`Vec2Gloss`: definition modeling leveraging contextualized vectors with Wordnet gloss

Yu-Hsiang Tseng, Mao-Chang Ku, Wei-Ling Chen, Yu-Lin Chang, Shu-Kai Hsieh
Graduate Institute of Linguistics, National Taiwan University
seantyh@gmail.com, d08142002@ntu.edu.tw, d10142007@ntu.edu.tw,
cylchang37@gmail.com, shukaihsieh@ntu.edu.tw

Abstract

Contextualized embeddings have proven to be powerful tools in various NLP tasks. However, their interpretability and how they encode lexical semantics remain challenging issues. In this paper, we tackle this problem by using definition modeling, a technique that aims to generate human-readable definitions for words, as a means to evaluate and understand high-dimensional semantic vectors. We introduce the `Vec2Gloss` model, which generates glosses from the contextualized embeddings of target words. The systematic gloss patterns provided by Chinese Wordnet enable us to examine the mechanism behind the model's gloss generation. To delve deeper into this mechanism, we devise two dependency indices to measure the semantic and contextual dependencies of the generated glosses. These indices allow us to analyze the generated texts at both the gloss and token levels. Our results demonstrate that the proposed `Vec2Gloss` model enhances our understanding of lexical semantics in contextualized embeddings.

1 Introduction

The rapid advancement of distributed semantic models has led to remarkable achievements, with machine performance in some language-related benchmarks either matching or even surpassing that of human non-experts (Maru et al., 2022; Chowdhery et al., 2022). These successes are often attributed to the complex pretrained language models (Peters et al., 2018; Devlin et al., 2019; Radford et al., 2019; Raffel et al., 2020), which are commonly referred to as sentence encodings in the literature (Pavlick, 2022). In contrast to traditional distributional semantic models (Lenci, 2018; Boleda, 2020), sentence encodings adopt a top-down training approach, prioritizing sentence processing as the primary goal. As a result, word-level semantics naturally emerge as inherent properties (Pavlick, 2022).

Studies have demonstrated that sentence encodings do capture lexical semantics. Although the contextualized embeddings of each token are highly intertwined with both semantics and syntax (Yenicelik et al., 2020), one can still access a wealth of information on word-level lexical semantics by averaging the vectors across contexts and model layers. When appropriately configured, these emerging lexical representations outperform explicitly trained static word vector models (Vulić et al., 2020). It can be argued that these contextualized embeddings are possibly *sense-aware*. This means that one could build sense embeddings for word sense disambiguation tasks, where the goal is to find the nearest neighbor of the target word in the sense embedding space (Scarlini et al., 2020b). These studies have demonstrated that while sentence encodings are not explicitly trained for word-level semantics, they do capture the nuances of word usage to a certain degree.

Indeed, the interpretability of these models and their ability to represent lexical semantics remain significant challenges. Various evaluation methods have been proposed to address this issue. One unique approach is definition modeling, which aims to generate a definition for a given word. This approach is argued to offer a more transparent and direct evaluation of the word's semantic representation (Noraset et al., 2017; Gardner et al., 2022). In the context of distributional semantic models, definition modeling can be understood as first encoding the semantic representations into one or multiple vectors, based on which a language model generates the corresponding definitions. Previous studies have explored various model architectures with fruitful results. The key advantage of definition modeling lies in the ability to analyze the embeddings in a natural language form, i.e., the definitions. Instead of indirectly examining a high-dimensional vector through word analogies and similarities, we can now probe into (distributional) lexical semantics transparently using human language.

The subsequent challenge lies in systematically examining the generated definitions, especially when these are produced by a model that may or may not fully capture the intricacies of definitional language. In this paper, we address this challenge by investigating the model-generated definitions using a relatively standardized gloss language to train a definition generation model. Our gloss dataset comes from the Chinese Wordnet (CWN) (Huang et al., 2010)[1], where lexical senses of each word are differentiated and described with a relatively constrained set of glossing rules.

We formulate the definition modeling as a vector-to-text task. Inspired by the sense embedding and the sequence-to-sequence architecture of definition modeling (Scarlini et al., 2020b; Mickus et al., 2019), we further encode the context-sensitive word sense into an encoding vector, from which the model learns to decode the gloss sentences. To evaluate the generated definitions, we use human ratings and propose two indices to examine the contextual and semantic dependencies closely. With these two indices, we conduct gloss and token-level analyses of the generated definitions and show that they fairly reflect aspects of lexical semantics.

