Decompose, Fuse and Generate:
A Formation-Informed Method for Chinese Definition Generation

Hua Zheng[1,2*], Damai Dai[1,2*], Lei Li[1,2], Tianyu Liu[1,2],
Zhifang Sui[1,2,3], Baobao Chang[1,2,3], Yang Liu[1,2†]
[1]Department of Computer Science and Technology, Peking University
[2]Key Lab of Computational Linguistics (MOE), Peking University
[3]Peng Cheng Laboratory, China
`{zhenghua,daidamai,tianyu0421,szf,chbb,liuyang}@pku.edu.cn`
`lilei@stu.pku.edu.cn`

Abstract

In this paper, we tackle the task of Definition Generation (DG) in Chinese, which aims at automatically generating a definition for a word. Most existing methods take the source word as an indecomposable semantic unit. However, in parataxis languages like Chinese, word meanings can be composed using the word formation process, where a word ("桃花", *peach-blossom*) is formed by formation components ("桃", *peach*; "花", *flower*) using a formation rule (*Modifier-Head*). Inspired by this process, we propose to enhance DG with word formation features. We build a formation-informed dataset and propose a model **DeFT**, which Decomposes words into formation features, dynamically Fuses different features through a gating mechanism, and generaTes word definitions. Experimental results show that our method is both effective and robust.[1]

1 Introduction

Definition Generation (DG) aims at automatically generating an explanatory text for a word. This task is of practical importance to assist dictionary construction, especially in highly productive languages like Chinese (Yang et al., 2020). Most existing methods take the source word as an indecomposable lexico-semantic unit, using features like word embedding (Noraset et al., 2017) and context (Gadetsky et al., 2018; Ishiwatari et al., 2019). Recently, Yang et al. (2020) and Li et al. (2020) achieve improvement by decomposing the word meaning into different semantic components.

In decomposing the word meaning, the word formation process is an intuitive and informative way that has not been explored in DG by far. For parataxis languages like Chinese, a word is formed by formation components, i.e., **morphemes**, and a formation **rule**. As shown in Figure 1, the polysemous word "白花" holds two meanings "白花$_1$" and "白花$_2$", which can be distinguished by different morphemes ("白$_1$;花$_1$" vs. "白$_2$;花$_2$") and different rules (*Modifier-Head* vs. *Adverb-Verb*). Such intuitive formation process can clearly and unambiguously construct the word meaning.

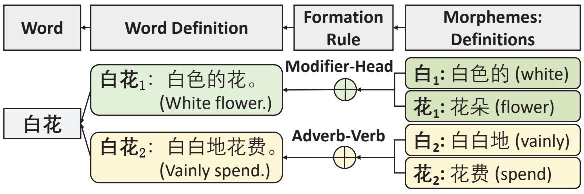

Figure 1: Word formation process for the polysemous "白花". With morphemes and a formation rule specified, the process can construct and distinguish each meaning. Definitions are simplified for ease of reading.

Inspired by the word formation process in Chinese, we propose to enhance DG with formation features. First, we build a formation-informed dataset under expert annotations. Next, we design a DG model **DeFT**, which Decomposes words into formation features, Fuses different features through a gating mechanism, and generaTes definitions.

Our contributions are as follows: (1) We first propose to use word formation features to enhance DG and design a formation-informed model **DeFT**. (2) We build a new formation-informed DG dataset under expert annotations. (3) Experimental results show that our method brings a substantial performance improvement, and maintains a robust performance even with only word formation features.

2 Related Work

Definition Generation: Noraset et al. (2017) first propose the DG task and use word embeddings as the main input. The following methods add contexts for disambiguation (Gadetsky et al., 2018; Ishiwatari et al., 2019) or word-pair embeddings to capture lexical relations (Washio et al., 2019).

[*] Equal contribution.

[†] Corresponding author.

[1] The code is available at `https://github.com/Hunter-DDM/DeFT-naacl2021`.

