Explicit Semantic Decomposition for Definition Generation

Jiahuan Li [*] **Yu Bao** [*] **Shujian Huang**[†] **Xinyu Dai** **Jiajun Chen**
National Key Laboratory for Novel Software Technology, Nanjing University, China
{lijh,baoy}@smail.nju.edu.cn,
{huangsj,daixinyu,chenjj}@nju.edu.cn

Abstract

Definition generation, which aims to automatically generate dictionary definitions for words, has recently been proposed to assist the construction of dictionaries and help people understand unfamiliar texts. However, previous works hardly consider explicitly modeling the "components" of definitions, leading to under-specific generation results. In this paper, we propose *ESD*, namely **E**xplicit **S**emantic **D**ecomposition for definition generation, which explicitly decomposes meaning of words into semantic components, and models them with discrete latent variables for definition generation. Experimental results show that *ESD* achieves substantial improvements on WordNet and Oxford benchmarks over strong previous baselines.

1 Introduction

Dictionary definition, which provides explanatory sentences for word senses, plays an important role in natural language understanding for human. It is a common practice for human to consult a dictionary when encountering unfamiliar words (Fraser, 1999). However, it is often the case that we cannot find satisfying definitions for words that are rarely used or newly created. To assist dictionary compilation and help human readers understand unfamiliar texts, generating definitions automatically is of practical significance.

Noraset et al. (2017) first propose definition modeling, which is the task of generating the dictionary definition for a given word with its embedding. Gadetsky et al. (2018) extend the work by incorporating word sense disambiguation to generate context-aware word definitions.Both methods adopt a variant of encoder-decoder architecture, where the word to be defined is mapped to a low-dimension semantic vector by an encoder, and the decoder is responsible for generating the definition given the semantic vector.

Word	captain
Reference	the person in charge of a ship
Generated	the person who is a member of a ship

Table 1: An example of the definitions of word "captain". *Reference* is from Oxford dictionary and *Generated* is from the method of Ishiwatari et al. (2019).

Although the existing encoder-decoder architecture (Gadetsky et al., 2018; Ishiwatari et al., 2019; Washio et al., 2019) yields reasonable generation results, it relies heavily on the decoder to extract thorough semantic components of the word, leading to under-specific definition generation results, i.e. missing some semantic components. As illustrated in Table 1, to generate a precise definition of the word "captain", one needs to know that "captain" refers to *a person*, "captain" is related to *ship*, and "captain" *manages* or *is in charge of* the *ship*, where *person*, *ship*, *manage* are three semantic components of word "captain". However, due to the lack of explicitly modeling of these semantic components, the model misses the semantic component "manage" for the word "captain".

Linguists and lexicographers define a word by decomposing its meaning into its semantic components and expressing them in natural language sentences (Wierzbicka, 1996). Inspired by this, Yang et al. (2019) incorporate sememes (Bloomfield, 1949; Dong and Dong, 2003), i.e. minimum units of semantic meanings of human languages, in the task of generating definition in Chinese. However, it is just as, if not more, time-consuming and expensive to label the components of words than to write definitions manually.

In this paper, we propose to explicitly decom-

[*] Equal contribution
[†] Corresponding author

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 708–717
July 5 - 10, 2020. ©2020 Association for Computational Linguistics

pose the meaning of words into semantic components for definition generation. We introduce a group of discrete latent variables to model the underlying semantic components.Extending the established training technique for discrete latent variable used in representation learning (Roy et al., 2018) and machine translation tasks (van den Oord et al., 2017; Kaiser et al., 2018; Shu et al., 2019), we further propose two auxiliary losses to ensure that the introduced latent variables capture the word semantics. Experimental results show that our method achieves significant improvements over previous methods on two definition generation datasets. We also show that our model indeed learns meaningful and informative latent codes, and generates more precise and specific definitions.

2 Background

In this section, we introduce the background of the original definition modeling task and two extensive works to original definition modeling.

2.1 Definition Modeling

Definition generation was firstly proposed by Noraset et al. (2017). The goal of the original task is to generate a natural language description $\mathcal{D} = d_{1:T}$ for a given word w_*. The authors view it as a conditional language modeling task:

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

The main drawback of Noraset et al. (2017) is that they cannot handle words with multiple different meanings such as "spring" and "bank", whose meanings can only be disambiguated using their contexts.

2.2 Word Context for Definition Modeling

To tackle the polysemous problem in the definition generation task, Gadetsky et al. (2018) introduce the task of Context-aware **D**efinition **G**eneration (**CDG**), in which a usage example $\mathcal{C} = c_{1:|\mathcal{C}|}$ of the target word is given to help disambiguate the meaning of the word.

