Fusing External Knowledge Resources for Natural Language Understanding Techniques: A Survey

Yuqi Wang[a], Wei Wang[a,*], Qi Chen[b], Kaizhu Huang[c], Anh Nguyen[d], Suparna De[e], Amir Hussain[f]

[a] School of Advanced Technology, Xi'an Jiaotong Liverpool University, 111 Ren'ai Road, Suzhou, 215123, Jiangsu, China
[b] School of AI and Advanced Computing, Xi'an Jiaotong Liverpool University, 111 Ren'ai Road, Suzhou, 215123, Jiangsu, China
[c] Data Science Research Center, Duke Kunshan University, No.8 Duke Avenue, Kunshan, 215316, Jiangsu, China
[d] Department of Computer Science, University of Liverpool, Liverpool, L69 3BX, UK
[e] Department of Computer Science, University of Surrey, Surrey, GU2 7XH, UK
[f] School of Computing, Edinburgh Napier University, 219 Colinton Road, Edinburgh, EH11 4BN, UK

Abstract

Knowledge resources, e.g. knowledge graphs, which formally represent essential semantics and information for logic inference and reasoning, can compensate for the unawareness nature of many natural language processing techniques based on deep neural networks. This paper provides a focused review of the emerging but intriguing topic that fuses quality external knowledge resources in improving the performance of natural language processing tasks. Existing methods and techniques are summarised in three main categories: 1) static word embeddings, 2) sentence-level deep learning models, and 3) contextualised language representation models, depending on when, how and where external knowledge is fused into the underlying learning models. We focus on the solutions to mitigate two issues: knowledge inclusion and inconsistency between language and knowledge. Details on the design of each representative method, as well as their strength and limitation, are discussed. We also point out some potential future directions in view of the latest trends in natural language processing research.

Keywords: Natural language understanding, knowledge graph, knowledge fusion, representation learning, deep learning

1. Introduction

Recent years have witnessed a thrilling development of deep learning in natural language processing (NLP) tasks, enabling machines to better comprehend and interpret human languages. However, many techniques are, in fact, solely based on the distributional hypothesis [1], [2], [3], thus lacking sufficient knowledge to capture true and intended semantic meanings from texts and to deal with knowledge-driven problems. Existing knowledge resources, such as WordNet [4], DBPedia [5] and Freebase [6], which contain plentiful quality and useful knowledge accumulated over the years, can be applied to many NLP applications [7]. With appropriate use of such knowledge, the performance of such tasks, e.g. classification [8], inference [9] and summarisation [10], could be greatly improved. It is particularly effective in low-resource learning applications [11], e.g. zero-shot to few-shot scenarios. Figure 1 shows an example that a structural resource provides the conceptual relations between entities mentioned in the given premise and hypothesis, which extensively benefits the inference process.

When fusing knowledge from external knowledge resources into NLP applications, two major challenges need to be considered.

- **Knowledge inclusion**: knowledge resources, such as knowledge graphs (KG), store an extraordinarily large number of entities, their literal information and relations. It is not uncommon to have millions or even billion of entities and their relations in many KGs. The scale and complexity pose major challenges to techniques for knowledge integration. How should we define the scope of the knowledge bases to be used and how to make use of such knowledge both effectively and efficiently?
- **Inconsistency of knowledge and language**: in previous research, a number of notable models have been developed for generating language representations, e.g. Word2Vec [12], [13] and GloVe [14], and knowledge representations, e.g. TransE [15] and its extensions [16], [17], [18]. These two types of representations, however, are generated in different and separate manners. How should we bridge the gap between knowledge and language representations generated in different semantic spaces?

Early research incorporates linguistic knowledge, e.g. synonyms and antonyms, from external knowledge resources for optimisation to improve the quality of word embedding [19], [20], [21], [22], [23], [24]. To bridge the gap between knowledge and language, methods based on joint representation learning have attracted a lot of interest [25], [26], [27], [28], [29], [30]. The basic idea is to align both words and entities from a knowledge base into a unified semantic space, or encode the textual data, e.g. documents and sentences, to the knowledge space using deep learning models such as Long Short-Term Memory (LSTM) [31] and Convolutional Neural Network

*Corresponding author
Email address: wei.wang03@xjtlu.edu.cn (Wei Wang)

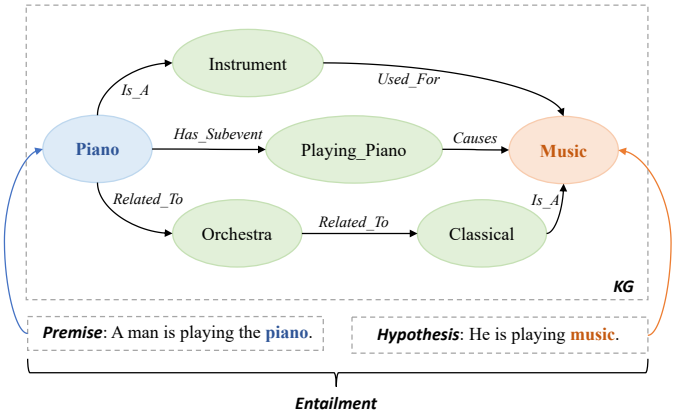

Figure 1: Illustration of how external knowledge resources can benefit NLP applications. A natural language inference (NLI) task is used as an example. With the background knowledge in the knowledge graph, relations between "*piano*" and "*music*" mentioned in the premise and hypothesis can be better understood. Apart from NLI, the knowledge graph also has the potential to facilitate a better understanding of the texts in many other NLP tasks. The figure is adapted from [9].