The overarching goal of this work is to explore the possibility of gloss generation using only one contextualized vector. We propose that a generation model can be trained on relatively constrained gloss patterns extracted from the fine-grained CWN glosses. To evaluate the performance of the model, we conduct human rating experiments, accompanied by a comprehensive analysis of the generated gloss patterns.[2]

2 Related Work

2.1 Patterns in gloss languages

Dictionary definitions, or word glosses, are often referred to as "language about language", or "metalanguage" (Sinclair, 1991; Johnson and Johnson, 1998; Hanks, 2013). One prominent theory in metalanguage is the Natural Semantic Metalanguage (NSM) (Wierzbicka, 1972; Durst, 2004), which posits that universal semantic primitives can account for the meanings of words. Additionally, Barque and Polguère (2004) have classified sense descriptions into "word paraphrases" and "word interpretations" based on their formal nature. (cf. Pottier, 1974 and Pustejovsky, 1998)

While previous studies on metalanguage often adopt a logical or formal semantic approach, the Corpus Pattern Analysis (CPA) proposed by Hanks (2004) offers a new direction for analyzing word glosses from the perspective of syntagmatic patterns. According to Firth (1957), the meanings of a word are influenced by the context formed by surrounding terms. In a similar vein, Hanks (2004, 2013) analyze concordance lines from corpora to generalize typical patterns of certain words. These groups of words constitute a *lexical set*, which is united by a common *semantic type*.

While not precisely following the methodology in CPA, the gloss language in CWN attempts to incorporate lexical sets and semantic types into its gloss. For example, one of the gloss patterns[3] for adverbial senses is shown below. Similar glossing guidelines are established across different lexical categories.

Word	ˆ *very*
Sense	h超N平常„程度 *describing exceeding normal extent*
Gloss Pattern	h ... „程度 *describing ... extent*

Therefore, the glosses in CWN provide a fertile ground to systematically model its gloss language. However, the complexity of the gloss patterns makes them challenging for logical or formal analyses. Therefore, utilizing deep learning for definition modeling is beneficial in exploring the hidden information within these gloss patterns.

2.2 Definition Modeling

Definition modeling aims to generate a definition for a given target word (Gardner et al., 2022; Noraset et al., 2017). Noraset et al. (2017) utilized hypernym embeddings to generate dictionary definitions. Gadetsky et al. (2018) incorporated context words' embeddings and an attention-based skip-gram model to improve definition modeling for polysemous words. More recent research in definition modeling has incorporated various architectures to better capture semantic vectors and improve definition generation. Recurrent neural networks, variational generative models, and pretrained language

[1]The data are accessible at https://lopentu.github.io/CwnWeb/

[2]The code and the rating material are available at the anonymized repository: https://github.com/seantyh/vec4gloss

[3]For more examples, please see the manual of CWN (in Chinese), https://lope.linguistics.ntu.edu.tw/cwn/documentation

models have been used to obtain semantic representations of the target word (Ishiwatari et al., 2019; Reid et al., 2020; Zhang et al., 2020). Additionally, some studies have leveraged lexical resources like HowNet and WordNet to construct latent vectors or use them as guiding signals (Dong and Dong, 2006; Luo et al., 2018a,b; Blevins and Zettlemoyer, 2020; Li et al., 2020; Scarlini et al., 2020a; Yang et al., 2020).

Contextualized embeddings have indeed demonstrated their ability to capture important aspects of lexical semantics (Peters et al., 2018; Loureiro and Jorge, 2019). For instance, Scarlini et al. (2020b) showed that a simple 1-nearest-neighbor algorithm using these sense vectors achieves comparable performance with other more complex supervised model architectures in the word sense disambiguation task. This nding indicates that the contextualized embeddings carry signi cant semantic information that can be effectively utilized not only for disambiguating polysemous words but also for improving de nition modeling tasks.

The proposed Vec2Gloss model is designed to tackle the de nition modeling task using a sequence-to-sequence approach with an encoder-decoder architecture (cf. Mickus et al., 2019; Bevilacqua et al., 2020). However, a key difference is that the objective of Vec2Gloss is to decode the de nition from the encoded vectors while simultaneously ne-tuning the encoder to optimize the semantic vector. To achieve this, we utilize the pretrained mT5 (Xue et al., 2021) text-to-text model architecture but introduce a tight bottleneck between the encoder and decoder. This design decision restricts the decoder's access to the full context of the input sentence, making it unable to rely on collocations directly for gloss generation. Therefore, the decoder must learn the gloss's regularities from the encoded vectors to generate accurate and contextually appropriate de nitions.

