Understanding Jargon: Combining Extraction and Generation for Definition Modeling

Jie Huang[1] **Hanyin Shao**[1] **Kevin Chen-Chuan Chang**[1]
Jinjun Xiong[2] **Wen-mei Hwu**[1,3]
[1]University of Illinois at Urbana-Champaign, USA
[2]University at Buffalo, USA
[3]NVIDIA, USA
{jeffhj, hanyins2, kcchang, w-hwu}@illinois.edu
jinjun@buffalo.edu

Abstract

Can machines know what *twin prime* is? From the composition of this phrase, machines may guess *twin prime* is a certain kind of prime, but it is still difficult to deduce exactly what *twin* stands for without additional knowledge. Here, *twin prime* is a jargon– a specialized term used by experts in a particular field. Explaining jargon is challenging since it usually requires domain knowledge to understand. Recently, there is an increasing interest in extracting and generating definitions of words automatically. However, existing approaches, either extraction or generation, perform poorly on jargon. In this paper, we propose to combine extraction and generation for jargon definition modeling: first extract self- and correlative definitional information of target jargon from the Web and then generate the final definitions by incorporating the extracted definitional information. Our framework is remarkably simple but effective: experiments demonstrate our method can generate high-quality definitions for jargon and outperform state-of-the-art models significantly, e.g., BLEU score from 8.76 to 22.66 and human-annotated score from 2.34 to 4.04.[1]

1 Introduction

Jargons are specialized terms associated with a particular discipline or field. To understand jargons, a straightforward approach is to read their definitions, which are highly summarized sentences that capture the main characteristics of them. For instance, given jargon *twin prime*, people can know its meaning by reading its definition: "*A twin prime is a prime number that is either 2 less or 2 more than another prime number.*"

Recently, acquiring definitions of words/phrases automatically has aroused increasing interest. There are two main approaches: *extractive*, corresponding to *definition extraction*, where definitions are extracted from existing corpora automatically (Anke and Schockaert, 2018; Veyseh et al., 2020; Kang et al., 2020); and *abstractive*, corresponding to *definition generation*, where definitions are generated conditioned with the target words/phrases and the contexts in which they are used (Noraset et al., 2017; Gadetsky et al., 2018; Bevilacqua et al., 2020; August et al., 2022; Gardner et al., 2022).

In this paper, we study **jargon definition modeling**, which aims to acquire definitions for jargon automatically. Jargon definition modeling is important since definitions of jargon are less likely to be organized in an existing dictionary/encyclopedia and such terms are difficult for non-experts to understand without explanations (Bullock et al., 2019). This is particularly true for new jargon from fast-advancing fields. For instance, neither Oxford dictionary (Butterfield et al., 2016) nor Wikipedia[2] includes *few-shot learning*– an important setup in machine learning.

However, to acquire definitions for jargon, both extractive and abstractive approaches may fail. Extracting high-quality definitions would be difficult due to the incompleteness and low quality of data sources (this issue is more serious for jargon since jargon is usually less frequently used than general words/phrases). E.g., a good definition may not be available in the corpus; even if it existed, it might be difficult to select from a large set of candidate sentences (Kang et al., 2020). Generating definitions for jargon would be challenging since jargons are usually technical terms that need domain knowledge to understand, while the contexts in which they are used cannot provide sufficient knowledge. For instance, it is almost impossible for a model to generate the definition for *twin prime* only with context "*proof of this conjecture would also imply the existence of an infinite number of twin primes*" since the context does not explain *twin prime*, and the specific meaning is difficult

[1]Code and data are available at https://github.com/jeffhj/CDM.

[2]https://en.wikipedia.org

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 3994 - 4004

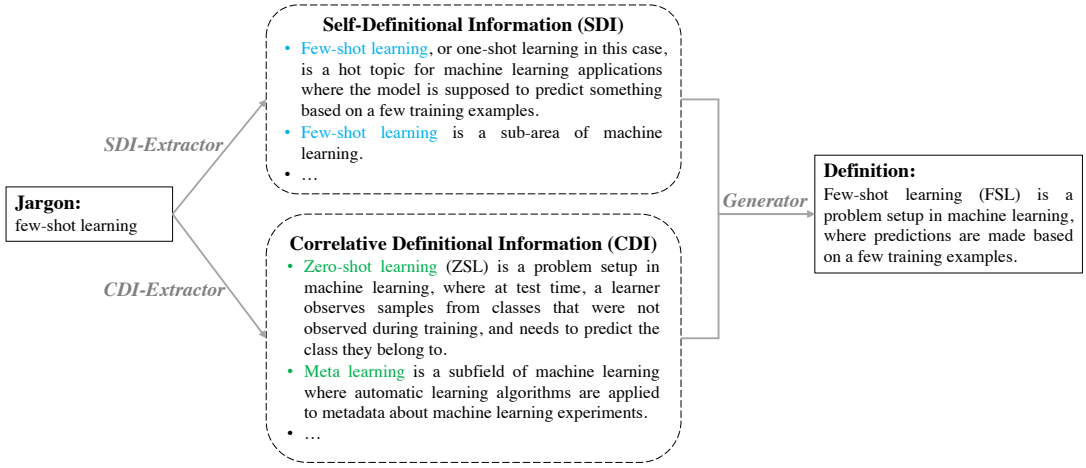

Figure 1: The overview of the proposed framework. In this example, the definition of *few-shot learning* is generated based on both the SDI (e.g., "*predictions are made based on a few training examples*") and the CDI (e.g., "*is a problem setup in machine learning*").

to infer from the surface form, leading to *hallucinations*, i.e., generating irrelevant or contradicted facts (Bevilacqua et al., 2020). Consequently, existing models designed for general words/phrases perform poorly on jargon. In our evaluation (Tables 5 and 6), we find most definitions produced by the state-of-the-art model contain wrong information.