(CNN) [32]. These methods have been substantially beneficial to not only common tasks within the scope of KG, e.g. triple classification [25], [27] and link prediction [28], [30], but also those in text mining, e.g. relation extraction [33], [34] and named entity disambiguation [26], [35].

Recently, large-scale pre-trained models (PTMs), such as OpenAI GPT [36], BERT [37] and Roberta [38], have become the dominant paradigms for NLP applications. They utilise the attention-based mechanisms, e.g. Transformers [39], to capture critical features from contextualised information, which have been used for many downstream tasks. Following the development of these techniques, there is an increasing interest in injecting structured knowledge into PTMs to adequately explore their capabilities of knowledge inference [40], [41], [42], [43], [44]. With this idea, better results for many classic NLP tasks such as text classification [40], [41] and question answering [45], [46] can be anticipated. A reasonable analogy is that a researcher in computer science is most likely to classify literature better and answer questions on computer science than others in economics or bioscience.

We focus on the convergence of knowledge resources, deep learning, and NLP applications and define the scope of the study as the methods and techniques that integrate external knowledge resources to enhance the performance of common NLP tasks. Although there are already some existing surveys with an emphasis on the KG embedding [47], NLP techniques [48] and applications [49], to the best of our knowledge, this is the first survey that provides a comprehensive review on the key techniques and methods for fusing external knowledge into deep learning-based NLP applications. Figure 2 demonstrates the taxonomy of the methods for knowledge integration in this paper. Our main contributions are summarised as follows:

- We present a taxonomy for knowledge integration into deep neural network-based NLP applications. More specifically, existing works are categorised into three groups: 1) static word embeddings, 2) sentence-level deep learning models, and 3) contextualised language representation models.
- A comprehensive review of many representative methods for fusing external knowledge is conducted. We provide a specific focus on the theoretical formulation and optimisation methods of the existing studies, e.g. knowledge constraints in NLP techniques, and alignment processes between knowledge and language. Mathematical notations across a plethora of publications are harmonised for better comparison and easier understanding.
- Several promising future directions regarding knowledge-integrated language models based on the recent research trends are discussed, e.g. knowledgeable prompt-based learning, continual knowledge fusion and neurosymbolic learning.

The rest of the paper is structured as follows. In Section 2, we briefly introduce the preliminary knowledge relating to our study. Section 3 reviews methods on static word embedding learning with external knowledge. Section 4 reviews methods to bridge knowledge and language. Section 5 reviews recent methods and techniques to integrate knowledge in PTMs. Section 6 discusses the future directions, and finally, Section 7 concludes this survey.

2. Preliminary

2.1. KG and Knowledge Resources

We categorise the KG and knowledge resources mentioned in this study into 3 groups.

- **Linguistic Knowledge Resources**
  Knowledge resources in this group provide useful linguistic information. Thesaurus [50] can serve as a dictionary, listing synonyms, antonyms and definitions of words. WordNet [4] is a large English lexical database that contains diverse semantic relations between words. The paraphrase database (PPDB) [51] contains a large collection of syntactic, phrasal and lexical paraphrase expression pairs, with scores to indicate the probability and similarity of each pair.
- **Common Sense Knowledge Resources**
  Common sense KG, expressing real-world truths in graph-based structures, is the most representative type of knowledge resource in this group. Each instance in the KG is usually represented as a Resource Description Framework (RDF) triple, consisting of two entities (a head entity and a tail entity) connected by a relation (or predicate), e.g. $(United States, is\_type\_of, Country)$. Freebase [6], DBPedia [5], Wikidata [52] and ConceptNet [53] are examples that are collaboratively developed, automatically extracted or transformed, and maintained by expert users. They provide different kinds of structured world knowledge with a

