Paradigm Shift in Natural Language Processing

Tianxiang Sun, Xiangyang Liu, Xipeng Qiu, Xuanjing Huang
School of Computer Science, Fudan University
Shanghai Key Laboratory of Intelligent Information Processing, Fudan University
{txsun19, xiangyangliu20, xpqiu, xjhuang}@fudan.edu.cn

Abstract

In the era of deep learning, modeling for most NLP tasks have converged to several mainstream paradigms. For example, we usually adopt the sequence labeling paradigm to solve a bundle of tasks such as POS-tagging, NER, Chunking, and adopt the classification paradigm to solve tasks like sentiment analysis. With the rapid progress of pre-trained language models, recent years have observed a rising trend of *Paradigm Shift*, which is solving one NLP task by reformulating it as another one. Paradigm shift has achieved great success on many tasks, 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]

1 Introduction

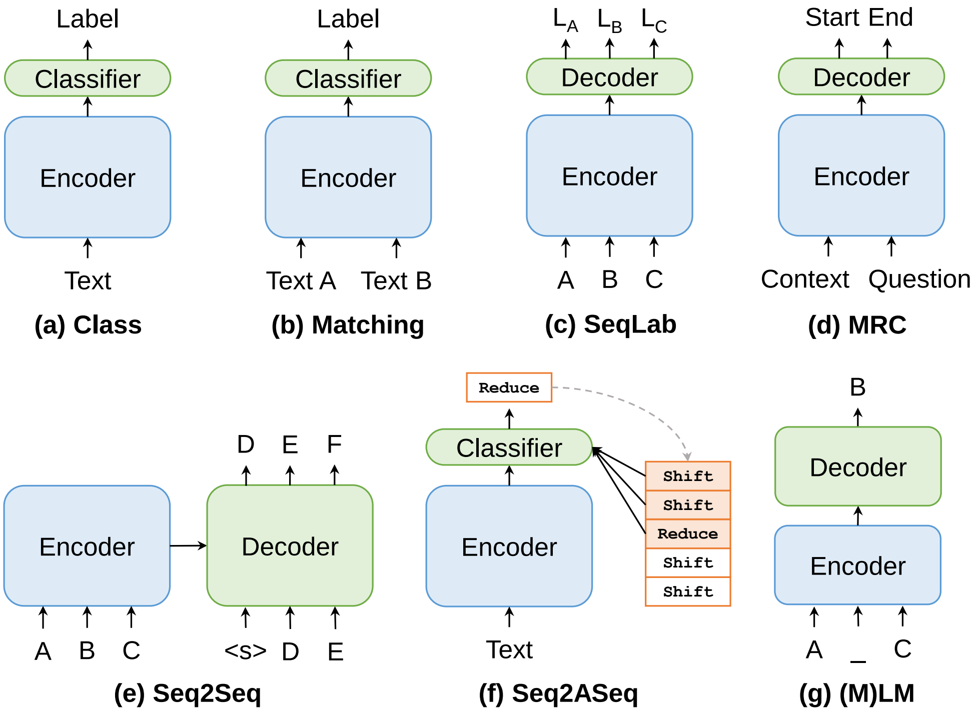
Paradigm is the general framework to model a class of tasks. For instance, sequence labeling is a mainstream paradigm for named entity recognition (NER). Different paradigms usually require different input and output, therefore highly depend on the annotation of the tasks. In the past years, modeling for most NLP tasks have 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` paradigm and the `Seq2Seq` paradigm can also achieve state-of-the-art performance on NER tasks (Li et al., 2020; Yan et al., 2021b), which are 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 NLP task as another one have achieved great success and gained increasing attention in the community. After the emergence of the pre-trained language models (PTMs) (Devlin et al., 2019; Raffel et al., 2020; Brown et al., 2020; Qiu et al., 2020), paradigm shift has 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 (Schick and Schütze, 2021a,b; Gao et al., 2021; Shin et al., 2020; Li and Liang, 2021; Liu et al., 2021b; Lester et al., 2021).

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 on this line of research, namely *paradigm shift* or *paradigm transfer*.

This paper is organized as follows. In section 2, we give formal definitions of the seven paradigms, and introduce their representative tasks and instance models. In section 3, we show recent paradigm shifts happened in different NLP tasks. In section 4, we discuss designs and challenges of several highlighted paradigms that have great potential to unify most existing NLP tasks. In section 5, we conclude with a brief discussion of recent trends and future directions.