For example, given the word "bank" and its context "a *bank* account", the goal of the task is to generate a definition like "an organization that provides financial services". However, if the input context has been changed to "He jumped into the river and swam to the opposite *bank*.", then the appropriate definition would be "the side of a river".

They extend Eqn. 1 to make use of the given context as follows:

$$p(\mathcal{D}|w_*, \mathcal{C}) = \prod_{t=1}^{T} p(d_t|d_{i<t}, w_*, \mathcal{C}) \qquad (2)$$

2.3 Decomposed Semantic for Definition Modeling

Linguists consider the process of defining a word is to decompose its meaning into constituent components and describe them in natural language sentences (Goddard and Wierzbicka, 1994; Wierzbicka, 1996). Previously, Yang et al. (2019) take *sememes* as one kind of such semantic components, and leverage external sememe annotations HowNet (Dong and Dong, 2003) to help definition generation. They formalize the task of definition generation given a word w_* and its sememes $\boldsymbol{s}$ as follows:

$$p(\mathcal{D}|w_*, \boldsymbol{s}) = \prod_{t=1}^{T} p(d_t|d_{i<t}, w_*, \boldsymbol{s}) \qquad (3)$$

Although it is shown their method can generate definitions more accurately, they assume that annotations of sememes are available for each word, which can be unrealistic in real-world scenarios.

3 Approach

In this section, we present *ESD*, namely **E**xplicit **S**emantic **D**ecomposition for context-aware definition generation.

3.1 Modeling Semantic Components with Discrete Latent Variables

It is linguistically motivated that to define a word is to decompose its meaning into constituent components and describe them in natural language sentences (Goddard and Wierzbicka, 1994; Wierzbicka, 1996). We assume that there exists a set of discrete latent variables $\boldsymbol{z} = \boldsymbol{z}_{1:M}$ that model the semantic components of w_*, where M is the hyperparameter denoting the number of decomposed components. Then the marginal likelihood of a definition $\mathcal{D}$ that we would like to maximize given a target word w_* and its context $\mathcal{C}$ can be written as follows:

$$p_\theta(\mathcal{D}|w_*, \mathcal{C}) = \sum_{\boldsymbol{z}} p_\theta(\boldsymbol{z}|w_*, \mathcal{C}) p_\theta(\mathcal{D}|w_*, \mathcal{C}, \boldsymbol{z})$$

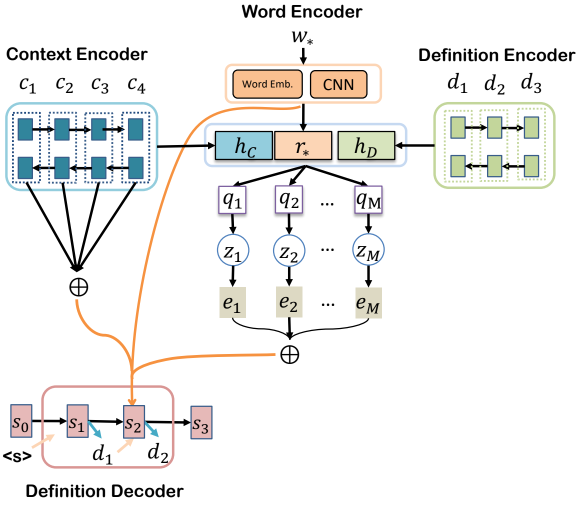

Figure 1: Neural architecture of *ESD*, including the word encoder, context encoder, the decoder and the definition encoder for the posterior networks.

However, it is generally computationally intractable to sum over all the configurations of latent variables. In order to address this issue, we instead introduce a approximate posterior $q_\phi(z|w_*, \mathcal{C}, \mathcal{D})$ and optimize the *evidence lower bound* (ELBO) of the log likelihood $\log p_\theta(\mathcal{D}|w_*, \mathcal{C})$ for training:

$$\begin{aligned}\mathcal{J}_{\text{ELBO}} = & \mathop{\mathbb{E}}_{q_\phi(\boldsymbol{z}|w_*,\mathcal{C},\mathcal{D})}\Big[\log p_\theta(\mathcal{D}|\boldsymbol{z}, w_*, \mathcal{C})\Big] \\ & - KL(q_\phi(\boldsymbol{z}|w_*, \mathcal{C}, \mathcal{D})||p_\theta(\boldsymbol{z}|w_*, \mathcal{C})) \\ & \leq \log p_\theta(\mathcal{D}|w_*, \mathcal{C})\end{aligned} \tag{4}$$

At the training phase, both posterior distribution $q_\phi(\boldsymbol{z}|w_*, \mathcal{C}, \mathcal{D})$ and prior distribution $p_\theta(\boldsymbol{z}|w_*, \mathcal{C})$ are computed and $\boldsymbol{z}$ is sampled from the posterior distribution.