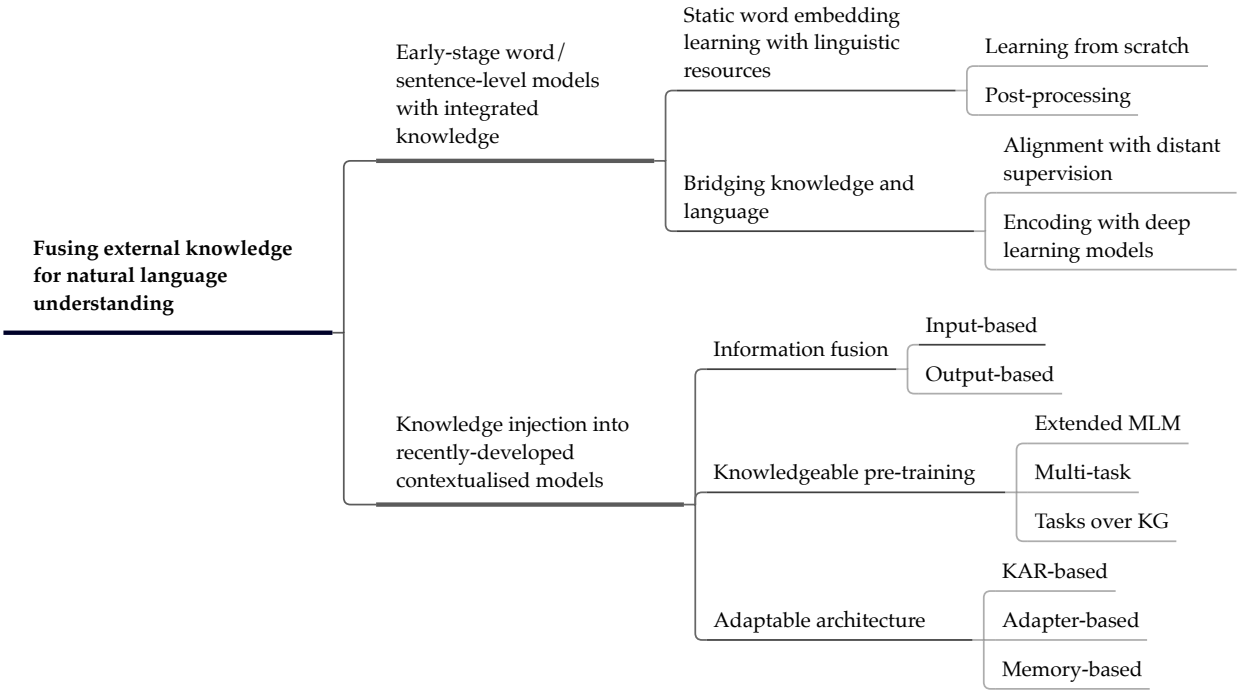

Figure 2: A taxonomy of methods and techniques for knowledge integration into the deep neural network-based NLP applications

great breadth (e.g. commonsense or complex), which can be incorporated and utilised in applications.

- **Domain Specific Knowledge Resources**
  Knowledge bases that provide domain-specific knowledge can be used to complement general knowledge when performing tasks relating to a specific domain. Some of the well-established domain-specific knowledge bases and ontologies include the Unified Medical Language System (UMLS)[1] for terminologies, classification and coding standards, and relations in the biomedical science domain; Medical Subject Headings (MeSH)[2] thesaurus for biomedical and health-related information; Gene Ontology (GO)[3] for functions of genes; and DrugBank[4] for the world's most robust drug knowledge.

2.2. KG Embeddings

To better integrate knowledge into deep learning models, KG embedding has been proposed as an effective technique which encodes elements of a KG, i.e. relations and entities, into a continuous low-dimensional vector representation. This technique can reduce computational complexity while preserving the original structural information of the KG (for comprehensive literature surveys on KG embedding algorithms, please refer to [47] and [54]). We only briefly describe some of the most well-known KG embedding techniques for knowledge integration.

TransE [15] is one of the early KG embedding techniques based on the notion of translational distance. It encodes relations and entities in a KG to a set of numerical vectors in the same semantic space. For each triple (h, r, t), the relation embedding $\mathbf{r}$ should be as close as possible to the translation from the head entity embedding $\mathbf{h}$ to the tail $\mathbf{t}$, i.e. $\mathbf{r} \approx \mathbf{t} - \mathbf{h}$. The distance function f is formulated as:

$$f_r(h, t) = \|\mathbf{r} - (\mathbf{t} - \mathbf{h})\|_{1/2} \tag{1}$$

During training, both positive and negative examples are used, and the objective function for the translational model is defined as follows:

$$\sum_{(h,r,t)\in\mathcal{S}} \sum_{(h',r',t')\in\mathcal{S}'} \max\left\{0, \gamma + f_r(h, t) - f_{r'}(h', t')\right\} \tag{2}$$

where $\mathcal{S}$ is the set of positive triples and $\mathcal{S}'$ is the set of negative triples; γ is the margin to separate the positive and negative examples.

An issue with TransE is that it is not able to handle complex relations, e.g. many-to-many relations, due to the simplicity of the algorithm [16]. To mitigate this issue, several extensions have been proposed; for instance, TransD [17], and TransR [18] suggest that the semantic space for entities and relations should be separated, thus projecting each entity to the new space for relation.

