Machine Intelligence Research
www.mi-research.net

19(3), June 2022, 169-183
DOI: 10.1007/s11633-022-1331-6

Paradigm Shift in Natural Language Processing

Tian-Xiang Sun[1,2] Xiang-Yang Liu[1,2] Xi-Peng Qiu[1,2] Xuan-Jing Huang[1,2]

[1]School of Computer Science, Fudan University, Shanghai 200438, China

[2]Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai 200438, China

Abstract: In the era of deep learning, modeling for most natural language processing (NLP) tasks has converged into several mainstream paradigms. For example, we usually adopt the sequence labeling paradigm to solve a bundle of tasks such as POS-tagging, named entity recognition (NER), and chunking, and adopt the classification paradigm to solve tasks like sentiment analysis. With the rapid progress of pre-trained language models, recent years have witnessed a rising trend of paradigm shift, which is solving one NLP task in a new paradigm by reformulating the task. The paradigm shift has achieved great success on many tasks and is becoming a promising way to improve model performance. Moreover, some of these paradigms have shown great potential to unify a large number of NLP tasks, making it possible to build a single model to handle diverse tasks. In this paper, we review such phenomenon of paradigm shifts in recent years, highlighting several paradigms that have the potential to solve different NLP tasks.[1]

Keywords: Natural language processing, pre-trained language models, deep learning, sequence-to-sequence, paradigm shift.

Citation: T. X. Sun, X. Y. Liu, X. P. Qiu, X. J. Huang. Paradigm shift in natural language processing. *Machine Intelligence Research*, vol.19, no.3, pp.169–183, 2022. http://doi.org/10.1007/s11633-022-1331-6

1 Introduction

In the scope of this paper, a paradigm is a general modeling framework or a distinct set of methodologies to solve a class of tasks. For instance, sequence labeling is a mainstream paradigm for named entity recognition (NER). Different paradigms usually require different formats of input and output, and therefore highly depend on the annotation of the tasks. In the past years, modeling for most NLP tasks has converged to several mainstream paradigms, as summarized in this paper, Class, Matching, SeqLab, MRC, Seq2Seq, Seq2ASeq, and (M)LM.

Though the paradigm for many tasks has converged and dominated for a long time, recent work has shown that models under some paradigms also generalize well on tasks with other paradigms. For example, the MRC and Seq2Seq paradigms can also achieve state-of-the-art performance on NER tasks[1, 2], which were previously formalized in the sequence labeling (SeqLab) paradigm. Such methods typically first convert the form of the dataset to the form required by the new paradigm, and then use the model under the new paradigm to solve the task. In recent years, similar methods that reformulate a natural language processing (NLP) task as another one have achieved great success and gained increasing attention in the community. After the emergence of pre-trained language models (PTMs)[3–6], paradigm shifts have been observed in an increasing number of tasks. Combined with the power of these PTMs, some paradigms have shown great potential to unify diverse NLP tasks. One of these potential unified paradigms, (M)LM (also referred to as prompt-based tuning), has made rapid progress recently, making it possible to employ a single PTM as the universal solver for various understanding and generation tasks[7–14].

Despite their success, these paradigm shifts scattering in various NLP tasks have not been systematically reviewed and analyzed. In this paper, we attempt to summarize recent advances and trends in this line of research, namely paradigm shift or paradigm transfer.

This paper is organized as follows. Section 2 gives formal definitions of the seven paradigms, and introduces their representative tasks and instance models. Section 3 shows recent paradigm shifts that happened in different NLP tasks. Section 4 discusses the designs and challenges of several highlighted paradigms that have great potential to unify most existing NLP tasks. Section 5 concludes with a brief discussion of recent trends and future directions.

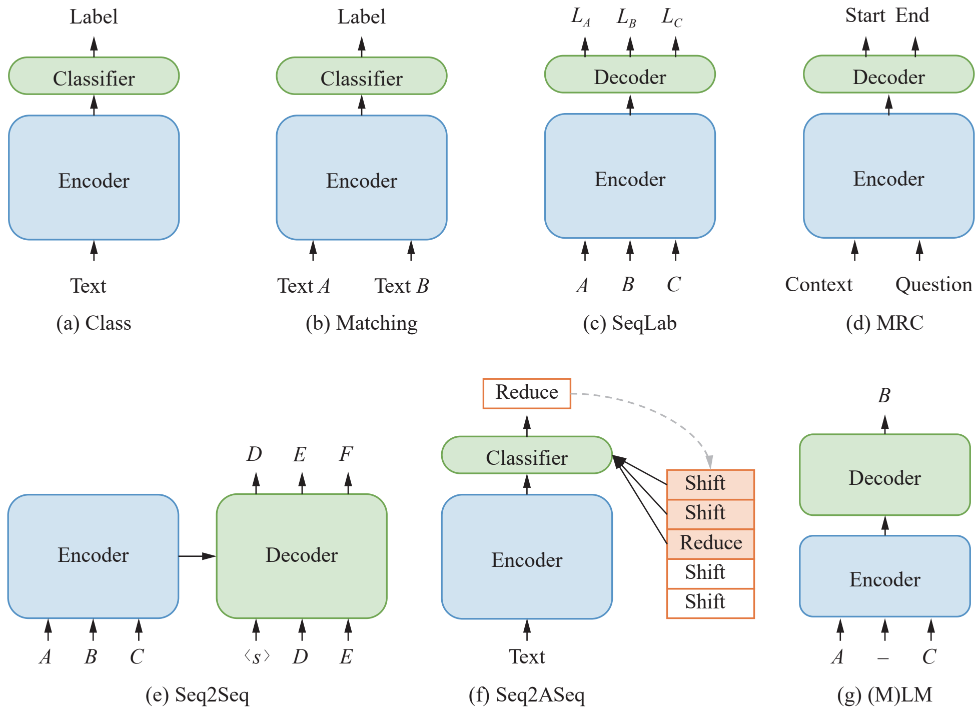
2 Paradigms in NLP

2.1 Paradigms, tasks, and models

Typically, a task corresponds to a dataset $\mathcal{D} =$

Review
Manuscript received March 1, 2022; accepted April 13, 2022
Recommended by Associate Editor Jun Zhao
Colored figures are available in the online version at https://link.springer.com/journal/11633

[1] A constantly updated website is publicly available at https://txsun1997.github.io/nlp-paradigm-shift.



(a) Class (b) Matching (c) SeqLab (d) MRC

(e) Seq2Seq (f) Seq2ASeq (g) (M)LM

Fig. 1 Illustration of the seven mainstream paradigms in NLP

$\{\mathcal{X}_i, \mathcal{Y}_i\}_{i=1}^N$. A paradigm is the general modeling framework to fit some datasets (or tasks) with a specific format (i.e., the data structure of $\mathcal{X}$ and $\mathcal{Y}$). Therefore, a task can be solved by multiple paradigms by transforming it into different formats. A paradigm can be used to solve multiple tasks that can be formulated in the same format. Moreover, a paradigm can be instantiated by a class of models with similar architectures.

2.2 Seven paradigms in NLP

In this paper, we mainly consider the following seven paradigms that are widely used in NLP tasks, i.e., Class, Matching, SeqLab, MRC, Seq2Seq, Seq2ASeq, and (M)LM. These paradigms have demonstrated strong dominance in many mainstream NLP tasks. Fig. 1 provides an illustration of the seven paradigms. Sections 2.2.1–2.2.7 briefly introduce the seven paradigms and their corresponding tasks and models.

2.2.1 Classification (Class)

Text classification, which is designating predefined labels for text, is an essential and fundamental task in various NLP applications such as sentiment analysis, topic classification, spam detection, etc. In the era of deep learning, text classification is usually done by feeding the input text into a deep neural-based encoder to extract the task-specific feature, which is then fed into a shallow classifier to predict the label, i.e.,

$$\mathcal{Y} = \mathrm{Cls}(\mathrm{Enc}(\mathcal{X})). \tag{1}$$

Note that $\mathcal{Y}$ can be one-hot or multi-hot (in which case we call multi-label classification). $\mathrm{Enc}(\cdot)$ can be instantiated as convolutional networks[15], recurrent networks[16], or transformers[17]. $\mathrm{Cls}(\cdot)$ is usually implemented as a simple multi-layer perceptron following a pooling layer. Note that the pooling layer can be performed on the whole input text or a span of tokens.

2.2.2 Matching

Text matching is a paradigm to predict the semantic relevance of two texts. It is widely adopted in many fields, such as information retrieval, natural language inference, question answering, and dialogue systems. A Matching model should not only extract the features of the two texts, but also capture their fine-grained interactions. The Matching paradigm can be simply formulated as

$$\mathcal{Y} = \mathrm{Cls}(\mathrm{Enc}(\mathcal{X}_a, \mathcal{X}_b)) \tag{2}$$

where $\mathcal{X}_a$ and $\mathcal{X}_b$ are two texts to be predicted, $\mathcal{Y}$ can be discrete (e.g., whether one text entails or contradicts the other text) or continuous (e.g., the semantic similarity between the two texts). The two texts are usually encoded separately and then interact with each other[18].