3 Vec2Gloss Model

The goal of the Vec2Gloss model is to generate a coherent gloss based on the semantic vector of a word, which is derived from CWN. This task is closely related to, yet distinct from, common NLP tasks. Unlike typical NLP tasks that involve obtaining an encoder representation and mapping a lexical word or sense into a vector, the primary objective of this model is to optimize the vector speci cally for decoding the gloss.

Figure 1: The model architecture of Vec2Gloss. The model follows a general encoder-decoder architecture but introduces a bottleneck between the encoder and decoder. The decoder is restricted to only seeing the target word's semantic vector (v_{sem}), rather than having access to the complete encoder states.

On the other hand, this task goes beyond a standard autoregressive approach, as the generated gloss must be conditioned on a vector rather than prompts or input sequences. While an encoder-decoder architecture might be the most suitable option, the standard task involves mapping between input and output text. As a result, it is unclear whether the model learns to decode the gloss directly from the semantic vector or simply translates it from the input text.

To leverage the encoder-decoder architecture while ensuring the model relies on the semantic vector to decode the gloss, we implement a tight bottleneck between the encoder and decoder (see Figure 1). The input to the model is a sentence containing a target word. The encoder processes the input sentence, resulting in a set of encoder states. We then apply a prede ned target mask to these encoder states, selecting only the vectors corresponding to the target word. These selected vectors are then averaged to create a single vector, which is fed into the decoder responsible for generating the gloss sequence.

Notably, unlike the standard architecture that incorporates cross attention between encoder states, the decoder in our model has access to only one encoder vector. As a result, the decoder cannot rely on the complete input sentences and is compelled to focus solely on the target word's semantic vector (v_{sem}). In this way, the encoder is encouraged to compress as much relevant information as possible into the target word's semantic vector, while the decoder must learn the regularities of gloss generation independently, without relying on potential collocation cues between word context and gloss. In summary, the model learns both the target word's semantic vector through the encoder and the gloss sequence through the decoder.

To enhance the model's ability to capture the patterns of gloss sequences, we propose a denoising stage *before* training for the vector-to-gloss task. In this denoising stage, a standard encoder-decoder architecture is employed, and the model is trained to reconstruct the corrupted spans in the glosses. The objective of this stage is to pretrain the model to better understand the regularities and structures of gloss language.

Following the denoising stage, we proceed to the ne-tuning stage, where we introduce the bottleneck between the encoder and decoder components. During ne-tuning, the model receives a sentence containing a target word, along with a target mask. It is tasked with learning the target word's semantic vector using the encoder and then generating the entire gloss sentence exclusively from this semantic vector using the decoder.

3.1 Denoising stage

To improve the model's ability to capture the underlying patterns in gloss language, we initiate the training process with a denoising objective. This approach has been used in previous studies (Lewis et al., 2020), and it involves preparing pairs of examples comprising corrupted spans as inputs and their corresponding dropped-out spans as outputs. The denoising objective has demonstrated its effectiveness in downstream tasks while also being computationally ef cient, as it reduces the length of decoding sequences (Raffel et al., 2020). An example of such a pair is provided below, with literal translations shown in italics:

Input	å ‡ W’ Ë <X>ú † „ o
	using text medium <X> -out information.
Target	<X>h T <Y>
	<X>express <Y>

The <X> and <Y> tokens are special sentinel tokens unique to each example. The spans used in the denoising objective are character-based and may not necessarily align with word boundaries. To introduce corruption, random locations within the spans are selected, and their lengths (measured in characters) are drawn from a Poisson distribution with a parameter $= 2$, ensuring that the length values are clipped between 1 and 4 (inclusive). If the input sequence is longer than 20 characters, an additional corrupted span is created using the same parameter settings. These examples are extracted from the word glosses in CWN, and 26,118 pairs are generated for the denoising objective.

In the denoising stage, we utilize the pretrained T5 encoder-decoder architecture (mt5-base) to train the denoising objective (Xue et al., 2021). During this stage, no bottleneck is applied between the encoder and decoder components. The model parameters are updated using the AdamW optimizer, with a learning rate of $1\ 10^{\ 4}$. The values of ${}_1$ and ${}_2$ in the optimizer are set to 0.9 and 0.999, respectively, and weight decay is con gured to 0.01.