At the testing phase, due to the lack of $\mathcal{D}$, we only compute the prior distribution $p_\theta(\boldsymbol{z}|w_*, \mathcal{C})$ and obtain $\boldsymbol{z}$ by applying $\arg\max$ to it.

Note that for the simplicity of notions, we denote $q_\phi(\boldsymbol{z}_i|w_*, \mathcal{C}, \mathcal{D})$ and $p_\theta(\boldsymbol{z}_i|w_*, \mathcal{C})$ as q_i and p_i in the following sections, respectively.

3.2 Model Architecture

As shown in Figure 1, *ESD* is composed of three modules: the encoder stack, a decoder, and a semantic components predictor. Before detailing each component of *ESD*, we overview the architecture for a brief understanding.

Following the common practice of context-aware definition models (Gadetsky et al., 2018; Ishiwatari et al., 2019), we first encode the source word w_* into its representation $\boldsymbol{r}_*$ and context C=$c_{1:|C|}$ into its contextual representation H=$h_{1:|C|}$. The semantic component predictor is responsible for predicting the semantic components $\boldsymbol{z}$=$z_{1:M}$. Finally, the decoder generates the target definition from the semantic components $\boldsymbol{z}$, the word representation r_* and the context representation H.

3.2.1 Encoder

Same as Ishiwatari et al. (2019), our encoder consists of two parts, namely word encoder and context encoder.

Word Encoder The word encoder is responsible for mapping the word w_* to a low-dimensional vector $\boldsymbol{r}_*$, and consists of a word embedding and a character level encoder. The word embedding is initialized by large-scale pretrained word embeddings such as *GloVe* (Pennington et al., 2014) or *FastText* (Bojanowski et al., 2017), and is kept fixed at the training time. Previous works (Noraset et al., 2017; Ishiwatari et al., 2019) also show that morphological information can be helpful for definition generation. We employ a convolutional neural network (Krizhevsky et al., 2012) to encode the character sequence of the word. We concatenate the word embedding and the character encoding to get the word representation r_*.

Context Encoder We adopt a standard bi-directional LSTM network (Sundermeyer et al., 2012) to encode the context, which takes word embedding sequence of the context $\mathcal{C}$=$c_{1:|\mathcal{C}|}$ and outputs a hidden state sequence H=$h_{1:|\mathcal{C}|}$.

3.2.2 Semantic Components Predictor

For the proposed *ESD*, we need to model both the semantic components posterior $q_\phi(\boldsymbol{z}|w_*, \mathcal{C}, \mathcal{D})$ and the prior $p_\theta(\boldsymbol{z}|w_*, \mathcal{C})$.

Semantic Components Posterior Approximator Exactly modeling the true posterior $q_\phi(\boldsymbol{z}|w_*, \mathcal{C}, \mathcal{D})$ is usually intractable. Therefore, we adopt an approximation method to simplify the posterior inference (Zhang et al., 2016) Following the spirit of VAE (Bowman et al., 2016), we use neural networks for better approximation in this paper.

Specifically, we first compute the representation $H_\mathcal{D}$=$h'_{1:T}$ of the definition $\mathcal{D} = d_{1:T}$ with a bi-directional LSTM network. We then obtain the representation of definition $\mathcal{D}$ and context $\mathcal{C}$ with *max-pooling* operation.

$$h_{\mathcal{D}} = \textit{max-pooling}(h'_{1:T}) \quad (5)$$

$$h_{\mathcal{C}} = \textit{max-pooling}(h_{1:|\mathcal{C}|}) \quad (6)$$

With these representations, as well the word representation r_*, we compute the posterior approximation q_i of $\boldsymbol{z}_i$ as follows:

$$q_i = \text{softmax}(W_i^q[r_*, h_{\mathcal{C}}, h_{\mathcal{D}}] + b_i^q)$$

where the W_i^q and b_i^q are the parameters of the semantic components posterior approximator.

Semantic Components Prior Model Similar to the posterior, we model the prior p_i of $\boldsymbol{z}_i$ by a neural network with the representation $h_{\mathcal{C}}$ (computed by Eqn 6) and r_* as follows:

$$p_i = \text{softmax}(W_i^p[r_*, h_{\mathcal{C}}] + b_i^p)$$

where the W_i^p and b_i^p are the parameters of the semantic components prior.