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5524–5531
June 6–11, 2021. ©2021 Association for Computational Linguistics

Recent methods attempt to decompose the word meaning by using HowNet sememes (Yang et al., 2020) or modeling latent variables (Li et al., 2020).
Semantic Components: To systematically define words, linguists decompose the word meaning into semantic components (Wierzbicka, 1996). Following this idea, HowNet (Dong and Dong, 2006) uses manually-created sememes to describe the semantic aspects of words. Recent studies also show that leveraging subword information produces better embeddings (Park et al., 2018; Lin and Liu, 2019; Zhu et al., 2019), but these methods lack a clear distinction among different formation rules.

3 Word Formation Process in Chinese

It is linguistically motivated to explore the word formation process to better understand words. Instead of combining roots and affixes, Chinese words are formed by characters in a parataxis way (Li et al., 2018). Here, we introduce two formation features and construct a formation-informed dataset.

3.1 Formation components and rules

Chinese formation components are **morphemes**, defined as the smallest meaning-bearing units (Zhu, 1982). Morphemes are unambiguous in representing word meanings, since they can distinguish different meanings and uses of each character in a word, like "白$_1$" and "白$_2$" in Figure 1. Morphemes are also productive in constructing words, since over 99.48% Chinese words are formed using a small set of nearly 20,000 morphemes (Fu, 1988). These properties make morphemes highly effective as formation components.

Formation **rules** specify how morphemes are combined to form words in a parataxis way. For example, the *Modifier-Head* rule uses the first morpheme to modify the second morpheme. Following the study of Liu et al. (2018), we adopt 16 Chinese formation rules and show the top 5 in instance percentage in Table 1. Complete descriptions of 16 formation rules are provided in Appendix A.

3.2 Formation-informed dataset

We construct a DG dataset under expert annotations, which contains morphemes and formation rules. Each entry consists of (1) source word, (2) morphemes and morpheme definitions, (3) formation rule, (4) context (a sentence containing the source word), (5) source word definition.

To ensure full coverage and fine granularity, we extract data from the 5th edition of the Contemporary Chinese Dictionary published by the Commercial Press[2], one of the most influential Chinese dictionaries. We collect 45,311 Chinese disyllabic word entries with contexts and definitions. To annotate them, we also collect 10,527 Chinese characters and 20,855 morphemes with definitions.

Formation Rule	Use Case	%
Modifier-Head	红花 (red-flower)	38.62
Parallel	昏花 (dizzy-dim)	22.87
Verb-Object	花钱 (spend-money)	16.44
Adverb-Verb	白花 (vainly-spend)	8.45
Single Morpheme	花生 (peanut)	3.51

Table 1: Examples of word formation rules and use cases. % denotes the instance percentage.

Morpheme (ID)	Morpheme Definition
花$_1$ (07361-01)	花朵 (flower)
花$_2$ (07361-06)	模糊；迷乱 (dim; blurred)
花$_3$ (07361-09)	用；耗费 (use; spend)

Table 2: Three example morphemes and definitions for the character "花". We give each morpheme a unique ID, C-M (C is character rank, M is morpheme rank).

Our annotators include two professors and six graduates major in Chinese linguistics. Given the definition, they annotate each word with its formation rule (as shown in Table 1) and morpheme IDs (as shown in Table 2). Each entry is cross-validated by three independent annotators and reviewed by one. The detailed annotation process includes the following three steps:

(1) Equipped with the definition, annotators annotate each entry with two morpheme IDs (select from the morphemes of each character) and a formation rule (select from 16 formation rules). Each entry is independently annotated by three annotators, who also note down a confidence score. If three annotations are the same, turn to (3); otherwise, turn to (2).

(2) Another annotator reviews the conflicting annotations and confidence scores, and decides the final annotation. Turn to (3).

(3) The annotation is collected as an entry into the final dataset.

It takes one minute on average for each annotator to annotate an entry. Only 8,193 out of 45,311 entries enter Phase (2) in the whole process.