2.2.3 Sequence labeling (SeqLab)

The sequence labeling (SeqLab) paradigm (also referred to as sequence tagging) is a fundamental paradigm modeling a variety of tasks such as part-of-speech (POS) tagging, NER, and text chunking. Conventional neural-based sequence labeling models are comprised of an encoder to capture the contextualized feature for each token in the sequence, and a decoder to take in the features and predict the labels, i.e.,

$$y_1, \cdots, y_n = \mathrm{Dec}(\mathrm{Enc}(x_1, \cdots, x_n)) \tag{3}$$

where $y_1, \cdots, y_n$ are the corresponding labels of $x_1, \cdots, x_n$. $\mathrm{Enc}(\cdot)$ can be instantiated as a recurrent network[19] or a transformer encoder[17]. $\mathrm{Dec}(\cdot)$ is usually implemented as conditional random fields (CRF)[20].

2.2.4 MRC

The machine reading comprehension (MRC) paradigm extracts contiguous token sequences (spans) from the input sequence conditioned on a given question. It is initially adopted to solve MRC tasks, and then generalized to other NLP tasks by reformulating them into the MRC format. To keep consistent with prior work and avoid confusion, we name this paradigm MRC, and distinguish it from the task MRC. The MRC paradigm can be formally described as follows:

$$y_k, \cdots, y_{k+l} = \mathrm{Dec}(\mathrm{Enc}(\mathcal{X}_p, \mathcal{X}_q)) \tag{4}$$

where $\mathcal{X}_p$ and $\mathcal{X}_q$ denote passage (also referred to context) and query, and $y_k, \cdots, y_{k+l}$ is a span from $\mathcal{X}_p$ or $\mathcal{X}_q$. Typically, Dec is implemented as two classifiers, one for predicting the starting position and one for predicting the ending position[21–23].

2.2.5 Sequence-to-sequence

The sequence-to-sequence (Seq2Seq) paradigm is a general and powerful paradigm that can handle a variety of NLP tasks. Typical applications of Seq2Seq include machine translation and dialogue, where the system is supposed to output a sequence (target language or response) conditioned on an input sequence (source language or user query). The Seq2Seq paradigm is typically implemented by an encoder-decoder framework[24–27]:

$$y_1, \cdots, y_m = \mathrm{Dec}(\mathrm{Enc}(x_1, \cdots, x_n)). \tag{5}$$

Different from SeqLab, the lengths of the input and output are not necessarily the same. Moreover, the decoder in Seq2Seq is usually more complicated and takes as input at each step the previous output (when inference) or the ground truth (when training).

2.2.6 Sequence-to-action-sequence

Sequence-to-action-sequence (Seq2ASeq) is a widely used paradigm for structured prediction. The aim of Seq2ASeq is to predict an action sequence (also called transition sequence) from some initial configuration c_0 to a terminal configuration. The predicted action sequence should encode some legal structure, such as a dependency tree. The instances of the Seq2ASeq paradigm are usually called transition-based models, which can be formulated as

$$\mathcal{A} = \mathrm{Cls}(\mathrm{Enc}(\mathcal{X}), \mathcal{C}) \tag{6}$$

where $\mathcal{A} = a_1, \cdots, a_m$ is a sequence of actions, $\mathcal{C} = c_0, \cdots, c_{m-1}$ is a sequence of configurations. At each time step, the model predicts an action a_t based on the input text and the current configuration c_{t-1}, which can be comprised of top elements in the stack, buffer, and previous actions[28, 29].

2.2.7 (M)LM

Language modeling (LM) is a long-standing task in NLP, which is to estimate the probability of a given sequence of words occurring in a sentence. Due to its self-supervised fashion, language modeling and its variants, e.g., masked language modeling (MLM), are adopted as training objectives to pre-train models on a large-scale unlabeled corpus. Typically, a language model can be simply formulated as

$$x_k = \mathrm{Dec}(x_1, \cdots, x_{k-1}) \tag{7}$$

where Dec can be any auto-regressive model such as recurrent networks[30, 31] and transformer decoder[32]. As a famous variant of LM, MLM can be formulated as

$$\bar{x} = \mathrm{Dec}(\mathrm{Enc}(\tilde{x})) \tag{8}$$

where $\tilde{x}$ is a corrupted version of x by replacing a portion of the tokens with a special token [MASK], and $\bar{x}$ denotes the masked tokens to be predicted. Dec can be implemented as a simple classifier as in bidirectional encoder representations from transformers (BERT)[3] or as an auto-regressive transformer decoder as in bidirectional and auto-regressive transformers (BART)[33] and text-to-text transfer transformer (T5)[4].

Though LM and MLM can be somehow different (LM is based on auto-regressive while MLM is based on auto-encoding), we categorize them into one paradigm, (M)LM, due to their same inherent nature, which is estimating the probability of some words given the context.

2.3 Compound paradigm

In this paper, we focus mainly on fundamental paradigms (as described above) and tasks. Nevertheless, it is worth noting that more complicated NLP tasks can be solved by combining multiple fundamental paradigms. For instance, HotpotQA[34], a multi-hop question answering task, can be solved by combining Matching and MRC, where Matching is responsible for finding relevant documents, and MRC is responsible for selecting the answer span[35].

3 Paradigm shift in NLP tasks

In this section, we review the paradigm shifts that occur in different NLP tasks: text classification, natural language inference, named entity recognition, aspect-based sentiment analysis, relation exaction, text summarization, and parsing. Table 1 provides a summary of paradigm shifts.

Table 1 Paradigms shift in NLP tasks. TC: Text classification. NLI: Natural language inference. NER: Named entity recognition. ABSA: Aspect-based sentiment analysis. RE: Relation extraction. Summ: Text summarization. Parsing: Syntactic/Semantic parsing. f and g indicate pre-processing and post-processing, respectively. $\mathcal{L}$ means label description. $\oplus$ means concatenation. $\mathcal{X}_{asp}, \mathcal{X}_{opin}, \mathcal{Y}_{sent}$ mean aspect, opinion, and sentiment, respectively. $\mathcal{S}_{aux}$ means auxiliary sentence. $\mathcal{X}_{sub}, \mathcal{X}_{obj}$ stand for subject entity and object entity, respectively. $\mathcal{S}_{cand}$ means candidate summary. $\mathcal{C}_t$ is the configuration at time step t and $\mathcal{A}$ is a sequence of actions.