3.2.3 Definition Decoder

Given the word w_*, the context $\mathcal{C}$ and the semantic component latent variables $\boldsymbol{z}$, our decoder adopt a LSTM to model the probability of generating definition $\mathcal{D}$ given word w_*, context $\mathcal{C}$, and semantic components $\boldsymbol{z}$:

$$p(\mathcal{D}|w_*, \mathcal{C}, \boldsymbol{z}) = \prod_{t=1}^{T} p(d_t|d_{<t}, w_*, \mathcal{C}, \boldsymbol{z}) \quad (7)$$

At each decoding time step, we first obtain the context vector c_t as follows:

$$\alpha_{ti} = \frac{\exp(s_t^T h_i)}{\sum_{j=1}^{|\mathcal{C}|} \exp(s_t^T h_j)}$$

$$c_t = \sum_{i}^{|\mathcal{C}|} \alpha_{ti} h_i$$

Moreover, it is intuitive that at different time steps the decoder is describing different semantic perspectives of the word, thus needing different semantic components (Yang et al., 2019). We embed each z_i using a latent embedding matrix $\mathrm{E}_i \in \mathrm{R}^{K \times D}$ and get M semantic component vectors $\{E_1(z_1), E_2(z_2), \cdots, E_M(z_M)\}$. We then apply an attention mechanism over the semantic component vectors and obtain a semantic context vector o_t :

$$\beta_{ti} = \frac{\exp(s_t^T E_i(z_i))}{\sum_{j=1}^{M} \exp(s_t^T E_i(z_i))}$$

$$o_t = \sum_{i}^{M} \beta_{ti} E_i(z_i)$$

Finally, we adopt a GRU-like (Cho et al., 2014) gate mechanism to allow the decoder to dynamically fuse information from the word representation $\boldsymbol{r}_*$, context vector $\boldsymbol{c}_t$, and semantic context vector $\boldsymbol{o}_t$, which can be calculated as follows:

$$\begin{aligned}
\boldsymbol{f}_t &= [\boldsymbol{r}_*; \boldsymbol{c}_t; \boldsymbol{o}_t] \\
\boldsymbol{u}_t &= \sigma(\mathbf{W}_u[\boldsymbol{f}_t; \boldsymbol{s}_t] + \boldsymbol{b}_u) \\
\boldsymbol{v}_t &= \sigma(\mathbf{W}_r[\boldsymbol{f}_t; \boldsymbol{s}_t] + \boldsymbol{b}_r) \\
\hat{\boldsymbol{s}}_t &= \tanh(\mathbf{W}_s[(\boldsymbol{v}_t \odot \boldsymbol{f}_t]; \boldsymbol{s}_t] + \boldsymbol{b}_s) \\
\boldsymbol{s}'_t &= (1 - \boldsymbol{u}_t) \odot \boldsymbol{s}_t + \boldsymbol{u}_t \odot \hat{\boldsymbol{s}}_t
\end{aligned}$$

where, $\boldsymbol{W}_*$ and $\boldsymbol{b}_*$ are weight matrices and bias terms, respectively.

3.3 Learning

The loss function in Eqn. 4 serves as our primary training objective. Besides, since the latent variables are designed to model the semantic components, we propose two auxiliary losses to ensure that these latent variables can learn informative codes and capture the decomposed semantics.

Semantic Completeness Objective In order to generate accurate definitions, the introduces latent variables must capture all perspectives of the word semantics. For example, it is impossible to precisely define the word "captain" in the context "The captain gave the order to abandon the ship" without knowing that (1) a captain is a person, (2) a captain works in a ship, and (3) a captain usually is in charge of a ship. Therefore, an ideal $\boldsymbol{z}$ should contain sufficient information for predicting the definition.

We first propose to leverage sememe annotations of HowNet (Dong and Dong, 2003) as an external signal to guide the learning of latent variables. As we mentioned in Section 2.3, sememes are also known to be helpful for definition generation (Yang et al., 2019). Previously, Xie et al. (2017) show that it is possible to predict sememes of words from large scale pretrained distributional representations.

Suppose the set of sememes in HowNet are denoted by $\mathcal{S} = \{s_1, s_2, \cdots, s_n\}$, and each word w in HowNet is annotated by a small subset of $\mathcal{S}$, denoted by $\mathcal{S}_w = \{s_i | s_i \in \mathcal{S}\}$. Inspired by Weng et al. (2017), we adopt a bag-of-word loss to ensure that $\boldsymbol{z}$ is informative enough to be predictive about sememe annotations S_w:

$$\mathcal{L}_{\text{com}}^{(\text{sem})} = -\log \sum_{s_i \in \mathcal{S}_w} p(s_i|\boldsymbol{z}) \quad (8)$$

Our next motivation is that the sememes annotation is still expensive, while definitions of words are off-the-shelf when training. Inspired by Bao et al. (2019) and John et al. (2019), we enforce the model to predict every words in the target definition $\mathcal{D}=d_{1:T}$ to ensure that $\boldsymbol{z}$ is informative enough:

$$\mathcal{L}_{\text{com}}^{(\text{def})} = -\log \sum_{i=1}^{T} p(d_i|\boldsymbol{z}) \tag{9}$$

Semantic Diversity Objective To achieve the goal of decomposing semantics, it is crucial that there are several different latent variables that separately model different semantic components. In order to prevent that multiple latent variables degenerate to one, we encourage the semantic vectors to be dissimilar from each other by introducing a disagreement loss:

$$\mathcal{L}_{\text{div}} = - \sum_{1 \le i < j \le M} dist(E_i(z_i), E_j(z_j)) \tag{10}$$

where, $dist(\cdot,\cdot)$ is a distance function between two distributions. We adopt cosine distance as the distance function in this paper.

