A Survey of Multi-task Learning in Natural Language Processing: Regarding Task Relatedness and Training Methods

Zhihan Zhang, Wenhao Yu, Mengxia Yu, Zhichun Guo, Meng Jiang
University of Notre Dame, Notre Dame, IN, USA
{zzhang23, wyu1, myu2, zguo5, mjiang2}@nd.edu

Abstract

Multi-task learning (MTL) has become increasingly popular in natural language processing (NLP) because it improves the performance of related tasks by exploiting their commonalities and differences. Nevertheless, it is still not understood very well how multi-task learning can be implemented based on the relatedness of training tasks. In this survey, we review recent advances of multi-task learning methods in NLP, with the aim of summarizing them into two general multi-task training methods based on their task relatedness: (i) joint training and (ii) multi-step training. We present examples in various NLP downstream applications, summarize the task relationships and discuss future directions of this promising topic.

1 Introduction

Machine learning generally involves training a model to perform a single task. By focusing on one task, the model ignores knowledge from the training signals of *related tasks* (Ruder, 2017). There are a great number of tasks in NLP, from syntax parsing to information extraction, from machine translation to question answering: each requires a model dedicated to learning from data. Biologically, humans learn natural languages, from basic grammar to complex semantics in a single brain (Hashimoto et al., 2017). In the field of machine learning, multi-task learning (MTL) aims to leverage useful information shared across multiple related tasks to improve the generalization performance on all tasks (Caruana, 1997). In deep neural networks, it is generally achieved by sharing part of hidden layers between different tasks, while keeping several task-specific output layers. MTL offers advantages like improved data efficiency, reduced overfitting, and fast learning by leveraging auxiliary information (Crawshaw, 2020).

There have been relevant surveys that looked into architecture designs and optimization algorithms in MTL. Ruder (2017) classified different MTL frameworks into two categories: hard parameter sharing and soft parameter sharing, as well as some earlier MTL examples in both non-neural and neural models; Zhang and Yang (2018) expanded such two "how to share" categories into five categories, including feature learning approach, low-rank approach, task clustering approach, task relation learning approach, and decomposition approach; Crawshaw (2020) presented more recent models in both single-domain and multi-modal architectures, as well as an overview of optimization methods in MTL. Nevertheless, it is still not clearly understood *how* to design and train a single model to handle a variety of NLP tasks according to **task relatedness**. Especially when faced with a set of tasks that are seldom simultaneously trained previously, it is of crucial importance that researchers find proper auxiliary tasks and assess the feasibility of such multi-task learning attempt.

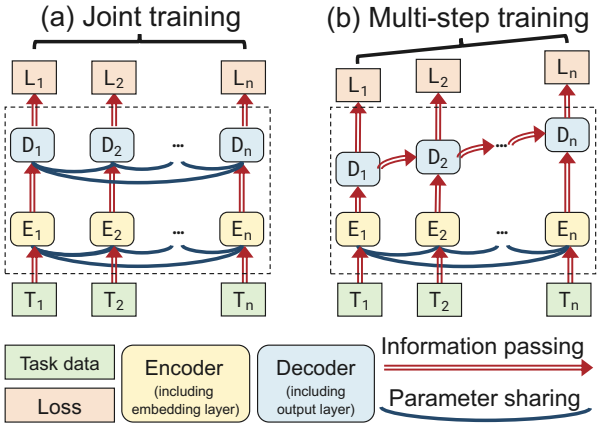

Figure 1: Two multi-task learning frameworks.

In this paper, we first review recent approaches on multi-task training methods in popular NLP applications. We find that these approaches can be categorized into *two multi-task training methods* according to the types of task relatedness: (i) joint training methods and (ii) multi-step training meth-

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 943–956

Multi-task training methods in our survey	Multi-task frameworks defined in Ruder (2017)	Multi-task frameworks defined in Crawshaw (2020)	Related papers
Joint Training	Deep relationship network	Shared trunk	(Liu et al., 2015; Dong et al., 2015)
	Cross-stitch network	Cross-talk	(Liu et al., 2016; Xiao et al., 2018)
	Weighting losses with uncertainty		(Xiong et al., 2018; Xu et al., 2019b)
	Sluice network		(Ruder et al., 2017)
		Adversarial feature separation	(Liu et al., 2017; Mao et al., 2020)
Multi-step Training		Prediction distillation	(Dinan et al., 2019; Lewis et al., 2020)
	Low supervision	Cascaded information	(Søgaard et al., 2016; Hashimoto et al., 2017)

Table 1: Categories of multi-task learning frameworks in two related surveys can be merged into our proposed joint training and multi-step training frameworks.

ods. **Joint training** is commonly used when all given tasks can be performed simultaneously and all the task-specific data can be learned simultaneously. In joint training, model parameters are shared (either via soft or hard parameter sharing[1]) among encoders and decoders so that the tasks can be jointly trained to benefit from shared representations, as shown in Figure 1(a). In contrast, **multi-step training** is used when some task's input needs to be determined by the outputs or hidden representations of previous task(s). Due to such task dependencies, the task-specific decoders are connected as a multi-step path starting from the encoder "node", as shown in Figure 1(b).

Therefore, different from previous surveys which focus on architecture designs (*e.g.*, how to share parameters in (Ruder, 2017) and (Zhang and Yang, 2018)) and optimization methods (*e.g.*, loss weighting and regularization in (Crawshaw, 2020)), our motivation lies in categorizing two major multi-task training methods in NLP, according to **task relatedness**. In fact, task relatedness is the key to determine what training method to use, then the training method decides the general scope of available architecture designs. With specific application tasks, readers are able to identify the ideal training method from our review before looking for detailed module design or loss optimization in previous surveys. We also show that how the MTL techniques covered in previous surveys can be matched with the two training methods in Table 1.

The remainder of this survey is organized as follows. Section 2 includes an overview of MTL models in NLP and the rationales of using MTL. Section 3 presents a number of joint and multi-step training applications in different fields of NLP. Section 4 analyzes the task relatedness involved in these MTL approaches. Section 5 discusses future directions. Section 6 concludes the paper.

[1]We do not specifically distinguish different parameter sharing designs, since this topic is not the focus of our survey. We refer readers to learn details in Ruder (2017).