Task		Original paradigm			Shifted paradigm		
TC	Paradigm	Class	$\Rightarrow$	Matching	Seq2Seq	(M)LM	
	Input	$\mathcal{X}$		$\mathcal{X}, \mathcal{L}$	$\mathcal{X}$	$f_{prompt}(\mathcal{X})$	
	Output	$\mathcal{Y}$		$\mathcal{Y} \in \{0, 1\}$	$y_1, \cdots, y_m$	$g(\mathcal{Y})$	
	Example	[3]		[36]	[37]	[7]	
NLI	Paradigm	Matching	$\Rightarrow$	Class	Seq2Seq	(M)LM	
	Input	$\mathcal{X}_a, \mathcal{X}_b$		$\mathcal{X}_a \oplus \mathcal{X}_b$	$f_{prompt}(\mathcal{X}_a, \mathcal{X}_b)$	$f_{prompt}(\mathcal{X}_a, \mathcal{X}_b)$	
	Output	$\mathcal{Y}$		$\mathcal{Y}$	$\mathcal{Y}$	$g(\mathcal{Y})$	
	Example	[18]		[3]	[38]	[7]	
NER	Paradigm	SeqLab	$\Rightarrow$	Class	MRC	Seq2Seq	(M)LM
	Input	$x_1, \cdots, x_n$		$\mathcal{X}_{span}$	$\mathcal{X}, \mathcal{Q}_y$	$\mathcal{X}$	$\mathcal{X}$
	Output	$y_1, \cdots, y_n$		$\mathcal{Y}$	$\mathcal{X}_{span}$	$(\mathcal{X}_{ent_i}, \mathcal{Y}_{ent_i})_{i=1}^m$	$g(\mathcal{Y})$
	Example	[19]		[39]	[1]	[2]	[40]
ABSA	Paradigm	Class	$\Rightarrow$	Matching	MRC	Seq2Seq	(M)LM
	Input	$\mathcal{X}_{asp}$		$\mathcal{X}, \mathcal{S}_{aux}$	$\mathcal{X}, \mathcal{Q}_{asp}, \mathcal{Q}_{opin}, \mathcal{Q}_{sent}$	$\mathcal{X}$	$f_{prompt}(\mathcal{X})$
	Output	$\mathcal{Y}$		$\mathcal{Y}$	$\mathcal{X}_{asp}, \mathcal{X}_{opin}, \mathcal{Y}_{sent}$	$(\mathcal{X}_{asp_i}, \mathcal{X}_{opin_i}, \mathcal{Y}_{sent_i})_{i=1}^m$	$g(\mathcal{Y})$
	Example	[41]		[42]	[43]	[44]	[45]
RE	Paradigm	Class	$\Rightarrow$	MRC	Seq2Seq	(M)LM	
	Input	$\mathcal{X}$		$\mathcal{X}, \mathcal{Q}_y$	$\mathcal{X}$	$f_{prompt}(\mathcal{X})$	
	Output	$\mathcal{Y}$		$\mathcal{X}_{ent}$	$(\mathcal{Y}_i, \mathcal{X}_{sub_i}, \mathcal{X}_{obj_j})_{i=1}^m$	$g(\mathcal{Y})$	
	Example	[46]		[47]	[48]	[49]	
Summ	Paradigm	SeqLab / Seq2Seq	$\Rightarrow$	Matching	(M)LM		
	Input	$\mathcal{X}_1, \cdots, \mathcal{X}_n$ / $\mathcal{X}, \mathcal{Q}_{summ}$		$(\mathcal{X}, \mathcal{S}_{cand_i})_{i=1}^n$	$\mathcal{X}$, Keywords/Prompt		
	Output	$\mathcal{Y}_1, \cdots, \mathcal{Y}_n \in \{0, 1\}^n$ / $\mathcal{Y}$		$\hat{\mathcal{S}}_{cand}$	$\mathcal{Y}$		
	Example	[38, 50]		[51]	[52]		
Parsing	Paradigm	Seq2ASeq	$\Rightarrow$	(M)LM	SeqLab	MRC	Seq2Seq
	Input	$(\mathcal{X}, \mathcal{C}_t)_{t=0}^{m-1}$		$(\mathcal{X}, \mathcal{Y}_i)_{i=1}^k$	$x_1, \cdots, x_n$	$\mathcal{X}, \mathcal{Q}_{child}$	$\mathcal{X}$
	Output	$\mathcal{A} = a_1, \cdots, a_m$		$\hat{\mathcal{Y}}$	$g(y_1, \cdots, y_n)$	$\mathcal{X}_{parent}$	$g(y_1, \cdots, y_m)$
	Example	[28]		[53]	[54]	[55]	[56]

3.1 Text classification

Text classification is an essential task in various NLP applications. Conventional text classification tasks can be well solved by the Class paradigm. Nevertheless, its variants, such as multi-label classification, can be challenging, in which case Class may be sub-optimal. To that end Yang et al. [37] propose to adopt the Seq2Seq paradigm to better capture interactions between labels for multi-label classification tasks.

In addition, the semantics hidden in the labels cannot be fully exploited in the Class paradigm. Chai et al.[36] and Wang et al.[57] adopt the Matching paradigm to predict whether the pair-wise input $(\mathcal{X}, \mathcal{L}_y)$ is matched, where $\mathcal{X}$ is the original text and $\mathcal{L}_y$ is the label description for class y. Though the semantic meaning of a label can be exactly defined by the samples that it is associated with, incorporating prior knowledge of the label is also helpful when training data is limited.

With the rise of pre-trained language models (LMs), text classification tasks can also be solved in the (M)LM paradigm[5, 7–9]. By reformulating a text classification task into a (masked) language modeling task, the gap between LM pre-training and fine-tuning is narrowed, resulting in improved performance when training data is limited.

3.2 Natural language inference

Natural language inference (NLI) is typically modeled in the Matching paradigm, where the two input texts $(\mathcal{X}_a, \mathcal{X}_b)$ are encoded and interact with each other, followed by a classifier to predict the relationship between them[18]. With the emergence of powerful encoders such as BERT[3], NLI tasks can be simply solved in the Class paradigm by concatenating the two texts as one. In the case of few-shot learning, NLI tasks can also be formulated in the (M)LM paradigm by modifying the input, e.g., "$\mathcal{X}_a$? [MASK], $\mathcal{X}_b$". The unfilled token [MASK] can be predicted by the MLM head as Yes/No/Maybe, corresponding to Entailment/Contradiction/Neutral[7–9].

3.3 Named entity recognition

NER is also a fundamental task in NLP. NER can be categorized into three subtasks: flat NER, nested NER, and discontinuous NER. Traditional methods usually solve the three NER tasks based on three paradigms, respectively, i.e., SeqLab[19, 58], Class[59, 60], and Seq2ASeq[58, 61].

Fu et al.[39] and Yu et al.[62] solve flat NER and nested NER with the Class paradigm. The main idea is to predict the label for each span in the input text. This paradigm shift introduces the span overlapping problem: The predicted entities may overlap, which is not allowed in the flat NER. To handle this, Fu et al.[39] adopt a heuristic decoding method: For these overlapped spans, only keep the span with the highest prediction probability.

Li et al.[1] propose to formulate flat NER and nested NER as an MRC task. They reconstruct each sample into a triplet $(\mathcal{X}, \mathcal{Q}_y, \mathcal{X}_{span})$, where $\mathcal{X}$ is the original text, $\mathcal{Q}_y$ is the question for the entity y, and $\mathcal{X}_{span}$ is the answer. Given the context, question, and answer, the MRC paradigm can be adopted to solve this. Since there can be multiple answers (entities) in a sentence, an index matching module is developed to align the start and end indexes.

Yan et al.[2] use a unified model based on the Seq2Seq paradigm to solve the three types of NER subtasks. The input of the Seq2Seq paradigm is the original text, while the output is a sequence of span-entity pairs, for instance, "*Barack Obama* ⟨Person⟩ *US* ⟨Location⟩". Due to the versatility of the Seq2Seq paradigm and the great power of BART[33], this unified model achieved state-of-the-art performance on various datasets spanning all three NER subtasks.

3.4 Aspect-based sentiment analysis

Aspect-based sentiment analysis (ABSA) is a fine-grained sentiment analysis task with seven subtasks, i.e., aspect term extraction (AE), opinion term extraction (OE), aspect-level sentiment classification (ALSC), aspect-oriented opinion extraction (AOE), aspect term extraction and sentiment classification (AESC), pair extraction (Pair), and triplet extraction (Triplet). These subtasks can be solved using different paradigms. For example, ALSC can be solved by the Class paradigm, and AESC can be solved using the SeqLab paradigm.

ALSC is to predict the sentiment polarity for each target-aspect pair, e.g., (LOC1, price), given a context, e.g., "LOC1 *is often considered the coolest area of London*". Sun et al.[42] formulate such a classification task into a sentence-pair matching task, and adopt the Matching paradigm to solve it. In particular, they generate auxiliary sentences (denoted as $\mathcal{S}_{aux}$) for each target-aspect pair. For example, $\mathcal{S}_{aux}$ for (LOC1, price) can be "*What do you think of the price of* LOC1?". The auxiliary sentence is then concatenated with the context as $(\mathcal{S}_{aux}, \mathcal{X})$, which is then fed into BERT[3] to predict the sentiment.

Mao et al.[43] adopt the MRC paradigm to handle all of the ABSA subtasks. In particular, they construct two queries to sequentially extract the aspect terms and their corresponding polarities and opinion terms. The first query is "*Find the aspect terms in the text.*" Assume that the answer (aspect term) predicted by the MRC model is AT, then the second query can be constructed as "*Find the sentiment polarity and opinion terms for* AT *in the text.*" Through such dataset conversion, all ABSA subtasks can be solved in the MRC paradigm.

Yan et al.[44] solve all ABSA subtasks with the Seq2Seq paradigm by converting the original label of a subtask into a sequence of tokens, which is used as a target to train a Seq2Seq model. Take the triplet extraction subtask as an example, for the input sentence, "*The drinks are always well made and the wine selection is fairly priced* ", the output target is constructed as "*drinks well made* Positive *wine selection fairly priced* Positive". Equipped with BART[33] as the backbone, they achieved competitive performance on most ABSA subtasks.

Recently, Li et al.[45] propose formulating ABSA subtasks in the (M)LM paradigm. In particular, for the input text $\mathcal{X}$, and the aspect A and opinion O of interest, they construct a consistency prompt, and a polarity prompt as *The* A *is* O? [MASK]. *This is* [MASK], where the first [MASK] can be filled with *yes* or *no* for consistent or inconsistent A and O, and the second [MASK] can be filled with sentiment polarity words.

