these predictions are highly plausible and contextually accurate. Other examples include 躺在床上半 [MASK] 半 [MASK]'lying on the bed half [MASK]half [MASK]', CxLM predicts 睡 (*shuì* 'asleep') and 醒 (*xǐng* 'awake') from the input . On the contrary, the predictions become 肥 (*féi* 'fat') and 瘦 (*shòu* 'lean') when the input is 這塊肉半 [MASK] 半 [MASK]'The meat is half [MASK]half [MASK]'. These examples highlight the variable slots, although being a structural or semantic element of a construction, are still influenced by the overall sentential context. This dependency can only be inferred by collocation statistics, but is directly observed in CxLM.

Finally, it will be informative to visualize the probability landscape of CxLM. Although the categorical nature of discrete tokens prevents us from directly visualizing the joint distributions used in sample generations, we can still inspect a more confined view of such distribution. Figure 2 shows the joint distribution of CxLM predicting the first example of Table 3, along with the ones of two other models, base-bert and unigram models, for comparison. All three panels plot the same function. For CxLM and base-bert, the function is defined as follows:

$$p(x_1, x_2|\mathbf{t}) = \text{softmax}(f(x_1) + f(x_2))$$

where $f(.)$ is the logit from CxLM or base-bert, x_1 and x_2 stand for the tokens in the masked site 1 and site 2 respectively; and $\mathbf{t}$ denotes the rest of the unmasked tokens. For the unigram model, we assume independence of two masked tokens; therefore the function is defined as the product of two probabilities.

$$p_{\text{uni}}(x_1, x_2|\mathbf{t}) = p_{\text{uni}}(x_1) \cdot p_{\text{uni}}(x_2)$$

Both x_1 and x_2 are distributed over the same set of tokens, which include the candidates from the union of three models' top 10 predictions. These candidates are arbitrarily ordered in the x- and y-axes (the correct token is positioned at the center for better visualization). The plots are color-coded with the respective model's log-probabilities. The white blocks on the figures denote the locations of the correct tokens.

Figure 2 shows a clear difference between unigram models and the ones with language models. The difference is expected as the unigram model is context-free; therefore the possibilities of all the candidates only reflect the frequency distributions in the variable slots. On the contrary, two context-sensitive language models, CxLM and base-bert, have more focused frequency distributions. It can also be observed that two distributions have different *band* on the plots, exemplifying their different underlying distributions. These figures complement the accuracies and qualitative sample observations. Together, these results demonstrate that CxLM does learn and tune conditional probability distributions underlying the constructions' variable slots.

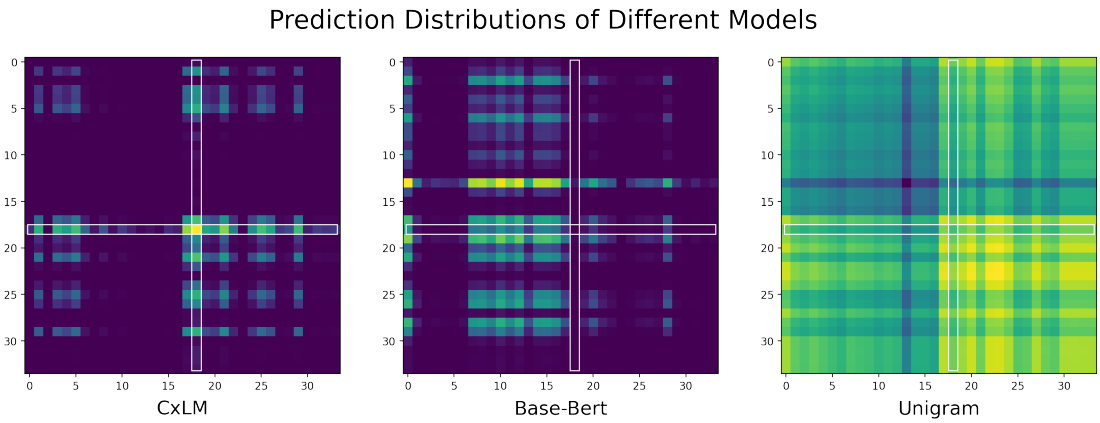

Figure 2: The prediction distribution of different models.

6. Conclusion

Constructions are conventionalized form-meaning pairs with varying degree of schematicities. Those schematic slots raise interesting questions, as they are constrained by the constructions but still influenced by the higher sentential contexts. We aim to characterize the interaction between the opening slots, constructions, and the embedded context as a conditional probability distribution. This distribution is inevitably complex; therefore we build CxLM, a masked language model specifically tuned for constructions' variable slots.