2 Multi-task Training Methods

2.1 Encoder-Decoder Architecture and Two Multi-Task Training Frameworks

Suppose we train a model on n NLP tasks $T_1, \cdots, T_n$ on a dataset $\mathcal{D} = \{(X^{(i)}, Y^{(i)})\}|_{i=1}^{N}$ with N data points. For the j-th NLP task, the model is trained with $\{(X_j^{(i)}, Y_j^{(i)})\}|_{i=1}^{N}$, where $X_j^{(i)}$ is a component of the input $X^{(i)}$, and $Y_j^{(i)}$ is the desired output. The input components of different tasks can be the same, but the desired outputs are usually different. We formulate the multi-task frameworks discussed in this paper under the popular encoder-decoder architecture which are mainly composed of three components: (a) the encoder layer (including the embedding layer), (b) the decoder layer (including the output layer for classification or generation), and (c) loss and optimization.

Encoder layer. In NLP networks, an embedding layer is usually applied to generate the embedding vectors of the basic elements of the input $X^{(i)}$ For the j-th task, the encoder layer learns the hidden state of $X_j^{(i)}$ as a vector $\mathbf{h}_j^{(i)}$:

$$\mathbf{h}_j^{(i)} = \text{Encoder}(X_j^{(i)}, \Theta_{E_j}), \quad (1)$$

where Θ_{E_j} denotes the parameters of j-th task's encoder. Parameters of different encoders can be shared. Popular encoder modules include BiLSTM and BERT (Devlin et al., 2019).

Decoder layer. When the tasks are *independent* with each other at decoding, the decoder of the j-th task transforms the hidden state into an output:

$$\hat{Y}_j^{(i)} = \text{Decoder}_j(\mathbf{h}_j^{(i)}, \Theta_{D_j}). \quad (2)$$

When the tasks are *sequentially dependent*, the decoder of the j-th task needs the output of the $(j-1)$-th task, then we have

$$\hat{Y}_j^{(i)} = \text{Decoder}_j(\hat{Y}_{j-1}^{(i)}, \mathbf{h}_j^{(i)}, \Theta_{D_j}). \quad (3)$$

where Θ_{D_j} denotes the parameters of j-th task's decoder. Practically, $\hat{Y}_{j-1}^{(i)}$ are often presented as hidden representations of the decoder prediction to enable end-to-end training. Parameters of different decoders can be shared. Popular decoder choices include MLP, LSTM and the Transformer decoder.

According to the two types of task dependencies, the multi-task learning frameworks define and organize the decoders in two different ways. As shown in Figure 1, (i) the **joint training** framework is for the tasks that are independent at decoding; and (ii) the **multi-step training** framework is for tasks that are sequentially dependent. It can be easily generalized when the task dependencies form a directed acyclic graph, in which sequential dependence is a special and common case.

Optimization. A common optimization approach of MTL is to optimize the weighted sum of loss functions from different tasks, (i.e., Loss $= \lambda_j \sum_{j=1}^{n} \text{Loss}_j$) then compute the gradient descent to update all trainable parameters ($\{\Theta_{E_j}\}_{j=1}^{n}$, $\{\Theta_{D_j}\}_{j=1}^{n}$). The weights of $\{\lambda_j\}_{j=1}^{n}$ can be either pre-defined or dynamically adjusted (Kendall et al., 2018; Xiong et al., 2018). It is worth mentioning that optimization in MTL has many alternative ways. For example, Søgaard et al. (2016) choose a random task t from a pre-defined task sets to optimize its loss at each iteration. Readers can find a more detailed review of MTL optimization methods in Crawshaw (2020), which is not the main focus of this paper.

2.2 How does MTL Work

One of the prerequisites of multi-task learning is the relatedness among different tasks and their data. Most work prefers to train positively correlated tasks in a multi-task setting. Such tasks have similar objectives or relevant data, and can boost each other to form consistent predictions through shared lower-level representations. According to Caruana (1997), in MTL, tasks prefer hidden representations that other tasks prefer. MTL enables shared representations to include features from all tasks, thus improving the consistency of task-specific decoding in each sub-task. Furthermore, the co-existence of features from different objectives naturally performs feature crosses, which enables the model to learn more complex features.

According to the experiments by Standley et al. (2020), tasks are more likely to benefit from MTL when using a larger network. This can be achieved as the emergence of deep neural frameworks in recent years. Many deep models, like BERT (Devlin et al., 2019) and T5 (Raffel et al., 2020), have strong generalization ability to fit a variety of tasks with minute changes. Therefore, different tasks can be learned through similar models, especially in the field of NLP where the encoder-decoder architecture has become a norm.

With the above premises, deep models are able to benefit from MTL in multiple perspectives. First, MTL improves data efficiency for each sub-task. Different tasks provide different aspects of information, enriching the expression ability of the hidden representation to the input text. Besides, different tasks have different noise patterns, which acts as an implicit data augmentation method. This encourages the multi-task model to produce more generalizable representations in shared layers. Thus, the model is prevented from overfitting to a single task and gains stronger generalization ability, which helps itself perform well when faced with new tasks from a similar environment (Baxter, 2000). Multi-task learning are also effective for low-resource tasks (Lin et al., 2018b; Johnson et al., 2017). Co-training with a high-resource task in a similar domain, low-resource tasks receive ampler training signals which prevents the model from overfitting on the limited data.

Auxiliary tasks in MTL can serve as conditions or hints for the main task. Such setting usually falls into the category of *multi-step training*. Providing additional conditions reduces the distribution space of possible outputs, thus lower the prediction difficulty of the main task. Such conditions can serve as additional features during decoding, including external knowledge pieces, low-level NLP tasks (*e.g.*, part-of-speech tagging or syntactic parsing) or relevant snippets extracted from long documents. When some features are difficult for the main task to directly learn, explicit supervision signals of such features, if available, enables the model to "eavesdrop", *i.e.*, obtaining these features through the learning of auxiliary task (Ruder, 2017).

3 Training Methods: Applications

3.1 Joint Training Applications

In this section, we list a series of recent approaches of joint training in different fields of NLP (shown in Figure 2), including information extraction, spoken language understanding, text classification, machine translation and language generation.