3.5 Relation extraction

Relation extraction (RE) has two main subtasks: relation prediction (predicting the relationship r of two given entities s and o conditioned on their context) and triplet extraction (extracting the triplet (s, r, o) from the input text). The former subtask is solved mainly with the Class paradigm[46, 63], while the latter subtask is often solved in the pipeline style that first uses the SeqLab paradigm to extract the entities and then uses the Class paradigm to predict the relationship between the entities. Recent years have seen paradigm shifts in relation extraction, especially in triplet extraction.

Zeng et al.[48] solve the triplet extraction task with the Seq2Seq paradigm. In their framework, the input of the Seq2Seq paradigm is the original text, while the output is a sequence of triplets $\{(r_1, s_1, o_1), \cdots, (r_n, s_n, o_n)\}$. The copy mechanism[64] is adopted to extract entities from the text.

Levy et al.[47] address the RE task via the MRC paradigm by generating relation-specific questions. For instance, for relation *educated_at*(s, o), a question such as "*Where did s graduate from?*" can be crafted to query an MRC model. Moreover, they demonstrate that formulating the RE task with MRC has a potential for zero-shot generalization to unseen relation types. Furthermore, Li et al.[65] and Zhao et al.[66] formulate the triplet extraction task as multi-turn question answering and solve it with the MRC paradigm. They extract entities and relations from the text by progressively asking the MRC model with different questions.

Recently, Han et al.[49] formulat the RE task as an MLM task using logic rules to construct prompts with multiple sub-prompts. By encoding prior knowledge of entities and relations into prompts, their proposed model, prompt tuning with rules (PTR), achieve state-of-the-art performance on multiple RE datasets.

3.6 Text summarization

Text summarization aims to generate a concise and informative summary of large texts. There are two different approaches to solving the text summarization task: extractive summarization and abstractive summarization. Extractive summarization approaches extract the clauses of the original text to form the final summary, which usually lies in the SeqLab paradigm. In contrast, abstractive summarization approaches usually adopt the Seq2Seq paradigm to directly generate a summary conditioned on the original text.

McCann et al.[38] reformulate the summarization task as a question answering task, where the question is "*What is the summary?*". Since the answer (i.e., the summary) is not necessarily comprised of the tokens in the original text, traditional MRC models cannot handle this. Therefore, the authors developed a Seq2Seq model to solve the summarization task in such a format.

Zhong et al.[51] propose to solve the extractive summarization task in the Matching paradigm instead of the SeqLab paradigm. The main idea is to match the semantics of the original text and each candidate summary, finding the summary with the highest matching score. Compared to traditional methods of extracting sentences individually, the matching framework enables the summary extractor to work at a summary level rather than a sentence level.

Aghajanyan et al.[52] formulate the text summarization task in the (M)LM paradigm. They pre-train a BART-style model directly on large-scale structured HTML web pages. Due to the rich semantics encoded in the HTML keywords, their pre-trained model is able to perform zero-shot text summarization by predicting the ⟨title⟩ element given the ⟨body⟩ of the document.

3.7 Parsing

Parsing (constituency parsing, dependency parsing, semantic parsing, etc.) plays a crucial role in many NLP applications such as machine translation and question answering. This family of tasks is to derive a structured syntactic or semantic representation from a natural language utterance. Two commonly used approaches for parsing are transition-based methods and graph-based methods. Typically, transition-based methods lie in the Seq2ASeq paradigm, and graph-based methods lie in the Class paradigm.

By linearizing the target tree-structure to a sequence, parsing can be solved in the Seq2Seq paradigm[56, 67–69], the SeqLab paradigm[54, 70–72], and the (M)LM paradigm[53]. In addition, Gan et al.[55] employ the MRC paradigm to extract the parent span given the original sentence as the context and the child span as the question, achieving state-of-the-art performance on dependency parsing tasks across various languages.

3.8 Trends of paradigm shift

To intuitively depict the trend of paradigm shifts, we draw a Sankey diagram[2] in Fig. 2. We track the development of the NLP tasks considered in this section, along with several additional common tasks such as event extraction. When a task is solved using a paradigm that is different from its original paradigm, some of the values of the original paradigm are transferred to the new paradigm. In particular, for each NLP task of interest, we collect published papers that solve this task from 2012 to 2021 and denote the paradigm used in 2012 as the original paradigm of this task. Then we track the paradigm shifts in all the tasks with the same original paradigm and count the number of tasks that observed paradigm shifts until 2021. For each paradigm, we denote N as the total number of tasks that branched out to new paradigms. Assume that the initial value of each paradigm is 100, and the transferred value for each outbranch is defined as $100/(N+1)$. Therefore, each branch in Fig. 2 indicates a task that shifted its paradigm. Table 2 lists the source data of Fig. 2.

As shown in Fig. 2, we find that: 1) The frequency of paradigm shifts has been increasing in recent years, especially after the emergence of pre-trained language models (PTMs). Therefore, to fully utilize the power of these PTMs, a better way is to reformulate various NLP tasks into the paradigms that PTMs are good at. 2) More and more NLP tasks have shifted from traditional paradigms such as Class, SeqLab, and Seq2ASeq, to paradigms that are more general and flexible, i.e., (M)LM, Matching, MRC, and Seq2Seq, which will be discussed in Section 4.

4 Potential unified paradigms in NLP

Some of the paradigms have demonstrated the potential ability to formulate various NLP tasks into a unified framework. Instead of solving each task separately, such paradigms provide the possibility that a single deployed model can serve as a unified solver for diverse NLP tasks. The advantages of a single unified model over multiple task-specific models can be summarized as follows:

1) **Data efficiency.** Training task-specific models usually requires large-scale task-specific labeled data. In contrast, the unified model has shown its ability to achieve considerable performance with much less labeled data.

2) **Generalization.** Task-specific models are hard to transfer to new tasks, whereas the unified model can generalize to unseen tasks by formulating them into proper formats.

3) **Convenience.** The unified models are easier and cheaper to deploy and serve, making them favorable as commercial black-box APIs.

In this section, we discuss the following general paradigms that have the potential to unify diverse NLP tasks: (M)LM, Matching, MRC, and Seq2Seq.

4.1 (M)LM

Reformulating downstream tasks into an (M)LM task is a natural way to utilize the pre-trained LMs. The original input is modified with a pre-defined or learned prompt with some unfilled slots, which can be filled by the pre-trained LMs. Then the task labels can be derived from the filled tokens. For instance, a movie review "*I love this movie*" can be modified by appending a prompt as "*I love this movie. It was* [MASK]", in which [MASK] may be predicted as "*fantastic*" by the LM. Then the word "*fantastic*" can be mapped to the label "*positive*" by a verbalizer. Solving downstream tasks in the (M)LM paradigm is also referred to as prompt-based learning. By fully utilizing the pre-trained parameters of the MLM head instead of training a classification head from scratch, prompt-based learning has demonstrated great power in few-shot and even zero-shot settings[76].

1) Prompt

The choice of prompt is critical for the performance of a particular task. A good prompt can be **i) Manually designed**. Brown et al.[5, 7, 8] manually craft task-specific prompts for different tasks. Though it is heuristic and sometimes non-intuitive, hand-crafted prompts have already achieved competitive performance on various few-shot tasks. **ii) Mined from corpora**. Jiang et al.[77] construct prompts for relation extraction by mining sentences with the same subject and object in the corpus. **iii) Generated by paraphrasing**. Jiang et al.[77] use back translation to paraphrase the original prompt into multiple new prompts. **iv) Generated by another pre-trained language model**. Gao et al.[9] generate prompts using T5[4] since it is pre-trained to fill in missing spans in the input. **v) Learned by gradient descent**. Shin et al.[10] automatically construct prompts based on gradient-guided search. If the prompt is not necessarily discrete, it can be optimized efficiently in the continuous space. Recent works[11, 12, 78–80] have shown that continuous prompts can also achieve competitive or even better performance.

2) Verbalizer

The design of the verbalizer also has a strong influence on the performance of prompt-based learning[9]. A verbalizer can be **i) Manually designed**. Schick and Schütze[7] manually design verbalizers for different tasks and achieved competitive results. However, it is not always intuitive for many tasks (e.g., when class labels do not directly correspond to words in the vocabulary) to manually design proper verbalizers. **ii) Automatically searched** on a set of labelled data by minimizing some objective, such as negative log likelihood[9, 10, 12, 81]. **iii) Constructed and refined with knowledge base**[82].

[2] Sankey diagram is a visualization used to depict data flows. Our sankey diagram is generated by http://sankey-diagram-generator.acquireprocure.com.