Fortunately, definition extraction and definition generation can complement each other naturally. On one hand, definition generator has the potential to help the extractor by refining and synthesizing the extracted definitions; therefore, the extracted sentences are not required to be perfect definitions of the target jargon. On the other hand, definition extractor can retrieve useful definitional information as knowledge for the generator to produce definitions of jargon. However, surprisingly, existing works are either extractive or abstractive, even do not connect and compare them.

Therefore, in this work, we propose to combine definition extraction and definition generation for jargon definition modeling. We achieve this by introducing a framework consisting of two processes: *extraction*, where definitional information of jargon is extracted from the Web; and *generation*, where the final definition is generated with the help of the extracted definitional information.

We build models for extraction and generation based on Pre-Trained Language Models (Devlin et al., 2019; Lewis et al., 2020a; Brown et al., 2020). Specifically, for extraction, we propose a BERT-based definition extractor to extract *self-definitional information* (i.e., definitional sentences of the target jargon). We also suggest that related terms can help define the target jargon and leverage Wikipedia as the external knowledge source to retrieve *correlative definitional information* (i.e., definitions of related terms). For generation, we design a BART-based definition generator to produce the final definition by incorporating the extracted knowledge. An example is shown in Figure 1.

Our framework for jargon definition modeling is remarkably simple that can easily be further expanded by leveraging more advanced language models, e.g., we can replace the BART generator with larger models such as Meta OPT (Zhang et al., 2022) with a simple modification. Besides, since our framework does not require a domain-specific corpus or ontology like the ones used in Vanetik et al. (2020); Liu et al. (2021), it is easy to apply to a variety of domains. Experimental results on four datasets demonstrate our ***simple*** model outperforms state-of-the-art models ***significantly*** (e.g., BLEU score from 8.76 to 22.66, human-annotated score from 2.34 to 4.04).

Our contributions are summarized as follows:

- We report the first attempt to connect and combine definition extraction and definition generation.
- We introduce jargon definition modeling and solve it by incorporating both self- and correlative definitional information of jargon.
- Experimental results show that our simple model substantially outperforms SOTA models for definition modeling.
- We publish several datasets, along with definitions (e.g., of `~75,600` computer science terms) generated by our proposed model.

2 Related Work

Definition Extraction. Definition extraction, which aims to extract definitions from corpus automatically, has been studied for a long period. Existing works for definition extraction can be roughly divided into three categories: 1) *rule-based*, which extracts definitions with defined linguistic rules and templates (Klavans and Muresan, 2001; Cui et al., 2004; Fahmi and Bouma, 2006); 2) *machine learning-based*, which extracts definitions by statistical machine learning with carefully designed features (Westerhout, 2009; Jin et al., 2013); 3) *deep learning-based*, the state-of-the-art approach for definition extraction, which is based on deep learning models such as CNN, LSTM, and BERT (Anke and Schockaert, 2018; Veyseh et al., 2020; Kang et al., 2020; Vanetik et al., 2020).

Definition Generation. Definition generation, or definition modeling, has aroused increasing interest in recent years. The first study on definition generation was presented in Noraset et al. (2017), which aims to generate definitions of words with word embeddings. Later works on definition generation put more emphasis on generating definitions of words/phrases with given contexts (Gadetsky et al., 2018; Ishiwatari et al., 2019; Washio et al., 2019; Mickus et al., 2019; Li et al., 2020; Reid et al., 2020; Bevilacqua et al., 2020; Huang et al., 2021a). For example, Bevilacqua et al. (2020) apply pre-trained BART (Lewis et al., 2020a) for definition generation with a simple context encoding scheme. Huang et al. (2021a) employ three T5 models (Raffel et al., 2020) for definition generation with a re-ranking mechanism to model specificity of definitions. Liu et al. (2021) study the graph-aware definition modeling problem by incorporating biomedical ontology. August et al. (2022) study the problem of generating definitions of scientific and medical terms with varying complexity. Huang et al. (2022) propose to generate definitional-like sentences to describe relations between entities. There are also recent works on definition modeling for other languages, e.g., Chinese, by incorporating the special properties of the specific language (Yang et al., 2020; Zheng et al., 2021).

However, although definition extraction and definition generation are quite relevant tasks, surprisingly, existing works do not connect and compare them. In this work, we report the first attempt to combine them.

3 Methodology

Our framework for jargon definition modeling consists of two processes: *extraction*, which extracts self- and correlative definitional information of the target jargon from the Web; and *generation*, which generates the final definition by incorporating the extracted definitional information. The overview of the framework is shown in Figure 1.

