Character-level Chinese Backpack Language Models

Hao Sun
Stanford University
haosun@stanford.edu

John Hewitt
Stanford University
johnhew@cs.stanford.edu

Abstract

The Backpack is a Transformer alternative shown to improve interpretability in English language modeling by decomposing predictions into a weighted sum of token sense components. However, Backpacks' reliance on token-defined meaning raises questions as to their potential for languages other than English, a language for which subword tokenization provides a reasonable approximation for lexical items. In this work, we train, evaluate, interpret, and control Backpack language models in character-tokenized Chinese, in which words are often composed of many characters. We find that our (134M parameter) Chinese Backpack language model performs comparably to a (104M parameter) Transformer, and learns rich character-level meanings that log-additively compose to form word meanings. In SimLex-style lexical semantic evaluations, simple averages of Backpack character senses outperform input embeddings from a Transformer. We find that complex multi-character meanings are often formed by using the same per-character sense weights consistently across context. Exploring interpretability-through control, we show that we can localize a source of gender bias in our Backpacks to specific character senses and intervene to reduce the bias.

1 Introduction

Language modeling is a crucial task in natural language processing, where the goal is to compute the probability of the next word in a sequence given the preceding words. Recently, large language models based on the Transformer architecture (Vaswani et al., 2017) have achieved remarkable success in various NLP applications, including text generation (Radford et al., 2018b; Brown et al., 2020; Wang and Komatsuzaki, 2021), machine translation (Bawden et al., 2019; Lewis et al., 2019), and question-answering (Miller et al., 2017; Karpukhin et al., 2020; Ram et al., 2021). However, Transformers are notoriously hard to interpret and control. Their non-linear contextualization functions imply that intervening on their internal activations can have unpredictable consequences.

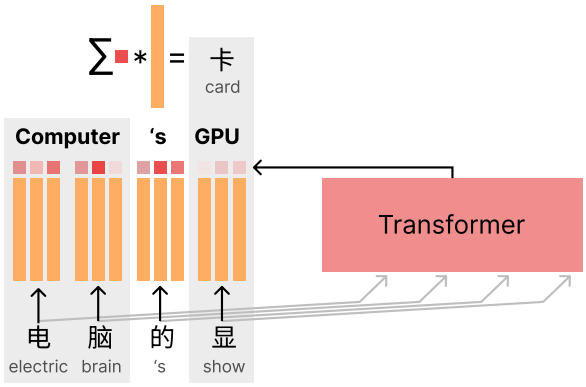

Figure 1: The general structure of the character-level Chinese Backpack Language Model. The next character is predicted by the weight sum of the senses of characters in the previous context. The sense vector of "显" (show) provides information for word composition, while the senses of "电" and "脑" (computer) provide semantic information through linear combination.

The recently proposed Backpack architecture (Hewitt et al., 2023) tackles the interpretability problem by decomposing its predictions as a sum of non-contextual vectors, which then provide an interface for interpretability. Intuitively, it combines the expressivity of Transformers with some of the interpretability and control benefits of log-linear models. It was shown to have similar language modeling capacity to Transformers on English, and performed comparably on perplexity and LAMBADA (Paperno et al., 2016) tests, at a tax of 1.4x more parameters.

The effectiveness of the Backpack architecture in languages with different morphological structures than English remains uncertain due to challenges in interpreting and controlling individual tokens without stable explicit semantics. In Chinese, most vocabulary consists of compound words with mul-

Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP, pages 106–119

tiple characters. However, these characters often have implicit meanings (Packard, 2011; Cui et al., 2018), making it challenging to infer the meaning of these words based solely on the individual meanings of their constituent characters. Additionally, some characters represent the pronunciation of foreign words and lack semantic associations, which requires characters to learn more complex semantic connections within limited sense vectors. The English-based Backpack model is trained on often complete words with more explicit meanings, making it uncertain whether Backpacks will perform well in character-level Chinese.

In this paper, we trained the first non-English (and first character-based language) Backpack language model and evaluate its performance and learned lexical semantics on character level.[1] We trained several Backpack and Transformer baseline models and evaluated them on perplexity and word prediction accuracy tasks. Our experiments show that our pretrained 134M Backpack Language Model with 16 sense vectors, which uses character-based tokenization, performs comparably to a 104M Transformer model.

To understand the Backpack's success, we first study how it composes word meaning from non-contextual token senses. We hypothesize word meaning is formed because tokens of a multi-character word receive similar weighting in the Backpack's sum across all contexts the word appears in. We find that indeed the proportion of these composed characters on each sense vector changes by no more than 20% in over 90% of cases. Moreover, we achieve better word representation under three Chinese corpora by simply averaging the sense vectors of composed characters compared to the character embeddings of the pretrained Transformer model. Additionally, we propose and evaluate character-level interventions to mitigate gender bias and control how word meaning is composed from character meaning , which demonstrate promising results for generating controllable text in character-based Chinese Backpack models. These experiments show that our Chinese Backpack Model learns the implicit semantics of characters, making it possible to control the emphasis or weakening of certain characteristics of a word during generation tasks.

[1]Our code, weights, and demos are available at https://github.com/SwordElucidator/nanoBackpackLM

2 Related Work

2.1 Word Representation with Deep Learning

Numerous word embedding techniques have been proposed in the early stages of natural language processing with deep learning, including Word2Vec (Mikolov et al., 2013) and GLoVe (Pennington et al., 2014), which represent words as vectors. Word2Vec learns word embeddings by predicting the probability of a word's occurrence given its context words or predicting the context words given a central word. Hewitt et al. (2023) showed that the Backpack is a generalization of Word2Vec. While these methods produce high-quality word representations that capture the semantic and syntactic relationships between words and have enabled rich interpetability studies as well as bias auditing (Senel et al., 2017; Subramanian et al., 2017; Swinger et al., 2018), they are not suited to language modeling tasks due to a lack of expressivity.