Overall Objectives With the different overall training loss used, there are two variants of *ESD*. The original loss of *ESD* is

$$\mathcal{L}_{\text{base}} = -\mathcal{J}_{\text{ELBO}}$$

The first variant of *ESD* (denoted by *ESD*-def) includes the optimization of semantic completeness and semantic diversity, which is optimized with:

$$\mathcal{L}_{ESD\text{-def}} = \mathcal{L}_{\text{base}} + \alpha \mathcal{L}_{\text{com}}^{(\text{def})} + \beta \mathcal{L}_{\text{div}}$$

Grounding on the annotated sememes, the second variant of *ESD* (denoted by *ESD*-sem) is optimized with:

$$\mathcal{L}_{ESD\text{-sem}} = \mathcal{L}_{\text{base}} + \alpha \mathcal{L}_{\text{com}}^{(\text{sem})} + \beta \mathcal{L}_{\text{div}}$$

4 Experiments

4.1 Experimental Setting

Datasets To demonstrate the effectiveness of our method, we conduct experiments on two datasets used in previous work (Ishiwatari et al., 2019): WordNet [1] and Oxford [2]. Each entry in the datasets is a triple of a word, a piece of its usage example, and its corresponding dictionary definition.

Sememe Annotation Resources Following previous work (Yang et al., 2019), we take HowNet as the sememe annotation resource, which is an ontology that contains annotations for over 100,000 words with sememes. Each word in HowNet may have several senses, and each sense is annotated with several sememes explaining the meaning of it.

Hyperparameters We adopt a two-layer LSTM network as our context encoder and definition decoder. We set the hidden dim to 300. Following Ishiwatari et al. (2019), we set the CNN kernel for character encoder of length $2, 3, 4, 5, 6$ and size $10, 30, 40, 40, 40$ respectively with a stride of 1. The dimension of the final character level encoding is 160. We set the number of latent variables M and the number of categories K to 8 and 256, respectively.

Optimization We adopt Adam (Kingma and Ba, 2014) to optimize our model. The learning rate is set to 0.001. The α and β we used in the overall objective are set to 1.0 and 0.1, respectively. All hyperparameters are chosen based on the performance on the validation set and are used across all the experiments.

Competitors We compare our model with several baseline models:

1. **I-Attention** (Gadetsky et al., 2018) uses the context to disambiguate the word embedding and cannot utilize context information at the decoding time.

2. **LOG-CaD** (Ishiwatari et al., 2019) is similar to our architecture, without modeling the semantic component.

3. **Pip-sem** is our intuitive pipeline that consists of a sememe predictor and a definition generator. The sememe predictor is trained on HowNet and is responsible for annotating words in definition generation datasets. The definition generator is used to generate definitions given the word, context, and pseudo annotations of sememes.

Metrics We adopt two several automatic metrics that are often used in generation tasks: BLEU (Papineni et al., 2002) and Meteor (Denkowski and Lavie, 2014). BLEU considers the exact match between generation results and references and is the most common metric used to evaluate generation systems. Following previous work, we compute

[1] https://wordnet.princeton.edu/

[2] https://en.oxforddictionaries.com/

Model	WordNet		Oxford	
	BLEU	METEOR	BLEU	METEOR
I-Attention (Gadetsky et al., 2018)	23.77	/	17.25	/
LOG-CaD (Ishiwatari et al., 2019)	24.79	/	18.53	/
*LOG-CaD	24.70	8.66	18.24	8.43
†Pip-sem	25.52	11.33	19.89	11.10
ESD-def	25.75	11.52	19.98	10.79
†*ESD*-sem	**26.48**	**12.45**	**20.86**	**11.86**

Table 2: BLEU and Meteor scores on WordNet and Oxford dataset. '$\dagger$' indicates models that incorporate external sememe annotations while training. '*' denotes our reimplementation of previous model.

Model	Fluency	Semantic Completeness
LOG-CaD	3.53	3.01
ESD-def	3.55	3.45

Table 3: Human annotated scores on Oxford dataset.

the sentence level BLEU score. We also consider Meteor (Denkowski and Lavie, 2014), a metric that takes synonyms, stemming, and paraphrases into consideration while calculating the score. Meteor score is said to favor word choices than word orders and favor recall over precision (Denkowski and Lavie, 2014). We use the recommended hyperparameters to compute Meteor scores.