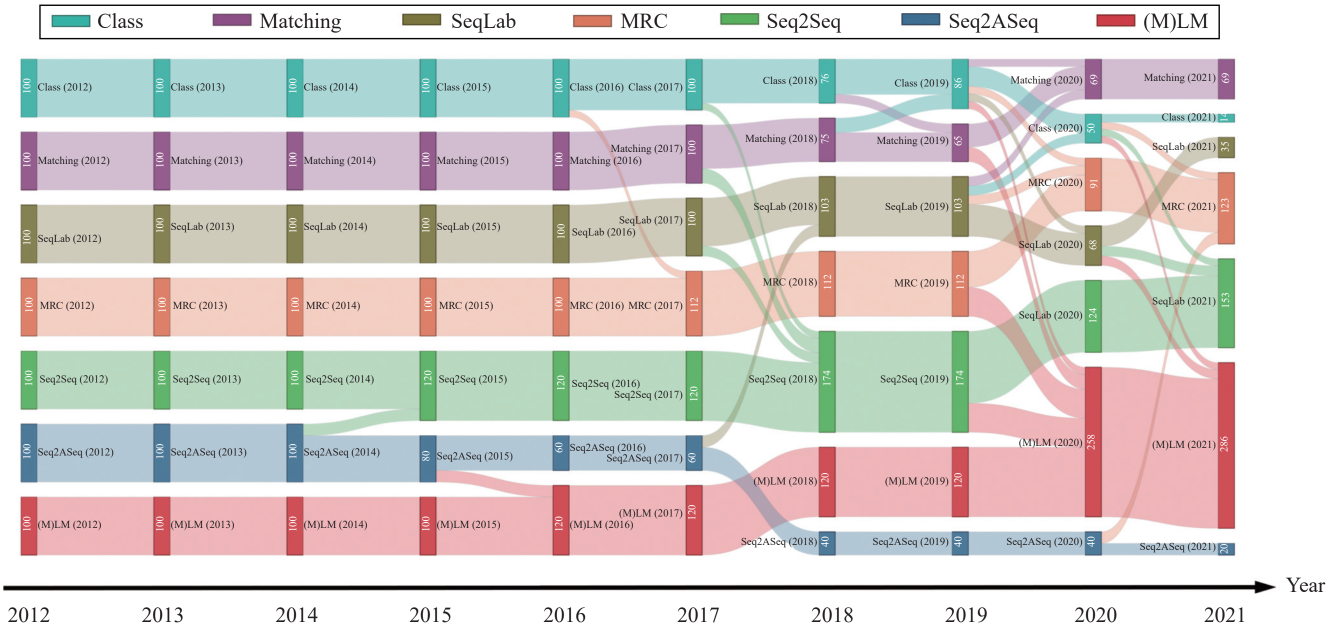


Fig. 2 Sankey diagram depicts the trend of paradigm shifting and unifying in natural language processing tasks. In Section 3.8, we show how this diagram is drawn.

3) Parameter-efficient prompt tuning

Compared with fine-tuning, where all model parameters need to be tuned for each task, prompt-based tuning is also favorable in its parameter efficiency. A recent study[13] has demonstrated that tuning only the prompt parameters while keeping the backbone model parameters fixed can achieve a comparable performance with standard fine-tuning when models exceed billions of parameters. Due to the parameter efficiency, prompt-based tuning is a promising technique for the deployment of large-scale pre-trained LMs. **In traditional fine-tuning**, the server has to maintain a task-specific copy of the entire pre-trained LM for each downstream task, and the inference has to be performed in separate batches. **In prompt-based tuning**, only a single pre-trained LM is required, and different tasks can be performed by modifying the inputs with task-specific prompts. Besides, inputs from different tasks can be mixed in the same batch, making the service highly efficient. In the case of extremely large PTMs, due to their small intrinsic dimensionalities, the prompt can be optimized with derivative-free optimization methods[14], which encourages a novel scenario, namely Language-Model-as-a-Service (LMaaS).

4.2 Matching

Another potential unified paradigm is Matching, or more specifically, textual entailment (a.k.a. natural language inference). Textual entailment is the task of predicting two given sentences, premise and hypothesis: Whether the premise entails the hypothesis, contradicts the hypothesis, or neither. Almost all text classification tasks can be reformulated as a textual entailment task[57, 75, 83, 84]. For example, a labeled movie review {x: *I love this movie*, y: *positive*} can be modified as {x: *I love this movie* [SEP] *This is a great movie*, y: *entailment*}. Similar to pre-trained LMs, entailment models are also widely accessible. Such universal entailment models can be pre-trained LMs that are fine-tuned on some large-scale annotated entailment datasets such as the multi-genre natural language inference (MultiNLI) dataset[85]. In addition to obtaining the entailment model in a supervised fashion, Sun et al.[86] show that the next sentence prediction head of BERT, without training on any supervised entailment data, can also achieve competitive performance on various zero-shot tasks.

1) Domain adaptation

The entailment model may be biased to the source domain, resulting in poor generalization to target domains. To mitigate the domain difference between the source task and the target task, Yin et al.[75] propose the cross-task nearest neighbor module that matches instance representations and class representations in the source domain and the target domain, such that the entailment model can generalize well to new NLP tasks with limited annotations.

2) Label descriptions

For single-sentence classification tasks, the label descriptions for each class are required to be concatenated with the input text to be predicted by the entailment model. Label descriptions can be regarded as a kind of prompt to trigger the entailment model. Wang et al.[57] show that hand-crafted label descriptions with minimum domain knowledge can achieve state-of-the-art performance on various few-shot tasks. Nevertheless, human-written label descriptions can be sub-optimal, Chai et al.[36]

Table 2 Source data of Fig. 2. We only list the first work for each paradigm shift.

Year	Task	Original paradigm	Shifted paradigm	Paper
2015	Parsing	Seq2ASeq	Seq2Seq	[56]
2016	Parsing	Seq2ASeq	(M)LM	[53]
2017	Relation extraction	Class	MRC	[47]
2018	Text summarization	SeqLab	Seq2Seq	[38]
2018	Parsing	Seq2ASeq	SeqLab	[70]
2018	Natural language inference	Matching	Seq2Seq	[38]
2018	Text classification	Class	Seq2Seq	[38]
2018	Relation extraction	Class	Seq2Seq	[48]
2019	Sentiment analysis	Class	Matching	[42]
2019	Natural Language inference	Matching	Class	[3]
2020	Named entity recognition	SeqLab	Class	[62]
2020	Named entity recognition	SeqLab	MRC	[1]
2020	Text summarization	SeqLab	Matching	[51]
2020	Event extraction	Class	MRC	[73]
2020	Event extraction	Class	SeqLab	[74]
2020	Text classification	Class	Matching	[75]
2020	Text classification	Class	(M)LM	[5]
2020	Question answering	MRC	(M)LM	[5]
2020	Machine translation	Seq2Seq	(M)LM	[5]
2020	Natural language inference	Matching	(M)LM	[5]
2021	Named entity recognition	SeqLab	Seq2Seq	[2]
2021	Named entity recognition	SeqLab	(M)LM	[40]
2021	Sentiment analysis	Class	MRC	[43]
2021	Sentiment analysis	Class	Seq2Seq	[44]
2021	Sentiment analysis	Class	(M)LM	[7]
2021	Parsing	Seq2ASeq	MRC	[55]

utilize reinforcement learning to generate label descriptions.

3) Comparison with prompt-based learning

In both paradigms ((M)LM and Matching), the goal is to reformulate the downstream tasks into the pre-training task (language modeling or entailment). To achieve this, both of them need to modify the input text with some templates to prompt the pre-trained language or entailment model. In prompt-based learning, the prediction is conducted by the pre-trained MLM head on the [MASK] token, while in matching-based learning, the prediction is conducted by the pre-trained classifier on the [CLS] token. In prompt-based learning, the output prediction is over the vocabulary, such that a verbalizer is required to map the predicted word in vocabulary into a task label. In contrast, matching-based learning can simply reuse the output (Entailment/Contradiction/Neutral, or Entailment/NotEntailment). Another benefit of matching-based learning is that one can construct pairwise augmented data to perform contrastive learning, achieving a further improvement of few-shot performance. However, matching-based learning requires large-scale human-annotated entailment data to pre-train an entailment model, and domain difference between the source domain and target domain needs to be handled. Besides, matching-based learning can only be used in understanding tasks, while prompt-based learning can also be used for generation[11, 12].

4.3 MRC

MRC is also an alternative paradigm to unify various NLP tasks by generating task-specific questions and training an MRC model to select the correct span from the input text conditioned on the questions. Take NER as an example, one can recognize the organization entity in the input "*Google was founded in 1998*" by querying an MRC model with "*Google was founded in 1998. Find organizations in the text, including companies, agencies and institutions*" as in [1]. In addition to NER, the MRC framework has also demonstrated competitive performance in entity-relation extraction[65], coreference resolution[87], entity linking[88], dependency parsing[55], dialog state tracking[89], event extraction[73, 90], aspect-based sentiment analysis[43], etc.