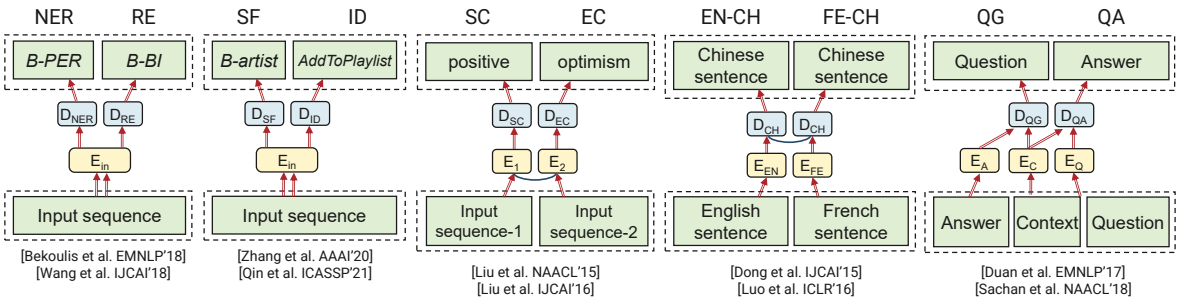

Figure 2: Five joint training NLP applications that have been discussed from §3.1.1 to §3.1.5.

3.1.1 Information Extraction (IE)

Two popular tasks that are usually jointly performed in IE are named entity recognition (NER) and relation extraction (RE). NER seeks to locate and classify named entities in text into pre-defined categories, such as names and locations. NER is often tackled by sequence labeling methods, or say, token-wise classifiers. RE aims to extract semantic relationships between two or more entities, and there are multiple ways to define the RE task in the multi-task training approach.

First, Zhou et al. (2019) predicted the type of relation mentioned in a sentence by the RE decoder. It works for simple sentences such as those that have a pair of entities and one type of relation, e.g., "[President Obama] was *born in* [Honolulu]." However, one sentence may have multiple types of relations. Second, Zheng et al. (2017) predicted a relation tag for every pair of tokens. If the decoder performs perfectly, it can identify any number and any types of relations in a sentence. However, the complexity is too high to be effectively trained with annotated data. Third, Bekoulis et al. (2018); Wang et al. (2018a) treated RE as a sequence labeling problem. So both NER and RE decoders are token-wise classifiers. As shown in Figure 2, for example, $B\text{-}BI$ tag represents the beginning word of subject entity (person) or object entity (location) in the "born_in" (BI) relation. Therefore, if multiple tag sequences can be generated, they can identify any number, and any type of relations in the input sentence.

3.1.2 Spoken Language Understanding

Spoken Language Understanding (SLU) plays an important role in spoken dialogue system (Qin et al., 2021c). SLU aims at extracting the semantics from user utterances, which is a critical component of task-oriented dialogue. Concretely, it captures semantic constituents of the utterance and identifies the user's intent. These two tasks are typically known as slot filling (SF) and intent detection (ID), respectively. Each word in the utterance corresponds to one slot label, and a specific intent is assigned to the whole utterance. An example of these two sub-tasks is given below:

Word	Put	Kanye	into	my	rap	playlist
	↓	↓	↓	↓	↓	↓
Slot	O	B-artist	O	O	B-playlist	O
Intent	AddToPlaylist					

Since two sub-tasks share the same input utterance, they usually share a single utterance encoder and are jointly trained (Liu and Lane, 2016; Castellucci et al., 2019). Recent state-of-the-art SLU models build bi-directional interactions during encoding (Liu et al., 2019b; Zhang et al., 2020; Qin et al., 2021a). Therefore, two tasks mutually impact each other before making respective predictions. It is worth noting that there is also a line of work that uses the hidden states of intent detection to assist slot filling (Goo et al., 2018; Qin et al., 2019, 2021b). This can be considered as a combination of joint training and multi-step training: intent detection helps the prediction of slot filling, but finally their predictions are integrated to perform the parent (larger) SLU task.

3.1.3 Sentence/Document Classification

Sentence/document classification is one of the fundamental tasks in NLP with broad applications such as sentiment classification (SC), emotion classification (EC), and stance detection. However, the construction of large-scale high-quality datasets is extremely labor-intensive. Therefore, multi-task learning plays an important role in leveraging potential correlations among related classification tasks to extract common features, increase corpus size implicitly and yield classification improvements. Popular multi-task learning setting in text classification has two categories. First, one dataset is annotated with multiple labels and one input is associated with multiple outputs (Liu et al., 2015;

Yu et al., 2018a; Gui et al., 2020). Second, multiple datasets have their respective labels, i.e., multiple inputs with multiple outputs, where samples from different tasks are jointly learned in parallel (Liu et al., 2016, 2017). Most existing work leverages joint training for different sentence/document classification tasks. Specifically, Liu et al. (2016) proposed three different parameter sharing designs under the joint training framework, and further compared their performances.

3.1.4 Multilinguality

Languages differ lexically but are closely related on the semantic and/or the syntactic levels. Such correlation across different languages motivates the multi-task learning on multilingual data. Neural machine translation (NMT) is the most important application. Dong et al. (2015) first proposed a multi-task learning framework based on Seq2Seq to conduct NMT from one source language to multiple target languages. Luong et al. (2016) extended it with many-to-one and many-to-many approaches. Many-to-one is useful for translating multi-source languages to the target language, in which only the decoder is shared. Many-to-many studies the effect of unsupervised translation between multiple languages. Zhu et al. (2019) proposed to improve cross-lingual summarization by jointly training with monolingual summarization and machine translation. Arivazhagan et al. (2019) built a massive multi-lingual translation model handling 103 languages, and conducted experiments on multiple sampling schema for building joint training dataset.

Besides, unlabelled data from the target language is also a common source of multi-task cross-lingual training. Ahmad et al. (2019) collected unannotated sentences from auxiliary languages to assist learning language-agnostic representations. Van Der Goot et al. (2021) incorporated a masked language modeling objective using unlabeled data from target language to perform zero-shot transfer.

3.1.5 Natural Language Generation (NLG)

Recent success in deep generative modeling have led to significant advances in NLG, motivated by an increasing need to understand and derive meaning from language (Yu et al., 2020b). The relatedness between different generation tasks promotes the application of multi-task learning in NLG.

For example, Guo et al. (2018) proposed to jointly learn abstractive summarization (AS) and question generation (QG). An accurate summary of a document is supposed to contain all its salient information. This goal is consistent with that of question generation (QG), which looks for salient questioning-worthy details. Besides, QG and question answering (QA) are often trained as dual tasks. Tang et al. (2017) proposed a joint learning framework that connects QG and QA. QA improves QG through measuring the relevance between the generated question and the answer. QG improves QA by providing additional signal which stands for the probability of generating a question given the answer. A similar framework was also employed in Duan et al. (2017); Sachan and Xing (2018).