[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

Figure 1: Illustration of the seven mainstream paradigms in NLP.

2 Paradigms in NLP

2.1 Paradigms, Tasks, and Models

Typically, a task corresponds to a dataset $\mathcal{D} = \{\mathcal{X}_i, \mathcal{Y}_i\}_{i=1}^{N}$. 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, and a paradigm can be used to solve multiple tasks that can be formulated as the same format. A paradigm can be instantiated by a class of models with similar architectures.

2.2 The 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`, `Seq2ASeq`, and `(M)LM`. These paradigms have demonstrated strong dominance in many mainstream NLP tasks. In the following sections, we 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.*

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

Note that $\mathcal{Y}$ can be one-hot or multi-hot (in which case we call multi-label classification). $\textsc{Enc}(\cdot)$ can be instantiated as convolutional networks (Kim, 2014), recurrent networks (Liu et al., 2016), or Transformers (Vaswani et al., 2017). $\textsc{Cls}(\cdot)$ 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

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

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.* semantic similarity between the two texts). The two texts can be separately encoded and then interact with each other (Chen et al., 2017b), or be concatenated to be fed into a single deep encoder (Devlin et al., 2019).

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, named entity recognition (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.*

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

where $y_1, \cdots, y_n$ are the corresponding labels of $x_1, \cdots, x_n$. $\textsc{Enc}(\cdot)$ can be instantiated as a recurrent network (Ma and Hovy, 2016) or a Transformer encoder (Vaswani et al., 2017). $\textsc{Dec}(\cdot)$ is usually implemented as conditional random fields (CRF) (Lafferty et al., 2001).

2.2.4 MRC

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 task, then is generalized to other NLP tasks by reformulating them into the MRC format. Though, 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,

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

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, $\textsc{Dec}$ is implemented as two classifiers, one for predicting the starting position and one for predicting the ending position (Xiong et al., 2017; Seo et al., 2017; Chen et al., 2017a).

2.2.5 Sequence-to-Sequence (Seq2Seq)

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 a input sequence (source language or user query). Seq2Seq paradigm is typically implemented by an encoder-decoder framework (Sutskever et al., 2014; Bahdanau et al., 2015; Luong et al., 2016; Gehring et al., 2017):

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

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 testing) or the ground truth (with teacher forcing when training).

2.2.6 Sequence-to-Action-Sequence (Seq2ASeq)

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 dependency tree. The instances of the Seq2ASeq paradigm are usually called transition-based models, which can be formulated as

$$\mathcal{A} = \textsc{Cls}(\textsc{Enc}(\mathcal{X}), \mathcal{C}), \qquad (6)$$

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 current configuration c_{t-1}, which can be comprised of top elements in stack, buffer, and previous actions (Chen and Manning, 2014; Dyer et al., 2015).

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 large-scale unlabeled corpus. Typically, a language model can be simply formulated as

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

where Dec can be any auto-regressive model such as recurrent networks (Bengio et al., 2000; Grave et al., 2017) and Transformer decoder (Dai et al., 2019). As a famous variant of LM, MLM can be formulated as

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

where $\tilde{x}$ is a corrupted version of x by replacing a portion of 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 BERT (Devlin et al., 2019) or an auto-regressive Transformer decoder as in BART (Lewis et al., 2020) and T5 (Raffel et al., 2020).

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 mainly focus 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 (Yang et al., 2018b), 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 (Wu et al., 2021).

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.

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. (2018a) propose to adopt the Seq2Seq paradigm to better capture interactions between the labels for multi-label classification tasks.

In addition, the semantics hidden in the labels can not be fully exploited in the Class paradigm. Chai et al. (2020) and Wang et al. (2021) 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 belonging to it, incorporating prior knowledge of the label is also helpful when training data is limited.

As the rise of pre-trained language models (LMs), text classification tasks can also be solved in the (M)LM paradigm (Brown et al., 2020; Schick and Schütze, 2021a,b; Gao et al., 2021). 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 (Chen et al., 2017b). With the emergence of powerful encoder such as BERT (Devlin et al., 2019), 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 (Schick and Schütze, 2021a,b; Gao et al., 2021).