[2] https://www.cp.com.cn

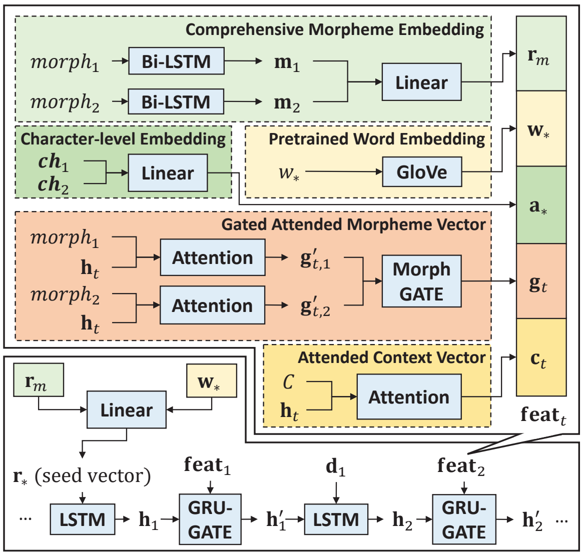

Figure 2: An illustration of DeFT. The definition generator uses the seed vector as input, and dynamically fuses different features using a gating mechanism.

4 Approach

4.1 Task formulation

We extend the DG setting in Ishiwatari et al. (2019) to incorporate the word formation features, $F = \{morph_1, morph_2, rule\}$, where $morph_i$ is the i^{th} morpheme definition sentence and $rule$ is the formation rule. The training goal is to maximize the likelihood of the ground-truth definition $D = d_{1:T}$ given the source word w_*, the context sentence $C = c_{1:n}$, and the word formation features F:

$$p(D|w_*, C, F) = \prod_{t=1}^{T} p(d_t|d_{i<t}, w_*, C, F).$$

Our optimization objective is to minimize the cross-entropy loss $\mathcal{L}$:

$$\mathcal{L} = -\sum_{t=1}^{T} \log\left(p(d_t|d_{i<t}, w_*, C, F)\right),$$

where $d_{1:T}$ is the ground-truth definition, w_* is the pretrained embedding of the source word, C is the context sentence, F is the formation information.

4.2 Proposed model: DeFT

As shown in Figure 2, DeFT first produces a seed vector in a rule-specific manner as global supervision. Then we feed it into the definition generator, which uses a gating mechanism to dynamically fuse different features and generate definitions.

4.2.1 Seed vector

We first employ a Bi-LSTM (Graves and Schmidhuber, 2005) to encode $morph_i$. Then, we combine $morph_i$ into a comprehensive morpheme embedding $\mathbf{r}_m$ with a rule-specific linear layer, which captures different semantic relations:

$$\begin{aligned}\mathbf{m}_i &= \text{Bi-LSTM}([morph_i]),\\ \mathbf{r}_m &= \mathbf{W}_m^{(rule)}[\mathbf{m}_1; \mathbf{m}_2] + \mathbf{b}_m^{(rule)}.\end{aligned}$$

We then use a linear layer to combine $\mathbf{r}_m$ and the pretrained source word embedding $\mathbf{w}_*$ to obtain the seed vector $\mathbf{r}_*$ as the initial generator input:

$$\mathbf{r}_* = \mathbf{W}_r[\mathbf{r}_m; \mathbf{w}_*] + \mathbf{b}_r.$$

4.2.2 Definition generator

We employ an LSTM followed by a GRU-like (Cho et al., 2014) gate GRU-GATE$(\cdot)$, which dynamically fuses different features, as the generator:

$$\begin{aligned}\mathbf{h}_t &= \text{LSTM}(\mathbf{d}_{t-1}, \mathbf{h}'_{t-1}),\\ \mathbf{h}'_t &= \text{GRU-GATE}(\mathbf{h}_t, \mathbf{feat}_t),\\ \mathbf{feat}_t &= [\mathbf{r}_m; \mathbf{w}_*; \mathbf{a}_*; \mathbf{g}_t; \mathbf{c}_t],\end{aligned}$$

where $\mathbf{h}_t$ is the LSTM hidden state at the t^{th} step, $\mathbf{h}'_t$ is the gated hidden state, $\mathbf{d}_{t-1}$ is the embedding of the previous definition word, specially, $\mathbf{d}_0 \triangleq r_*$, and $\mathbf{feat}_t$ denotes the features that dynamically control the generation process. We explain $\mathbf{a}_*$, $\mathbf{g}_t$, and $\mathbf{c}_t$ as follows.