In other applications, semantic parsing is gaining attention for knowledge-based question answering since it does not rely on hand-crafted features. (Shen et al., 2019) developed a joint learning approach where a pointer-equipped semantic parsing model is designed to resolve coreference in conversations, and naturally empower joint learning with a novel type-aware entity detection model. Researchers also found NLU tasks, *e.g.*, input meaning representation learning (Qader et al., 2019) or entity mention prediction (Dong et al., 2020), can improve the performance of generating sentences

Multi-view learning is also applied in NLG approaches for auxiliary learning objectives. Input data are erased partially to create distinct views, and divergence metrics are usually learned along with the main loss to force the model generate consistent predictions across different views of the same input. Typical approaches include Clark et al. (2018) which built up the multi-view learning paradigm in IE and NLG tasks. In addition, Shen et al. (2020) upgraded the network by combining multiple cutoff methods to create augmented data, and achieved success in translation tasks.

3.2 Multi-step Training: Applications

We list recent approaches of multi-step training in different field of NLP (as shown in Figure 3), such as language understanding, multi-passage question answering and natural language generation.

3.2.1 Multi-level Language Understanding

The potential for leveraging multiple levels of representations has been demonstrated in various ways in the field of NLP. For example, Part-Of-Speech (POS) tags are used for syntactic parsers. The parsers are used to improve higher-level tasks, such as natural language inference. Søgaard et al. (2016) showed when learning POS tagging and chunking, it is consistently better to have POS supervision

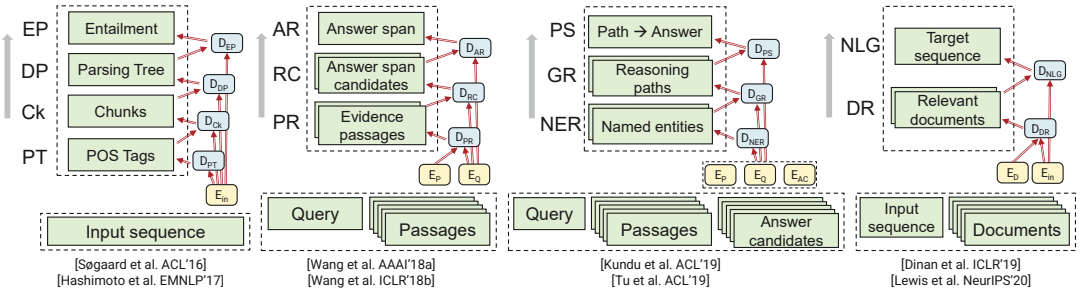

Figure 3: Four multi-step training NLP applications discussed at §3.2.1, §3.2.2 (2nd and 3rd subfigures) and §3.2.3.

at the innermost rather than the outermost layer. Hashimoto et al. (2017) predicted increasingly complex NLP tasks at successively deeper layers for POS tagging, chunking, dependency parsing, semantic relatedness, and textual entailment, by considering linguistic hierarchies. Lower level predictions may influence predictions in higher levels, *e.g.*, if the semantic relatedness between two sentences is very low, they are unlikely to entail each other. Similar architecture can be found in Sanh et al. (2019b), where low-level tasks are name entity recognition and entity mention detection, with coreference resolution and relation extraction supervising at higher levels.

3.2.2 Multi-Passage Question Answering

Question answering (QA) models may need to construct answers by querying multiple passages (*e.g.*, paragraphs, documents). Given a question, multi-passage QA (MPQA) requires AI models identify an answer span from multiple evidence passages. Due to the complexity of MPQA, it is usually achieved by multiple sub-tasks. Thus, multi-step training is utilized by many approaches in MPQA.

Typically, MPQA can be split into a 3-phase task. *Passage retrieval* (PR) is to select relevant evidence passages according to the question. *Reading comprehension* (RC) is to extract multiple answer span candidates from the retrieved set of relevant passages. *Answer reranking* (AR) is to re-score multiple answer candidates based on the question and evidence passages. There exist dependencies between these tasks: evidence passages are generated by PR and fed into RC as input; the answer span candidates are generated by RC and given into AR as input. So, as shown in Figure 3, the decoders form a multi-hop path starting from the shared encoder. Hu et al. (2019) proposed a typical approach called RE^3 (for REtriever, REader, and REranker). The retriever used TF-IDF cosine similarities to prune irrelevant passages. The reader is a token classifier that predicts the start and end indices of answer candidates per segment. The reranker prunes redundant span candidates and then predict the reranking scores. Other works in MPQA also considered 2-phase approaches, such as PR+RC (Wang et al., 2018b) or RC+AR (Wang et al., 2018d), which are simplified versions of the above framework. Similar approaches have been developed for many domains such as news (Nishida et al., 2018), and web questions (Lin et al., 2018a).

Another branch of this task is *multi-choice* MPQA, where we have a set of answer candidates for the given question. (Kundu et al., 2019) proposed to exploit explicit paths for multi-hop reasoning over structured knowledge graphs. The model attempted to extract implicit relations from text through entity pair representations, and compose them to encode each path. It composed the passage representations along each path to compute a passage-based representation. Then it can explain the reasoning via these explicit paths through the passages. The sub-tasks are *named entity recognition* (NER), *graph-based reasoning* (GR) to extract and encode paths, and passage-based *path scoring* (PS). So, the multi-task QA systems perform interpretable and accurate reasoning (Welbl et al., 2018; Tu et al., 2019).

3.2.3 Retrieval-augmented Text Generation

In NLG, the input sequence alone often contains limited knowledge to support neural generation models to produce the desired output, so the performance of generation is still far from satisfactory in many real-world scenarios (Yu et al., 2020b). Retrieval-augmented generation models use the input sequence to retrieve relevant information (e.g., a background document) and use it as additional contexts when generating the target sequence. For example, Dinan et al. (2019) proposed to tackle the knowledge-aware dialogue by first selecting knowledge from a large pool of document candidates and generate a response based on the selected knowledge and context. To enhance the aforementioned

idea, Kim et al. (2020) presented a sequential latent variable model to keep track of the prior and posterior distribution over knowledge. It not only reduced the ambiguity caused from the diversity in knowledge selection of conversation but also better leveraged the response information for proper choice of knowledge. Similar retrieval-augmented generation approaches have been applied in question generation (Lewis et al., 2020), comment generation (Lin et al., 2019b), image captioning (Xu et al., 2019a), summarization (Cao et al., 2018), and long form QA (Krishna et al., 2021).

4 NLP Task Relatedness

In this section, we summarize the characteristics of the aforementioned MTL approaches, and look into the task relatedness between the sub-tasks.