To schedule the learning rate, a linear schedule is employed. The batch size used for training is set to 8. The model is trained for 3 epochs, and the training process takes approximately 30 minutes when executed on an A5000 GPU. The parameters obtained after training in this denoising stage serve as the starting point for the subsequent ne-tuning stage.

3.2 Fine-tuning stage

In the ne-tuning stage, the primary objective is to establish the relationships between the target words embedded in the sentences and their corresponding glosses in CWN. To achieve this, we maintain the standard T5 encoder-decoder transformer-based architecture while simultaneously introducing a tight bottleneck between the encoder and decoder components.

Speci cally, during ne-tuning, we select and average only the target word's encoder states from the input sentence. These encoder states might consist of more than one token, depending on the length and complexity of the target word. The resulting averaged encoder states serve as the semantic vector representation of the target word. The decoder is then trained to generate a complete gloss sentence based solely on this semantic vector.

The training data is sourced from the sense inventories of CWN. For each example sentence in a CWN sense, a training instance is created, consisting of a pair of input and target sequences. The input sequence is an example sentence where the target words are identi ed by enclosing them within a pair of angular brackets. On the other hand, the target sequences are composed of glosses associated with the corresponding senses, preceded by their respective part-of-speeches, and followed by a Chinese full-width period. In total, there are

76,969 instances in the training dataset, while the evaluation dataset comprises 8,553 pairs. A sample instance is provided below:

Input	y å S º †À ¼‹ ØØ <ž > She didn't<say > a word for some reason.
Target	VA N | r h ˜ (ž ó ³ o VA. Using vocal organs to convey a message with speech.

The model architecture closely follows the standard T5, allowing the trained weights from the denoising stage to directly apply to this model. During preprocessing, the target words' angular brackets are removed to create the target mask. This mask is crucial for selecting the relevant encoder states and generating the semantic vector, which serves as the input to the decoder. As a result, the decoder's cross-attention will always receive a single vector as input.

During training, the model is treated as a text-to-text task, where the objective is to generate the gloss sequence from the given input sentence. However, during inference, the encoder and decoder can operate independently. That is, the encoder can be used to obtain a semantic vector from a given sentence. This semantic vector can then be exibly transformed or manipulated before being passed to the decoder for gloss generation.

During the ne-tuning stage, the training procedure remains the same as the denoising stage, with the only difference being the number of epochs. In this stage, the model is trained for 10 epochs. The training process takes approximately 100 minutes when executed on an A5000 GPU.

3.3 Automatic evaluations

The automatic evaluation of de nition generation is presented in Table 1, which displays the BLEU and METEOR scores for each lexical category. The overall BLEU score is .41, and the overall METEOR score is .62. Notably, the noun category (N) has the lowest score, while the proper name category (Nb) has the highest score.

The higher score for proper names may be attributed to their speci c characteristics in CWN. Many proper names used in CWN are family names or foreign names, which tend to have shorter and more standardized de nitions. As a result, the model might nd it easier to capture these shorter glosses, leading to higher scores for the proper name category. The proper names category comprises 188 items.

For other categories, the interpretation of the automatic metrics is less straightforward. The scores only indicate the textual similarity between the generated and reference glosses. However, at a given score level, the generated gloss might still be unintelligible to human readers or merely a paraphrased version of the reference gloss. Therefore, to gain deeper insights and to assess the quality of the generated glosses, additional human evaluations are conducted, including a rating experiment, a gloss dependency analysis, and a token dependency analysis.

POS	N	BLEU	METEOR
N	2,801	.35(.01)	.59(.01)
V	4,376	.43(.01)	.63(.01)
D	432	.41(.02)	.62(.02)
O	530	.41(.02)	.63(.01)
Nb	414	.63(.02)	.74(.02)
All	8,553	.41(.01)	.62(.01)

Table 1: Automatic evaluation metrics on different lexical categories, which are nouns (N), verbs (V), adverbs (D), others (O), and proper names (Nb). Numbers in parentheses are standard errors.