3.1 Extraction

3.1.1 Self-Definitional Information

Since jargons are specialized terms used in a particular field, to understand jargon, we need background knowledge of jargon. To acquire useful information for defining jargon, it is natural to refer to definitional sentences containing the target jargon, named ***Self-Definitional Information (SDI)***. We achieve SDI by first extracting sentences containing the target jargon from the Web (more details are in Section 4.1) and then using a classifier to rank the extracted sentences.

To build the classifier, we apply the BERT model (Devlin et al., 2019), which has achieved excellent results on various text classification tasks. We adopt a simple encoding scheme, which is "*[CLS]* $jargon$ *[DEF]* $sentence$", e.g., "*[CLS] machine learning [DEF] machine learning is the study of computer algorithms that improve automatically through experience and by the use of data.*" The final hidden state of the first token `[CLS]` is used as the representation of the whole sequence and a classification layer is added. After fine-tuning on the jargon-sentence pairs, the model has a certain ability to distinguish whether the sentence contains representative definitional information of the target jargon. SDI is then obtained as the top definitional sentences by ranking the sentences according to the confidence of the prediction. We refer to this model as **SDI-Extractor**.

3.1.2 Correlative Definitional Information

To explain a jargon, in addition to utilizing SDI, we can also refer to the definitions of its related terms, i.e., ***Correlative Definitional Information (CDI)***. For instance, to define *few-shot learning*, we can incorporate definitions of *zero-shot learning* and *meta learning*, with which we can know the meaning of "shot" and "learning" and may define *few-shot learning* similarly to *zero-shot learning*.

To get related terms and their definitions, we leverage Wikipedia as the external knowledge

source, which covers a wide range of domains and contains high-quality definitions for a large number of terms. Specifically, we follow the *core-fringe* notion in Huang et al. (2021b), where *core terms* are terms that have corresponding Wikipedia pages, and *fringe terms* are ones that are not associated with a Wikipedia page. For each jargon, we treat it as query to retrieve the most relevant core terms via document ranking based on Elasticsearch (Gormley and Tong, 2015), and extract first sentences on the corresponding Wikipedia pages as the definitions of related terms. We refer to this model as **CDI-Extractor**.

3.2 Generation

After extraction, we acquire the self- and correlative definitional information of jargon. This kind of information captures important characteristics of jargon and can be further refined and synthesized into the final definition by a definition generator.

Definition generation can be formulated as a conditioned sentence generation task– generating a coherent sentence to define the target jargon. Formally, we apply the standard sequence-to-sequence formulation: given jargon x, combining with the extracted sentences $\mathcal{S}_s$ (for SDI) and $\mathcal{S}_c$ (for CDI), the probability of the generated definition d is computed auto-regressively:

$$P(d|x, \mathcal{S}_s, \mathcal{S}_c) = \prod_{i=1}^{m} P(d_i|d_{0:i-1}, x, \mathcal{S}_s, \mathcal{S}_c),$$

where m is the length of d, d_i is the ith token of d, and d_0 is a special start token.

Following Bevilacqua et al. (2020), to build the generator, we employ BART (Lewis et al., 2020a), a pre-trained transformer-based encoder-decoder model that can be fine-tuned to perform specific conditional language generation tasks with specific training input-output pairs. Different from existing works (Gadetsky et al., 2018; Ishiwatari et al., 2019; Bevilacqua et al., 2020) which aim to learn to define a word/phrase in a given context, we propose to learn to define a jargon using the extracted knowledge. To be specific, we aim to fine-tune the BART model to generate the definition of the target jargon based on the surface name of the jargon and the extracted definitional information.

To apply the BART model, for a target jargon, we adopt the following encoding scheme: "$jargon$ *[DEF]* $sent_1$ *[SEP]* $sent_2$ ... *[SEP]* $sent_k$ *[DEF]* $sent'_1$ *[SEP]* $sent'_2$ ... *[SEP]* $sent'_{k'}$", where $sent_i$ and $sent'_i$ are the ith sentences ranked by *SDI-Extractor* and *CDI-Extractor*, respectively. We fine-tune BART to produce the ground-truth definition conditioned with the encoded input.

After training, given a new jargon, we get corresponding SDI and CDI according to Section 3.1. We encode the jargon and the top k ranked sentences of SDI and top k' ranked sentences of CDI as described above and use the generator to produce the final definition. We refer to this model as **CDM-Sk,Ck'**, i.e., **C**ombined **D**efinition **M**odeling.

Here we would like to mention that our combined definition modeling framework is modular and can be applied to different extractor-generator combinations commonly proposed for definition extraction/generation, which means that the proposed framework can improve the performance for a variety of definition modeling systems. For instance, we can replace the BART generator with GPT-2/3 generator (Radford et al., 2019; Brown et al., 2020) or DMAS (Huang et al., 2021a) by simply modifying the encoding scheme.