4.1 Joint Training

Joint training with similar tasks. Joint training with a similar task is the classical choice for multi-task learning. According to Caruana (1997), more similar tasks share more hidden units. Hence, similar tasks are more likely to benefit from shared generic representations. However, what kind of tasks can be considered as "similar" are not always evident in the deep learning era. Empirically selecting similar tasks is still the most mainstream method (Ruder, 2017; Worsham and Kalita, 2020). To get some intuitions what a similar task can be, here we introduce some prominent examples. (Dong et al., 2015) proposed training neural machine translation from one language into multiple languages simultaneously; (Yu et al., 2018a) proposed a joint training framework for sentiment classification and emotion classification; (Guo et al., 2018) proposed abstractive summarization can be jointly learned with question generation. (Yang et al., 2019) jointly trained question categorization and answer retrieval.

Recently, Aribandi et al. (2021) attempted to empirically select a set of tasks (from 107 NLP tasks) to transfer from, using a multi-task objective of mixing supervised tasks with self-supervised objectives for language model pre-training. In addition, Guo et al. (2019) used multi-armed bandits to select tasks and a Gaussian Process to control the mixing rates. Besides, some recent work tried to select appropriate sub-tasks based on manually defined features (Lin et al., 2019a; Sun et al., 2021). Aside from NLP, Fifty et al. (2021) proposed a method to select sub-tasks based on task gradients.

Auxiliary task for adversarial learning. Partial sharing of model parameters is the mainstream in multi-task learning, which attempts to divide the features of different tasks into private and shared spaces. However, the shared feature space could contain some unnecessary task-specific features, while some sharable features could also be mixed in private space, suffering from feature redundancy. To alleviate this problem, Liu et al. (2017) adds an adversarial task via a discriminator to estimate what task the encoding sequence comes from. Such learning strategy prevents the shared and private latent feature spaces from interfering with each other. This setup has also received success in multi-task multi-domain training for domain adaptation (Yu et al., 2018b). The adversarial task in this case is to predict the domain of the input. By reversing the gradient of the adversarial task, the adversarial task loss is maximized, which is beneficial for the main task as it forces the model to learn representations that are indistinguishable between domains.

Auxiliary task to boost representation learning. While auxiliary tasks are utilized to assist the main task, they are usually expected to learn representations shared or helpful for the main task (Ruder, 2017). Self-supervised, or unsupervised tasks, therefore, are often considered as a good choice. Self-supervised objectives allow the model to learn beneficial representations without leveraging expensive downstream task labels. For example, language modeling can help to learn transferable representations. In BERT (Devlin et al., 2019) pre-training, the next sentence prediction task is used to learn sentence-level representations, which is complementary to the masked language model task that mainly targets at word-level contextual representations. Besides, Rei (2017) showed that learning with a language modeling objective improves performance on several sequence labelling tasks. An auto-encoder objective can also be used as an auxiliary task. Yu et al. (2020a) demonstrated adding the auto-encoder objective improves the quality of semantic representations for questions and answers in the task of answer retrieval.

Another branch of auxiliary tasks used to facilitate representation learning is knowledge distillation. This is achieved by forcing a smaller student model to learn a larger teacher model's output distribution or hidden representation using additional training objectives (Hinton et al., 2015). Knowledge contained in the hidden representations is then

transferred from the teacher to the student. Thus, the student model gains the generalization ability of the teacher model, but still preserving its small size which is more suitable for deployment. Such distillation idea has been verified on popular NLP models such as BERT (Sanh et al., 2019a).

4.2 Multi-step Training

Narrow the search space of the subsequent decoder. In some cases, it is not easy to predict the original task directly due to the large search space of the potential outputs (Lewis et al., 2020). For example, in open-domain QA, directly answering a given question is hard. So, multi-stage methods (e.g., retrieve-then-read) are often used to tackle open-domain QA problems: a retriever component finding documents that might contain the answer from a large corpus, followed by a reader component finding the answer in the retrieved documents. Documents provided by the retriever serves as conditions to the reader, which narrows the search space and thus reduces the difficulty of open-domain QA (Wang et al., 2018b,c).

In another example about pre-trained language models, BERT only learns from 15% of the tokens that are masked in the input. ELECTRA (Clark et al., 2020) proposed a two-step self-supervised training to improve training efficiency. The masked language modeling task performed by an auto-encoder serves as an auxiliary task. It reconstructs the masked tokens in the input. Then, a discriminative model in the second step predicts whether each token in the corrupted input was replaced by the auto-encoder. The design of such classification task allows supervision on all tokens in the example.

Select focused contents from the input. The auxiliary task can be used to focus attention on parts of the input text that can be leveraged for the main task. For example, humans tend to write summaries containing certain keywords and then perform necessary modifications to ensure the fluency and grammatical correctness of the summary. Thus, keyword extraction could help the model to focus on salient information that can be used in the summary (Li et al., 2020). A similar approach can be found in Cho et al. (2019), where the authors used a flexible continuous latent variable for content selection to deal with different focuses on the context in question generation.

Predict attributes of the output. In some NLG scenarios, it may be hard to guarantee the output sequence contains certain desired patterns or features (e.g., emotion, sentiment) if no explicit signals are given. Therefore, an attribute classifier could be used for predicting whether the output sequence contains the desired objective, either before or after the prediction is made. For example, Fan et al. (2018) predicted which question type should be used before generating diverse questions for an image. The predicted question type acts as an additional condition while the decoder is searching for the best question sequence. Besides, Song et al. (2019) used a emotion classifier after the decoder to discriminate whether the generated sentence expresses the desired emotion. The post-decoder classifier guides the generation process to generate dialogue responses with specific emotions.

Introduce external knowledge. Precisely manipulating world knowledge is extremely hard for a single neural network model. One could devise learning tasks informed by the knowledge so that the model is trained to acquire and utilize external knowledge. This research direction is known as "Knowledge-enhanced NLP" (Yu et al., 2020b). The knowledge-related tasks can be combined as auxiliary to the main task, resulting in a multi-task learning setting (Dinan et al., 2019; Kim et al., 2020; Zhang et al., 2021). For instance, Wu et al. (2019) uses the input sequence to query the candidate knowledge pieces via attention mechanism, then fuses the selected knowledge into decoder. The knowledge selection phase is trained by minimizing the KL-divergence between the prior distribution (queried by the input) and the posterior distribution (queried by the output).

5 Future Directions

In this section, we will discuss some promising directions regarding either task relatedness or training methods of multi-task training in NLP.