4 Human Evaluations

4.1 Rating experiment

In the rating experiment, human raters are employed to assess the quality of the generated definitions, speci cally focusing on their semantic interpretability and syntactic well-formedness. The task is designed as a multiple-choice task, with each entry comprising a de nition in Chinese and a list of four-word options. Among the four options provided, only one is correct, representing a well-formed and semantically accurate de nition. A total of 140 entries are used in the evaluation, and these entries are derived from two sources: the Academia Sinica Balanced Corpus of Modern Chinese (ASBC) (Huang and Chen, 1998) and CWN. To ensure consistency, we only select words composed entirely of Chinese characters, exclude proper nouns, and lter out words with less than 10 occurrences in the corpus. Among the 140 test items, 40 are new words with their de nitions generated by ouVec2Gloss model, which we refer to asV2G:ex vivo For each test item, the correct answer (target word) is randomly and equally chosen from four different lexical categories: nouns, verbs, adverbs, and other word classes. The incorrect options for each question are also from the same word class, randomly selected from the collection of words derived from ASBC. Among the remaining 100 words, 20 use definitions from CWN, and 80 are generated by the model, which we refer to as V2G:in vivo. The word class composition is identical for the words from CWN, and the target words are randomly selected from the dataset and evenly distributed across the different word classes.

The experiment involved five native Chinese speakers majoring in linguistics, who were recruited as raters. They were assigned several tasks to assess the quality of the definitions generated by the Vec2Gloss model. In the first task, raters were presented with a set of four options, and they had to determine the most suitable term from those options based on the given definition. The second task focused on evaluating the semantic interpretability of a definition. Raters were asked to rate on a five-point acceptability judgment scale to what extent the definition could well explain the word that had been selected as the correct answer in the previous task. Similarly, in the third task, raters were asked to evaluate the syntactic well-formedness of a definition. They rated the well-formedness of the definition based on their internal grammar, again using a five-point acceptability judgment scale. The evaluation results are presented in Table 2. The Vec2Gloss model achieved promising performance compared to the original glosses in CWN.

Table 3 presents more detailed results for the evaluations of vector-generated glosses. The mean values for syntactic well-formedness are considerably high across all four lexical categories, both for V2G:in vivo and V2G:ex vivo. This indicates that the model-generated definitions are generally well-formed from a syntactic perspective. However, the results show that the semantic interpretability scores for V2G:ex vivo are lower than those for V2G:in vivo. This indicates that the model may face challenges in generating semantically interpretable definitions for new words. Despite the lower interpretability scores for V2G:ex vivo, the multiple-choice task still achieves over 80% correct rates in every category, similar to the results of V2G:in vivo. This suggests that even though the generated definitions for new words might be less semantically interpretable, they are still often correct and align with the correct word class. Additionally, the semantic scores of nouns are lower than those of other categories for both V2G:in vivo and V2G:ex vivo. This implies that the model may struggle more with generating semantically interpretable definitions for nouns compared to other word classes. To gain further insights and investigate possible reasons for the evaluation results, a gloss dependency analysis is conducted.

Source	Correctness	Mean_{sem}	Mean_{syn}
CWN	.95(.02)	4.47(.15)	4.82(.10)
V2G:in vivo	.88(.03)	3.51(.16)	4.58(.09)
V2G:ex vivo	.86(.04)	2.53(.22)	4.51(.12)

Table 2: Human evaluation results for definitions generated from different sources, with Mean_{sem} and Mean_{syn} representing the mean value of semantic interpretability and syntactic well-formedness, respectively.

4.2 Gloss dependency analysis

In the gloss dependency analysis, two indices are computed for each token in the generated glosses to represent their reliance on the preceding contexts and the semantic vector, respectively. First, the token likelihood under the full context and the original semantic vector (p_{full}) is compared to the likelihood when all of its preceding contexts are masked during decoding (p_{mask}). If a token is mostly determined by the context alone, masking the preceding contexts would significantly impact the token likelihood (p_{mask}). Hence, the negative likelihood ratio ($_{sem}$) will be larger, indicating a higher reliance on the context. Similarly, if a token is primarily driven by the semantic vector, replacing it while leaving the preceding context intact will lower the likelihood (p_{rep}), and the ratio ($_{ctx}$) will be larger, signifying a higher reliance on the semantic vector. To calculate these indices, the semantic vector (v_{sem}) obtained from the encoder is replaced with another word's semantic vector from the same lexical category. The indices are all calculated using the shifted reference glosses of each sense as the decoder inputs, ensuring a consistent comparison.

$$_{sem} = \log(p_{rep}/p_{full})$$
$$_{ctx} = \log(p_{mask}/p_{full})$$

The gloss-level indices are computed by averaging the token-level indices for each token, $_{sem}$ and $_{ctx}$, in the generated glosses. The results are shown in Figure 2. One notable observation is that the contextual dependency scores are comparable across the four different lexical categories, indicating that the preceding contexts play a similar role