Subsequently, modern language models with the Transformer architecture (Vaswani et al., 2017) build contextualized word embeddings that are useful for modeling language in a variety of settings. However, as noted by Hewitt et al. (2023), these models' monolithic, non-linear processing of token sequences eschew any meaningful word-level semantics, so word-level interpretability has no direct connection to model behavior. Separately, interpreting contextual representations is difficult because each context maps arbitrarily to different representations, making it difficult for word embeddings to directly represent non-contextual semantic information and challenging to achieve predictable intervention across all contexts.

2.2 Language Modeling with Deep Learning

Language modeling is a fundamental task in natural language processing, involving computing the probability of the next word in a sequence given the previous words. Early neural approaches to language modeling used feed-forward networks (Bengio et al., 2000), various Recurrent Neural Networks (RNNs) (Elman, 1990; Sutskever et al., 2011) and attention mechanisms (Bahdanau et al., 2014). More recently, modern language models have adopted the Transformer architecture (Vaswani et al., 2017), with the GPT series (Radford et al., 2018a,b; Brown et al., 2020) by OpenAI achieving notable success in generating high-quality and coherent text. This success has led

to applications in various areas, such as story generation (Xu et al., 2020b; Chen et al., 2021) and chatbots (Lin et al., 2020; Roller et al., 2020; Shuster et al., 2022). However, as previously discussed, interpreting word embeddings in Transformer-based language models poses a challenge.

2.3 The Backpack Architecture

Hewitt et al. (2023) introduced the Backpack, a neural architecture which achieves high performance on contextualization and non-contextual word representations. This approach represents each word in a sequence as a linear combination of sense vectors, with weights computed by an expressive network such as the Transformer. (We'll review the Backpack in detail in Section 3.) The linearity of the contributions of sense vectors to predictions encourages the sense vectors to specialize and encode rich notions of word meaning during pretraining. Furthermore, the authors conducted experiments on sense vectors, demonstrating their potential for predictable control across all contexts. We reproduced and pretrained it on character-based Chinese language, demonstrating the Backpack model's potential for application to languages of this type.

2.4 Chinese Tokenization and Embeddings

One common approach for tokenization in Chinese involves sub-word tokenization methods, such as WordPiece (Schuster and Nakajima, 2012), byte pair encoding (Sennrich et al., 2016), and unigram language model segmentation (Kudo, 2018), which were adopted by recent Chinese Pretrained Language Models such as CPM (Zhang et al., 2020). Furthermore, Si et al. (2023) proposed Sub-Character Tokenization, which encodes each Chinese character into a sequence of phonetic or stroke symbols, and then utilizes a sub-word tokenization method to construct the vocabulary. In our research, to understand the performance of character-level sense vectors, we used single Chinese character tokenization method proven to be effective by Li et al. (2019) and utilized by Chinese GPT2 (Du, 2019) and MacBERT (Cui et al., 2021, 2019).

Various studies have explored embeddings at the word (Rumelhart et al., 1986; Bengio et al., 2000; Mnih and Hinton, 2008), phrase (Socher et al., 2010; Zhang et al., 2014; Yu and Dredze, 2015), sentence (Le and Mikolov, 2014; Socher et al., 2013; Kalchbrenner et al., 2014), and document (Srivastava et al., 2013; Le and Mikolov, 2014; Hermann and Blunsom, 2014) levels for representing knowledge and semantics. In the case of Chinese, character-level embeddings (Chen et al., 2015; Li et al., 2015) have also been investigated in relation to compounded word embeddings (Xu et al., 2016). We investigated on character embeddings and conducted two methods for representing compounded words using the contextualization weights learned during pretraining.

3 Approach

3.1 Backpack language model

Drawing directly from Hewitt et al. (2023), a Backpack language model is a probabilistic model

$$p(\mathbf{x}_i \mid \mathbf{x}_{<i}) = \text{softmax}(E^\top \mathbf{o}_{i-1}), \qquad (1)$$

where $\mathbf{x}_{1:i}$ is a sequence of elements from finite vocabulary $\mathcal{V}$, $E \in \mathbb{R}^{d \times |\mathcal{V}|}$, and $\mathbf{o}_{i-1}$ is a *Backpack representation* of $\mathbf{x}_{<i}$. In turn, a Backpack representation is constructed in two pieces:

Sense vectors. For each word in the vocabulary $\mathcal{V}$, a backpack learns k *sense vectors*, each like a specialized word2vec vector. We write the sense vectors for $\mathbf{x} \in \mathcal{V}$ as $\{C(\mathbf{x})_\ell\}_{\ell=1}^k$. When presented with a sequence $\mathbf{x}_{1:i}$, the Backpack constructs its sense vectors for the words in the sequence:

$$C(\mathbf{x}_1), \ldots, C(\mathbf{x}_i). \qquad (2)$$

Weighted sum. The Backpack representation $\mathbf{o}_i$ is just a weighted sum of the sense vectors of the sequence:

$$\mathbf{o}_i = \sum_{j=1}^{n} \sum_{\ell=1}^{k} \alpha_{\ell i j} C(\mathbf{x}_j)_\ell, \qquad (3)$$

where $\alpha_{\ell i j}$ is defined by a contextualization function $\alpha = A(\mathbf{x}_{1:n})$, and $A : \mathcal{V}^n \to \mathbb{R}^{k \times n \times n}$, and all $\alpha_{\ell i j} \geq 0$.

3.2 A note on Backpack token semantics

Intuitively, the contribution of each sense $C(\mathbf{x})\ell$ to any prediction is *independent of context*. We find it instructive to write out what this means for token-level semantics. The score $(E^\top \mathbf{o}_i)_\mathbf{w}$ of a word $\mathbf{w} \in \mathcal{V}$ in context $\mathbf{x}_{<i}$ is the unnormalized log-probability of that word. Because of linearity, we have:

$$E^\top \mathbf{o}_i = \sum_{j=1}^{n} \sum_{\ell=1}^{k} \alpha_{\ell i j} E^\top C(\mathbf{x}_j)_\ell, \qquad (4)$$

The contribution of a sense $C(\mathbf{x}_{j'})_\ell$ to that word's score is thus

$$\alpha_{\ell i j'} E^\top C(\mathbf{x}_{j'})_\ell \in \mathbb{R}^{|\mathcal{V}|}. \quad (5)$$

Because all α are non-negative, the *meaning* or use of a sense is simply its set of scores over the vocabulary $E^\top C(\mathbf{x}_{j'})_\ell$, which depends only on the word (not the context); only the *importance* of that meaning is determined by context. As such, visualizations of the "highest-scoring words" for a sense—as we provide in future sections—have a particularly transparent connection to model behavior.