5.1 Regarding Task Relatedness

Task-specific multi-task pre-training. Under a typical *"pre-train then fine-tune"* paradigm, many NLP works attempted to design pre-training tasks that are relevant to downstream objectives (Févry et al., 2020; Wang et al., 2021b). Such approaches endow the model with task-specific knowledge acquired from massive pre-training data. For example, Wang et al. (2021b) learned a knowledge embedding objective besides masked language modeling (MLM) to assist relation classification and

entity typing tasks; Févry et al. (2020) and Zhang et al. (2022) added an entity linking objective into pre-training for fact checking and question answering applications. These results have shown that designing proper downstream-oriented pre-training tasks is a promising direction. Such pre-training tasks are jointly trained with the MLM objective to learn relevant knowledge of downstream tasks, which can greatly reduce the gap between pre-training and fine-tuning. These tasks need to be self-supervised on pre-training corpus, while sharing a similar learning objective with downstream tasks so that relevant knowledge can be transferred.

Learning to multi-tasking. One critical issue in MTL is how to *train* a multi-task model. Existing works typically design MTL training strategies (e.g., weighing losses or task grouping) by human intuition and select the best framework through cross validation. Such model selection suffers from heavy computational cost when considering every possibility. Thus, a promising direction is to learn to multi-tasking. Meta learning is a popular approach while encountering "learning to learn" problems (Hospedales et al., 2021). It aims to allow the model to quickly learn a new task, given the experience of multiple learning episodes on different tasks. Wang et al. (2021a) tried to fuse the feature of fast adaptation of meta learning into an efficient MTL model. This approach preliminarily proved that meta-learning philosophy can benefit the training of MTL models. For future directions, using meta-learning to learn a general purpose multi-task learning algorithm is a promising route to "learning to multi-tasking". Besides, learning to group tasks through meta-learning is worthy of exploration.

5.2 Regarding Training Methods

Adaptive parameter sharing. Parameter sharing is believed to be an effective technique in improving the generalizability of multi-task learning models and reducing training time and memory footprint. Two popular parameter sharing methods are hard and soft sharing (Ruder, 2017). Hard parameter sharing (Bekoulis et al., 2018) means all tasks share a certain number of model layers before branching out. Soft parameter sharing (Duong et al., 2015) adds constraints to the distances between specific layers of different tasks. However, hard sharing suffers from finding the optimal configuration while soft sharing does not reduce the number of parameters. Therefore, in addition to empirically tuning which layers to share, learning adaptable sharing for efficient MTL is a promising solution. Sun et al. (2020) tried to allow adaptive sharing by learning which layers are used by each task through model training. This approach suits in the field of computer vision where many models have the architecture of stacking the same layer. However, in NLP neural networks, layers are functionally and structurally discrepant, such as the encoder-decoder framework. The development of proper adaptive sharing methods to improve parameter sharing in multi-task NLP models is needed.

Mutli-task leaning in training a universal model. Recently, training a universal model to perform a variety of tasks becomes an emerging trend in NLP. Multi-task supervised learning helps the model fuse knowledge from different domains, and encourages it to obtain universal representations that generalize to different downstream tasks. For example, Liu et al. (2019a) unified the input format of GLUE tasks to feed into a single model before fine-tuning on individual tasks. However, the role of multi-task learning is still unclear in training a universal model, with different approaches adopting MTL in different phases of transfer learning. Among recent works, Aribandi et al. (2021) preferred multi-task pre-training over multi-task fine-tuning for smaller gaps between pre-training and fine-tuning. Aghajanyan et al. (2021) used multi-task pre-finetuning on a self-supervised pre-trained model before further fine-tuning on downstream tasks. Sanh et al. (2022) used prompted multi-task fine-tuning over a pre-trained T5 (Raffel et al., 2020) in order to perform zero-shot transfer on out-of-domain tasks. Therefore, future researches may dive deeper into maximizing the benefits of MTL in the transfer learning paradigm, including the choice of properly including MTL in pre-training or fine-tuning for better generalization. Besides, a theoretical analysis of transfer learning regarding the benefits of MTL is also desired.

6 Conclusions

In this paper, we reviewed recent work on multi-task learning for NLP tasks. According to the types of task relatedness, we categorized multi-task NLP approaches into two typical frameworks: joint training and multi-step training. We presented the design of each framework in various NLP applications, and discussed future directions of this interesting topic.

7 Limitations

Due to the space constraint, we are only able to show some prominent application scenarios of joint training and multi-step training, in which we may not cover all existing fields with multi-task approaches. For example, in dialogue systems, dialogue act recognition and sentiment recognition can be jointly trained to capture speakers' intentions. Besides, zero-shot and few-shot approaches in the multi-task setting are also interesting directions.

As for another limitation, this work is purely theoretical without any software-level implementation of the mentioned framework. In addition, we did not list the experimental results of the mentioned models on benchmark datasets because of the space limit.

Acknowledgements

This work was supported by NSF IIS-2119531, IIS-2137396, IIS-2142827, CCF-1901059, and ONR N00014-22-1-2507. Wenhao Yu is also supported in part by Bloomberg Data Science Ph.D Fellowship. We would also like to thank Libo Qin from Harbin Institute of Technology for his valuable suggestions to this paper.

References

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. Muppet: Massive multi-task representations with pre-finetuning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Wasi Uddin Ahmad, Zhisong Zhang, Xuezhe Ma, Kai-Wei Chang, and Nanyun Peng. 2019. Cross-lingual dependency parsing with unlabeled auxiliary languages. In *The SIGNLL Conference on Computational Natural Language Learning (CoNLL)*.

Vamsi Aribandi, Yi Tay, Tal Schuster, Jinfeng Rao, Huaixiu Steven Zheng, Sanket Vaibhav Mehta, Honglei Zhuang, Vinh Q Tran, Dara Bahri, Jianmo Ni, et al. 2021. Ext5: Towards extreme multi-task scaling for transfer learning. In *ArXiv preprint*.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. In *ArXiv preprint*.

Jonathan Baxter. 2000. A model of inductive bias learning. In *Journal of artificial intelligence research*.

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. Adversarial training for multi-context joint entity and relation extraction. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2018. Retrieve, rerank and rewrite: Soft template based neural summarization. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Rich Caruana. 1997. Multitask learning. In *Machine learning*.

Giuseppe Castellucci, Valentina Bellomaria, Andrea Favalli, and Raniero Romagnoli. 2019. Multi-lingual intent detection and slot filling in a joint bert-based model. In *ArXiv preprint*.