4 Experiments

4.1 Datasets

Existing datasets for definition modeling are mainly for general words/phrases. In this paper, we build several datasets (**UJ-CS**, **UJ-Math**, **UJ-Phy**) for jargon based on Wikipedia and CFL (Huang et al., 2021b). Compared to general words/phrases, jargons are less ambiguous but more specialized, i.e., a jargon usually only has one meaning, but it requires domain knowledge to understand. We also conduct experiments on the dataset (**Sci&Med**) provided in August et al. (2022), which contains definitions of scientific and medical terms derived from Wikipedia science glossaries and MedQuAD (Ben Abacha and Demner-Fushman, 2019).

Definition Extraction. We build a dataset for jargon definition extraction with Wikipedia. We first collect jargons with Wikipedia Category. Specifically, we traverse from three root categories, including *Category:Subfields of computer science*[3], *Category:Fields of mathematics*[4], and *Category:Subfields of physics* [5], and collect pages

[3] https://en.wikipedia.org/wiki/Category:Subfields_of_computer_science

[4] https://en.wikipedia.org/wiki/Category:Fields_of_mathematics

[5] https://en.wikipedia.org/wiki/Category:Subfields_of_physics

Data	Source of Jargon	Train	Valid	Test
UJ-CS	Springer	11,738	1,671	3,349
UJ-Math	CFL	4,247	583	1,019
UJ-Phy	CFL	4,157	573	1,026

Table 1: The statistics of the data.

	Precision	Recall	F1
CNN	91.84	90.66	91.25
C-BLSTM	91.59	88.93	90.24
SDI-Extractor	**96.72**	**97.67**	**97.19**

Table 2: Results of definition extraction.

at the first three levels of the hierarchies. For each page, we process the title with lemmatization as the jargon, extract the first sentence in the summary section as the corresponding definition, and sample ≤ 5 sentences containing the target jargon from other sections as negatives (they are less likely to be definitional sentences). We filter out jargons with surface name frequency < 5 in the arXiv corpus[6] (to filter out some noisy phrases, e.g., *List of artificial intelligence projects*). The dataset contains 26,559 positive and 121,975 negative examples, and the train/valid/test split is 0.8/0.1/0.1.

Definition Generation. Following (Huang et al., 2021b), we focus on generating definitions for jargon in three fields: computer science (**UJ-CS**), mathematics (**UJ-Math**), and physics (**UJ-Phy**). We collect jargons in two ways. For computer science, we collect jargons (author-assigned keywords) by web scraping from Springer publications on computer science. We filter out jargons with frequency < 5. For mathematics and physics, we collect jargons with the CFL model proposed in Huang et al. (2021b). Specifically, we collect terms with domain relevance score > 0.5 as jargons. For each jargon in the list, URLs of the top 20 results from Google search are visited. Then the sentences containing the target jargon are extracted. For training and evaluation, we only keep jargons that have a corresponding Wikipedia page and extract the first sentence on each page as the ground-truth definition. Table 1 summarizes the statistics of the data.

4.2 Experimental Setup

Baselines. For extraction, we compare SDI-Extractor with a CNN baseline and a CNN-BiLSTM baseline proposed in Anke and Schockaert (2018). Here we should mention that the more recent models (Veyseh et al., 2020; Kang et al., 2020) cannot be compared directly since these works focus on a fine-grained sequence labeling task, where the training data also requires additional labeling. Besides, extraction is not the focus of this paper; therefore, we put more emphasis on the evaluation for generation. For generation, we evaluate on the following models:

- **Gen (w/o context)**: A simple version of Generationary (Bevilacqua et al., 2020), where BART (Lewis et al., 2020a) is fine-tuned on jargon-definition pairs.
- **Gen (w/ context)**: Generationary with a sentence containing the target jargon as context, where BART is fine-tuned on context-definition pairs.
- **DMAS** (Huang et al., 2021a): A definition modeling model with three T5 (Raffel et al., 2020), where a re-ranking mechanism is included to model the specificity of definitions. Context is given by a sentence containing the target jargon.
- **BART NO SD and BART SD**: For the *Sci&Med* dataset (August et al., 2022), we also compare with the two best methods introduced in their paper: BART SD, where BART is fine-tuned with the term question, e.g., *What is (are) carbon nanotubes?*, concatenated with the supporting document; and BART NO SD, where BART is fine-tuned with just the question and definition, without the support documents.
- **Extractive**: An extractive baseline, which outputs the candidate definition with the highest confidence score predicted by SDI-Extractor (Section 3.1.1).
- **CDM-Sk,Ck'**: The combined definition modeling model introduced in Section 3.2. Sk or Ck' is omitted when k or k' is equal to 0.

Metrics. For extraction, we use the standard precision, recall, and F1 scores to evaluate the performance. For generation, we follow Bevilacqua et al. (2020) and apply several automatic metrics, including BLEU (BL)[7] (Papineni et al., 2002), ROUGE-L (R-L) (Lin, 2004), METEOR (MT) (Banerjee and Lavie, 2005), and BERTScore (BS) (Zhang et al., 2019). BLEU, ROUGE-L, and METEOR focus on measuring surface similarities between the generated definitions and the ground-truth definitions, and BERTScore is based on the similarities of contextual token embeddings. The signa-

[6] https://www.kaggle.com/Cornell-University/arxiv

[7] The version implemented on https://github.com/mjpost/sacrebleu.