3.3 Parameterizing Backpack Language Models

The sense function is parameterized $C(x) = \text{FF}(Ex)$ where FF: $\mathbb{R}^d \to \mathbb{R}^{d\times k}$ is a a feed-forward network, and contextualization weights $A(\mathbf{x}_{1:n}) = \alpha$ where

$$\alpha_\ell = \text{softmax}(\mathbf{h}_{1:n}^\top K^{(\ell)\top} Q^{(\ell)} \mathbf{h}_{1:n}) \quad (6)$$

for each predictive sense ℓ with matrices $K^{(\ell)}, Q^{(\ell)} \in \mathbb{R}^{d\times d/k}$ and $\mathbf{h}_{1:n}$ calculated by a Transformer (Vaswani et al., 2017) with autoregressive masking, i.e.

$$\mathbf{h}_{1:n} = \text{Transformer}(E\mathbf{x}_{1:n}) \quad (7)$$

We introduced a series of minor adjustments to the implementation details of the original backpack language model with the objective of enhancing training stability and facilitating a more comprehensive comparison between our model and the GPT model as discussed in Appendix A.

3.4 Baselines

We employed a GPT2-like Transformer model (Radford et al., 2018b) as a baseline, pretrained using the same datasets, hyperparameters, and random seed as our Backpack model. The Transformer and Backpack models have equal contextual parameters in the Transformer structure, whereas the Backpack model contains additional non-contextual parameters for the sense vectors. The Transformer and Backpack models share the same tokenizer and have an identical embedding size, as well as the same number of layers and heads for contextualization.

4 Experiment Training Backpack LMs

To compare the performance of our models against the baseline models in general language modeling evaluations, We first pretrained our 134M "Backpack-small" and 27M "Backpack-micro" Chinese Backpack language models and the baseline 104M "GPT2-small" and 18M "GPT2-micro" GPT2 models on large Chinese corpus. These sizes are set so the Transformer used in the Backpack's weight computation is the same size as the corresponding GPT2-like Transformer model.

4.1 Data

For pretraining, we employed three corpora: wiki2019zh (Xu, 2019a), news2016zh (Xu, 2019a), and webtext2019zh (Xu, 2019a), which are composed of 1.04 million Wikipedia entries, 2.5 million news articles, and 4.1 million Q&As, respectively, resulting in a total dataset size of 14.3G. This dataset was used to pretrain ALBERT Chinese (Xu, 2019b; Lan et al., 2020). To prepare the data, we set aside 1% of the data for the test set and 0.5% for the development set. The data was randomly partitioned into blocks of size 1,024 for each training step on each GPU.

4.2 Evaluation method

To evaluate the contextual performance of the Backpack and Transformer baseline models, we computed perplexities on the test set of our web corpus. We also used the Chinese WCPC dev set (Ge et al., 2021), an open-ended Chinese cloze task similar to LAMBADA (Paperno et al., 2016), which includes 4,827 test cases and is used for assessing top-1 word accuracy in word prediction with long-term context, to evaluate the models' ability to contextualize and predict words accurately. Specifically, each test case comprised a long sentence with at least 150 Chinese characters, with the last significant word being masked and having a length of 2 to 4 characters. The objective of the task was to predict the masked word, and we evaluated the performance of the models based on their top-1 and top-3 accuracy. As this task was originally tested on masked language models which can see the sentence's ending tokens, we designed a sampling method to evaluate our autoregressive models more fairly: we generated characters with beam search until the length of the output tokens equaled the length of the original sentence. We retained ten generations from the beam in every step, penalized

the outputs by adding the log probability of the original ending characters, and then selected the top generations.

4.3 Experimental details

The pretraining process for the Backpack-micro and the GPT2-micro baseline models involved training on 3× RTX3090 GPUs, using a batch size of 184,320 tokens for 500,000 gradient steps with cross-entropy loss, the AdamW (Loshchilov and Hutter, 2017) optimizer, 2,000 warmup steps, and linear decay on the learning rate starting from 6e-4 used by karpathy (2023). The model with the best performance on the dev set was retained by evaluating at intervals of 1000 steps. The Transformer structure comprised 6 layers, 6 heads, and an embedding size of 384, with dropout disabled for flash attention (Dao et al., 2022) in Torch 2.0. Three attempts were made to improve parameterization of the Backpack language model. Compared to the original paper, one layer was removed from the contextualization layer of the Transformer structure to match the size of the corresponding Transformer model. 134M Backpack-small and GPT2-small were pretrained on one A100 GPU with a batch size of 245,760 tokens for 500,000 gradient steps, using 16 sense vectors and a Transformer structure comprising 12 layers, 12 heads, and an embedding size of 768.

4.4 Results

During the experiment, it was observed that pretraining the Backpack model was more challenging to stabilize compared to the Transformer model, although the overall loss curve of the 16-sense vector Backpack LM was similar to the Transformer. Specifically, in the Backpack architecture, the lack of layer normalization in the representation $\mathbf{o}_i$'s weighted sum computation can cause dramatic changes in the sense vectors and lead to gradient explosion during pretraining when encountering low-quality batches.

In general, the Backpack models achieve similar perplexity scores compared to the GPT2-like Transformer model of similar scale and demonstrate significantly improved accuracy in WCPC (Ge et al., 2021) (Table 1).