$\mathbf{a}_*$ is the character-level embedding, obtained by combining the embedding $\mathbf{ch}_i$ of each character in w_* with a rule-specific linear layer:

$$\mathbf{a}_* = \mathbf{W}_a^{(rule)}[\mathbf{ch}_1; \mathbf{ch}_2] + \mathbf{b}_a^{(rule)}.$$

$\mathbf{g}_t$ is the gated attended morpheme vector that dynamically focuses on the most relevant parts in morphemes during generation. We first calculate attended morpheme vectors $\mathbf{g}'_{t,i}$ by the attention mechanism (Bahdanau et al., 2015):

$$\mathbf{g}'_{t,i} = \text{Attention}(\mathbf{h}_t, morph_i),$$

where Attention$(\mathbf{h}, seq)$ denotes the function that uses $\mathbf{h}$ to attend over the Bi-LSTM encoded seq. We then design a **MorphGATE** to compute $\mathbf{g}_t$ by assigning different weights to two morphemes:

$$\begin{aligned}\mathbf{z}_t &= \sigma(\mathbf{W}_z[\mathbf{g}'_{t,1}; \mathbf{g}'_{t,2}; \mathbf{h}_t] + \mathbf{b}_z),\\ \mathbf{g}_t &= (\mathbf{1} - \mathbf{z}_t) \odot \mathbf{g}'_{t,1} + \mathbf{z}_t \odot \mathbf{g}'_{t,2},\end{aligned}$$

Split	#Words	#Entries	Context Length	Morph$_1$ Length	Morph$_2$ Length	Definition Length
Train	29,169	36,248	7.22	7.69	7.29	12.02
Valid	3,673	4,531	7.32	7.45	7.30	11.91
Test	3,666	4,532	7.26	7.51	7.01	12.03

Table 3: Statistics of our formation-informed dataset. Morph$_i$ denotes the definition of the i^{th} morpheme. The length is calculated as the average number of Chinese characters.

where $\sigma(\cdot)$ is Sigmoid and $\odot$ is Hadamard product.

$\mathbf{c}_t$ is the attended context vector. Following Ishiwatari et al. (2019), we take $\mathbf{c}_t = \text{Attention}(\mathbf{h}_t, C)$ as a feature since it may assist disambiguation.

Finally, GRU-GATE$(\mathbf{h}_t, \mathbf{feat}_t)$ takes the LSTM hidden state $\mathbf{h}_t$ and the dynamically controlled features $\mathbf{feat}_t$ as input, and updates $\mathbf{h}_t$ to $\mathbf{h}'_t$ by fusing different features:

$$
\begin{aligned}
\mathbf{u}_t &= \sigma(\mathbf{W}_u[\mathbf{h}_t; \mathbf{feat}_t] + \mathbf{b}_u),\\
\mathbf{v}_t &= \sigma(\mathbf{W}_r[\mathbf{h}_t; \mathbf{feat}_t] + \mathbf{b}_r),\\
\hat{\mathbf{h}}_t &= \tanh(\mathbf{W}_h[(\mathbf{v}_t \odot \mathbf{feat}_t); \mathbf{h}_t] + \mathbf{b}_h),\\
\mathbf{h}'_t &= \mathbf{u}_t \odot \mathbf{h}_t + (\mathbf{1} - \mathbf{u}_t) \odot \hat{\mathbf{h}}_t,
\end{aligned}
$$

where σ denotes the Sigmoid and $\odot$ denotes the Hadamard product. The gate $\mathbf{u}_t$ controls how much the original state $\mathbf{h}_t$ is remained, and the gate $\mathbf{v}_t$ controls the contribution from features $\mathbf{feat}_t$.

5 Experiments

5.1 Experimental settings

Dataset: We split the dataset described in Section 3 into training, validation and test sets by 8:1:1, as shown in Table 3. Note that we treat polysemous words as different entries, and the words are mutually exclusive across three sets.

Hyper-parameters: We tune hyper-parameters to achieve the best BLEU score on the validation set. We use Adam (Kingma and Ba, 2015) with an initial learning rate of 10^{-3} as the optimizer. We set hidden size to 300, batch size to 64 and dropout rate to 0.2. Word embeddings are 300-dimensional, pretrained by fastText (Bojanowski et al., 2017). We train for up to 50 epochs, and early stop the training process once the performance does not improve for 10 consecutive epochs. We run our experiments on a single NVIDIA GeForce GTX 2080Ti GPU with 11 GB memory.