The MRC paradigm can be applied as long as the task input can be reformulated as context, question, and answer. Due to its universality, McCann et al.[38] proposed decaNLP to unify ten NLP tasks, including question answering, machine translation, summarization, natural language inference, sentiment analysis, semantic role labeling, relation extraction, goal-oriented dialogue, semantic parsing, and commonsense pronoun resolution in a unified QA format. However, different from previously mentioned works, the answer may not appear in the context and question for some tasks of decaNLP, such as semantic parsing. Therefore, the framework is strictly not an MRC paradigm.

Comparison with prompt-based learning. It is worth noticing that the designed question can be analogous to the prompt in (M)LM. The verbalizer is not necessary for MRC since the answer is a span in the context or question. The predictor, MLM head in the prompt-based learning, can be replaced by a start/end classifier as in traditional MRC models or a pointer network as in [38].

4.4 Seq2Seq

Seq2Seq is a general and flexible paradigm that can handle any task whose input and output can be recast as a sequence of tokens. Early work[38] has explored using the Seq2Seq paradigm to simultaneously solve different classes of tasks. Powered by recent advances of sequence-to-sequence pre-training, such as masked sequence to sequence pre-training (MASS)[91], T5[4], and BART[33], the Seq2Seq paradigm has shown great potential in unifying diverse NLP tasks. Paolini et al.[92] use T5[4] to solve many structured prediction tasks, including joint entity and relation extraction, nested NER, relation classification, semantic role labeling, event extraction, coreference resolution, and dialogue state tracking. Yan et al.[2, 44] use BART[33], equipped with the copy network[64], to unify all NER tasks (flat NER, nested NER, discontinuous NER) and all ABSA tasks (AE, OE, ALSC, AOE, AESC, Pair, Triplet), respectively.

Comparison with other paradigms. Compared with other unified paradigms, Seq2Seq is particularly suited for complicated tasks such as structured prediction. Another benefit is that Seq2Seq is also compatible with other paradigms such as (M)LM[4, 33], MRC[38], etc. Nevertheless, what comes with its versatility is its high latency. Currently, most successful Seq2Seq models are in an auto-regressive fashion, where each generation step depends on the previously generated tokens. Such sequential nature results in inherent latency at inference time. Therefore, more work is needed to develop efficient Seq2Seq models through non-autoregressive methods[93, 94], early exiting[95], or other alternative techniques.

5 Conclusions

Recently, prompt-based tuning, which is to formulate some NLP tasks into an (M)LM task, has exploded in popularity. They can achieve considerable performance with much less training data. In contrast, other potential unified paradigms, i.e., Matching, MRC, and Seq2Seq, are under-explored in the context of pre-training. One of the main reasons is that these paradigms require large-scale annotated data to conduct pre-training, especially Seq2Seq is notorious for being data-hungry.

Nevertheless, these paradigms have their advantages over (M)LM: Matching requires less engineering, MRC is more interpretable, and Seq2Seq is more flexible to handle complicated tasks. Besides, by combining with self-supervised pre-training (e.g., BART[33] and T5[4]), or further pre-training on annotated data with existing language model as initialization (e.g., [57]), these paradigms can achieve competitive performance or even better performance than (M)LM. Therefore, we argue that more attention is needed for the exploration of more powerful entailment, MRC, or Seq2Seq models through pre-training or other alternative techniques.

Acknowledgements

This work was supported by National Natural Science Foundation of China (No. 62022027).

Open Access

References

[1] X. Y. Li, J. R. Feng, Y. X. Meng, Q. H. Han, F. Wu, J. W. Li. A unified MRC framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, ACL, pp. 5849–5859, 2020. DOI: 10.18653/v1/2020.acl-main.519.

[2] H. Yan, T. Gui, J. Q. Dai, Q. P. Guo, Z. Zhang, X. P. Qiu. A unified generative framework for various NER subtasks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, ACL, pp. 5808–5822, 2021. DOI: 10.18653/v1/2021.acl-long.451.

[3] J. Devlin, M. W. Chang, K. Lee, K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, ACL, Minneapolis, USA, pp. 4171–4186, 2019. DOI: 10.18653/v1/N19-1423.

[4] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Q. Zhou, W. Li, P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[5] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei. Language models are few-shot learners. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2020.

[6] X. P. Qiu, T. X. Sun, Y. G. Xu, Y. F. Shao, N. Dai, X. J. Huang. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, vol. 63, no. 10, pp. 1872–1897, 2020. DOI: 10.1007/s11431-020-1647-3.

[7] T. Schick, H. Schütze. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, ACL, pp. 255–269, 2021. DOI: 10.18653/v1/2021.eacl-main.20.

[8] T. Schick, H. Schütze. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, ACL, pp. 2339–2352, 2021. DOI: 10.18653/v1/2021.naacl-main.185.

[9] T. Y. Gao, A. Fisch, D. Q. Chen. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, ACL, pp. 3816–3830, 2021. DOI: 10.18653/v1/2021.acl-long.295.

[10] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, S. Singh. AutoPrompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, ACL, pp. 4222–4235, 2020. DOI: 10.18653/v1/2020.emnlp-main.346.

[11] X. L. Li, P. Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, ACL, pp. 4582–4597, 2021. DOI: 10.18653/v1/2021.acl-long.353.

[12] X. Liu, Y. N. Zheng, Z. X. Du, M. Ding, Y. J. Qian, Z. L. Yang, J. Tang. GPT understands, too. [Online], Available: https://arxiv.org/abs/2103.10385, 2021.

[13] B. Lester, R. Al-Rfou, N. Constant. The power of scale for parameter-efficient prompt tuning. [Online], Available: https://arxiv.org/abs/2104.08691, 2021.

[14] T. X. Sun, Y. F. Shao, H. Qian, X. J. Huang, X. P. Qiu. Black-box tuning for language-model-as-a-service. [Online], Available: https://arxiv.org/abs/2201.03514, 2022.

[15] Y. Kim. Convolutional neural networks for sentence classification. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, ACL, Doha, Qatar, pp. 1746–1751, 2014. DOI: 10.3115/v1/D14-1181.

[16] P. F. Liu, X. P. Qiu, X. J. Huang. Recurrent neural network for text classification with multi-task learning. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, New York, USA, pp. 2873–2879, 2016.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, \L. Kaiser, I. Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Long Beach, USA, pp. 6000–6010, 2017.

[18] Q. Chen, X. D. Zhu, Z. H. Ling, S. Wei, H. Jiang, D. Inkpen. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, ACL, Vancouver, Canada, pp. 1657–1668, 2017. DOI: 10.18653/v1/P17-1152.

[19] X. Z. Ma, E. Hovy. End-to-end sequence labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, ACL, Berlin, Germany, pp. 1064–1074, 2016. DOI: 10.18653/v1/P16-1101.

[20] J. D. Lafferty, A. McCallum, F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, Morgan Kaufmann Publishers Inc., Williamstown, USA, pp. 282–289, 2001.

[21] C. M. Xiong, V. Zhong, R. Socher. Dynamic coattention networks for question answering. In *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France, 2017.

[22] M. J. Seo, A. Kembhavi, A. Farhadi, H. Hajishirzi. Bidirectional attention flow for machine comprehension. In *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France, 2017.

[23] D. Q. Chen, A. Fisch, J. Weston, A. Bordes. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, ACL, Vancouver, Canada, pp. 1870–1879, 2017. DOI: 10.18653/v1/P17-1171.

[24] I. Sutskever, O. Vinyals, Q. V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Pro-*

cessing Systems, Montreal, Canada, pp. 3104–3112, 2014.

[25] D. Bahdanau, K. Cho, Y. Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, USA, 2015.

[26] M. T. Luong, Q. V. Le, I. Sutskever, O. Vinyals, L. Kaiser. Multi-task sequence to sequence learning. In *Proceedings of the 4th International Conference on Learning Representations*, San Juan, Puerto Rico, 2016.

[27] J. Gehring, M. Auli, D. Grangier, D. Yarats, Y. N. Dauphin. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia, pp. 1243–1252, 2017.

[28] D. Q. Chen, C. D. Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, ACL, Doha, Qatar, pp. 740–750, 2014.

[29] C. Dyer, M. Ballesteros, W. Ling, A. Matthews, N. A. Smith. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, ACL, Beijing, China, pp. 334–343, 2015. DOI: 10.3115/v1/P15-1033.

[30] Y. Bengio, R. Ducharme, P. Vincent. A neural probabilistic language model. In *Proceedings of the 13th International Conference on Neural Information Processing Systems*, Denver, USA, pp. 932–938, 2000.

[31] E. Grave, A. Joulin, N. Usunier. Improving neural language models with a continuous cache. In *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France, 2017.