	UJ-CS				UJ-Math				UJ-Phy			
	BL	**R-L**	**MT**	**BS**	**BL**	**R-L**	**MT**	**BS**	**BL**	**R-L**	**MT**	**BS**
Extractive	15.62	29.41	16.41	79.02	13.04	25.95	13.88	75.97	9.75	20.30	12.48	75.27
Gen (w/o context)	8.31	28.02	12.83	77.97	6.89	28.50	10.97	76.45	5.28	25.75	10.57	76.88
Gen (w/ context)	8.76	30.00	13.15	78.73	10.00	31.18	12.67	77.24	8.71	29.68	12.94	78.37
DMAS	4.98	26.05	10.60	78.09	1.58	24.10	7.97	75.75	2.70	24.59	9.59	77.43
CDM-C5	12.26	29.90	14.55	79.09	12.54	32.17	14.10	78.22	9.36	28.87	13.26	78.62
CDM-S1	17.12	34.67	17.46	80.75	16.33	36.07	16.22	78.94	12.42	31.48	14.70	78.96
CDM-S3	19.08	35.48	18.44	81.16	19.35	38.56	17.88	79.90	16.54	34.54	17.00	80.03
CDM-S5	20.21	35.98	19.06	81.33	20.76	39.87	18.63	80.31	18.58	35.15	18.00	80.38
CDM-S10	19.27	36.34	18.79	81.51	21.71	**40.43**	19.28	**80.68**	20.66	36.92	19.18	81.03
CDM-S5,**C**5	**22.66**	**38.12**	**20.30**	**82.00**	**23.22**	39.39	**19.61**	80.30	**20.84**	**37.66**	**19.26**	**81.18**

Table 3: Results of definition generation on automatic metrics. The best results are **bold** and second best ones are underlined.

	BL	**R-L**	**MT**	**BS**
Extractive	8.75	17.79	12.32	74.68
Gen (w/o context)	13.13	31.75	13.30	79.31
Gen (w/ context)	12.50	31.50	13.86	79.54
DMAS	9.12	28.43	11.04	79.21
BART NO SD	10.68	30.89	13.18	79.19
BART SD	11.11	32.34	13.97	80.12
CDM-C5	13.50	32.19	15.00	80.24
CDM-S1	11.91	33.14	15.31	80.24
CDM-S3	17.97	35.60	17.23	81.30
CDM-S5	20.18	37.25	18.52	81.75
CDM-S10	20.35	**37.98**	19.22	**82.19**
CDM-S5,**C**5	**20.55**	37.70	**19.24**	81.98

Table 4: Results of definition generation on **Sci&Med** (August et al., 2022).

ture of BERTScore is: roberta-large-mnli L19 no-idf version=0.3.0(hug trans=2.8.0). We also ask three human annotators (graduate students doing research on computational linguistics) to evaluate the output definitions with a 1-5 rating scale used in Ishiwatari et al. (2019): 1) completely wrong or self-definition; 2) correct topic with wrong information; 3) correct but incomplete; 4) small details missing; 5) correct.

Implementation Details. For SDI extraction, we adopt BERT-base-uncased from huggingface transformers framework (Wolf et al., 2020). We apply the BertForSequenceClassification in huggingface (with a linear layer on top of the pooled output). We use the default hyperparameters and fine-tune the model using Adam (Kingma and Ba, 2015) with learning rate of 2×10^{-6}. All the layers of the BERT model are fine-tuned. For the two baselines, we train the models on our data with the official implementation. For the extracted SDI, we exclude sentences from Wikipedia to avoid the models to see the ground truth.

For CDI extraction, following Huang et al. (2021a), we use the built-in Elasticsearch-based Wikipedia search engine[8] to collect related core terms for jargon; and then, we extract the first sentence on the corresponding Wikipedia page as the definition of each related term.

For generation, we employ the fairseq library[9] to build the BART-base generator and adopt the hyperparameters and settings as suggested in Bevilacqua et al. (2020). We set the learning rate as 5×10^{-5} and use batch size of $1,024$ tokens, updating every 16 iterations, with the number of warmup steps as $1,000$. For all the datasets, we use the same trained SDI-extractor as described above to extract SDI. We adopt the default/suggested hyperparameters for the baselines. We train and evaluate all the baselines and variants on the same train/valid/test split on NVIDIA Quadro RTX 5000 GPUs. The training of CDM can be finished in one hour.

4.3 Definition Extraction

Table 2 reports the results of definition extraction. We observe that SDI-Extractor outperforms baselines significantly and the performance is quite satisfactory (with an F1 score higher than 0.97), which indicates our definition extractor can extract useful self-definitional information for jargon.

4.4 Definition Generation

We provide both quantitative and qualitative evaluations for definition generation.