Baselines: We compare with two reproducible baselines that have a similar model framework with us but using different features, including SG (Noraset et al., 2017) that uses only the word feature, and LOG-CaD (Ishiwatari et al., 2019) that uses both the word and context features.

	Automatic		Human	
	BLEU	ROUGE-L	Coverage	Overall
W (SG*)	23.63	34.77	2.04	2.11
W+C (LOG-CaD*)	24.06	34.93	2.52	2.45
F	25.55	37.27	2.61	2.58
F+W	25.93	37.94	2.78	2.71
F+C	25.69	37.29	2.89	2.78
F+W+C (DeFT)	**26.42**	**38.58**	**3.29**	**3.19**

Table 4: Automatic and human evaluation results. Best results are in **Bold**. W, C and F are word, context and formation features, respectively. * denotes baselines.

5.2 Evaluation results

We conduct both automatic and human evaluations to validate our method, and show results in Table 4.

For automatic evaluation, we select BLEU-4 (Bahdanau et al., 2015) and ROUGE-L (Lin, 2004) as metrics. We find that (1) our formation-informed DeFT (F+W+C) significantly outperforms baselines and other simplified versions (F, F+W, F+C); (2) based on W or W+C, adding formation features introduces significant improvement; (3) formation features are robust, since using only F can outperform LOG-CaD by 9.8% and 10.45% in BLEU and ROUGE-L, respectively. These findings validate that formation features can effectively enhance DG by assisting word meaning construction.

For human evaluation, we measure semantic **coverage** and **overall** quality. The **coverage** metric measures how much ground-truth information is mentioned in the predicted definition. To be specific, the scores are given based on how many semantic aspects in the ground truth definition are covered by the predicted definition. The **overall** metric measures the overall quality of the predicted definition, referencing the ground-truth definition. We randomly select 100 entries from the test set, and hire three raters to rate the predicted definitions on a scale of 1 to 5, where each entry includes (1) the source word, (2) the ground-truth definition, and (3) the predicted definition to the raters. We show in Table 5 the detailed guideline for raters on each point.

The inter-rater kappa (Fleiss and Cohen, 1973) is 0.65 for coverage and 0.66 for overall. We average scores of raters and obtain consistent results with

Point	Coverage	Overall
1	Nothing is covered.	Completely wrong or not related to the ground-truth.
2	Some semantic aspects are similar, but not the same.	Almost wrong but has some correct information.
3	Some semantic aspects are covered.	Basically correct with some minor errors.
4	Almost all semantic aspects are covered.	Correct but redundant or missing details.
5	Everything is covered.	Exactly correct.

Table 5: Human evaluation guideline for raters. Note that the evaluation results on these two metrics may show different trends. For example, a predicted definition with an opposite meaning to the ground-truth definition may receive a high coverage score but a low overall score.

	BLEU (Δ)	**ROUGE-L** (Δ)
DeFT	26.42	38.58
w/o MorphGATE	25.81 (0.61↓)	38.01 (0.57↓)
w/o Formation Rules	25.52 (0.90↓)	37.40 (1.18↓)

Table 6: Ablation study of DeFT.

INPUT	**Word** (w_*)	零用$_1$	零用$_2$
	Morph$_1$	零$_1$: piecemeal	零$_1$: piecemeal
	Morph$_2$	用$_1$: use	用$_2$: expense
	Rule	Adverb-Verb	Modifier-Head
	Context	你可以 $[w_*]_1$ 50 元。 (You can $[w_*]_1$ 50 yuan.)	拿更多 $[w_*]_2$。 (Get more $[w_*]_2$.)
	Definition	零碎地使用。 (Use in a piecemeal way.)	零碎的费用。 (Piecemeal expense.)
OUTPUT	**W**	钱用钱。 (Money uses money.)	钱用钱。 (Money uses money.)
	F	零碎的使用。 (Piecemeal use.)	零碎的钱。 (Piecemeal money.)
	F+W+C (DeFT)	零碎地使用。 (Use in a piecemeal way.)	零碎的费用。 (Piecemeal expense.)

Figure 3: Generation examples for a polysemous word "零用" using different features.

the automatic evaluation: formation features are effective and DeFT performs the best.