Recently, Graph Neural Network (GNN) [55] based models,

[1] https://www.nlm.nih.gov/research/umls/
[2] https://www.nlm.nih.gov/mesh/
[3] http://geneontology.org/
[4] https://www.drugbank.com/

such as R-GCN [56] and TransGCN [57], are becoming increasingly popular. They can directly process graph structures to model the entities and relations in a KG. The neighbourhood information is aggregated and accumulated throughout the KG via the message passing framework [58], i.e.

$$\mathbf{h}^{l+1} = \sigma\left(\sum_{m\in\mathcal{M}_i} g_m\left(\mathbf{h}^l, \mathbf{t}^l\right)\right) \tag{3}$$

where $\mathbf{h}^l$ is the representation of head entity h at the l-th layer of the graph neural network; $\mathcal{M}_i$ stands for the set of all incoming messages for h and $g_m(\cdot)$ is the message-oriented function of message m from $\mathcal{M}$. $\sigma(\cdot)$ is a non-linear function, e.g. Sigmoid function, to activate the l-th layer. The accumulated information from neighbours contributes to the representation of entities at the next layer, $\mathbf{h}^{l+1}$. In this way, the topological structure can be well leveraged to produce more effective representations. During training, a similar objective function as Equation 2 is usually used [59] to construct entity embeddings.

2.3. Static Word Embeddings

Word embedding is an important language representation technique for text analysis. It has a long history of development, from static to contextualised [60]. The objective of static word embedding is to learn only one representation for each word in a vocabulary $\mathcal{V}$. The pre-trained embedding for each word remains unchanged regardless of where it appears. On the contrary, contextualised word embedding generates different representations for each word based on its context.

Word2Vec [12], [13] is one of the most representative static word embedding techniques. It has two model architectures for pre-training: Continuous Bag-Of-Words (CBOW) and Skip-Gram (SG). Both of them are 3-layer neural networks and learn embeddings with a context window from a text corpus $\mathcal{T}$. The CBOW model maximises the log-likelihood of each centre word given its context. The objective function for this model can be defined as:

$$\frac{1}{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{T}|}\log p(w_i|\mathcal{W}_i^c) \tag{4}$$

where c is the size of the context window; $\mathcal{W}_i^c$ represents context words of the centre word w_i, i.e. $\mathcal{W}_i^c = \{w_{i-c}, ...w_{i-1}, w_{i+1}, ..., w_{i+c}\}$.

The SG model is the reversed version of the CBOW model, which uses the centre word to predict surrounding words. It has the following objective function.

$$\frac{1}{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{T}|}\sum_{-c\le k\le c, k\neq 0}\log p(w_{i+k}|w_i) \tag{5}$$

The likelihood of the word w_j given the word w_i is estimated as:

$$p(w_j|w_i) = \frac{\exp\left(\mathbf{w}_i^\top\widetilde{\mathbf{w}}_j\right)}{\sum_{k=1}^{|\mathcal{V}|}\exp\left(\mathbf{w}_i^\top\widetilde{\mathbf{w}}_k\right)} \tag{6}$$

where $\mathbf{w}_i$ and $\widetilde{\mathbf{w}}_i$ are input and output representations of w_i, respectively.

GloVe [14] is another commonly used static word embedding technique. While the training of Word2Vec is over local context windows, GloVe utilises the global information by including a co-occurrence matrix $\mathbf{M}$. Similar to Word2Vec, each word w_i is associated with two representations. Here $\mathbf{w}_i$ is defined as the target representation, and $\widetilde{\mathbf{w}}_i$ serves as the representation in the context of other words. The objective function for GloVe is defined as:

$$\sum_{i,j=1}^{|\mathcal{V}|} f\left(\mathbf{M}_{i,j}\right)\left(\mathbf{w}_i^\top\widetilde{\mathbf{w}}_j + \mathbf{b}_i + \widetilde{\mathbf{b}}_j - \log\mathbf{M}_{i,j}\right)^2 \tag{7}$$

where $\mathbf{b}_i$ and $\widetilde{\mathbf{b}}_j$ are biases for the target word w_i and its context word w_j, respectively; f is a function to prevent the overweighting of co-occurrences.

2.4. Sentence-level Deep Learning Models

One of the main limitations of static word embeddings is that such representations are context-independent, while the same word may convey slightly or completely different meanings based on the context where it appears. To address this issue, methods for sentence-level representations can be used.

LSTM [31] and CNN [32] are two popular models to encode languages. Normally, the representations of each word in the input are initialised with static word embeddings, and then sent to the neural networks for encoding into the deep semantic space. LSTM is usually used for sequential or temporal data modelling and contains three types of gates: input gate, forget gate and output gate. With sufficient supervision, it can properly perform time-series retrieval and control how much information is remembered/forgotten. CNN can better extract position-invariant textual features. The core components are the convolution and pooling layers, which generate feature maps from text input with kernels and downsample the output to preserve the most important information.

2.5. Contextualised Language Representation Models

OpenAI GPT [36], BERT[37] are transformer-based contextualised language representation models. OpenAI GPT captures critical features from left to right, while BERT incorporates contextual information from both directions. Specifically, given an input sequence $s = \{w_1, w_2, ..., w_n\}$, BERT computes the token-level representation $\mathbf{W}^0 = \{\mathbf{w}_1^0, \mathbf{w}_2^0, ..., \mathbf{w}_n^0\}$, where $\mathbf{w}_1^0$ is the summation of token embedding, segment embedding and position embedding of w_1. The contextualised representation is calculated recursively with bidirectional transformers [39], i.e.

$$\mathbf{W}^l = \text{TransformerBlock}_l(\mathbf{W}^{l-1}) \tag{8}$$

where $\mathbf{W}^l$ stands for the contextualised representation in the l-th layer. BERT is pre-trained over two unsupervised tasks: Next Sentence Prediction (NSP) and Masked Language Modelling (MLM) simultaneously.