POS	V2G:in vivo			V2G:ex vivo		
	Correctness	Mean_{sem}	Mean_{syn}	Correctness	Mean_{sem}	Mean_{syn}
N	.94 (.04)	3.18 (.35)	4.14 (.25)	.86 (.08)	1.92 (.40)	4.32 (.34)
V	.89 (.06)	3.63 (.34)	4.79 (.10)	.86 (.08)	2.74 (.46)	4.48 (.27)
D	.84 (.06)	3.75 (.31)	4.69 (.18)	.84 (.07)	2.76 (.43)	4.74 (.16)
O	.85 (.06)	3.47 (.32)	4.70 (.16)	.86 (.10)	2.70 (.45)	4.50 (.20)

Table 3: Human evaluation results for different lexical categories of definitions generated from *V2G:in vivo* and *V2G:ex vivo*. The semantic evaluation scores of nouns are lower than those of other categories for both sources.

in shaping the generated glosses across all categories. However, the semantic vector dependency indices show more significant differences. Specifically, the glosses of nouns have higher semantic dependency scores, followed by verbs, adverbs, and others. These results align with the human ratings, where the syntactic ratings are similar across all categories, but nouns receive significantly lower semantic rating scores. The higher semantic dependency scores for nouns may suggest that nouns are more likely to be used as nominal predicates, *categorizing* referents into a class with a holistic set of properties. On the other hand, adverbs, which have the lowest semantic dependency scores, primarily serve to *describe* things by adding a single property to the characterization of the referent (Baker and Croft, 2017; Bolinger, 1980). However, despite the relatively low semantic dependency scores, adverbs still carry semantic meaning, such as indicating manner, means, or instrumentality (Lyons, 1977; Lakoff, 1968). Therefore, further analysis is needed to understand why certain tokens are more pertinent to the semantic vector than others within the adverb category.

4.3 Token dependency analysis

The gloss dependency analysis is followed by a manual identification of chunks (referred to as semantic constituency in the gloss), where each chunk is annotated with its corresponding semantic type. Here, a *chunk* is defined as a *significant element* that functions as a unit carrying a semantic type (cf. Gerdes and Kahane, 2013).

Specifically, 244 adverbs are selected from CWN whose gloss contains the word ö“ ” shìjiàn `event', as these adverbs describe an explicit event structure in their glosses. Each gloss is manually segmented into length-variant chunks, and each chunk is manually tagged with its corresponding semantic type. Notably, the glosses of the first words are not annotated with semantic types since they typically follow a pattern based on their lexical category. For example, the glosses of adverbs often start with the word h biao `indicate', as seen in the example below (the gloss of ¥f# jiēlián `in a row'):

Gloss	h / ‹ö /(/Œð B µ /- / Œ/|
	To express the same event continuously happens during the later-mentioned period.
Annot.	--/Event/Preposition/Time/Preposition/Modifier/Action

In this dataset, there are a total of 905 chunks, where each chunk represents a significant element that functions as a semantic type-carrying unit. These chunks have been annotated with 19 unique semantic types. From this set of semantic types, we have selected six types (*event*, *action*, *modifier*, *preposition*, *negation*, *others*) that occur at least 25 times (representing 10% of the glosses count) for further analysis. These six selected semantic types account for 59% of all the annotated chunks in the dataset.

The token-level indices are computed as described in Section 4.2. It is important to note that the annotated glosses may contain multiple example sentences in CWN. Therefore, we extract and average the semantic vectors from each sentence to represent the target words. Subsequently, the context and semantic vector dependency scores are computed for each token, and these scores are then averaged based on their corresponding semantic types. The resulting averaged scores are presented in Figure 3.

The gloss-level analysis is further supported by distinctive dependency patterns observed across different semantic types. Specifically, the *action* types exhibit higher contextual dependency but relatively lower semantic dependency scores. This pattern aligns with the fact that *action* words typically serve as the main verbs in glosses. However, it's worth noting that the distribution of *action*

Figure 2: Dependency scores by each lexical category. The left panel shows the semantic dependency and the right one shows the context dependency scores. The letters along the vertical axis denote the lexical categories: nouns (N), verbs (V), adverbs (D), and others (O).

Figure 3: The dependency scores of six annotated semantic types. The error bars denote one standard error of semantic or context dependency scores. Numbers in parentheses are the member count of the type.

words is highly skewed, with a few common *action* words accounting for a signi cant portion of all *action* words. As a result, the higher contextual dependency scores may re ect the constrained word usage when generating glosses with *action* words.