Jaemin Cho, Minjoon Seo, and Hannaneh Hajishirzi. 2019. Mixture content selection for diverse sequence generation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference for Learning Representation (ICLR)*.

Kevin Clark, Minh-Thang Luong, Christopher D Manning, and Quoc V Le. 2018. Semi-supervised sequence modeling with cross-view training. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Michael Crawshaw. 2020. Multi-task learning with deep neural networks: A survey. In *ArXiv preprint*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference for Learning Representation (ICLR)*.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Xiangyu Dong, Wenhao Yu, Chenguang Zhu, and Meng Jiang. 2020. Injecting entity types into entity-guided text generation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. 2017. Question generation for question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Zhihao Fan, Zhongyu Wei, Piji Li, Yanyan Lan, and Xuanjing Huang. 2018. A question type driven framework to diversify visual question generation. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Thibault Févry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. 2020. Entities as experts: Sparse memory access with entity supervision. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Chris Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. 2021. Efficiently identifying task groupings for multi-task learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Lin Gui, Leng Jia, Jiyun Zhou, Ruifeng Xu, and Yulan He. 2020. Multi-task learning with mutual learning for joint sentiment classification and topic detection. In *IEEE Transactions on Knowledge and Data Engineering (TKDE)*.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2018. Soft layer-specific multi-task summarization with entailment and question generation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2019. Autosem: Automatic task selection and mixing in multi-task learning. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2017. A joint many-task model: Growing a neural network for multiple nlp tasks. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. In *Deep Learning Workshop at Advances in Neural Information Processing Systems (NeurIPS)*.

Timothy M Hospedales, Antreas Antoniou, Paul Micaelli, and Amos J Storkey. 2021. Meta-learning in neural networks: A survey. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Minghao Hu, Yuxing Peng, Zhen Huang, and Dongsheng Li. 2019. Retrieve, read, rerank: Towards end-to-end multi-document reading comprehension. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. In *Transactions of the Association for Computational Linguistics (TACL)*.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Byeongchang Kim, Jaewoo Ahn, and Gunhee Kim. 2020. Sequential latent knowledge selection for knowledge-grounded dialogue. In *International Conference for Learning Representation (ICLR)*.

Kalpesh Krishna, Aurko Roy, and Mohit Iyyer. 2021. Hurdles to progress in long-form question answering. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Souvik Kundu, Tushar Khot, Ashish Sabharwal, and Peter Clark. 2019. Exploiting explicit paths for multi-hop reading comprehension. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Haoran Li, Junnan Zhu, Jiajun Zhang, Chengqing Zong, and Xiaodong He. 2020. Keywords-guided abstractive sentence summarization. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Yankai Lin, Haozhe Ji, Zhiyuan Liu, and Maosong Sun. 2018a. Denoising distantly supervised open-domain question answering. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Ying Lin, Shengqi Yang, Veselin Stoyanov, and Heng Ji. 2018b. A multi-lingual multi-task architecture for low-resource sequence labeling. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, et al. 2019a. Choosing transfer languages for cross-lingual learning. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Zhaojiang Lin, Genta Indra Winata, and Pascale Fung. 2019b. Learning comment generation by leveraging user-generated data. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Bing Liu and Ian R. Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. In *17th Annual Conference of the International Speech Communication Association*.

Pengfei Liu, Xipeng Qiu, and Xuan-Jing Huang. 2017. Adversarial multi-task learning for text classification. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-yi Wang. 2015. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Multi-task deep neural networks for natural language understanding. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yijin Liu, Fandong Meng, Jinchao Zhang, Jie Zhou, Yufeng Chen, and Jinan Xu. 2019b. Cm-net: A novel collaborative memory network for spoken language understanding. In *Conference on Empirical Methods in Natural Language Processing, (EMNLP)*.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *International Conference for Learning Representation (ICLR)*.

Yuren Mao, Weiwei Liu, and Xuemin Lin. 2020. Adaptive adversarial multi-task representation learning. In *International Conference on Machine Learning (ICML)*.

Kyosuke Nishida, Itsumi Saito, Atsushi Otsuka, Hisako Asano, and Junji Tomita. 2018. Retrieve-and-read: Multi-task learning of information retrieval and reading comprehension. In *International Conference on Information and Knowledge Management (CIKM)*.

Raheel Qader, François Portet, and Cyril Labbé. 2019. Semi-supervised neural text generation by joint learning of natural language generation and natural language understanding models. In *Proceedings of the 12th International Conference on Natural Language Generation*.

Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2019. A stack-propagation framework with token-level intent detection for spoken language understanding. In *Conference on Empirical Methods in Natural Language Processing(EMNLP)*.

Libo Qin, Tailu Liu, Wanxiang Che, Bingbing Kang, Sendong Zhao, and Ting Liu. 2021a. A co-interactive transformer for joint slot filling and intent detection. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Libo Qin, Fuxuan Wei, Tianbao Xie, Xiao Xu, Wanxiang Che, and Ting Liu. 2021b. GL-GIN: fast and accurate non-autoregressive model for joint multiple intent detection and slot filling. In *Annual Meeting of the Association for Computational Linguistics(ACL)*.

Libo Qin, Tianbao Xie, Wanxiang Che, and Ting Liu. 2021c. A survey on spoken language understanding: Recent advances and new frontiers. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

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. In *Journal of Machine Learning Research*.

Marek Rei. 2017. Semi-supervised multitask learning for sequence labeling. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. In *ArXiv preprint*.

Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. 2017. Sluice networks: Learning what to share between loosely related tasks. In *ArXiv preprint*.

Mrinmaya Sachan and Eric Xing. 2018. Self-training for jointly learning to ask and answer questions. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019a. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *Workshop on Energy Efficient Machine Learning and Cognitive Computing at Advances in Neural Information Processing Systems (NeurIPS)*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference for Learning Representation (ICLR)*.

Victor Sanh, Thomas Wolf, and Sebastian Ruder. 2019b. A hierarchical multi-task approach for learning embeddings from semantic tasks. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Dinghan Shen, Mingzhi Zheng, Yelong Shen, Yanru Qu, and Weizhu Chen. 2020. A simple but tough-to-beat data augmentation approach for natural language understanding and generation. In *ArXiv preprint*.