WCPC is a challenging evaluation task as it requires the model to have long-distance contextualization ability and some world knowledge to determine the masked word. For the WCPC score, we found that our 134M Backpack-small tied with 223M ALBERT-xxlarge Chinese (Xu et al., 2020a) on top-1 accuracy and tied with the most performed MacBERT-large(Cui et al., 2021) in Chinese BERT family baselines (Devlin et al., 2018; Liu et al., 2019; Cui et al., 2020, 2021) on top-3 accuracy using the ending words penalizing strategy. Our strategy penalizes language models for generating predictions that do not end the sentence, improving evaluation alignment with masked language models.

5 Analysis of Lexical Structure

5.1 Sense Vectors

5.1.1 Visualizing Senses

Following the Backpack paper, we projected the sense vectors of characters onto the vocabulary, denoted as $E^\top C(\mathbf{x})_\ell \in \mathbb{R}^{|\mathcal{V}|}$, to illustrate the contribution of the sense vectors towards predictions. The outcomes are in Table 2 and you can find a detailed version in the appendix (see Table 7). As hypothesized, specific sense vectors automatically captured word composition rules during pretraining, whereas others captured semantic relatedness or associations.

5.1.2 Word Representations

In character-based languages, words are constructed through one or several characters in a complex manner. Linguistic studies have examined the morphological, orthographic, and phonological information within compound words (Zhou et al., 1999; Packard, 2011). However, we distinguish them into the following categories based on whether the characters convey meaning individually and the implicit information density within the characters. In detail, some words are composed of characters with sub-meanings ("compound word"), while some borrowed words from foreign languages only use the pronunciations of the characters ("loanword"). There are also four-character words that represent lengthy allusions, with the characters representing the critical objects in the allusion ("idiom").

We explored methods for better representing these vocabularies based on the sense vectors of the compositional characters to test lexical relationship on the words with explicit meanings. Here are the two methods that we explored.

Firstly, We purposed a method which involved simply computing the average value of the sense

Model	PPL ↓	WCPC top-1 ACC ↑	WCPC top-3 ACC ↑
Backpack-micro	16.25	2.98%	7.46%
GPT2-micro	16.66	2.44%	5.51%
Backpack-small	9.18	4.16%	10.6%
GPT2-small	8.87	4.27%	10.42%
BERT-base, Chinese	-	7.3%	10.1%
RoBERTa-wwm-ext-base	-	6.5%	9.8%
MacBERT-large, Chinese	-	6.8%	10.6%
ALBERT-xxlarge, Chinese	-	4.5%	6.5%

Table 1: Language modeling performance. The baseline WCPC accuracies are from the original paper. For perplexity, lower is better; for accuracy, higher is better.

Sense Vector 10 (*Word Composition*)		Sense Vector 12 (*Character Meaning Relatedness*)	
天 (sky / day)	进 (enter / advance / come in)	天 (sky / day)	进 (enter / advance / come in)
(天)涯 (distant land)	(进)驻 (settle in)	早 (early)	步 (walk / step / pace)
(天)津 (Tianjin City)	(进)入 (enter)	夜 (night)	必 (must / will / certainly)
(天)竺 (Ancient India)	(进)军 (march)	醒 (wake up)	毯 (blanket / carpet)
(天)骄 (exceptional talent)	(进)攻 (attack)	晚 (night)	卧 (lie / crouch)
(天)籁 (beautiful voice)	(进)展 (make progress)	凌 (approach / rise high)	涡 (eddy / whirlpool)

Table 2: The sense vectors in the same index learned a particular facet of character usage in pretraining. Each column contains the characters with the highest scores under the projection of the sense vectors on the vocabulary. Sense vector 10 excels in composing two-character words, while sense vector 12 demonstrates strong character-level semantic correlations.

vectors of the constituent characters to represent the word's sense vector.

Secondly, we hypothesize that words with a complicated, non-systematic function from characters to the word meaning will have their constituent character senses weighted similarly no matter what context they appear in—thus constructing the non-systematic meaning. Suppose we have a context $\mathbf{c}$ that contains a target word with p constituent characters $w = \mathbf{x}_1, \ldots, \mathbf{x}_p$, with the index of these characters in the context $\mathbf{c}$ as $j_{\mathbf{x}_1}, \ldots, j_{\mathbf{x}_p}$, we calculate the average contextual composition ratio $\lambda(\mathbf{c})_\ell$ on sense vector ℓ as

$$\frac{\lambda(\mathbf{c})_{\ell j_{\mathbf{x}_1}}}{\sum_{s=1}^{p} \lambda(\mathbf{c})_{\ell j_{\mathbf{x}_s}}}, \ldots, \frac{\lambda(\mathbf{c})_{\ell j_{\mathbf{x}_p}}}{\sum_{s=1}^{p} \lambda(\mathbf{c})_{\ell j_{\mathbf{x}_s}}} \tag{8}$$

where

$$\lambda(\mathbf{c})_{\ell j_{\mathbf{x}_s}} = \frac{1}{|\mathbf{c}| - j_{\mathbf{x}_p}} \sum_{i=j_{\mathbf{x}_p}+1}^{|\mathbf{c}|+1} \frac{\alpha_{\ell i j_{\mathbf{x}_s}}}{\sum_{k=1}^{p} \alpha_{\ell i j_{\mathbf{x}_k}}} \tag{9}$$

We expect the ratios $\lambda(\mathbf{c}_1)_\ell \approx \cdots \approx \lambda(\mathbf{c}_q)_\ell$ for any q contexts without any significant semantic amplifications on the meaning any of the constituent characters. Assuming this hypothesis holds, a word w could be represented as

$$C(w)_\ell = \frac{1}{q} \sum_{m=1}^{q} \sum_{s=1}^{p} \lambda(\mathbf{c}_m)_{\ell j_{\mathbf{x}_s}} C(\mathbf{x}_s)_\ell \tag{10}$$

for samples of context $\mathbf{c}_1, \ldots, \mathbf{c}_q$.