3.3 Named Entity Recognition

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 (Ma and Hovy, 2016; Lample et al., 2016), Class (Xia et al., 2019; Fisher and Vlachos, 2019), and Seq2ASeq (Lample et al., 2016; Dai et al., 2020).

Yu et al. (2020) and Fu et al. (2021) 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

Task		`Class`	`Matching`	`SeqLab`	`MRC`	`Seq2Seq`	`Seq2ASeq`	`(M)LM`
TC	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	Devlin et al. (2019)	Chai et al. (2020)			Yang et al. (2018a)		Schick and Schütze (2021a)
NLI	Input	$\mathcal{X}_a \oplus \mathcal{X}_b$	$\mathcal{X}_a, \mathcal{X}_b$			$f_{prompt}(\mathcal{X}_a, \mathcal{X}_b)$		$f_{prompt}(\mathcal{X}_a, \mathcal{X}_b)$
	Output	$\mathcal{Y}$	$\mathcal{Y}$			$\mathcal{Y}$		$g(\mathcal{Y})$
	Example	Devlin et al. (2019)	Chen et al. (2017b)			McCann et al. (2018)		Schick and Schütze (2021a)
NER	Input	$\mathcal{X}_{span}$		$x_1, \cdots, x_n$	$\mathcal{X}, \mathcal{Q}_y$	$\mathcal{X}$	$(\mathcal{X}, \mathcal{C}_t)_{t=0}^{m-1}$	
	Output	$\mathcal{Y}$		$y_1, \cdots, y_n$	$\mathcal{X}_{span}$	$(\mathcal{X}_{ent_i}, \mathcal{Y}_{ent_i})_{i=1}^{m}$	$\mathcal{A} = a_1, \cdots, a_m$	
	Example	Fu et al. (2021)		Ma and Hovy (2016)	Li et al. (2020)	Yan et al. (2021b)	Lample et al. (2016)	
ABSA	Input	$\mathcal{X}_{asp}$	$\mathcal{X}, \mathcal{S}_{aux}$		$\mathcal{X}, \mathcal{Q}_{asp}, \mathcal{Q}_{opin\&sent}$	$\mathcal{X}$		$f_{prompt}(\mathcal{X})$
	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	Wang et al. (2016)	Sun et al. (2019)		Mao et al. (2021)	Yan et al. (2021a)		Li et al. (2021)
RE	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	Zeng et al. (2014)			Levy et al. (2017)	Zeng et al. (2018)		Han et al. (2021)
Summ	Input		$(\mathcal{X}, \mathcal{S}_{cand_i})_{i=1}^{n}$	$\mathcal{X}_1, \cdots, \mathcal{X}_n$		$\mathcal{X}, \mathcal{Q}_{summ}$		$\mathcal{X}$, Keywords/Prompt
	Output		$\hat{\mathcal{S}}_{cand}$	$\mathcal{Y}_1, \cdots, \mathcal{Y}_n \in \{0, 1\}^n$		$\mathcal{Y}$		$\mathcal{Y}$
	Example		Zhong et al. (2020)	Cheng and Lapata (2016)		McCann et al. (2018)		Aghajanyan et al. (2021)
Parsing	Input			$x_1, \cdots, x_n$	$\mathcal{X}, \mathcal{Q}_{child}$	$\mathcal{X}$	$(\mathcal{X}, \mathcal{C}_t)_{t=0}^{m-1}$	$(\mathcal{X}, \mathcal{Y}_i)_{i=1}^{k}$
	Output			$g(y_1, \cdots, y_n)$	$\mathcal{X}_{parent}$	$g(y_1, \cdots, y_m)$	$\mathcal{A} = a_1, \cdots, a_m$	$\hat{\mathcal{Y}}$
	Example			Strzyz et al. (2019)	Gan et al. (2021)	Vinyals et al. (2015)	Chen and Manning (2014)	Choe and Charniak (2016)

Table 1: Paradigms shift in natural language processing 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. In `(M)LM`, $f(\cdot)$ is usually implemented as a template and $g(\cdot)$ is a verbalizer. In parsing tasks, $g(\cdot)$ is a function that reconstructs the structured representation (*e.g.* dependency tree) from the output sequence. $\mathcal{L}$ 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 configuration t and $\mathcal{A}$ is a sequence of actions. More details can be found in Section 3.

the span overlapping problem: The predicted entities may be overlapped, which is not allowed in flat NER. To handle this, Fu et al. (2021) adopt a heuristic decoding method: For these overlapped spans, only keep the span with the highest prediction probability.