[32] Z. H. Dai, Z. L. Yang, Y. M. Yang, J. Carbonell, Q. Le, R. Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, ACL, Florence, Italy, pp. 2978–2988, 2019. DOI: 10.18653/v1/P19-1285.

[33] M. Lewis, Y. H. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer. 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*, ACL, pp. 7871–7880, 2020. DOI: 10.18653/v1/2020.acl-main.703.

[34] Z. L. Yang, P. Qi, S. Z. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, C. D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, ACL, Brussels, Belgium, pp. 2369–2380, 2018. DOI: 10.18653/v1/D18-1259.

[35] B. H. Wu, Z. S. Zhang, H. Zhao. Graph-free multi-hop reading comprehension: A select-to-guide strategy. [Online], Available: https://arxiv.org/abs/2107.11823, 2021.

[36] D. Chai, W. Wu, Q. H. Han, F. Wu, J. W. Li. Description based text classification with reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 1371–1382, 2020.

[37] P. C. Yang, X. Sun, W. Li, S. M. Ma, W. Wu, H. F. Wang. SGM: Sequence generation model for multi-label classification. In *Proceedings of the 27th International Conference on Computational Linguistics*, ACL, Santa Fe, USA, pp. 3915–3926, 2018.

[38] B. McCann, N. S. Keskar, C. M. Xiong, R. Socher. The natural language decathlon: Multitask learning as question answering. [Online], Available: https://arxiv.org/abs/1806.08730, 2018.

[39] J. L. Fu, X. J. Huang, P. F. Liu. SpanNER: Named entity re-/recognition as span prediction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, ACL, pp. 7183–7195, 2021. DOI: 10.18653/v1/2021.acl-long.558.

[40] L. Y. Cui, Y. Wu, J. Liu, S. Yang, Y. Zhang. Template-based named entity recognition using BART. In *Proceedings of the Findings of the Association for Computational Linguistics*, ACL, pp. 1835–1845, 2021. DOI: 10.18653/v1/2021.findings-acl.161.

[41] Y. Q. Wang, M. L. Huang, L. Zhao, X. Y. Zhu. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, ACL, Austin, USA, pp. 606–615, 2016. DOI: 10.18653/v1/D16-1058.

[42] C. Sun, L. Y. Huang, X. P. Qiu. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, ACL, Minneapolis, USA, pp. 380–385, 2019. DOI: 10.18653/v1/N19-1035.

[43] Y. Mao, Y. Shen, C. Yu, L. J. Cai. A joint training dual-MRC framework for aspect based sentiment analysis. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, Palo Alto, USA, pp. 13543–13551, 2021.

[44] H. Yan, J. Q. Dai, T. Ji, X. P. Qiu, Z. Zhang. A unified generative framework for aspect-based sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, ACL, pp. 2416–2429, 2021. DOI: 10.18653/v1/2021.acl-long.188.

[45] C. X. Li, F. Y. Gao, J. J. Bu, L. Xu, X. Chen, Y. Gu, Z. R. Shao, Q. Zheng, N. Y. Zhang, Y. P. Wang, Z. Yu. SentiPrompt: Sentiment knowledge enhanced prompt-tuning for aspect-based sentiment analysis. [Online], Available: https://arxiv.org/abs/2109.08306, 2021.

[46] D. J. Zeng, K. Liu, S. W. Lai, G. Y. Zhou, J. Zhao. Relation classification via convolutional deep neural network. In *Proceedings of the 25th International Conference on Computational Linguistics*, ACL, Dublin, Ireland, pp. 2335–2344, 2014.

[47] O. Levy, M. Seo, E. Choi, L. Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning*, ACL, Vancouver, Canada, pp. 333–342, 2017. DOI: 10.18653/v1/K17-1034.

[48] X. R. Zeng, D. J. Zeng, S. Z. He, K. Liu, J. Zhao. Extract-

ing relational facts by an end-to-end neural model with copy mechanism. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, ACL, Melbourne, Australia, pp. 506–514, 2018. DOI: 10.18653/v1/P18-1047.

[49] X. Han, W. L. Zhao, N. Ding, Z. Y. Liu, M. S. Sun. PTR: Prompt tuning with rules for text classification. [Online], Available: https://arxiv.org/abs/2105.11259, 2021.

[50] J. P. Cheng, M. Lapata. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, ACL, Berlin, Germany, pp. 484–494, 2016. DOI: 10.18653/v1/P16-1046.

[51] M. Zhong, P. F. Liu, Y. R. Chen, D. Q. Wang, X. P. Qiu, X. J. Huang. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, ACL, pp. 6197–6208, 2020. DOI: 10.18653/v1/2020.acl-main.552.

[52] A. Aghajanyan, D. Okhonko, M. Lewis, M. Joshi, H. Xu, G. Ghosh, L. Zettlemoyer. HTLM: Hyper-text pre-training and prompting of language models. [Online], Available: https://arxiv.org/abs/2107.06955, 2021.

[53] D. K. Choe, E. Charniak. Parsing as language modeling. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, ACL, Austin, USA, pp. 2331–2336, 2016. DOI: 10.18653/v1/D16-1257.

[54] M. Strzyz, D. Vilares, C. Gómez-Rodríguez. Viable dependency parsing as sequence labeling. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, ACL, Minneapolis, USA, pp. 717–723, 2019. DOI: 10.18653/v1/N19-1077.

[55] L. L. Gan, Y. X. Meng, K. Kuang, X. F. Sun, C. Fan, F. Wu, J. W. Li. Dependency parsing as MRC-based span-span prediction. [Online], Available: https://arxiv.org/abs/2105.07654, 2021.

[56] O. Vinyals, L. Kaiser, T. Koo, S. Petrov, I. Sutskever, G. E. Hinton. Grammar as a foreign language. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, Montreal, Canada, pp. 2773–2781, 2015.

[57] S. N. Wang, H. Fang, M. Khabsa, H. Z. Mao, H. Ma. Entailment as few-shot learner. [Online], Available: https://arxiv.org/abs/2104.14690, 2021.

[58] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer. Neural architectures for named entity recognition. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, ACL, San Diego, USA, pp. 260–270, 2016. DOI: 10.18653/v1/N16-1030.

[59] C. Y. Xia, C. W. Zhang, T. Yang, Y. L. Li, N. Du, X. Wu, W. Fan, F. L. Ma, P. Yu. Multi-grained named entity recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, ACL, Florence, Italy, pp. 1430–1440, 2019. DOI: 10.18653/v1/P19-1138.

[60] J. Fisher, A. Vlachos. Merge and label: A novel neural network architecture for nested NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, ACL, Florence, Italy, pp. 5840–5850, 2019. DOI: 10.18653/v1/P19-1585.

[61] X. Dai, S. Karimi, B. Hachey, C. Paris. An effective transition-based model for discontinuous NER. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, ACL, pp. 5860–5870, 2020. DOI: 10.18653/v1/2020.acl-main.520.

[62] J. T. Yu, B. Bohnet, M. Poesio. Named entity recognition as dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, ACL, pp. 6470–6476, 2020. DOI: 10.18653/v1/2020.acl-main.577.

[63] T. X. Sun, Y. F. Shao, X. P. Qiu, Q. P. Guo, Y. R. Hu, X. J. Huang, Z. Zhang. CoLAKE: Contextualized language and knowledge embedding. In *Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, Spain, pp. 3660–3670, 2020. DOI: 10.18653/v1/2020.coling-main.327.

[64] J. T. Gu, Z. D. Lu, H. Li, V. O. K. Li. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, ACL, Berlin, Germany, pp. 1631–1640, 2016. DOI: 10.18653/v1/P16-1154.

[65] X. Y. Li, F. Yin, Z. J. Sun, X. Y. Li, A. Yuan, D. Chai, M. X. Zhou, J. W. Li. Entity-relation extraction as multi-turn question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, ACL, Florence, Italy, pp. 1340–1350, 2019. DOI: 10.18653/v1/P19-1129.

[66] T. Y. Zhao, Z. Yan, Y. B. Cao, Z. J. Li. Asking effective and diverse questions: A machine reading comprehension based framework for joint entity-relation extraction. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, Yokohama, Japan, pp. 3948–3954, 2020.

[67] J. Andreas, A. Vlachos, S. Clark. Semantic parsing as machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, ACL, Sofia, Bulgaria, pp. 47–52, 2013.

[68] Z. C. Li, J. X. Cai, S. X. He, H. Zhao. Seq2seq dependency parsing. In *Proceedings of the 27th International Conference on Computational Linguistics*, ACL, Santa Fe, USA, pp. 3203–3214, 2018.

[69] S. Rongali, L. Soldaini, E. Monti, W. Hamza. Don't parse, generate! A sequence to sequence architecture for task-oriented semantic parsing. In *Proceedings of the Web Conference 2020*, ACM, Taipei, China, pp. 2962–2968, 2020. DOI: 10.1145/3366423.3380064.