To prove the feasibility of the second method, we designed several prompts (Appendix 9) that fit different types of words and calculate the average contribution ratio of each character's sense vectors among all constituent characters in the word and how much each contribution is away from the average value. We created a dataset containing 120 compound words, 102 loanwords, and 104 idioms, and validated the above hypothesis on this dataset. Our experimental results showed that each character's contribution ratio in a word on each sense vector for prediction remained stable across various contexts. Furthermore, the stability of word compositions was observed to follow the order of idiom > compound word > loanword as shown in Table 3. However, we also observed that while the senses of most vocabulary items are highly stable across different contexts, there exists a subset of vocabulary items that exhibit poor stabilities. The underlying reasons for this phenomenon warrant further investigation. More word examples are in the Appendix 8.

Type	$\leq \pm 10\%$	$\leq \pm 20\%$	$\geq \pm 20\%$
compound words	69.53%	26.61%	4.06%
loanwords	60.60%	29.64%	9.76%
idioms	84.94%	13.94%	1.20%

Table 3: How the contribution ratio of sense vectors on characters of a word varies among the different contexts. A more minor variation in the contribution ratio indicates a more stable word composition.

5.1.3 Lexical Relationship Test

We evaluated the lexical relationship of the sense vectors using two datasets: Wordsim-240 and Wordsim-297 (Niu et al., 2017), and represent a word by averaging all the sense vectors of the constituent characters. To assess the quality of the resulting lexical representations, we computed Spearman rank-order correlation coefficient between the relationship scores in the datasets and the cosine similarities of each word pair across all the sense vectors of our models. For the GPT2 model, we represented each word by averaging the embeddings of the constituent characters.

Our results in table 4 show that our Backpack Model outperformed the same-scaled GPT2 model, but the results were significantly inferior to word embeddings trained *directly on words* using methods such as word2vec (Mikolov et al., 2013) or GLoVe (Pennington et al., 2014).

Representation	WS240	WS297
Backpack-micro #14	0.335	0.226
GPT2-micro	0.164	0.271
Backpack-small #9	**0.384**	**0.426**
GPT2-small	0.225	0.334
Word-tokenized models (not comparable)		
CBOW	0.561	0.626
GloVe	0.558	0.584

Table 4: Pearson product-moment correlation coefficients between the provided scores and the cosine similarities of the word pairs are calculated. Character-tokenized Backpack LMs outperform GPT2 but are inferior to word-tokenized models.

5.2 Sense Vectors for Control

In this section, we showcase two character-level interventions on the sense vectors as proof-of-concept.

5.2.1 Mitigating gender bias

In Modern Chinese, most professions are composed of two or more Chinese characters, making direct debiasing of stereotypically gendered profession nouns difficult. To address this issue, we attempted two approaches: 1) identifying the characters within the composed words that contain gender-biased meanings and debiasing them from their sense vectors, and 2) directly debiasing the sense vectors of the composed words using the method discussed in Word Representations.

We hypothesized that the first approach could be practical because many Modern Chinese words are combined from ancient single-character words that represent a relevant meaning to the composed words. For example, the word "士兵" (soldier) is composed of "士" (man/warrior) and "兵" (arms), both of which carry stereotypical male bias. In our experiments, we attempted to identify the sense vectors of characters that contain gender stereotypes and compared $|(EC(\mathbf{x_{he}})_\ell - EC(\mathbf{x_{she}})_\ell)|$ to determine which sense vectors contribute to gender bias. We found that sense 3 contributed the most bias. Using the method described in the Backpack paper, we reduced the weight of sense 3 on these characters. We evaluated how the composed words were gender debiased by creating several prompts (Appendix 10) that fit all the profession words, filling in the target word, and computing the average bias probability score of "他 (he/him)" versus "她 (she/her)" as $\mathbb{E}_{X \in \text{prompts}}[\max(\frac{p(\text{he}|\mathbf{x})}{p(\text{she}|\mathbf{x})}, \frac{p(\text{she}|\mathbf{x})}{p(\text{he}|\mathbf{x})})]$.

Baseline. We employed a similar approach as described in the Backpack paper, which was inspired by the work of (Bolukbasi et al., 2016). Specifically, we computed the gender bias direction using the difference between the embeddings of the words "他 (he/him)" and "她 (she/her)," denoted as $E\mathbf{X_{he}} - E\mathbf{X_{she}}$, and then projected the embeddings of the biased characters onto the nullspace of this direction.

Results. We experimented with investigating the effect of removing sense 15 from several characters on bias scores of profession words containing those characters. The bias ratios resulting from this experiment are reported in the table 5. Our experimentation demonstrated that removing sense 15 substantially decreased the bias in words that were originally more biased while producing a considerably lesser impact on words with lower levels of bias. Nonetheless, this approach yielded significant

		Transformers		Backpacks (ours)		
Character	**Target Word**	**GPT2**	**GPT2 proj**	**Backpack**	**half #15**	**remove #15**
兵 (arms)	士兵 (soldier)	70.32	55.55	58.13	34.95	21.34
警 (alert)	警察 (police)	20.93	20.47	23.62	14.90	9.47
演 (act)	演员 (actor / actress)	6.58	6.19	4.92	4.50	4.13
会 (teach)	教师 (teacher)	2.45	2.40	4.69	4.13	3.65

Table 5: Character-level bias ratio; by partially or totally removing sense 15, the character and the words composed by the character get debiased. A perfect unbiased model would achieve a ratio of 1.

Multipliers 沙(sand),滩(beach)	撒 (sanding)	粒 (particle)	堡 (castle)	丘 (dune)	石 (stone)	人 (people)	球 (ball)	海 (sea)	晒 (bask)
1,1	1	1	1	1	1	1	1	1	1
4,1	2.13	1.74	1.42	1.27	1.14	0.78	0.71	0.62	0.61
1,4	0.54	0.55	0.70	0.71	0.71	1.23	1.25	1.24	1.48

Table 6: The ratio of probabilities on predicting certain characters by amplifying the sense vectors with multipliers for the characters "沙" (sand) and "滩" (beach) compared to the original probabilities.

improvements compared to the GPT2 baseline.