5.3 Analysis

Ablation study: Based on DeFT, we perform ablation study regarding MorphGATE and the formation rule in Table 6. (1) For MorphGATE, we replace it with a simple average function, which leads to a drop in performance. This reveals that different morphemes take effect in different generation phases. (2) For formation rule, we replace the rule-specific layers with a rule-shared layer, leading to a more serious performance drop. This verifies that distinguishing the specific formation rule can assist word meaning construction.

Formation features can assist disambiguation: We present the generated definitions for a polysemous word in Figure 3. The example shows that using only the word feature (W) cannot distinguish different meanings. By contrast, using only the formation features (F) can capture the meaning difference and disambiguate the word (*use* vs. *money*). Further, DeFT (F+W+C) generates the exactly correct definition by fusing different features. Due to space limits, we put two additional interesting analyses on formation rules in Appendix B.

Formation features are more feasible and effective compared with sememes: Sememes are expert-crafted words to describe the semantic aspects of words. For annotation cost, annotating sememes is as expensive as writing definitions (Li et al., 2020), whereas annotating formation features is a simple multiple-choice task with 1.98 choices on average. For effectiveness, we conduct experiments using sememe embeddings from Yang et al. (2020) as additional features. Results show that, based on W, adding sememes brings a BLEU improvement of 0.52, lower than that of 2.30 from F+W. Further, based on DeFT, adding sememes even brings noises and decreases BLEU by 0.35. This indicates that, compared with sememes, formation features are more feasible and effective.

6 Conclusion

In this paper, we propose to use formation features to enhance DG. We build a formation-informed dataset and design a model DeFT, which decomposes words into formation features and fuses features via a gating mechanism. Experimental results show that our method is both effective and robust.

Acknowledgments

We would like to thank all the anonymous reviewers and annotators for their helpful advice on various aspects of this work, including Fuqian Wu, Ming Liu, Yaqi Yin, Yue Wang, etc. This paper is supported by the National Natural Science Foundation of China (No. 62036001, U19A2065) and the National Social Science Foundation of China (No. 16YY137, 18ZDA295).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR 2015*.

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

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP 2014*, pages 1724–1734.

Zhendong Dong and Qiang Dong. 2006. *Hownet and the computation of meaning*. World Scientific.

Joseph L Fleiss and Jacob Cohen. 1973. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and Psychological Measurement*, 33(3):613–619.

Yonghe Fu. 1988. Modern chinese general character list: Appendix of the most and secondly frequently used characters. *Yuwen Jianshe*.

Artyom Gadetsky, Ilya Yakubovskiy, and Dmitry P. Vetrov. 2018. Conditional generators of words definitions. In *ACL 2018*, pages 266–271.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

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 *NAACL-HLT 2019*, pages 3467–3476.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR 2015*.

Jiahuan Li, Yu Bao, Shujian Huang, Xinyu Dai, and Jiajun Chen. 2020. Explicit semantic decomposition for definition generation. In *ACL 2020*, pages 708–717.

Shen Li, Zhe Zhao, Renfen Hu, Wensi Li, Tao Liu, and Xiaoyong Du. 2018. Analogical reasoning on chinese morphological and semantic relations. In *ACL 2018*, pages 138–143.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *ACL 2004 Workshop on Text Summarization Branches Out*, pages 74–81.

Zi Lin and Yang Liu. 2019. Implanting rational knowledge into distributed representation at morpheme level. In *AAAI 2019*, pages 2954–2961.

Yang Liu, Zi Lin, and Sichen Kang. 2018. Towards a description of chinese morpheme conceptions and semantic composition of word. *Journal of Chinese Information Processing*, 32(2):12–21.

Thanapon Noraset, Chen Liang, Larry Birnbaum, and Doug Downey. 2017. Definition modeling: Learning to define word embeddings in natural language. In *AAAI 2017*, pages 3259–3266.

Hyun-jung Park, Min-chae Song, and Kyung-Shik Shin. 2018. Sentiment analysis of korean reviews using cnn: Focusing on morpheme embedding. *Journal of Intelligence and Information Systems*, 24(2):59–83.