Li et al. (2020) propose to formulate flat NER and nested NER as a 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 entity y, $\mathcal{X}_{span}$ is the answer. Given 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. (2021b) use a unified model based on the Seq2Seq paradigm to solve all the three kinds 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 (Lewis et al., 2020), this unified model achieved state-of-the-art performance on various datasets spanning all the 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 by different paradigms. For example, ALSC can be solved by the Class paradigm, and AESC can be solved by 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. (2019) 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 (Devlin et al., 2019) to predict the sentiment.

Mao et al. (2021) 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 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. (2021a) solve all the ABSA subtasks with the Seq2Seq paradigm by converting the original label of a subtask into a sequence of tokens, which is used as the target to train a seq2seq model. Take the Triplet Extraction subtask as an example, for a input sentence, "*The drinks are always well made and wine selection is fairly priced*", the output target is constructed as "*drinks well made* Positive *wine selection fairly priced* Positive". Equipped with BART (Lewis et al., 2020) as the backbone, they achieved competitive performance on most ABSA subtasks.

Very recently, Li et al. (2021) propose to formulate the 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 Exaction

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 triplet (s, r, o) from the input text). The former subtask is mainly solved with the Class paradigm (Zeng et al., 2014; Sun et al., 2020), 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 shift in relation extraction, especially in triplet extraction.

Zeng et al. (2018) solve the triplet extraction task with the Seq2Seq paradigm. In their framework, the input of the Seq2Seq paradigm is the origi-

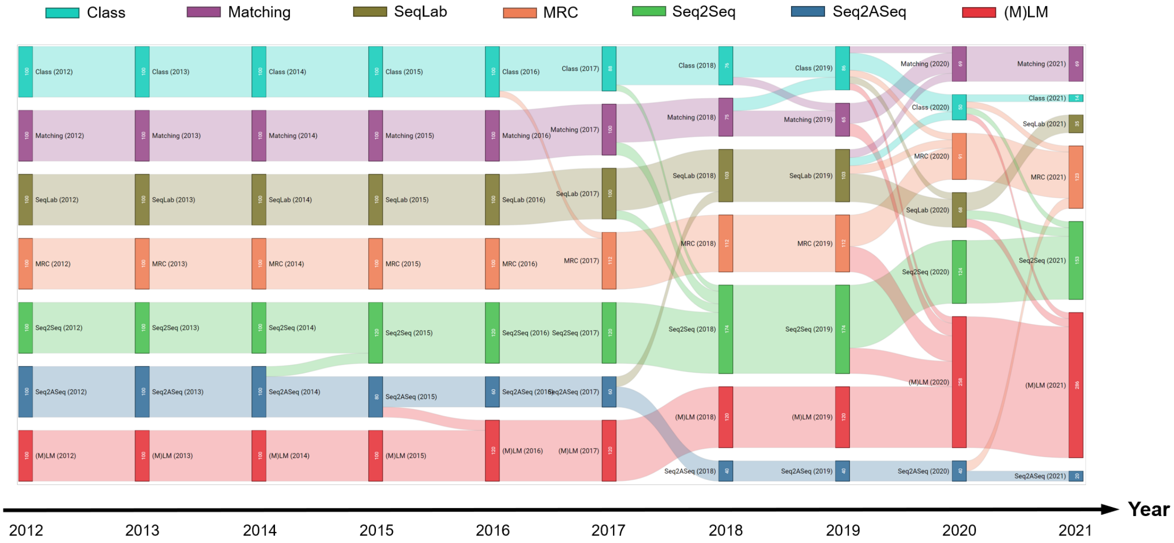


Figure 2: Sankey diagram to depict the trend of paradigm shifting and unifying in natural language processing tasks. In Section 3.8 we show how this diagram is drawn.

nal 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 (Gu et al., 2016) is adopted to extract entities in the text.

Levy et al. (2017) 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 a MRC model. Moreover, they demonstrate that formulating the RE task with MRC has a potential of zero-shot generalization to unseen relation types. Further, Li et al. (2019) and Zhao et al. (2020) 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.