Besides, we explored the second approach by removing sense 15 for both constituent characters. Surprisingly, this approach was less effective than the first approach. To investigate whether there exists a specific sense vector to remove for all characters in all compositional words for gender debiasing, we experimented and observed that reducing sense 3 significantly reduced the bias in the word 警察 (police); however, the reducing sense 3 method did not generalize to other words. We hypothesize that the model might not effectively learn the gender-representing information due to the limited model size and pretraining steps. Some critical gender-related information might still distribute among several sense vectors.

5.2.2 Character Amplification Control

Focusing on sub-meanings or properties in a word constructed by multiple characters makes more sense in character-based languages. For instance, the Chinese word "词典" which means "dictionary," is composed of the characters "词" (word) and "典" (book, in ancient Chinese), and when generating text from input containing this word, the model could focus on either the "word" or "book" property. By adjusting the weights of the sense vectors of the constituent characters, we were able to amplify implicit meaning of a constituent character and bias the model toward generating text related to a specific property. Specifically, we conducted experiments to amplify the contribution of the first or second character four times each while keeping the total contribution of the word unchanged in the output. We found that the model tended to generate sentences that relate to the amplified character with greater probability, as shown in Appendix 11. We assessed the efficacy of the proposed method by computing the ratio of expectations for the controlled model relative to an uncontrolled model in the context of predicting semantically related characters from an open-topic prompt as $\mathbb{E}_{\mathbf{c}_{target}}[\frac{p(\mathbf{c}_{target}|\mathbf{x}_{amp})}{p(\mathbf{c}_{target}|\mathbf{x})}]$. Table 6 illustrates an instance of the outcome of amplifying characters in the word "沙滩" (beach). Notably, the findings indicate that character-specific semantics were the most amplified. We hypothesize that this work can assist in scenarios where it is necessary to precisely generate expressions that convey the author's intended meaning in a short sequence, such as poetry, songwriting, or beginning a discourse around one of the meanings in a polysemous word.

6 Conclusion

In this paper, we presented implementing, pretraining, and evaluating a character-based Chinese Backpack language model. We conducted extensive experiments on sense vector visualizations, word representations, lexical relationships, and idiom compositions and explored two approaches to character-level interventions. Our results demonstrate the potential of Backpack LM in language modeling tasks for character-based languages, the

interpretability of the sense vectors on the character and word level, and the potential of character-level interventions across various contexts.

7 Limitations

Despite these promising results, there are several limitations to our study. First, we had limited GPU resources, which prevented us from attempting a larger batch size during pretraining. Second, our word interventions depend on the sub-meanings of the characters, and we currently have no solution to effectively intervene in transliterated words by modifying the sense vectors of the characters that only represent phonetic information. Therefore, intervening in character-based languages where many words are transliterated, such as Korean, remains challenging. Third, we observed that although our approach enables greater flexibility in character-level sense vectors to represent richer morphological structures, word representations by characters are less interpretable than word sense vectors learned by models using word tokenizations, particularly for complex words such as idiomatic phrases. We believe that this issue could be mitigated by increasing the number of sense vectors with a larger contextualization model and pretraining with more data. Further research is required to address these limitations and explore the potential of word representations and interventions in character-based languages.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Rachel Bawden, Nikolay Bogoychev, Ulrich Germann, Roman Grundkiewicz, Faheem Kirefu, Antonio Valerio Miceli Barone, and Alexandra Birch. 2019. The University of Edinburgh's submissions to the WMT19 news translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 103–115, Florence, Italy. Association for Computational Linguistics.

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2000. A neural probabilistic language model. In *Advances in Neural Information Processing Systems*, volume 13. MIT Press.

Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam Tauman Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *NIPS*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Hong Chen, Raphael Shu, Hiroya Takamura, and Hideki Nakayama. 2021. Graphplan: Story generation by planning with event graph.

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

Lei Cui, Fengjiao Cong, Jue Wang, Wenxin Zhang, Yuwei Zheng, and Jukka Hyönä. 2018. Effects of grammatical structure of compound words on word recognition in chinese. *Frontiers in Psychology*, 9.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 657–668, Online. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. 2021. Pre-training with whole word masking for chinese BERT. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3504–3514.

Yiming Cui, Ting Liu, Wanxiang Che, Li Xiao, Zhipeng Chen, Wentao Ma, Shijin Wang, and Guoping Hu. 2019. A span-extraction dataset for Chinese machine reading comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5883–5889, Hong Kong, China. Association for Computational Linguistics.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. Cite arxiv:1810.04805Comment: 13 pages.

Zeyao Du. 2019. Gpt2-chinese: Tools for training gpt2 model in chinese language. `https://github.com/Morizeyao/GPT2-Chinese`.

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science*, 14(2):179–211.

Huibin Ge, Chenxi Sun, Deyi Xiong, and Qun Liu. 2021. Chinese WPLC: A Chinese dataset for evaluating pretrained language models on word prediction given long-range context. In *Proceedings of the*

2021 Conference on Empirical Methods in Natural Language Processing, pages 3770–3778, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Karl Moritz Hermann and Phil Blunsom. 2014. Multilingual models for compositional distributed semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 58–68, Baltimore, Maryland. Association for Computational Linguistics.

John Hewitt, John Thickstun, Christopher D. Manning, and Percy Liang. 2023. Backpack language models. In *Proceedings of the Association for Computational Linguistics*. Association for Computational Linguistics.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences.

karpathy. 2023. nanogpt. `https://github.com/karpathy/nanoGPT`.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *CoRR*, abs/2004.04906.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Xiaoya Li, Yuxian Meng, Xiaofei Sun, Qinghong Han, Arianna Yuan, and Jiwei Li. 2019. Is word segmentation necessary for deep learning of Chinese representations? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3242–3252, Florence, Italy. Association for Computational Linguistics.

Yanran Li, Wenjie Li, Fei Sun, and Sujian Li. 2015. Component-enhanced Chinese character embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 829–834, Lisbon, Portugal. Association for Computational Linguistics.