4.2 Automatic Evaluation

The results, as measured by the automatic evaluation metrics, i.e. BLEU and Meteor, are presented in Table2.

***ESD* significantly improves the quality of definition generation with a large margin.** On all the benchmark datasets, our *ESD* that incorporates sememes achieves the best generation performance, both in BLEU and Meteor scores. It is worth noting that the improvement of the Meteor score is more significant than the BLEU score, i.e. 3.79 vs. 1.78 on WordNet, and 3.43 vs. 2.62 on Oxford, indicating that our model is better at recalling semantically correct words, which is consistent with our motivation to address the under-specific problem.

Decomposing semantics is indeed helpful to definition modeling. The models that generate definition with the explicit decomposed semantics (Pip-sem, *ESD*-def and *ESD*-sem) leads to remarkable improvements over the competitor without decomposed component modeling (I-Attention and LOG-CaD). The comparison between the *ESD*-def, I-Attention and LOG-CaD is fair because all of them do not have the external sememe annotation during training and testing. Notably, *ESD*-sem also improves over Pip-sem by a large margin. This shows that the way our method leverages the sememe annotations, i.e. using them as external signals of word semantics, is more effective than simple annotate-then-generate pipeline methods.

4.3 Human Evaluation

In order to further compare the proposed methods and the strongest previous method (i.e., the Log-CaD model), we performed a human evaluation of the generated definitions. We randomly selected 100 samples from the test set of Oxford dataset, and invited four people with at least CET6 level English skills to rate the output definitions in terms of fluency and semantic completeness from 1 to 5 points. The averaged scores are presented in Table3. As can be seen from the table, definitions generated by our methods are rated higher in terms of semantic completeness while achieving comparable fluency.

4.4 Ablation Study

We also perform an ablation study to quantify the effect of different model components.

Semantic completeness objective We can see that the semantic completeness objective, i.e. $\mathcal{L}_{com}^{(*)}$ leads to a substantial improvement in terms of Meteor score (Line 3 and Line 4 vs. Line 1), which indicates that the gain obtained by our model is not by trivially adopting the conditional VAE framework to definition generation task.

Semantic diversity objective The experimental results show that although independently using the semantic diversity objective leads to no gains (Line 2 vs. Line 1), regularizing the model to learn diverse latent codes when using semantic completeness objective can improve the generation perfor-

	$\mathcal{L}_{\text{base}}$	$\mathcal{L}_{\text{div}}$	$\mathcal{L}_{\text{com}}^{(\text{def})}$	$\mathcal{L}_{\text{com}}^{(\text{sem})}$	Meteor
1	✓				8.99
2	✓	✓			9.15
3	✓		✓		11.09
4	✓			✓	11.88
5	✓	✓	✓		11.56
6	✓	✓		✓	12.43
7	✓	✓	✓	✓	12.87

Table 4: Ablation study on the development set of Oxford dataset.

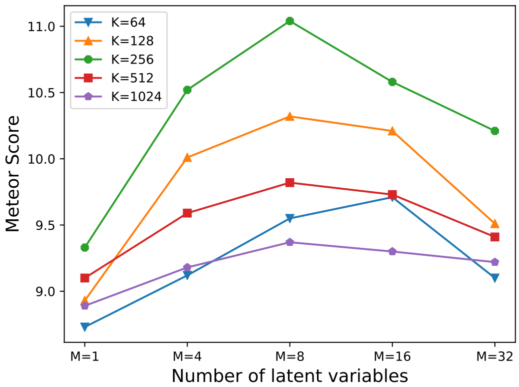

Figure 2: The Meteor scores of *ESD* on Oxford test dataset with different M and K, where M is the number of discrete latent variables used in *ESD*, and K is the number of categories.

mance of the model (Line 5 vs. Line 3 and Line 6 vs. Line 4).

5 Analysis

To gain more insight into the improvement provided by the proposed method, we perform several analyses in this section.

5.1 Influence of the number of components

To validate that explicit decomposition of word semantics is beneficial for definition generation, we compare the performances of several models with different number of latent variables, and plot the result in Figure 2.

Overall, setting multiple latent variables given the same categories achieves noticeable improvements over M=1, i.e. encoder-decoder model with word prediction mechanism. However, it is not the case we should adopt as many latent variables as possible. The reason for it is that generally a word has a limited number of semantic components (3-10 in HowNet), and having too many components in the latent models would damage the performance.