We first conducted an experiment to examine to what extent a pretrained language model would be aware of constructions. The results show that while the model is aware of the construction, it is confused at the variable slots. Therefore, we fine-tuned the model to learn the probability distributions underlying the variable site. The quantitative evaluation shows CxLM achieves higher accuracies in predicting masked words. More importantly, CxLM also generates semantically and structurally plausible samples at the variable slots. Future works include expanding the coverage of CxLM's constructions types and using CxLM to build a higher precision model of construction extraction. The CxLM model and its construction data are released as public resources. We hope CxLM would provide another tool and perspective on studying the constraints of opening slots in schematic constructions.

Acknowledgements

This work was supported by Ministry of Science and Technology (MOST), Taiwan, Grant Number MOST 110-2634-F-001-011.

7. Bibliographical References

Alvarez-Melis, D., Garg, V., and Kalai, A. T. (2020). When not to use an adversarial approach to generative modeling.

Boas, H. C. (2013). Cognitive construction grammar. In *The Oxford handbook of construction grammar*. Oxford University Press.

Bybee, J. (2006). From usage to grammar: the mind's response to repetition. *Language*, 82(4):711–733.

Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. (2020). ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Croft, W. (2001). *Radical construction grammar: Syntactic theory in typological perspective*. Oxford University Press on Demand.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dunn, J. (2017). Computational learning of construction grammars. *Language and cognition*, 9(2):254–292. Publisher: Cambridge University Press.

Fillmore, C. J. (1988). The mechanisms of "construction grammar". In *Annual Meeting of the Berkeley Linguistics Society*, volume 14, pages 35–55.

Goldberg, A. E. (2006). *Constructions at work: The nature of generalization in language*. Oxford University Press on Demand.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Z. Ghahramani, et al., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Gries, S. T. and Stefanowitsch, A. (2004). Extending collostructional analysis: A corpus-based perspective onalternations'. *International journal of corpus linguistics*, 9(1):97–129.

Gui, J., Sun, Z., Wen, Y., Tao, D., and Ye, J. (2021). A review on generative adversarial net-

works: Algorithms, theory, and applications. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1.

Guo, J., Lu, S., Cai, H., Zhang, W., Yu, Y., and Wang, J. (2018). Long text generation via adversarial training with leaked information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Hanna, P., Stewart, D., Smith, F. J., et al. (2006). Reduced n-gram models for english and chinese corpora. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 309–315.

Hoffmann, T. and Trousdale, G. (2013). *Construction Grammar*. Oxford University Press, December.

Kaschak, M. P. and Glenberg, A. M. (2000). Constructing meaning: The role of affordances and grammatical constructions in sentence comprehension. *Journal of memory and language*, 43(3):508–529. Publisher: Elsevier.

Liu, T.-J. (2014). Ptt corpus: Construction and applications. Master's thesis, National Taiwan University.

Madabushi, H. T., Romain, L., Divjak, D., and Milin, P. (2020). CxGBERT: BERT meets Construction Grammar. *arXiv preprint arXiv:2011.04134*.

Manning, C. D., Clark, K., Hewitt, J., Khandelwal, U., and Levy, O. (2020). Emergent linguistic structure in artificial neural networks trained by self-supervision. *Proceedings of the National Academy of Sciences*, 117(48):30046–30054.

McMahon, J. and Smith, F. J. (1998). A review of statistical language processing techniques. *Artificial Intelligence Review*, 12(5):347–391.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training gans. *Advances in neural information processing systems*, 29:2234–2242.

Schmid, H.-J. (2010). Does frequency in text instantiate entrenchment in the cognitive system? In *Quantitative methods in cognitive semantics: Corpus-driven approaches*, pages 101–134. De Gruyter Mouton.

Stefanowitsch, A. and Gries, S. T. (2003). Collostructions: Investigating the interaction of words and constructions. *International journal of corpus linguistics*, 8(2):209–243.

Stefanowitsch, A. and Gries, S. T. (2005). Covarying collexemes.

Stefanowitsch, A. (2013). *Collostructional Analysis*. Oxford University Press, December.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October. Association for Computational Linguistics.

Zhan, W. (2017). Some Key Issues on Building A Knowledge Database of Chinese Constructions. *Journal of Chinese Information Processing*, 31(1):230–238.