Very recently, Han et al. (2021) formulate the RE task as a MLM task by using logic rules to construct prompts with multiple sub-prompts. By encoding prior knowledge of entities and relations into prompts, their proposed model, PTR, achieved 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 solve 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. (2018) 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 model cannot handle this. Therefore, the authors developed a seq2seq model to solve the summarization task in such format.

Zhong et al. (2020) 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 with traditional methods of extracting sentences individually, the matching framework enables the summary extractor to work at summary level rather than sentence level.

Aghajanyan et al. (2021) 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 (Andreas et al., 2013; Vinyals et al., 2015; Li et al., 2018; Rongali et al., 2020), the `SeqLab` paradigm (Gómez-Rodríguez and Vilares, 2018; Strzyz et al., 2019; Vilares and Gómez-Rodríguez, 2020; Vacareanu et al., 2020), and the `(M)LM` paradigm (Choe and Charniak, 2016). In addition, Gan et al. (2021) 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 shift, we draw a sankey diagram[2] in Figure 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 by a paradigm that is different with its original paradigm, some of the values is transferred from the original paradigm to the new paradigm. We initialize the value of each paradigm as 100, and the transferred value is defined as $100/N$, where N is the total number of paradigms that have been used to solve the task. Therefore, each branch in Figure 2 indicates a task that shifts its paradigm.

As shown in Figure 2, we find that: (1) The frequency of paradigm shift is increasing in recent years, especially after the emergence of pre-trained language models (PTMs). 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`, `Seq2ASeq`, to paradigms that are more general and flexible, *i.e.*, `(M)LM`, `Matching`, `MRC`, and `Seq2Seq`, which will be discussed in the following section.

4 Potential Unified Paradigms in NLP

Some of the paradigms have demonstrated 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:

- **Data efficiency.** Training task-specific models usually requires large-scale task-specific labeled data. In contrast, unified model has shown its ability to achieve considerable performance with much less labeled data.
- **Generalization.** Task-specific models are hard to transfer to new tasks while unified model can generalize to unseen tasks by formulating them into proper formats.
- **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 a (M)LM task is a natural way to utilizing 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 *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

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

few-shot and even zero-shot settings (Scao and Rush, 2021).

Prompt. The choice of prompt is critical to the performance of a particular task. A good prompt can be **(1) Manually designed**. Brown et al. (2020); Schick and Schütze (2021a,b) manually craft task-specific prompts for different tasks. Though it is heuristic and sometimes non-intuitive, hand-crafted prompts already achieved competitive performance on various few-shot tasks. **(2) Mined from corpora**. Jiang et al. (2020) construct prompts for relation extraction by mining sentences with the same subject and object in the corpus. **(3) Generated by paraphrasing**. Jiang et al. (2020) use back translation to paraphrase the original prompt into multiple new prompts. **(4) Generated by another pre-trained language model**. Gao et al. (2021) generate prompts using T5 (Raffel et al., 2020) since it is pre-trained to fill in missing spans in the input. **(5) Learned by gradient descent**. Shin et al. (2020) automatically construct prompts based on gradient-guided search. If prompt is not necessarily discrete, it can be optimized efficiently in continuous space. Recent works (Li and Liang, 2021; Qin and Eisner, 2021; Hambardzumyan et al., 2021; Liu et al., 2021b; Zhong et al., 2021) have shown that continuous prompts can also achieve competitive or even better performance.

Verbalizer. The design of verbalizer also has a strong influence on the performance of prompt-based learning (Gao et al., 2021). A verbalizer can be **(1) Manually designed**. Schick and Schütze (2021a) heuristically designed verbalizers for different tasks and achieved competitive results. However, it is not always intuitive for many tasks (*e.g.*, when class labels not directly correspond to words in the vocabulary) to manually design proper verbalizers. **(2) Automatically searched** on a set of labelled data (Schick et al., 2020; Gao et al., 2021; Shin et al., 2020; Liu et al., 2021b). **(3) Constructed and refined with knowledge base** (Hu et al., 2021).

Parameter-Efficient 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. Recent study (Lester et al., 2021) has demonstrated that tuning only prompt parameters while keeping the backbone model parameters fixed can achieve 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 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 of different tasks can be mixed in the same batch, which makes the service highly efficient.[3]

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 one (Dagan et al., 2005; Poliak et al., 2018; Yin et al., 2020; Wang et al., 2021). 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 MNLI (Williams et al., 2018). In addition to obtaining the entailment model in a supervised fashion, Sun et al. (2021) 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.