Koki Washio, Satoshi Sekine, and Tsuneaki Kato. 2019. Bridging the defined and the defining: Exploiting implicit lexical semantic relations in definition modeling. In *EMNLP-IJCNLP 2019*, pages 3519–3525.

Anna Wierzbicka. 1996. *Semantics: Primes and universals: Primes and universals*. Oxford University Press, UK.

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.

Dexi Zhu. 1982. *Yufa Jiangyi (Lectures on Grammar)*. The Commercial Press, China.

Yi Zhu, Ivan Vulic, and Anna Korhonen. 2019. A systematic study of leveraging subword information for learning word representations. In *NAACL-HLT 2019*, pages 912–932.

A Formation-Informed Dataset Details

We show the complete descriptions of 16 formation rules in Table 7.

B Additional Analysis on Formation Rules

Here we provide some additional interesting analyses that reveal the specific properties and influences of word formation rules.

B.1 The similarity among different formation rules

In Section 4.2.1, we produce a comprehensive morpheme embedding using a rule-specific linear layer. We study the relations of the weight matrices $\mathbf{W}_m^{(rule)}$ in these layers by resizing them into vectors and calculating their pairwise cosine similarity, as shown in Figure 4.

Figure 4 (a) shows that Overlapping is most similar to Suffixation, Prefixation, and Single Morpheme. Interestingly, word meanings constructed

Word Formation Rule	Explanation	Use Case	%
定中 dìng·zhōng (Modifier-Head)	morph$_1$ modifies morph$_2$ (noun).	红花 hóng·huā (red-flower)	38.62
联合 lián·hé (Parallel)	morph$_1$ and morph$_2$ are similar, contrasting or complementary.	昏花 hūn·huā (dizzy-dim)	22.87
述宾 shù·bīn (Verb-Object)	morph$_1$ operates on morph$_2$.	花钱 huā·qián (spend-money)	16.44
状中 zhuàng·zhōng (Adverb-Verb)	morph$_1$ modifies morph$_2$ (verb).	白花 bái·huā (vainly-spend)	8.45
单纯 dān·chún (Single Morpheme)	The word is a single morpheme.	花生 huā·shēng (peanut)	3.51
连谓 lián·wèi (Verb-Consequence)	morph$_2$ is the consequence of morph$_1$.	休息 xiū·xi (stop-rest)	3.43
后缀 hòu·zhuì (Suffixation)	morph$_2$ is the suffix of morph$_1$.	花头 huā·tóu (trick-$\emptyset$)	2.70
述补 shù·bǔ (Verb-Complement)	morph$_2$ is the action follows morph$_1$.	压低 yā·dī (press-down)	1.28
主谓 zhǔ·wèi (Subject-Predicate)	morph$_1$ is the subject of morph$_2$.	眼花 yǎn·huā (eyesight-dim)	1.06
重叠 chóng·dié (Overlapping)	morph$_1$ and morph$_2$ are the same.	白白 bái·bái (vainly-vainly)	0.59
方位 fāng·wèi (Entity-Position)	morph$_1$ is an entity, morph$_2$ is a position.	期中 qī·zhōng (semester-mid)	0.37
介宾 jiè·bīn (Preposition-Object)	morph$_1$ is a preposition, morph$_2$ is an object.	凭空 píng·kōng (from-nowhere)	0.31
名量 míng·liàng (Noun-Quantifier)	morph$_2$ is the quantifier of morph$_1$.	花朵 huā·duǒ (flower-bud)	0.13
数量 shù·liàng (Number-Quantifier)	morph$_1$ is a number, morph$_2$ is a quantifier.	一点 yì·diǎn (one-dot)	0.10
前缀 qián·zhuì (Prefixation)	morph$_1$ is the prefix of morph$_2$.	老师 lǎo·shī ($\emptyset$-teacher)	0.10
复量 fù·liàng (Quantifier-Quantifier)	Both morph$_1$ and morph$_2$ are quantifiers.	千米 qiān·mǐ (kilo-meter)	0.03

Table 7: Descriptions of the total 16 formation rules. $\emptyset$ denotes the affix and % denotes the instance percentage. The first and the third columns are in the format of "Chinese characters - Chinese phonetic notation - (English translation)". To help understand these rules, we give a simple explanation to describe the relation between two morphemes in the second column.

by these four formation rules share a similar pattern of using only one morpheme. For example, in Suffixation, only the first morpheme carries the meaning, like "花头" (*trick-$\emptyset$*).