Zhaojiang Lin, Peng Xu, Genta Indra Winata, Farhad Bin Siddique, Zihan Liu, Jamin Shin, and Pascale Fung. 2020. Caire: An empathetic neural chatbot.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. Cite arxiv:1907.11692.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

A. H. Miller, W. Feng, A. Fisch, J. Lu, D. Batra, A. Bordes, D. Parikh, and J. Weston. 2017. Parlai: A dialog research software platform. *arXiv preprint arXiv:1705.06476*.

Andriy Mnih and Geoffrey E Hinton. 2008. A scalable hierarchical distributed language model. In *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc.

Yilin Niu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2017. Improved word representation learning with sememes. In *Annual Meeting of the Association for Computational Linguistics*.

Jerome L Packard. 2011. *New Approaches to Chinese Word Formation: Morphology, phonology and the lexicon in modern and ancient Chinese*, volume 105. Walter de Gruyter.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernandez. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, Berlin, Germany. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018a. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018b. Language models are unsupervised multitask learners.

Ori Ram, Yuval Kirstain, Jonathan Berant, Amir Globerson, and Omer Levy. 2021. Few-shot question answering by pretraining span selection. *CoRR*, abs/2101.00438.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M. Smith, Y-Lan Boureau, and Jason Weston. 2020. Recipes for building an open-domain chatbot.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152.

Lutfi Kerem Senel, Ihsan Utlu, Veysel Yücesoy, Aykut Koç, and Tolga Çukur. 2017. Semantic structure and interpretability of word embeddings. *CoRR*, abs/1711.00331.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, Morteza Behrooz, William Ngan, Spencer Poff, Naman Goyal, Arthur Szlam, Y-Lan Boureau, Melanie Kambadur, and Jason Weston. 2022. Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage.

Chenglei Si, Zhengyan Zhang, Yingfa Chen, Fanchao Qi, Xiaozhi Wang, Zhiyuan Liu, Yasheng Wang, Qun Liu, and Maosong Sun. 2023. Sub-character tokenization for chinese pretrained language models.

Richard Socher, Christopher D. Manning, and A. Ng. 2010. Learning continuous phrase representations and syntactic parsing with recursive neural networks.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Nitish Srivastava, Ruslan R Salakhutdinov, and Geoffrey E. Hinton. 2013. Modeling documents with deep boltzmann machines.

Anant Subramanian, Danish Pruthi, Harsh Jhamtani, Taylor Berg-Kirkpatrick, and Eduard H. Hovy. 2017. SPINE: sparse interpretable neural embeddings. *CoRR*, abs/1711.08792.

Ilya Sutskever, James Martens, and Geoffrey E Hinton. 2011. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1017–1024.

Nathaniel Swinger, Maria De-Arteaga, Neil Thomas Heffernan IV, Mark D. M. Leiserson, and Adam Tauman Kalai. 2018. What are the biases in my word embedding? *CoRR*, abs/1812.08769.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`.

Bright Xu. 2019a. Nlp chinese corpus: Large scale chinese corpus for nlp.

Bright Liang Xu. 2019b. https://github.com/brightmart/albert_zh. *GitHub repository*.

Jian Xu, Jiawei Liu, Liangang Zhang, Zhengyu Li, and Huanhuan Chen. 2016. Improve Chinese word embeddings by exploiting internal structure. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1041–1050, San Diego, California. Association for Computational Linguistics.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Shaoweihua Liu, Zhe Zhao, Qipeng Zhao, Cong Yue, Xinrui Zhang, Zhengliang Yang, Kyle Richardson, and Zhenzhong Lan. 2020a. CLUE: A Chinese language understanding evaluation benchmark. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4762–4772, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Peng Xu, Mostofa Patwary, Mohammad Shoeybi, Raul Puri, Pascale Fung, Anima Anandkumar, and Bryan Catanzaro. 2020b. Megatron-cntrl: Controllable story generation with external knowledge using large-scale language models.

Mo Yu and Mark Dredze. 2015. Learning composition models for phrase embeddings. *Transactions of the Association for Computational Linguistics*, 3:227–242.

Jiajun Zhang, Shujie Liu, Mu Li, M. Zhou, and Chengqing Zong. 2014. Bilingually-constrained phrase embeddings for machine translation. In *Annual Meeting of the Association for Computational Linguistics*.

Zhengyan Zhang, Xu Han, Hao Zhou, Pei Ke, Yuxian Gu, Deming Ye, Yujia Qin, Yusheng Su, Haozhe Ji, Jian Guan, Fanchao Qi, Xiaozhi Wang, Yanan Zheng, Guoyang Zeng, Huanqi Cao, Shengqi Chen, Daixuan Li, Zhenbo Sun, Zhiyuan Liu, Minlie Huang, Wentao Han, Jie Tang, Juanzi Li, Xiaoyan Zhu, and Maosong Sun. 2020. Cpm: A large-scale generative chinese pre-trained language model.

Xiaolin Zhou, William Marslen-Wilson, Marcus Taft, and Hua Shu. 1999. Morphology, orthography, and phonology reading chinese compound words. *Language and cognitive processes*, 14(5-6):525–565.

A Language Modeling Details

A.1 Residual Connection

We started our experiment with no second residual connection. However, we found that adding second residual connection by unsqueezing the output from the first feed-forward layer by dimension k to match $k * d$ dimensions improved training stability compared to the specification of Hewitt et al. (2023).

A.2 Comparison of Parameter Numbers

The contextualization weight function was defined with mask filling and an extra dropout layer included after the Softmax function.

To make a fair comparison with the corresponding GPT2 model, we analysed the number of parameters and removed one block from the Transformer structure of the Backpack model. As discussed, the contextualization weight of each sense vector is calculated with additional matrices $K, Q \in \mathbb{R}^{d \times d}$. The first feed forward layer in the sense vector layer involves an up projection matrix $\in \mathbb{R}^{d \times 4d}$ and a down projection matrix $\in \mathbb{R}^{4d \times d}$. Summing up these parameters, we have a $10d^2$ additional parameter size, which is close to the $12d^2$ parameter size in a single Transformer block so that by removing one block, we will only add $(k-2) * d^2 \approx k * d^2$ parameters which are necessary for representing the sense vectors.