Given two sentences s_a and s_b constituting the input s, the objective of NSP is to predict whether s_a is followed by s_b, which can be formalised as:

$$\log p(y|s_a, s_b) \tag{9}$$

where $y = 1$ if s_a and s_b are consecutive sentences, otherwise 0. This task can help the model better understand the relationship between sentences.

MLM is first proposed by Talyor *et al.*[61] to measure the learning ability of a language model. In the pre-training stage of the BERT model, some words in a sequence s will be randomly masked, and MLM aims to predict the original words. The loss function of MLM is defined as follows:

$$\sum_{w \in \mathcal{M}(s)} \log p\left(w|s_{\setminus \mathcal{M}(s)}\right) \tag{10}$$

where $\mathcal{M}(s)$ is the set of all randomly masked words in the sequence s; $s_{\setminus \mathcal{M}(s)}$ is the input with all masked words removed from s.

3. Static Word Embeddings with Linguistic Resources

As discussed in Section 2.3, static word embeddings are generated according to the distribution of words, i.e. frequently co-occurred words tend to have similar representations. This shows that the reliability of word embeddings is subject to the pre-training corpus. Moreover, the linguistic relations between words are not taken into consideration.

In this section, we describe the methods and techniques that employ knowledge from linguistic resources to improve the quality of static word embedding. Existing research in this line can be divided into two categories: *learning from scratch* and *post-processing*. The former introduces additional training objectives to the distribution-based word embeddings, while the latter refines pre-trained word embeddings with knowledge from external resources. Such knowledge-aware word embeddings usually have the capability to distinguish between senses of words, e.g. synonyms and antonyms, and can be applied to important lexical semantic tasks, such as word analogy reasoning, word similarity measurement and synonym selection, to benefit higher-level NLP applications.

3.1. Learning from Scratch

Learning representation for each word from scratch usually employs the linear combination of the original training objectives for word embeddings and additional objectives. Here we only focus on the additional objectives as the commonly-used static word embedding models have already been presented in Section 2.3. We assume that the linguistic knowledge resources that identify relations between words are given.

Yu and Dredze [19] proposed a relation-constrained objective to maximise the log-likelihood of w_j given w_i if there is a lexical relation between w_i and w_j in the linguistic knowledge resources (PPDB and WordNet are used in this research) during word embedding learning, i.e.

$$\frac{1}{|\mathcal{V}|} \sum_{w_i \in \mathcal{V}} \sum_{w_j \in \mathcal{R}(w_i)} \log p(w_j|w_i) \tag{11}$$

where $p(w_j|w_i)$ can be obtained using Eq. (6), which is consistent with Word2Vec [13]. $\mathcal{R}(w_i)$ is the set of words are linked with word w_i by a lexical relation. The aim of this method is to produce a higher probability of one word given another word which has a relation with the previous word, complementing the original Word2Vec [13], where only words with co-occurrence in the pre-training corpus will be considered to be related.

Bian *et al.*[62] presented an auxiliary objective for CBOW. They assign a weight λ_r to each relation r in the set $\mathcal{R}$. The weight indicates the importance of the relation r in linguistic knowledge. Moreover, they predict the related word w_j with the surroundings of w_i from a given training corpus $\mathcal{T}$, i.e. $\mathcal{W}_i^c$, rather than a single word w_i to be compatible with the original CBOW model. This objective function is defined as:

$$\frac{1}{|\mathcal{T}|} \sum_{i=1}^{|\mathcal{T}|} \sum_{r \in \mathcal{R}} \lambda_r \sum_{w_j \in \mathcal{R}_r(w_i)} \log p\left(w_j|\mathcal{W}_i^c\right) \tag{12}$$

Intuitively, they believed that if two words have a relation with each other in the linguistic knowledge resource, given the surroundings of one word in the training corpus, the probability of another word should also be higher.

Inspired by the translational model TransE [15], Xu *et al.*[20] proposed R-NET to incorporate relational knowledge into SG. Each word itself can be regarded as an entity so that each triple comes from the triple set $\mathcal{S}$. The additional objective is similar to Eq. (2), i.e.

$$\sum_{(w_i,r,w_j) \in \mathcal{S}} \sum_{\left(w_i',r',w_j'\right) \in \mathcal{S}'} \max\left\{0, \gamma + f_r\left(w_i, w_j\right) - f_{r'}\left(w_i', w_j'\right)\right\} \tag{13}$$

where w_i and w_j are words that serve as head and tail entities, respectively; (w_i, r, w_j) represents a triple from the knowledge resources. In this way, multiple relations can be better incorporated into the objective function.