Tao Shen, Xiubo Geng, QIN Tao, Daya Guo, Duyu Tang, Nan Duan, Guodong Long, and Daxin Jiang. 2019. Multi-task learning for conversational question answering over a large-scale knowledge base. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Anders Søgaard, Yoav Goldberg, et al. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Zhenqiao Song, Xiaoqing Zheng, Lu Liu, Mu Xu, and Xuan-Jing Huang. 2019. Generating responses with a specific emotion in dialog. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. 2020. Which tasks should be learned together in multi-task learning? In *International Conference on Machine Learning (ICML)*.

Jimin Sun, Hwijeen Ahn, Chan Young Park, Yulia Tsvetkov, and David R Mortensen. 2021. Ranking transfer languages with pragmatically-motivated features for multilingual sentiment analysis. In *European Chapter of the Association for Computational Linguistics (EACL)*.

Ximeng Sun, Rameswar Panda, Rogerio Feris, and Kate Saenko. 2020. Adashare: Learning what to share for efficient deep multi-task learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Duyu Tang, Nan Duan, Tao Qin, Zhao Yan, and Ming Zhou. 2017. Question answering and question generation as dual tasks. In *ArXiv preprint*.

Ming Tu, Guangtao Wang, Jing Huang, Yun Tang, Xiaodong He, and Bowen Zhou. 2019. Multi-hop reading comprehension across multiple documents by reasoning over heterogeneous graphs. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Rob Van Der Goot, Ibrahim Sharaf, Aizhan Imankulova, Ahmet Üstün, Marija Stepanović, Alan Ramponi, Siti Oryza Khairunnisa, Mamoru Komachi, and Barbara Plank. 2021. From masked language modeling to translation: Non-english auxiliary tasks improve zero-shot spoken language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Haoxiang Wang, Han Zhao, and Bo Li. 2021a. Bridging multi-task learning and meta-learning: Towards efficient training and effective adaptation. In *International Conference on Machine Learning (ICML)*.

Shaolei Wang, Yue Zhang, Wanxiang Che, and Ting Liu. 2018a. Joint extraction of entities and relations based on a novel graph scheme. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Gerald Tesauro, Bowen Zhou, and Jing Jiang. 2018b. R3: Reinforced ranker-reader for open-domain question answering. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Shuohang Wang, Mo Yu, Jing Jiang, Wei Zhang, Xiaoxiao Guo, Shiyu Chang, Zhiguo Wang, Tim Klinger, Gerald Tesauro, and Murray Campbell. 2018c. Evidence aggregation for answer re-ranking in open-domain question answering. In *International Conference for Learning Representation (ICLR)*.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021b. Kepler: A unified model for knowledge embedding and pre-trained language representation. In *Transactions of the Association for Computational Linguistics (TACL)*.

Zhen Wang, Jiachen Liu, Xinyan Xiao, Yajuan Lyu, and Tian Wu. 2018d. Joint training of candidate extraction and answer selection for reading comprehension. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. In *Transactions of the Association for Computational Linguistics (TACL)*.

Joseph Worsham and Jugal Kalita. 2020. Multi-task learning for natural language processing in the 2020s. In *Pattern Recognition*.

Wenquan Wu, Zhen Guo, Xiangyang Zhou, Hua Wu, Xiyuan Zhang, Rongzhong Lian, and Haifeng Wang. 2019. Proactive human-machine conversation with explicit conversation goal. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Liqiang Xiao, Honglun Zhang, and Wenqing Chen. 2018. Gated multi-task network for text classification. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Caiming Xiong, Victor Zhong, and Richard Socher. 2018. Dcn+: Mixed objective and deep residual coattention for question answering. In *International Conference for Learning Representation (ICLR)*.

Chunpu Xu, Wei Zhao, Min Yang, Xiang Ao, Wangrong Cheng, and Jinwen Tian. 2019a. A unified generation-retrieval framework for image captioning. In *International Conference on Information and Knowledge Management (CIKM)*.

Yichong Xu, Xiaodong Liu, Yelong Shen, Jingjing Liu, and Jianfeng Gao. 2019b. Multi-task learning with sample re-weighting for machine reading comprehension. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Min Yang, Lei Chen, Xiaojun Chen, Qingyao Wu, Wei Zhou, and Ying Shen. 2019. Knowledge-enhanced hierarchical attention for community question answering with multi-task and adaptive learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Jianfei Yu, Luis Marujo, Jing Jiang, Pradeep Karuturi, and William Brendel. 2018a. Improving multi-label emotion classification via sentiment classification with dual attention transfer network. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Jianfei Yu, Minghui Qiu, Jing Jiang, Jun Huang, Shuangyong Song, Wei Chu, and Haiqing Chen. 2018b. Modelling domain relationships for transfer learning on retrieval-based question answering systems in e-commerce. In *International Conference on Web Search and Data Mining (WSDM)*.

Wenhao Yu, Lingfei Wu, Qingkai Zeng, Yu Deng, Shu Tao, and Meng Jiang. 2020a. Crossing variational autoencoders for answer retrieval. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, and Meng Jiang. 2020b. A survey of knowledge-enhanced text generation. In *ACM Computing Survey (CSUR)*.

Linhao Zhang, Dehong Ma, Xiaodong Zhang, Xiaohui Yan, and Houfeng Wang. 2020. Graph LSTM with context-gated mechanism for spoken language understanding. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Yu Zhang and Qiang Yang. 2018. An overview of multi-task learning. In *National Science*.

Zhihan Zhang, Xiubo Geng, Tao Qin, Yunfang Wu, and Daxin Jiang. 2021. Knowledge-aware procedural text understanding with multi-stage training. In *Proceedings of the Web Conference*.

Zhihan Zhang, Wenhao Yu, Chenguang Zhu, and Meng Jiang. 2022. A unified encoder-decoder framework with entity memory. In *Conference on Empirical Methods in Natural Language Processing(EMNLP)*.

Suncong Zheng, Yuexing Hao, Dongyuan Lu, Hongyun Bao, Jiaming Xu, Hongwei Hao, and Bo Xu. 2017. Joint entity and relation extraction based on a hybrid neural network. In *Neurocomputing*.

Xin Zhou, Luping Liu, Xiaodong Luo, Haiqiang Chen, Linbo Qing, and Xiaohai He. 2019. Joint entity and relation extraction based on reinforcement learning. In *IEEE Access*. IEEE.

Junnan Zhu, Qian Wang, Yining Wang, Yu Zhou, Jiajun Zhang, Shaonan Wang, and Chengqing Zong. 2019. Ncls: Neural cross-lingual summarization. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.