B Interpreting Idiom Composition

We investigated which sense vectors played a dominant role when the model used the first three characters of idiomatic phrases as input to predict the last character. However, we encountered difficulty in interpreting the character composition of idiomatic phrases. For example, when analyzing the phrase "画蛇添(足)" i.e., "drawing legs on a snake," which means "an unnecessary and redundant act that spoils the original effect or even makes it worse," by stacking weights of the first three characters on 16 or 64 sense vectors, we found that using any single sense vector for prediction did not significantly lead the model to output the target character, even though the model correctly outputted "足" i.e., "leg" after performing a weighted sum of these sense vectors. We projected 16 sense vectors onto the vocabulary and examined their projections onto the character; however, we observed that none exhibited a disproportionately large or small projection onto the resulting character. This experiment provides evidence that the top components of sense vectors may not effectively capture how they will compose to make predictions.

Sense Vector 10 (*Word Composition*)

天 (sky / day)	沙 (sand)	进 (enter / advance / come in)	自 (from / self)
(天)涯 (distant land)	(沙)漠 (desert)	(进)驻 (settle in)	(自)由 (freedom)
(天)津 (Tianjin City)	(沙)鸥 (gull)	(进)入 (enter)	(自)慰 (console)
(天)竺 (Ancient India)	(沙)哑 (hoarse)	(进)军 (march)	(自)如 (the App Ziroom)
(天)骄 (exceptional talent)	(沙)溢 (actor Yi Sha)	(进)攻 (attack)	(自)拍 (selfie)
(天)籁 (beautiful voice)	(沙)滩 (Beach)	(进)展 (make progress)	(自)卸 (self-dumping)

Sense Vector 12 (*Character Meaning Relatedness or Composition*)

天 (sky / day)	沙 (sand)	进 (enter / advance / come in)	自 (from / self)
早 (early)	04 (FC Schalke 04)	步 (walk / step / pace)	从 (from)
夜 (night)	(沙)箱 (sandbox)	必 (must / will / certainly)	之 (he / she / it / go / 's)
醒 (wake up)	(沙)盒 (sandbox)	毯 (blanket / carpet)	打 (since)
晚 (night)	毒 (poison)	卧 (lie / crouch)	感 (sense / feel)
凌 (approach / rise high)	铂 (platinum)	洄 (eddy / whirlpool)	蚂 (ant)

Sense Vector 15 (*Character Meaning Relatedness or Composition*)

天 (sky / day)	沙 (sand)	进 (enter / advance / come in)	自 (from / self)
黑 (black)	潇 (drizzle)	(进)展 (progress)	(自)大 (arrogant)
亮 (light)	浏 (clear)	顺 (smooth)	(自)满 (complacent)
昨 (yesterday)	湖 (lake)	神 (magical / god)	狠 (ruthless)
黑 (black)	岳 (mountain)	慢 (slow)	(自)暴 (Give up on yourself)
今 (today)	橘 (tangerine)	缓 (delay)	(自)免 (to resign voluntarily)

Table 7: The sense vectors in the same index are considered to have a particular facet of character usage. Each column contains the characters with the highest scores under the projection of the sense vectors on the vocabulary.

Type	**Word**	**Stablility**	$\leq \pm 10\%$	$\leq \pm 20\%$	$\geq \pm 20\%$
compound words	手机 (telephone) = 手 (hand) + 机 (machine)	high	16	0	0
	大学 (university) = 大 (large) + 学 (learn)	high	16	0	0
	孤独 (lonely) = 孤 (isolated) + 独 (alone)	low	1	6	9
loanwords	马赛克 (Mosaic)	high	16	0	0
	迷你 (mini)	high	12	4	0
	夸克 (quark)	low	5	7	4
idioms	骑虎难下 (in a difficult situation with no easy way out)	high	16	0	0
	画蛇添足(to do something unnecessary even harmful)	high	14	2	0
	韬光养晦 (to wait for the right moment to shine)	low	12	2	2

Table 8: How many sense vectors for each range of the contribution ratio on characters of a word varies among the different contexts. A more minor variation in the contribution ratio indicates a more stable word composition.

prompt	**English**
WORD	WORD
"WORD"的意思是	The meaning of "WORD" is
老师曾教育，WORD	A teacher told that WORD
关于WORD，	About WORD,
电视里说，WORD	In TV, it is said that WORD
WORD是	WORD is
我觉得WORD	I think WORD

Table 9: General prompts for different type of nouns

prompt	English
那个WORD说，	That WORD said,
这个WORD相信	This WORD believes
WORD进到屋子里，	The WORD enters the house,
WORD坐在车里，然后	The WORD sat in the car, and then
WORD走了过来，	Then WORD came over,

Table 10: General prompts for gender bias evaluations

Word	Multiplier	Output
沙滩 (beach) 沙(sand) 滩(beach / puddle)	1,1	沙滩上有很多人。 (There are a lot of people on the beach.)
沙滩 (beach) 沙(**sand**) 滩(beach / puddle)	4,1	沙滩上有很多大大小小的沙堆。 (On the beach, there are many big and small **sand dunes**.)
沙滩 (beach) 沙(sand) 滩(**beach / puddle**)	1,4	沙滩上有很多人在海边钓鱼。 (There are many people fishing **by the seaside** on the beach.)
理想 (ideal) 理(principle / logic) 想(imagine / want)	1,1	理想是什么?我很迷茫，不知道自己喜欢什么。 (What is ideal? I am confused and unsure of what I truly like.)
理想 (ideal) 理(**principle / logic**) 想(imagine / want)	4,1	理想是什么？如何理解? (What is ideal? How to **understand** it?)
理想 (ideal) 理(principle / logic) 想(**imagine / want**)	1,4	理想是什么？如何做到? (What is ideal? How to **achieve** it?)

Table 11: Generative outputs on the character amplification control task with top probabilities. Note that the word "理想" means "ideal" but is combined with the characters meaning "principle / logic" and "imagine / want".