Apart from relational knowledge, they suggested that categorical knowledge can also be leveraged to further improve word representations. The idea is that words with similar attributes can be grouped into the same category. If a category only contains a few words, then it is more likely to be a specific one, and the words in it should be highly-related. On the contrary, if a category includes a large number of words, it tends to be general, which reflects the relatively low degree of similarity between words in this category [20]. Therefore, they proposed another additional objective to effectively make use of this heuristic:

$$\sum_{w_i \in \mathcal{V}} \sum_{w_j \in \mathcal{V}} \beta_{ij} \left\|\mathbf{w}_i - \mathbf{w}_j\right\|_2 \tag{14}$$

where β_{ij} stands for the weight, indicating the degree of similarity between w_i and w_j according to the categorical knowledge.

Ono *et al.*[21] and Nguyen *et al.*[63] focused on the most basic relations from linguistic knowledge resources: synonyms and antonyms. Instead of the probabilistic approaches, they mainly consider the semantic similarity of word embeddings. Specifically, the methods add a constraint to enforce synonyms to have similar representations and antonyms to have dissimilar ones, i.e.

$$\begin{aligned}&\sum_{w_i\in\mathcal{V}}\sum_{w_j\in\mathcal{R}_S(w_i)}\log\sigma\left(sim\left(\mathbf{w}_i,\mathbf{w}_j\right)\right)\\&+\alpha\sum_{w_i\in\mathcal{V}}\sum_{w_j\in\mathcal{R}_A(w_i)}\log\sigma\left(-sim\left(\mathbf{w}_i,\mathbf{w}_j\right)\right)\end{aligned}\tag{15}$$

where $sim(\cdot)$ is a function to calculate the similarity score between two vectors, and σ is the Sigmoid function. $\mathcal{R}_S(w_i)$ and $\mathcal{R}_A(w_i)$ are the synonym and antonym sets of word w_i, respectively. α is a parameter to control how much the antonym can contribute to the objective function.

Liu *et al.*[64] defined a set of ordinal rules that take into account both relational and categorical knowledge. They generated a set O, where each instance contains 3 words: w_i, w_j and w_k. These 3 words meet one of the following conditions: 1) w_i and w_j are synonyms while w_i and w_k are antonyms; 2) w_i and w_j belong to the same category while w_i and w_k belong to different category; 3) The distance between w_i and w_j is shorter than w_i and w_k in the hypernym tree. Therefore, the similarity score between w_i and w_j should be higher than w_i and w_k, which forms an inequality: $sim(\mathbf{w}_i,\mathbf{w}_k) > sim(\mathbf{w}_i,\mathbf{w}_j)$. Based on this, they proposed the additional ordinal constraint:

$$\sum_{(w_i,w_j,w_k)\in O}\sigma\left(sim\left(\mathbf{w}_i,\mathbf{w}_k\right)-sim\left(\mathbf{w}_i,\mathbf{w}_j\right)\right)\tag{16}$$

Bollegala *et al.*[65] believed that by employing the global co-occurrence instead of local co-occurrence, one might acquire better representations. Hence, they used GloVe [14] as the base model and proposed an additional objective to incorporate relational knowledge.

$$\frac{1}{2}\sum_{w_i\in\mathcal{V}}\sum_{w_j\in\mathcal{R}(w_i)}\left\|\mathbf{w}_i-\widetilde{\mathbf{w}}_j\right\|^2\tag{17}$$

where $\mathbf{w}_i$ is the target word embedding and $\widetilde{\mathbf{w}}_j$ is the context word embedding. In their work, both symmetric and asymmetric lexical relations are taken into consideration. The word embedding, under this definition, moves closer to the context embeddings of its related words.

3.2. Post-processing

In this category, word embeddings can be refined with important information from external knowledge resources.

Retrofitting, first proposed by Faruqui *et al.*[22], is a process of updating pre-trained word embeddings based on graph-structured data from external knowledge resources, e.g. WordNet. The fundamental idea is to place neighbour nodes in the graph closer. The main objective can be formulated as follows:

$$\sum_{w_i\in\mathcal{V}}\sum_{w_j\in\mathcal{R}(w_i)}\beta_{ij}\left\|\mathbf{w}_i-\mathbf{w}_j\right\|^2\tag{18}$$

where β_{ij} is the associative weight between words w_i and w_j.

Meanwhile, it is also important to preserve the distributional information gained during the pre-training phase. Therefore, a regularisation term is introduced into the post-processing to ensure that the updated word embeddings do not move too far from the original ones, i.e.

$$\sum_{w_i\in\mathcal{V}}\alpha_i\left\|\mathbf{w}_i-\hat{\mathbf{w}}_i\right\|^2\tag{19}$$

where $\hat{\mathbf{w}}_i$ is the original word embedding for w_i; α_i represents another associative weight between updated word embedding and original word embedding.