On the other hand, the *preposition* and *negation* types show relatively higher semantic vector dependency scores. This observation may be attributed to the fact that *prepositions* are used to introduce related complements, and the decoder requires guidance from the semantic vectors to select the precise relations for the gloss. Similarly, *negation* words are challenging to capture solely through syntagmatic relations from the context (Aina et al., 2019; Ettinger, 2020), leading the decoder to rely more on additional cues from the semantic vectors.

Interestingly, words that are highly predictable given the adverb glosses, such as the *event* type, display lower scores in both contextual and semantic dependencies. This lower dependency indicates that the decoder has suf cient information from the context and semantic vectors to predict these words accurately, resulting in reduced reliance on both contextual and semantic cues.

5 Conclusion

This paper introduces the *Vec2Gloss* model, a gloss generation model that directly decodes glosses from semantic vectors. The study bene ts from the systematic gloss patterns provided by Chinese Wordnet. Human evaluation of the generated glosses through a multiple-choice task demonstrates that the *Vec2Gloss*-generated glosses are both grammatically correct and semantically accurate. Furthermore, we devised two indices to measure the semantic and syntactic dependencies of the generated glosses. The results show that the glosses for nouns are more semantically dependent, and the *prepositions* and the *negation* words in the glosses also need more semantic guidance. These results shed light on how the model captures lexical-semantic information through the de nition modeling task.

Overall, this paper contributes to advancing the eld of gloss generation. The systematic study of glosses and the incorporation of semantic vectors provide a foundation for further research in understanding the intricacies of lexical-semantic information and re ning gloss generation approaches.

References

Laura Aina, Raffaella Bernardi, and Raquel Fernández. 2019. Negated adjectives and antonyms in distributional semantics: not similar? Italian Journal of Computational Linguistics, 5(1):57–71.

Mark Baker and William Croft. 2017. Lexical categories: Legacy, lacuna, and opportunity for functionalists and formalists. Annual Review of Linguistics, 3(1):179–197.

Lucie Barque and Alain Polguère. 2004. A definitional metalanguage for explanatory combinatorial lexicography.

Michele Bevilacqua, Marco Maru, Roberto Navigli, et al. 2020. Generationary or "how we went beyond word sense inventories and learned to gloss". In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 7207–7221.

Terra Blevins and Luke Zettlemoyer. 2020. Moving down the long tail of word sense disambiguation with gloss informed bi-encoders. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1006–1017, Online. Association for Computational Linguistics.

Gemma Boleda. 2020. Distributional semantics and linguistic theory. Annual Review of Linguistics, 6(1):213–234.

Dwight Bolinger. 1980. Language - the Loaded Weapon. Pearson Education Limited.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. arXiv preprint arXiv:2204.02311.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding 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, pages 4171–4186, Minneapolis, Minnesota.

Zhendong Dong and Qiang Dong. 2006. HowNet and the Computation of Meaning.

Uwe Durst. 2004. The natural semantic metalanguage approach to linguistic meaning. Theoretical Linguistics, 29(3).

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. Transactions of the Association for Computational Linguistics, 8:34–48.

John Firth. 1957. A synopsis of linguistic theory, 1930-1955. Studies in Linguistic Analysis, pages 10–32.

Artyom Gadetsky, Ilya Yakubovskiy, and Dmitry Vetrov. 2018. Conditional generators of words definitions. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 266–271, Melbourne, Australia. Association for Computational Linguistics.

Noah Gardner, Hafiz Khan, and Chih-Cheng Hung. 2022. Definition modeling: literature review and dataset analysis. Applied Computing and Intelligence, 2(1):83–98.

Kim Gerdes and Sylvain Kahane. 2013. Defining dependencies (and constituents). Frontiers in Artificial Intelligence and Applications, 258:1–25.

Patrick Hanks. 2004. Corpus pattern analysis. In Proceedings of the 11th EURALEX International Congress, pages 87–97, Lorient, France. Université de Bretagne-Sud, Faculté des lettres et des sciences humaines.

Patrick Hanks. 2013. Lexical analysis: Norms and exploitations. MIT Press.

Chu-Ren Huang and Keh-jiann Chen. 1998. Academia sinica balanced corpus of modern chinese. Technical report, Academia Sinica.