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. (2020) 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.

[3] The reader is referred to Liu et al. (2021a) for a more comprehensive survey of prompt-based learning.

Label Descriptions For single sentence classification tasks, 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. (2021) 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. (2020) utilize reinforcement learning to generate label descriptions.

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 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 (Li and Liang, 2021; Liu et al., 2021b).

4.3 MRC

MRC is also an alternative paradigm to unify various NLP tasks by generating task-specific questions and training a 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 a MRC model with "*Google was founded in 1998. Find organizations in the text, including companies, agencies and institutions*" as in Li et al. (2020). In addition to NER, MRC framework has also demonstrated competitive performance in entity-relation extraction (Li et al., 2019), coreference resolution (Wu et al., 2020), entity linking (Gu et al., 2021), dependency parsing (Gan et al., 2021), dialog state tracking (Gao et al., 2019), event extraction (Du and Cardie, 2020; Liu et al., 2020), aspect-based sentiment analysis (Mao et al., 2021), *etc.*

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. (2018) 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. 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 a 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 in 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 model or a pointer network as in McCann et al. (2018).

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 (McCann et al., 2018) has explored using the Seq2Seq paradigm to simultaneously solve different classes of tasks. Powered by recent advances of seq2seq pre-training such as MASS (Song et al., 2019), T5 (Raffel et al., 2020), and BART (Lewis et al., 2020), Seq2Seq paradigm has shown its great potential in unifying diverse NLP tasks. Paolini et al. (2021) use T5 (Raffel et al., 2020) 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. (2021a) and Yan et al. (2021b) use BART (Lewis et al., 2020), equipped with the copy network (Gu et al., 2016), 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 paradgms, 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 (Raffel et al., 2020; Lewis et al., 2020), MRC (McCann et al., 2018), etc. Nevertheless, what comes with its versatility is the high latency. Currently, most successful seq2seq models are in 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 (Gu et al., 2018; Qi et al., 2021), early exiting (Elbayad et al., 2020), or other alternative techniques.

5 Conclusion

Recently, prompt-based tuning, which is to formulate some NLP task into a (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 underexplored 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 data hungry.

Nevertheless, these paradigms have their advantages over (M)LM: Matching requires less engineering, MRC is more interpretable, Seq2Seq is more flexible to handle complicated tasks. Besides, by combining with self-supervised pre-training (*e.g.* BART (Lewis et al., 2020) and T5 (Raffel et al., 2020)), or further pre-training on annotated data with existing language model as initialization (*e.g.* Wang et al. (2021)), 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.

References

Armen Aghajanyan, Dmytro Okhonko, Mike Lewis, Mandar Joshi, Hu Xu, Gargi Ghosh, and Luke Zettlemoyer. 2021. HTLM: hyper-text pre-training and prompting of language models. *CoRR*, abs/2107.06955.

Jacob Andreas, Andreas Vlachos, and Stephen Clark. 2013. Semantic parsing as machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 2: Short Papers*, pages 47–52. The Association for Computer Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2000. A neural probabilistic language model. In *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*, pages 932–938. MIT Press.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Duo Chai, Wei Wu, Qinghong Han, Fei Wu, and Jiwei Li. 2020. Description based text classification with reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1371–1382. PMLR.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017a. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1870–1879. Association for Computational Linguistics.

Danqi Chen and Christopher D. Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on*

Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 740–750. ACL.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017b. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1657–1668. Association for Computational Linguistics.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Do Kook Choe and Eugene Charniak. 2016. Parsing as language modeling. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2331–2336. The Association for Computational Linguistics.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Machine Learning Challenges, Evaluating Predictive Uncertainty, Visual Object Classification and Recognizing Textual Entailment, First PASCAL Machine Learning Challenges Workshop, MLCW 2005, Southampton, UK, April 11-13, 2005, Revised Selected Papers*, volume 3944 of *Lecture Notes in Computer Science*, pages 177–190. Springer.

Xiang Dai, Sarvnaz Karimi, Ben Hachey, and Cécile Paris. 2020. An effective transition-based model for discontinuous NER. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5860–5870. Association for Computational Linguistics.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2978–2988. Association for Computational Linguistics.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 671–683. Association for Computational Linguistics.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. 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 of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 334–343. The Association for Computer Linguistics.