Figure 4 (b) shows that Noun-Quantifier is most similar to Quantifier-Quantifier and Number-Quantifier. Word meanings constructed by these three formation rules all have a quantifier morpheme. For example, in Number-Quantifier, the first morpheme is a number, and the second morpheme is a quantifier, like "一点" (*one-dot*).

Figure 4 (c) shows that Verb-Object is most similar to Modifier-Head, Parallel and Adverb-Verb. Word meanings constructed by Verb-Object, Modifier-Head and Adverb-Verb share a similar pattern of using the first morpheme to operate on the second morpheme. For example, in Adverb-Verb, the first morpheme modifies the second verb morpheme, like "白花" (*vainly-spend*).

B.2 The specific impact of formation rules

In Table 8, we generate definitions using different formation rules for the same word. Results show that each predicted definition indicates a clear pattern of the used formation rule. The Modifier-Head rule uses *touched* to modify *sad*, and outputs a noun; the Adverb-Verb rule outputs an adjective in a similar modifying way; the Parallel rule outputs a single meaning of *sad* with differ-

感伤	
Morph$_1$: 感触 (touched)	**Morph$_2$**: 伤心 (sad)
Verb-Consequence (✓)	因有所感触而悲伤。 (Sad due to being touched.)
Modifier-Head	感触的悲伤。 (Sadness of being touched.)
Parallel	伤心。 (Sad.)
Adverb-verb	难过的伤心。 (Woefully sad.)

Table 8: A case study of 4 predicted definitions of "感伤" using 1 correct formation rule (✓) and 3 others.

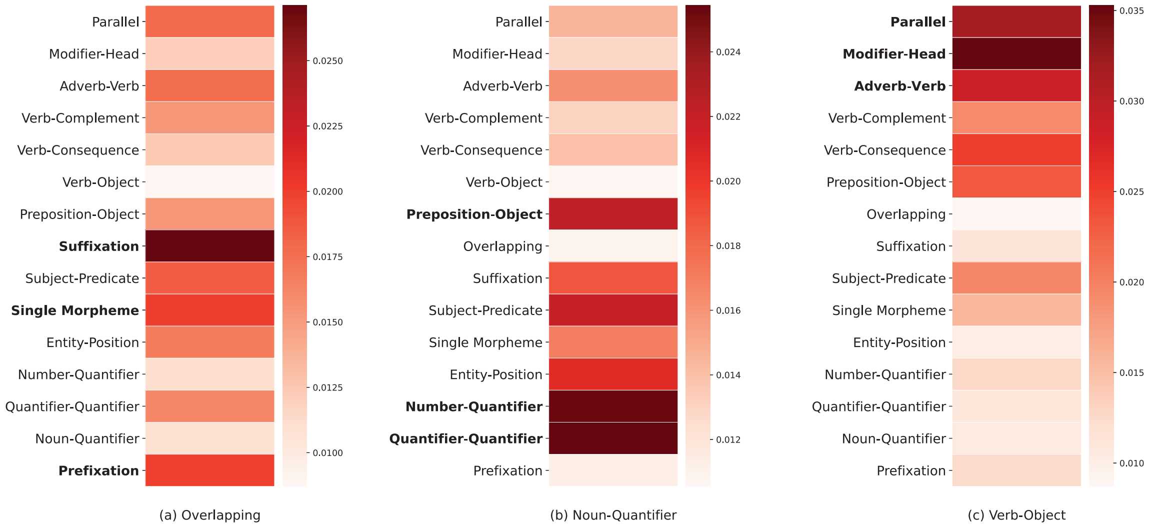

Figure 4: We take Verb-Object, Overlapping and Noun-Quantifier weight matrices as three examples and display their similarity with all the other 15 formation rules in the heatmap. The color goes deeper for more similar formation rules. The top 3 most similar formation rules are shown in **Bold**.

ent parts-of-speech. However, only the correct rule (Verb-Consequence) captures the **cause-and-effect** semantic aspect and outputs the correct definition. This reveals the impact of formation rules in the word formation process.