To specialise the word embedding based on linguistic information for word semantic similarity tasks, the method proposed by Kiela *et al.*[66] maximises the log-likelihood of word w_j given w_i after the pre-trained distributional word embeddings are generated, if these two words are associated in the knowledge resources. More specifically, it treats words connected by a specific relation as the context from a corpus and performs exactly the same optimisation as in the pre-training, i.e. Skip-Gram, during the post-processing. Compared to retrofitting [22], this method performs well in terms of incorporating the auxiliary thesaurus information. However, without any regularisation, the original information gained from pre-training may be lost after the post-processing.

Inspired by retrofitting, Mrksic *et al.*[23] proposed counter-fitting, a method that can inherently capture more accurate similarity by considering both synonyms and antonyms. They considered three terms in their design: Antonym Repel, Synonym Attract, and Vector Space Preservation. The main idea is to pull synonyms closer while pushing away antonyms from each other during post-processing. A regularisation term is also introduced to maintain the distributional information from the original embeddings, i.e.

$$\sum_{w_i\in\mathcal{V}}\sum_{w_j\in\mathcal{V}}\max\left\{0,sim(\hat{\mathbf{w}}_i,\hat{\mathbf{w}}_j)-sim(\mathbf{w}_i,\mathbf{w}_j)\right\}\tag{20}$$

This regularisation focuses on the semantic similarity between two words, which is more consistent with the type of knowledge used in this work. Later, they refined this approach and proposed the Attract-Repel model by introducing negative examples to fine-tune the word embeddings in a context-aware way [67]. In particular, they created mini-batches for synonyms and antonyms from the knowledge resources as well as negative examples. Ideally, synonyms are enforced to have more similar representations in the semantic space than negative examples, while antonyms are enforced to have more dissimilar representations than negative examples. The model can be formulated as follows:

Table 1: Summary of learning from the scratch methods

References	Base.	Constraints	Remarks
[19]	CBOW	$\frac{1}{\|\mathcal{V}\|}\sum_{w_i\in\mathcal{V}}\sum_{w_j\in\mathcal{R}(w_i)}\log p(w_j\|w_i)$	Maximise the log-likelihood of w_j given w_i if w_i has a relation with w_j during the word embedding learning.
[62]	CBOW	$\frac{1}{\|\mathcal{T}\|}\sum_{i=1}^{\|\mathcal{T}\|}\lambda_r\sum_{w_j\in\mathcal{R}_r(w_i)}\log p\left(w_j\|\mathcal{W}_i^c\right)$	Maximise the log-likelihood of w_j given the context of w_i from a training sequence $\mathcal{T}$ weighted by the relation r.
[20]	SG	$\sum_{(w_i,r,w_j)\in\mathcal{S}}\sum_{(w_i',r',w_j')\in\mathcal{S}'}\max\{0,\ \gamma+f_r\left(w_i,w_j\right)-f_{r'}\left(w_i',w_j'\right)\};$ $\sum_{w_i\in\mathcal{V}}\sum_{w_j\in\mathcal{V}}\beta_{ij}\left\|\mathbf{w}_i-\mathbf{w}_j\right\|_2$	Incorporate relational knowledge using TransE model [15] and categorical knowledge with the weight β_{ij} between w_i and w_j.
[21] [63]	SG	$\sum_{w_i\in\mathcal{V}}\sum_{w_j\in\mathcal{R}_S(w_i)}\log\sigma\left(sim\left(\mathbf{w}_i,\mathbf{w}_j\right)\right)$ $+\alpha\sum_{w_i\in\mathcal{V}}\sum_{w_j\in\mathcal{R}_A(w_i)}\log\sigma\left(-sim\left(\mathbf{w}_i,\mathbf{w}_j\right)\right)$	Enforce synonyms/antonyms to have similar/dissimilar embeddings.
[64]	SG	$\sum_{(w_i,w_j,w_k)\in\mathcal{O}}\sigma\left(sim\left(\mathbf{w}_i,\mathbf{w}_k\right)-sim\left(\mathbf{w}_i,\mathbf{w}_j\right)\right)$	Constrain the word embeddings based on ordinal rules.
[65]	GloVe	$\frac{1}{2}\sum_{w_i\in\mathcal{V}}\sum_{w_j\in\mathcal{R}(w_i)}\left\|\mathbf{w}_i-\widetilde{\mathbf{w}}_j\right\|^2$	Enforce words linked by relations to have similar embeddings.