Maha Elbayad, Jiatao Gu, Edouard Grave, and Michael Auli. 2020. Depth-adaptive transformer. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Joseph Fisher and Andreas Vlachos. 2019. Merge and label: A novel neural network architecture for nested NER. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5840–5850. Association for Computational Linguistics.

Jinlan Fu, Xuanjing Huang, and Pengfei Liu. 2021. 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/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 7183–7195. Association for Computational Linguistics.

Leilei Gan, Yuxian Meng, Kun Kuang, Xiaofei Sun, Chun Fan, Fei Wu, and Jiwei Li. 2021. Dependency parsing as mrc-based span-span prediction. *CoRR*, abs/2105.07654.

Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, and Dilek Hakkani-Tür. 2019. Dialog state tracking: A neural reading comprehension approach. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue, SIGdial 2019, Stockholm, Sweden, September 11-13, 2019*, pages 264–273. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. 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/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021, pages 3816–3830. Association for Computational Linguistics.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252. PMLR.

Carlos Gómez-Rodríguez and David Vilares. 2018. Constituent parsing as sequence labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1314–1324. Association for Computational Linguistics.

Edouard Grave, Armand Joulin, and Nicolas Usunier. 2017. Improving neural language models with a continuous cache. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O. K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Yingjie Gu, Xiaoye Qu, Zhefeng Wang, Baoxing Huai, Nicholas Jing Yuan, and Xiaolin Gui. 2021. Read, retrospect, select: An MRC framework to short text entity linking. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 12920–12928. AAAI Press.

Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. 2021. 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/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4921–4933. Association for Computational Linguistics.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. PTR: prompt tuning with rules for text classification. *CoRR*, abs/2105.11259.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Juanzi Li, and Maosong Sun. 2021. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. *CoRR*, abs/2108.02035.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know. *Trans. Assoc. Comput. Linguistics*, 8:423–438.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751. ACL.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 282–289. Morgan Kaufmann.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 260–270. The Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *CoRR*, abs/2104.08691.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada, August 3-4, 2017*, pages 333–342. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. 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, Online. Association for Computational Linguistics.

Chengxi Li, Feiyu Gao, Jiajun Bu, Lu Xu, Xiang Chen, Yu Gu, Zirui Shao, Qi Zheng, Ningyu Zhang, Yongpan Wang, and Zhi Yu. 2021. Sentiprompt: Sentiment knowledge enhanced prompt-tuning for aspect-based sentiment analysis. *CoRR*, abs/2109.08306.

Xiang Lisa Li and Percy Liang. 2021. 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/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4582–4597. Association for Computational Linguistics.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A unified MRC framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5849–5859. Association for Computational Linguistics.

Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019. Entity-relation extraction as multi-turn question answering. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1340–1350. Association for Computational Linguistics.

Zuchao Li, Jiaxun Cai, Shexia He, and Hai Zhao. 2018. Seq2seq dependency parsing. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 3203–3214. Association for Computational Linguistics.

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020. Event extraction as machine reading comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 1641–1651. Association for Computational Linguistics.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2873–2879. IJCAI/AAAI Press.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *CoRR*, abs/2107.13586.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. GPT understands, too. *CoRR*, abs/2103.10385.

Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Xuezhe Ma and Eduard H. Hovy. 2016. 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 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Yue Mao, Yi Shen, Chao Yu, and Longjun Cai. 2021. A joint training dual-mrc framework for aspect based sentiment analysis. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13543–13551. AAAI Press.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *CoRR*, abs/1806.08730.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cícero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Adam Poliak, Aparajita Haldar, Rachel Rudinger, J. Edward Hu, Ellie Pavlick, Aaron Steven White, and Benjamin Van Durme. 2018. Collecting diverse natural language inference problems for sentence representation evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 67–81. Association for Computational Linguistics.

Weizhen Qi, Yeyun Gong, Jian Jiao, Yu Yan, Weizhu Chen, Dayiheng Liu, Kewen Tang, Houqiang Li, Jiusheng Chen, Ruofei Zhang, Ming Zhou, and Nan Duan. 2021. BANG: bridging autoregressive and non-autoregressive generation with large scale pretraining. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8630–8639. PMLR.