[70] C. Gómez-Rodríguez, D. Vilares. Constituent parsing as sequence labeling. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, ACL, Brussels, Belgium, pp. 1314–1324, 2018. DOI: 10.18653/v1/D18-1162.

[71] D. Vilares, C. Gómez-Rodríguez. Discontinuous constituent parsing as sequence labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, ACL, pp. 2771–2785, 2020. DOI: 10.

18653/v1/2020.emnlp-main.221.

[72] R. Vacareanu, G. C. G. Barbosa, M. A. Valenzuela-Escárcega, M. Surdeanu. Parsing as tagging. In *Proceedings of the 12th Language Resources and Evaluation Conference*, Marseille, France, pp. 5225–5231, 2020.

[73] J. Liu, Y. B. Chen, K. Liu, W. Bi, X. J. Liu. Event extraction as machine reading comprehension. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, ACL, pp. 1641–1651, 2020. DOI: 10.18653/v1/2020.emnlp-main.128.

[74] A. Ramponi, R. van der Goot, R. Lombardo, B. Plank. Biomedical event extraction as sequence labeling. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, ACL, pp. 5357–5367, 2020. DOI: 10.18653/v1/2020.emnlp-main.431.

[75] W. P. Yin, N. F. Rajani, D. Radev, R. Socher, C. M. Xiong. Universal natural language processing with limited annotations: Try few-shot textual entailment as a start. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, ACL, pp. 8229–8239, 2020. DOI: 10.18653/v1/2020.emnlp-main.660.

[76] T. Le Scao, A. Rush. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, ACL, pp. 2627–2636, 2021. DOI: 10.18653/v1/2021.naacl-main.208.

[77] Z. B. Jiang, F. F. Xu, J. Araki, G. Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 423–438, 2020. DOI: 10.1162/tacl_a_00324.

[78] G. H. Qin, J. Eisner. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, ACL, pp. 5203–5212, 2021. DOI: 10.18653/v1/2021.naacl-main.410.

[79] K. Hambardzumyan, H. Khachatrian, J. May. WARP: Word-level adversarial ReProgramming. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, ACL, pp. 4921–4933, 2021. DOI: 10.18653/v1/2021.acl-long.381.

[80] Z. X. Zhong, D. Friedman, D. Q. Chen. Factual probing is [MASK]: Learning vs. learning to recall. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, ACL, pp. 5017–5033, 2021. DOI: 10.18653/v1/2021.naacl-main.398.

[81] T. Schick, H. Schmid, H. Schütze. Automatically identifying words that can serve as labels for few-shot text classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, Spain, pp. 5569–5578, 2020. DOI: 10.18653/v1/2020.coling-main.488.

[82] S. D. Hu, N. Ding, H. D. Wang, Z. Y. Liu, J. G. Wang, J. Z. Li, W. Wu, M. S. Sun. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. [Online], Available: https://arxiv.org/abs/2108.02035, 2021.

[83] I. Dagan, O. Glickman, B. Magnini. The PASCAL recognising textual entailment challenge. In *Proceedings of the 1st Machine Learning Challenges Workshop*, Springer, Southampton, UK, pp. 177–190, 2005. DOI: 10.1007/11736790_9.

[84] A. Poliak, A. Haldar, R. Rudinger, J. E. Hu, E. Pavlick, A. S. White, B. Van Durme. Collecting diverse natural language inference problems for sentence representation evaluation. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, ACL, Brussels, Belgium, pp. 67–81, 2018. DOI: 10.18653/v1/D18-1007.

[85] A. Williams, N. Nangia, S. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, ACL, New Orleans, USA, pp. 1112–1122, 2018. DOI: 10.18653/v1/N18-1101.

[86] Y. Sun, Y. Zheng, C. Hao, H. P. Qiu. NSP-BERT: A prompt-based zero-shot learner through an original pre-training task-next sentence prediction. [Online], Available: https://arxiv.org/abs/2109.03564, 2021.

[87] W. Wu, F. Wang, A. Yuan, F. Wu, J. W. Li. CorefQA: Coreference resolution as query-based span prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, ACL, pp. 6953–6963, 2020. DOI: 10.18653/v1/2020.acl-main.622.

[88] Y. J. Gu, X. Y. Qu, Z. F. Wang, B. X. Huai, N. J. Yuan, X. L. Gui. Read, retrospect, select: An MRC framework to short text entity linking. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, Palo Alto, USA, pp. 12920–12928, 2021.

[89] S. Y. Gao, A. Sethi, S. Agarwal, T. Chung, D. Hakkani-Tur. Dialog state tracking: A neural reading comprehension approach. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, ACL, Stockholm, Sweden, pp. 264–273, 2019. DOI: 10.18653/v1/W19-5932.

[90] X. Y. Du, C. Cardie. Event extraction by answering (almost) natural questions. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, ACL, pp. 671-683, 2020. DOI: 10.18653/v1/2020.emnlp-main.49.

[91] K. T. Song, X. Tan, T. Qin, J. F. Lu, T. Y. Liu. MASS: Masked sequence to sequence pre-training for language generation. In *Proceedings of the 36th International Conference on Machine Learning*, Long Beach, USA, pp. 5926–5936, 2019.

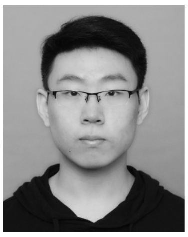
[92] G. Paolini, B. Athiwaratkun, J. Krone, J. Ma, A. Achille, R. Anubhai, C. N. dos Santos, B. Xiang, S. Soatto. Structured prediction as translation between augmented natural languages. In *Proceedings of the 9th International Conference on Learning Representations*, Vienna, Austria, 2021.

[93] J. T. Gu, J. Bradbury, C. M. Xiong, V. O. K. Li, R. Socher. Non-autoregressive neural machine translation. In *Proceedings of the 6th International Conference on Learning Representations*, Vancouver, Canada, 2018.

[94] W. Z. Qi, Y. Y. Gong, J. Jiao, Y. Yan, W. Z. Chen, D. Liu, K. W. Tang, H. Q. Li, J. S. Chen, R. F. Zhang, M. Zhou, N. Duan. BANG: Bridging autoregressive and non-autoregressive generation with large scale pretraining. In *Proceedings of the 38th International Conference on Machine*

Learning, pp. 8630–8639, 2021.

[95] M. Elbayad, J. T. Gu, E. Grave, M. Auli. Depth-adaptive transformer. In *Proceedings of the 8th International Conference on Learning Representations*, Addis Ababa, Ethiopia, 2020.



Tian-Xiang Sun received the B. Eng. degree in software engineering from Xidian University, China in 2019. During 2019−2020, he was an applied scientist intern at Amazon Shanghai AI Lab, China. Since 2019, he is the Ph. D. degree candidate in School of Computer Science, Fudan University, China. He serves as a reviewer of ICML, ACL, EMNLP, AAAI, IJCAI, and COLING.

His research interests include natural language processing and deep learning.

E-mail: txsun19@fudan.edu.cn

ORCID iD: 0000-0001-8291-820X

Xiang-Yang Liu received the B. Eng. degree in intelligent science and technology from Xidian University, China in 2020. Since 2020, he is a master student in School of Computer Science, Fudan University, China.

His research interests include natural language processing and deep learning.

E-mail: xiangyangliu20@fudan.edu.cn

ORCID iD: 0000-0003-0618-9919

Xi-Peng Qiu received the B. Sc. degree and Ph. D. degrees in computer science from Fudan University, China in 2001 and 2006, respectively. Currently, he is a professor in School of Computer Science, Fudan University, China.

His research interests include natural language processing and deep learning.

E-mail: xpqiu@fudan.edu.cn (Corresponding author)

ORCID iD: 0000-0001-7163-5247

Xuan-Jing Huang received the Ph. D. degree in computer science from Fudan University China in 1998. She is currently a professor of School of Computer Science, Fudan University, China. She has served as Program Co-Chair in EMNLP 2021, CCL 2019, CCL 2016, NLPCC 2017, SMP 2015, the organizer of WSDM 2015, and competition chair of CIKM 2014. She has been included in the 2020 Women in AI List and AI 2000 Most Influential Scholar Annual List, jointly announced by Tsinghua − Chinese Academy of Engineering's Joint Research Center for Knowledge and Intelligence, and the Institute for Artificial Intelligence of Tsinghua University, and 2020 Women in Tech List by Forbes China. She has published more than 100 papers in major computer science conferences and journals.

Her research interests include artificial intelligence, natural language processing, information retrieval and social media processing.

E-mail: xjhuang@fudan.edu.cn

ORCID iD: 0000-0001-9197-9426