4.4.1 Automatic Evaluation

Tables 3 and 4 show the results on automatic metrics[10]. We observe the proposed CDM model out-

[8] https://en.wikipedia.org/w/index.php?search

[9] https://github.com/pytorch/fairseq/tree/master/examples/bart

[10] For Table 4, August et al. (2022) use BERT-base for BERTScore, while we use RoBERTa-large for BERTScore to

	Score (1-5)
Extractive	3.57
Gen (w/ context)	2.34
CDM-S1	3.65
CDM-S5	3.99
CDM-S5**,C**5	**4.04**

Table 5: Averaged human annotated scores.

performs the SOTA baselines significantly. Comparing Gen (w/ context) with Gen (w/o context), we find contexts (random sentences containing the target jargon) only have limited help with jargon definition modeling. Besides, CDM-S5 outperforms CDM-S3, while CDM-S3 outperforms CDM-S1, which means the sentences extracted by SDI-Extractor can provide important definitional information. Comparing CDM-C5 with Gen (w/ context) and Gen (w/o context), we can verify CDI is also helpful for definition generation, while the improvement is not as significant as the models with SDI, e.g., CDM-S5. Among all the models, CDM-S5,C5 usually achieves the best performance, which demonstrates the combination of SDI and CDI is the most significant for jargon definition modeling.

An interesting finding is that our simple extractive model is comparable to the SOTA abstractive baselines (except for Table 4, because most of the definitions in the dataset are not complete sentences, e.g, "*the science of automatic control systems*" for *cybernetics*, while SDI-Extractor usually extracts complete sentences). We suppose this is because, compared to general words/phrases, jargons are more difficult to define without external knowledge. For instance, it is almost impossible for a model to generate the definition for *twin prime* only with context "*proof of this conjecture would also imply the existence of an infinite number of twin primes*", while the definition can possibly be retrieved from the Web. The results also demonstrate that existing context-aware definition modeling systems are hard to handle jargon, while our proposed extraction-generation framework is quite practical for jargon definition modeling.

4.4.2 Human Evaluation

We conduct human evaluation for the computer science field (UJ-CS). Specifically, we randomly sample 50 jargons from the test set, and ask three human annotators to evaluate the definitions produced by different models with the rating scale described in Section 4.2. Table 5 reports the human evaluation results, where the average pairwise Cohen's κ is 0.69 (good agreement). We observe the state-of-the-art baseline Gen (w/context) is difficult to generate reasonable definitions for jargon. In contrast, the proposed CDM-S5,C5 model can produce high-quality definitions in most cases (with a human-annotated score higher than 4). The human evaluation results are also consistent with the automatic evaluation results presented in Table 3.

be consistent with Table 3.

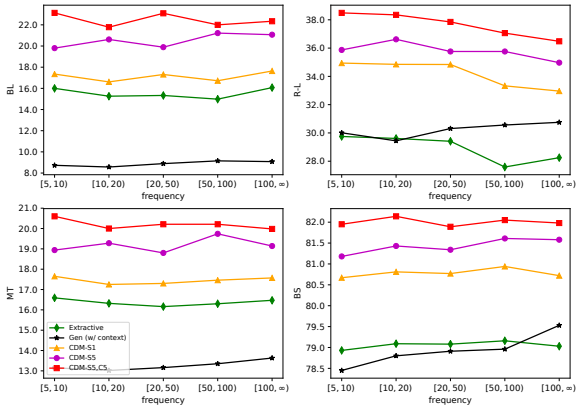

Figure 2: Results of definition generation with respect to jargon frequency in Springer (author-assigned keywords). Best viewed in color.

4.5 Sensitivity to Frequency

To investigate the sensitivity of the models with respect to the popularity of jargon, we report the results according to jargon frequency in Figure 2. We observe that Generationary (Bevilacqua et al., 2020) achieves slightly worse performance for less popular jargon on all metrics, while CDM performs well for low-frequency jargon, which indicates our framework can produce high-quality definitions for long-tail jargon. We suppose this is because, although long-tail jargon is less frequent, we can still extract useful definitional information from the entire Web and incorporate it for definition generation.

4.6 Generation Examples and Error Analysis

In Table 6, we show some sample outputs in the test set of three models: *Extractive*, *Gen (w/ context)*, and *CDM-S*5*,C*5, with ground-truth definitions in Wikipedia (*Gold*) as references.

From the results, we observe although the extractive baseline can produce reasonable sentences, the output sentences may not be high-quality definitional sentences of the target jargon. For instance,