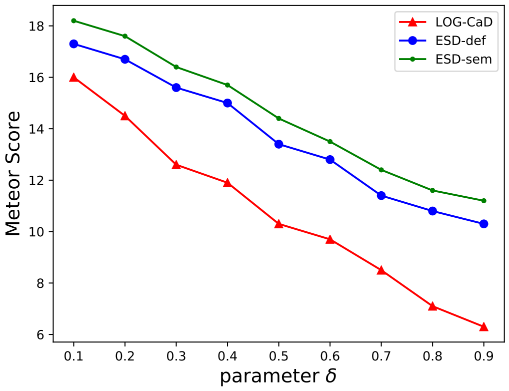

Figure 3: Comparison between LOG-CaD and *ESD*-def with different parameter δ. δ controls how much we prefer content words over function words. Larger δ implies we prefer content words more.

It is interesting to see that when we set the number of components M to 8, the optimal number of categories K is 256. As the total number of semantic units we are modeling is $M \times K$, this approximately equals to the number of sememes in HowNet.

5.2 Improvements on different word types

The goal of definition generation task is to accelerate dictionary compilation or to help humans with unfamiliar text. In both application scenarios, it is more important to generate content words that describe the semantic of the given word, rather than function words or phrases such as “refer to” and “of or relating to”. To understand which kind of word our model achieves the largest improvements on, we evaluate Meteor scores of the baseline model and our model under different values of δ, where δ is a hyperparameter used by Meteor that controls how much we prefer content words over function words. Figure 3 shows the results. We can see that as our preference over content words increases, both the performances of baseline model and our model decreases, indicating that it is more difficult for current definition generation models to generate useful content words than function words. However, the gap between the baseline model and our model becomes larger when δ increases, which shows that the gain of our model is mainly from the content words instead of function words.

Word	militia
Context	The militia repelled attacks from without and denied the executive the means to oppress from within.
Reference	a group of people who are not professional soldiers but who have had military training and can act as an army
LOG-CaD	a group of people engaged in a military force
ESD-def	a group of people engaged in a military force and not very skillful
Word	captain
Context	The captain gave the order to abandon ship
Reference	the person in charge of a ship
LOG-CaD	a person who is a member of ship
ESD-def	a person who is the leader of a ship

Table 5: Examples from LOG-CaD and *ESD*-def. We highlight the different part between two models in red.

word	z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_8
red	54	7B	9C	60	A1	A7	F5	C7
yellow	54	92	7F	22	A1	A7	F5	55
blue	6A	E5	7F	22	A1	A7	F5	C7
cat	7A	E3	C4	22	A1	A7	F5	3B
dog	7A	43	C4	60	A1	A7	F5	3B
penguin	7A	C3	C4	60	A1	BE	F5	3B

Table 6: Examples of the learned latent codes. Each line is a word with the hexadecimal identifier of its corresponding latent codes. Color words like "red", "yellow", "blue" share most parts of latent codes with each other, while words from different groups like "red" and "cat" share fewer parts of latent codes.

5.3 Case Studies

Examples of learned latent codes In Table 6, we show some examples of learned latent codes on WordNet dataset. We can see that our model does learn informative codes, i.e. words with similar meanings are assigned with similar latent codes, and codes of words with different meanings tend to differ.

Examples of generated definitions We also list several generation samples in Table 5. We can see that the definitions generated by our method are more semantically complete than those by previous works, and they indeed capture fine-grained semantic components that the baseline model ignores. For example, it is necessary to know that *militia* has unprofessional military skills, which distinguishes the meaning of *militia* and *army*. The definition generated by the baseline model ignores this perspective. However, our model does describe the *unprofessional* nature of *militia* by generating "not very skillful", thanks to the ability of modeling fine-grained semantic components.

6 Related Work

Definition Generation Definition modeling was firstly proposed by Noraset et al. (2017). They take a word embedding as input and generate a definition of the word. An obvious drawback is that their model cannot handle polysemous words. Recently several works (Ni and Wang, 2017; Gadetsky et al., 2018; Ishiwatari et al., 2019) consider the context-aware definition generation task, where the context is introduced to disambiguate senses of words. They all adopt a encoder-decoder architecture, and rely heavily on the decoder to extract semantic components of the word semantic, thus leading to under-specific definitions. In contrast, we introduce a group of discrete latent variables to model these semantic components explicitly.

Semantic decomposition and Decomposed Semantics It is recognized by linguists that human beings understand complex meaning by decomposing it into components that are latent in the meaning. Wierzbicka (1996) propose that different languages share a set of atomic concepts that cannot be further decomposed i.e. *semantic primitives*, and all complex concepts can be semantically composed by semantic primitives. Dong and Dong (2003) introduce a similar idea. They call the atomic concepts as *sememes*, and present a knowledge base HowNet in which senses of words are annotated with sememes. HowNet is shown to be helpful for many NLP tasks, such as word representation learning (Niu et al., 2017), relation extraction (Li et al., 2019), aspect extraction (Luo et al., 2019). Previously Yang et al. (2019) propose to use sememe annotations as a direct input when generating definitions, which can suffer from the data sparsity problem. In this paper, we instead leverage HowNet as the external supervising signals for latent variables when training and try to learn the knowledge into the model itself.