Guanghui Qin and Jason Eisner. 2021. 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, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5203–5212. Association for Computational Linguistics.

Xipeng Qiu, TianXiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *SCIENCE CHINA Technological Sciences*.

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. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Subendhu Rongali, Luca Soldaini, Emilio Monti, and Wael Hamza. 2020. Don't parse, generate! A sequence to sequence architecture for task-oriented semantic parsing. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 2962–2968. ACM / IW3C2.

Teven Le Scao and Alexander M. Rush. 2021. 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, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2627–2636. Association for Computational Linguistics.

Timo Schick, Helmut Schmid, and Hinrich Schütze. 2020. Automatically identifying words that can serve as labels for few-shot text classification. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 5569–5578. International Committee on Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021a. 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, EACL 2021, Online, April 19 - 23, 2021*, pages 255–269. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2339–2352. Association for Computational Linguistics.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4222–4235. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: masked sequence to sequence pre-training for language generation. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5926–5936. PMLR.

Michalina Strzyz, David Vilares, and Carlos Gómez-Rodríguez. 2019. Viable dependency parsing as sequence labeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 717–723. Association for Computational Linguistics.

Chi Sun, Luyao Huang, and Xipeng Qiu. 2019. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. 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 380–385, Minneapolis, Minnesota. Association for Computational Linguistics.

Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuanjing Huang, and Zheng Zhang. 2020. Colake: Contextualized language and knowledge embedding. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 3660–3670. International Committee on Computational Linguistics.

Yi Sun, Yu Zheng, Chao Hao, and Hangping Qiu. 2021. NSP-BERT: A prompt-based zero-shot learner through an original pre-training task–next sentence predictio. *CoRR*, abs/2109.03564.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Robert Vacareanu, George Caique Gouveia Barbosa, Marco Antonio Valenzuela-Escárcega, and Mihai Surdeanu. 2020. Parsing as tagging. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 5225–5231. European Language Resources Association.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural*

Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pages 5998–6008.

David Vilares and Carlos Gómez-Rodríguez. 2020. Discontinuous constituent parsing as sequence labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2771–2785. Association for Computational Linguistics.

Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey E. Hinton. 2015. Grammar as a foreign language. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2773–2781.

Sinong Wang, Han Fang, Madian Khabsa, Hanzi Mao, and Hao Ma. 2021. Entailment as few-shot learner. *CoRR*, abs/2104.14690.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 606–615. The Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Bohong Wu, Zhuosheng Zhang, and Hai Zhao. 2021. Graph-free multi-hop reading comprehension: A select-to-guide strategy. *CoRR*, abs/2107.11823.

Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li. 2020. Corefqa: Coreference resolution as query-based span prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6953–6963. Association for Computational Linguistics.

Congying Xia, Chenwei Zhang, Tao Yang, Yaliang Li, Nan Du, Xian Wu, Wei Fan, Fenglong Ma, and Philip S. Yu. 2019. Multi-grained named entity recognition. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1430–1440. Association for Computational Linguistics.

Caiming Xiong, Victor Zhong, and Richard Socher. 2017. Dynamic coattention networks for question answering. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Hang Yan, Junqi Dai, Tuo Ji, Xipeng Qiu, and Zheng Zhang. 2021a. 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/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 2416–2429. Association for Computational Linguistics.

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021b. 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/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5808–5822. Association for Computational Linguistics.

Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. 2018a. SGM: sequence generation model for multi-label classification. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 3915–3926. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018b. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.

Wenpeng Yin, Nazneen Fatema Rajani, Dragomir R. Radev, Richard Socher, and Caiming Xiong. 2020. Universal natural language processing with limited annotations: Try few-shot textual entailment as a start. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 8229–8239. Association for Computational Linguistics.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named entity recognition as dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6470–6476, Online. Association for Computational Linguistics.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *COLING*

2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland, pages 2335–2344. ACL.

Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. Extracting 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 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 506–514. Association for Computational Linguistics.

Tianyang Zhao, Zhao Yan, Yunbo Cao, and Zhoujun Li. 2020. Asking effective and diverse questions: A machine reading comprehension based framework for joint entity-relation extraction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3948–3954. ijcai.org.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online. Association for Computational Linguistics.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual probing is [MASK]: learning vs. learning to recall. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5017–5033. Association for Computational Linguistics.