	Extractive	**Gen (w/ context)**	**CDM-S5,C5**	***Gold***
intelligent user interfaces	ACM IUI 2021 is the 26th annual premier international forum for reporting outstanding research and development on intelligent user interfaces.	An intelligent user interface (IUI) is a user interface (UI) that is designed to respond to the needs and wants of the user.	In computing, an intelligent user interface (UI) is a user interface that has been developed by a group of researchers interested in enabling natural human-computer interaction by combining techniques from computer vision, machine learning, computer graphics, human-computer interaction and psychology.	An intelligent user interface (Intelligent UI, IUI, or sometimes Interface Agent) is a user interface (UI) that involves some aspect of artificial intelligence (AI or computational intelligence).
wear leveling	The term preemptive wear leveling (PWL) has been used by Western Digital to describe their preservation technique used on hard disk drives (HDDs) designed for storing audio and video data.	Wear leveling is the process of reducing the wear of a Flash die.	Wear leveling is a technique used to increase the lifetime of a solid-state drive (SSD).	Wear leveling (also written as wear levelling) is a technique for prolonging the service life of some kinds of erasable computer storage media, such as flash memory, which is used in solid-state drives (SSDs) and USB flash drives, and phase-change memory.
gittins index	In applied mathematics, the "Gittins index" is a real scalar value associated to the state of a stochastic process with a reward function and with a probability of termination.	The Gittins index is a decision-making tool used in decision-making and project management.	In applied mathematics, the Gittins index is a real scalar value associated to the state of a stochastic process with a reward function and with a probability of termination.	The Gittins index is a measure of the reward that can be achieved through a given stochastic process with certain properties, namely: the process has an ultimate termination state and evolves with an option, at each intermediate state, of terminating.
reduplication	The term ?€?compensatory reduplication?€? refers to duplication that serves a phonological purpose.	In mathematics, reduplication is a generalization of the concept of reduplication.	Reduplication is the repetition of an entire word, word stem (root with one or more affixes), or root.	In linguistics, reduplication is a morphological process in which the root or stem of a word (or part of it) or even the whole word is repeated exactly or with a slight change.
power delay profile	The power delay profile of a channel represents the average power of the received signal in terms of the delay with respect to the first arrival path in multi-path transmission.	A power delay profile (PDP) is a measure of the time delay between the transmission and reception of a signal.	In telecommunications, the power delay profile (PDP) of a multipath channel represents the average power of the received signal in terms of the delay with respect to the first arrival path in multi-path transmission.	The power delay profile (PDP) gives the intensity of a signal received through a multipath channel as a function of time delay.

Table 6: Sample of definitions produced by *Extractive*, *Gen (w/ context)*, and *CDM-S5,C5*.

the extracted sentence for *wear leveling* in fact is the definition of *preemptive wear leveling*. We also find Gen (w/ context) suffers severely from *hallucinations*, i.e., generating irrelevant or contradicted facts. For instance, *gittins index* is described as a decision-making tool instead of a measure/value, which is completely wrong. This is mainly because the contexts of jargon may not provide sufficient knowledge to define jargon. In contrast, the quality of definitions generated by CDM-S5,C5 is high–all the generated definitions capture the main characteristics of the target jargon correctly.

Error Analysis. To further understand the results and identify the remaining challenges, we analyze the human evaluation results. We find that errors could be introduced in either the extraction or the generation process. E.g., 1) for *intelligent user interfaces* in Table 6, the top 1 sentence extracted by SDI-Extractor ("*ACM IUI ... interfaces.*") cannot provide meaningful knowledge to the generator. Although by incorporating other sentences, CDM-S5,C5 can generate a reasonable definition, the definition still contains minor errors. 2) For *markup languages*, although SDI-Extractor extracts reasonable definitions (e.g., "*Markup languages are languages used by a computer to annotate a document.*"), the generator mistakenly synthesizes the SDI and CDI into "*A markup language is a series of tags mixed with plain text.*" Nonetheless, compared to existing models that do not combine extraction and generation, CDM greatly reduces hallucination.

5 Discussion

In this work, we focus on jargon definition modeling. The proposed framework can be further extended to general words/phrases in a context-aware setting (Gadetsky et al., 2018). For instance, to retrieve the definitional information, we can incorporate the context the target word/phrase used in. E.g., the BERT extractor can be trained with a modified encoding scheme: "*[CLS]* $word/phrase$ *[SEP]* $context$ *[DEF]* $sentence$". Similarly, the generator can produce the final definition conditioned on the context. E.g., the input of the generator can be encoded as "$word/phrase$ *[SEP]* $context$ *[DEF]* $sent_1$ *[SEP]* $sent_2$ *...* *[SEP]* $sent_k$ *[DEF]* $sent'_1$ *[SEP]* $sent'_2$ *...* *[SEP]* $sent'_{k'}$". Since our framework is modular, the BERT extractor and BART generator can also be replaced with more advanced language models. It is also interesting to train the extractor and generator jointly or iteratively (Guu et al., 2020; Lewis et al., 2020b). We keep the proposed model simple and leave context-aware combined definition modeling and more complicated combinations as future work.

6 Conclusion

We present the first combination of definition extraction and definition generation. We show that, by incorporating extracted self- and correlative definitional information, the generator can produce high-quality definitions for jargon. Experimental results demonstrate the effectiveness of our framework, where the proposed method outperforms recent baselines by a large margin. We also publish several datasets for jargon definition modeling. In future work, we plan to improve our framework as discussed in Section 5 and apply our methods to construct several online domain dictionaries.

Limitations

One limitation of this paper is that it does not consider the diversity of definitions. Definitions from different perspectives can facilitate a more comprehensive understanding. For instance, to define *artificial intelligence*, we may relate it to or contrast it with other concepts, e.g., "*artificial intelligence refers to systems or machines that mimic human intelligence to perform tasks and can iteratively improve themselves based on the information they collect.*" or "*artificial intelligence is intelligence demonstrated by machines, as opposed to the natural intelligence displayed by animals including humans.*" Recent work starts to model the specificity and complexity for definition modeling (Huang et al., 2021a; Gardner et al., 2022); however, the diversity of generative definitions is still limited. We believe our framework can benefit diversity since the generator has the potential to generate definitions with different styles by incorporating diverse definitional information extracted from the Web.