Chu-Ren Huang, Shu-Kai Hsieh, Jia-Fei Hong, Yun-Zhu Chen, I-Li Su, Yong-Xiang Chen, and Shen-Wei Huang. 2010. Constructing chinese wordnet: Design principles and implementation. (in chinese). Zhong-Guo-Yu-Wen, 24:2:169–186.

Shonosuke Ishiwatari, Hiroaki Hayashi, Naoki Yoshinaga, Graham Neubig, Shoetsu Sato, Masashi Toyoda, and Masaru Kitsuregawa. 2019. Learning to describe unknown phrases with local and global contexts. 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 3467–3476, Minneapolis, Minnesota. Association for Computational Linguistics.

Keith Johnson and Helen Johnson. 1998. Encyclopedic Dictionary of Applied Linguistics: A Handbook for Language Teaching. Blackwell Publishers.

George Lakoff. 1968. Instrumental adverbs and the concept of deep structure. Foundations of language, pages 4–29.

Alessandro Lenci. 2018. Distributional models of word meaning. Annual Review of Linguistics, 4(1):151–171.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jiahuan Li, Yu Bao, Shujian Huang, Xinyu Dai, and Jiajun Chen. 2020. Explicit semantic decomposition for definition generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 708–717.

Daniel Loureiro and Alípio Jorge. 2019. Language modelling makes sense: Propagating representations through WordNet for full-coverage word sense disambiguation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5682–5691, Florence, Italy. Association for Computational Linguistics.

Fuli Luo, Tianyu Liu, Zexue He, Qiaolin Xia, Zhifang Sui, and Baobao Chang. 2018a. Leveraging gloss knowledge in neural word sense disambiguation by hierarchical co-attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1402–1411, Brussels, Belgium. Association for Computational Linguistics.

Fuli Luo, Tianyu Liu, Qiaolin Xia, Baobao Chang, and Zhifang Sui. 2018b. Incorporating glosses into neural word sense disambiguation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2473–2482, Melbourne, Australia. Association for Computational Linguistics.

John Lyons. 1977. *Semantics*. Cambridge University Press.

Marco Maru, Simone Conia, Michele Bevilacqua, and Roberto Navigli. 2022. Nibbling at the hard core of Word Sense Disambiguation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4724–4737, Dublin, Ireland. Association for Computational Linguistics.

Timothee Mickus, Denis Paperno, and Matthieu Constant. 2019. Mark my word: A sequence-to-sequence approach to definition modeling. In *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*, pages 1–11, Turku, Finland. Linköping University Electronic Press.

Thanapon Noraset, Chen Liang, Larry Birnbaum, and Doug Downey. 2017. Definition modeling: Learning to define word embeddings in natural language. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Ellie Pavlick. 2022. Semantic structure in deep learning. *Annual Review of Linguistics*, 8(1):447–471.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Bernard Louis Pottier. 1974. *Linguistique generale: Theorie et description*. Klincksieck.

James Pustejovsky. 1998. *The Generative Lexicon*. MIT Press.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Machel Reid, Edison Marrese-Taylor, and Yutaka Matsuo. 2020. VCDM: Leveraging Variational bi-encoding and Deep contextualized Word Representations for Improved Definition Modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6331–6344, Online. Association for Computational Linguistics.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2020a. Sensembert: Context-enhanced sense embeddings for multilingual word sense disambiguation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8758–8765.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2020b. With more contexts comes better performance: Contextualized sense embeddings for all-round word sense disambiguation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3528–3539.

John Sinclair. 1991. *Corpus, Concordance, Collocation*. Oxford University Press.

Ivan Vulić, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020. Probing pretrained language models for lexical semantics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7222–7240, Online. Association for Computational Linguistics.

Anna Wierzbicka. 1972. *Semantic Primitives*. Athenaum-Verlag.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Liner Yang, Cunliang Kong, Yun Chen, Yang Liu, Qinan Fan, and Erhong Yang. 2020. Incorporating sememes into chinese definition modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:1669–1677.

David Yenicelik, Florian Schmidt, and Yannic Kilcher. 2020. How does BERT capture semantics? a closer look at polysemous words. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 156–162, Online. Association for Computational Linguistics.

Haitong Zhang, Yongping Du, Jiaxin Sun, and Qingxiao Li. 2020. Improving interpretability of word embeddings by generating definition and usage. *Expert Systems with Applications*, 160:113633.

A Appendix

Table 4 illustrates some examples of the model-generated glosses. Figure 4 and Figure 5 shows the statistics of semantic type annotations in Section 4.3.