7 Conclusion

We proposed *ESD*, a context-aware definition generation model that explicitly models the decomposed semantics of words. Specifically, we model the decomposed semantics as discrete latent variables, and training with auxiliary losses to ensure that the model learns informative latent codes for definition modeling. As a result, *ESD* leads to significant improvements over the previous strong baselines on two established definition datasets. Quantitative and qualitative analysis showed that our model could generate more meaningful, specific and accurate definitions.

In future work, we plan to seek better ways to guide the learning of latent variables, such as using dynamic routing (Sabour et al., 2017) method to align the latent variables and sememes, and learn more explainable latent codes.

Acknowledgments

We would like to thank the anonymous reviewers for their insightful comments. Shujian Huang is the corresponding author. This work is supported by National Science Foundation of China (No. U1836221, 61772261), the Jiangsu Provincial Research Foundation for Basic Research (No. BK20170074).

References

Yu Bao, Hao Zhou, Shujian Huang, Lei Li, Lili Mou, Olga Vechtomova, Xinyu Dai, and Jiajun Chen. 2019. Generating sentences from disentangled syntactic and semantic spaces. In *ACL*, pages 6008–6019.

Leonard Bloomfield. 1949. A set of postulates for the science of language. *IJAL*, 15(4):195–202.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *TACL*, 5:135–146.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *CoNLL*, pages 10–21.

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

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.

Zhendong Dong and Qiang Dong. 2003. HowNet - a hybrid language and knowledge resource. In *NLPKE*, pages 820–824.

Carol Fraser. 1999. The role of consulting a dictionary in reading and vocabulary learning. *Canadian Journal of Applied Linguistics*, 2:73–89.

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

Cliff Goddard and Anna Wierzbicka. 1994. *Semantic and Lexical Universals: Theory and Empirical Findings*, volume 25. John Benjamins Publishing.

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*, pages 3467–3476.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. Disentangled representation learning for text style transfer. In *ACL*.

Lukasz Kaiser, Samy Bengio, Aurko Roy, Ashish Vaswani, Niki Parmar, Jakob Uszkoreit, and Noam Shazeer. 2018. Fast decoding in sequence models using discrete latent variables. In *ICML*, pages 2395–2404.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105.

Ziran Li, Ning Ding, Zhiyuan Liu, Haitao Zheng, and Ying Shen. 2019. Chinese relation extraction with multi-grained information and external linguistic knowledge. In *ACL*, pages 4377–4386.

Ling Luo, Xiang Ao, Yan Song, Jinyao Li, Xiaopeng Yang, Qing He, and Dong Yu. 2019. Unsupervised neural aspect extraction with sememes. In *IJCAI*, pages 5123–5129.

Ke Ni and William Yang Wang. 2017. Learning to explain non-standard english words and phrases. *CoRR*, abs/1709.09254.

Yilin Niu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2017. Improved word representation learning with sememes. In *ACL*, pages 2049–2058.

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

Aaron van den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pages 6306–6315.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

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

Aurko Roy, Ashish Vaswani, Niki Parmar, and Arvind Neelakantan. 2018. Towards a better understanding of vector quantized autoencoders. *arXiv*.

Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic routing between capsules. *CoRR*, abs/1710.09829.

Raphael Shu, Jason Lee, Hideki Nakayama, and Kyunghyun Cho. 2019. Latent-variable non-autoregressive neural machine translation with deterministic inference using a delta posterior. In *AAAI*.

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. Lstm neural networks for language modeling. In *InterSpeech*.

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*, pages 3519–3525.

Rongxiang Weng, Shujian Huang, Zaixiang Zheng, XIN-YU DAI, and CHEN Jiajun. 2017. Neural machine translation with word predictions. In *EMNLP*, pages 136–145.

Anna Wierzbicka. 1996. *Semantics: Primes and Universals*. Oxford University Press.

Ruobing Xie, Xingchi Yuan, Zhiyuan Liu, and Maosong Sun. 2017. Lexical sememe prediction via word embeddings and matrix factorization. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4200–4206. AAAI Press.

Liner Yang, Cunliang Kong, Yun Chen, Yang Liu, Qinan Fan, and Erhong Yang. 2019. Incorporating sememes into chinese definition modeling. *arXiv preprint arXiv:1905.06512*.

Biao Zhang, Deyi Xiong, Jinsong Su, Hong Duan, and Min Zhang. 2016. Variational neural machine translation. In *EMNLP*.