Acknowledgements

We thank the reviewers for their constructive feedback. This material is based upon work supported by the National Science Foundation IIS 16-19302 and IIS 16-33755, Zhejiang University ZJU Research 083650, IBM-Illinois Center for Cognitive Computing Systems Research (C3SR)– a research collaboration as part of the IBM Cognitive Horizon Network, grants from eBay and Microsoft Azure, UIUC OVCR CCIL Planning Grant 434S34, UIUC CSBS Small Grant 434C8U, and UIUC New Frontiers Initiative. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the funding agencies.

References

Luis Espinosa Anke and Steven Schockaert. 2018. Syntactically aware neural architectures for definition extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 378–385.

Tal August, Katharina Reinecke, and Noah A Smith. 2022. Generating scientific definitions with controllable complexity. In *ACL*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Asma Ben Abacha and Dina Demner-Fushman. 2019. A question-entailment approach to question answering. *BMC bioinformatics*, 20(1):1–23.

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Olivia M Bullock, Daniel Colón Amill, Hillary C Shulman, and Graham N Dixon. 2019. Jargon as a barrier to effective science communication: Evidence from metacognition. *Public Understanding of Science*, 28(7):845–853.

Andrew Butterfield, Gerard Ekembe Ngondi, and Anne Kerr. 2016. *A dictionary of computer science*. Oxford University Press.

Hang Cui, Min-Yen Kan, and Tat-Seng Chua. 2004. Unsupervised learning of soft patterns for generating definitions from online news. In *Proceedings of the 13th international conference on World Wide Web*, pages 90–99.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the*

North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186.

Ismail Fahmi and Gosse Bouma. 2006. Learning to identify definitions using syntactic features. In *Proceedings of the Workshop on Learning Structured Information in Natural Language Applications*.

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

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

Clinton Gormley and Zachary Tong. 2015. *Elasticsearch: the definitive guide: a distributed real-time search and analytics engine*. " O'Reilly Media, Inc.".

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.

Han Huang, Tomoyuki Kajiwara, and Yuki Arase. 2021a. Definition modelling for appropriate specificity. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2499–2509.

Jie Huang, Kevin Chen-Chuan Chang, Jinjun Xiong, and Wen-mei Hwu. 2021b. Measuring fine-grained domain relevance of terms: A hierarchical core-fringe approach. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*.

Jie Huang, Kevin Chen-Chuan Chang, Jinjun Xiong, and Wen-mei Hwu. 2022. Open relation modeling: Learning to define relations between entities. In *Findings of the Association for Computational Linguistics: ACL 2022*.

Shonosuke Ishiwatari, Hiroaki Hayashi, Naoki Yoshinaga, Graham Neubig, Shoetsu Sato, Masashi Toyoda, and Masaru Kitsuregawa. 2019. Learning to describe unknown phrases with local and global contexts. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3467–3476.

Yiping Jin, Min-Yen Kan, Jun Ping Ng, and Xiangnan He. 2013. Mining scientific terms and their definitions: A study of the acl anthology. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 780–790.

Dongyeop Kang, Andrew Head, Risham Sidhu, Kyle Lo, Daniel S Weld, and Marti A Hearst. 2020. Document-level definition detection in scholarly documents: Existing models, error analyses, and future directions. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 196–206.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *Proceedings of the 3rd International Conference on Learning Representations*.

Judith L Klavans and Smaranda Muresan. 2001. Evaluation of the definder system for fully automatic glossary construction. In *Proceedings of the AMIA Symposium*, page 324. American Medical Informatics Association.

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

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*.

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

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Zequn Liu, Shukai Wang, Yiyang Gu, Ruiyi Zhang, Ming Zhang, and Sheng Wang. 2021. Graphine: A dataset for graph-aware terminology definition generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3453–3463.

Timothee Mickus, D Paperno, and Mathieu Constant. 2019. Mark my word: A sequence-to-sequence approach to definition modeling. In *Proceedings of The First NLPL Workshop on Deep Learning for Natural Language Processing*, page 1. Linköping University Electronic Press.

Thanapon Noraset, Chen Liang, Larry Birnbaum, and Doug Downey. 2017. Definition modeling: Learning to define word embeddings in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

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

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

Machel Reid, Edison Marrese-Taylor, and Yutaka Matsuo. 2020. Vcdm: Leveraging variational bi-encoding and deep contextualized word representations for improved definition modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6331–6344.

Natalia Vanetik, Marina Litvak, Sergey Shevchuk, and Lior Reznik. 2020. Automated discovery of mathematical definitions in text. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2086–2094.

Amir Veyseh, Franck Dernoncourt, Dejing Dou, and Thien Nguyen. 2020. A joint model for definition extraction with syntactic connection and semantic consistency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9098–9105.

Koki Washio, Satoshi Sekine, and Tsuneaki Kato. 2019. Bridging the defined and the defining: Exploiting implicit lexical semantic relations in definition modeling. 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 3521–3527.

Eline Westerhout. 2009. Definition extraction using linguistic and structural features. In *Proceedings of the 1st Workshop on Definition Extraction*, pages 61–67.

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

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.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Hua Zheng, Damai Dai, Lei Li, Tianyu Liu, Zhifang Sui, Baobao Chang, and Yang Liu. 2021. Decompose, fuse and generate: A formation-informed method for chinese definition generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5524–5531.