Table 2: Summary of Post-processing method

References	Main Objectives	Regularisations	Remarks
[22]	$\sum_{w_i\in\mathcal{V}}\sum_{w_j\in\mathcal{R}(w_i)}\beta_{ij}\left\|\mathbf{w}_i-\mathbf{w}_j\right\|^2$	$\sum_{w_i\in\mathcal{V}}\alpha_i\left\|\mathbf{w}_i-\hat{\mathbf{w}}_i\right\|^2$	Pull words linked by a relation closer.
[66]	$\frac{1}{\|\mathcal{V}\|}\sum_{w_i\in\mathcal{V}}\sum_{w_j\in\mathcal{R}(w_i)}\log p(w_j\|w_i)$	- -	Maximise the log-likelihood of w_j given w_i after the pre-trained distributional word embeddings are generated.
[23]	$\sum_{w_i\in\mathcal{V}}\sum_{w_j\in\mathcal{R}_A(w_i)}\max\{0,sim(\mathbf{w}_i,\mathbf{w}_j)\}$ $\sum_{w_i\in\mathcal{V}}\sum_{w_j\in\mathcal{R}_S(w_i)}\max\{0,-sim(\mathbf{w}_i,\mathbf{w}_j)\}$	$\sum_{w_i\in\mathcal{V}}\sum_{w_j\in\mathcal{V}}\max\{0,$ $sim(\hat{\mathbf{w}}_i,\hat{\mathbf{w}}_j)-sim(\mathbf{w}_i,\mathbf{w}_j)\}$	Pull synonyms closer while pushing away antonyms from each other.
[67]	$\sum_{(w_i,w_j)\in\mathcal{B}_S}\max\{0,(\mathbf{w}_i^\top\mathbf{w}_i'-\mathbf{w}_i^\top\mathbf{w}_j)$ $+(\mathbf{w}_j^\top\mathbf{w}_j'-\mathbf{w}_i^\top\mathbf{w}_j)\}$ $\sum_{(w_i,w_j)\in\mathcal{B}_A}\max\{0,(\mathbf{w}_i^\top\mathbf{w}_j-\mathbf{w}_i^\top\mathbf{w}_i')$ $+(\mathbf{w}_i^\top\mathbf{w}_j-\mathbf{w}_j^\top\mathbf{w}_j')\}$	$\sum_{w_i\in\mathcal{V}(\mathcal{B}_S\cup\mathcal{B}_A)}\alpha_i\left\|\mathbf{w}_i-\hat{\mathbf{w}}_i\right\|^2$	The idea is similar to [23]; introduce negative examples to fine-tune word embeddings in a context-aware way.
[68]	$\sum_{(w_i,w_j,d_{ij})\in\mathcal{B}}\left[cos\left(f\left(\mathbf{w}_i\right),f\left(\mathbf{w}_j\right)\right)-d_{ij}\right]^2$	$\sum_{(w_i,w_j)\in\mathcal{B}}[cos(\hat{\mathbf{w}}_i,f(\mathbf{w}_i))$ $+cos(\hat{\mathbf{w}}_j,f(\mathbf{w}_j))]$	The idea is similar to [23]; employ a non-linear specification function f to project each word embedding to a new semantic space.

$$\begin{aligned}\sum_{(w_i,w_j)\in\mathcal{B}_S} \max\left\{0,\left(\mathbf{w}_i^\top\mathbf{w}'_i-\mathbf{w}_i^\top\mathbf{w}_j\right)+\left(\mathbf{w}_j^\top\mathbf{w}'_j-\mathbf{w}_i^\top\mathbf{w}_j\right)\right\}\\ \sum_{(w_i,w_j)\in\mathcal{B}_A} \max\left\{0,\left(\mathbf{w}_i^\top\mathbf{w}_j-\mathbf{w}_i^\top\mathbf{w}'_i\right)+\left(\mathbf{w}_i^\top\mathbf{w}_j-\mathbf{w}_j^\top\mathbf{w}'_j\right)\right\}\end{aligned} \tag{21}$$

where w'_i is a negative sample for w_i, $\mathcal{B}_S$ and $\mathcal{B}_A$ are mini-batch for synonyms and antonyms, respectively.

The regularisation is similar to the retrofitting [22], avoiding each updated word embedding moving too far away from its original word embedding. During post-processing, the embedding will be fine-tuned with negative examples, which indicate a stronger association in the semantic space within each mini-batches. It also outperforms similar work in [69], which only takes synonyms into consideration, suggesting that employing both similarity and dissimilarity constraints can be more effective, especially while using cross-lingual or multilingual language resources.

Glavaš *et al.*[68] proposed a non-linear semantic specification function $f(\cdot)$ to map word embeddings to a deeper space to better capture their semantic similarity. Each instance in a batch $\mathcal{B}$ contains two words w_i and w_j from the vocabulary and their expected distance d_{ij} in the specified semantic space based on their relations. Both synonyms and antonyms are considered in the optimisation. The post-processing stage can be formulated as follows:

$$\sum_{(w_i,w_j,d_{ij})\in\mathcal{B}} \left[cos\left(f\left(\mathbf{w}_i\right),f\left(\mathbf{w}_j\right)\right)-d_{ij}\right]^2 \tag{22}$$

where $cos(\cdot)$ is the cosine distance function (which is computed as "1-cosine similarity"). The idea of this objective function is to make the cosine distance of two words as close as their expected distance in the specified semantic space. The regularisation in the method aims to minimise the cosine distance between the original word embedding in the original distributional space and the transformed word embedding by the semantic specification function $f(\cdot)$, i.e.

$$\sum_{(w_i,w_j)\in\mathcal{B}} \left[cos\left(\hat{\mathbf{w}}_i,f\left(\mathbf{w}_i\right)\right)+cos\left(\hat{\mathbf{w}}_j,f\left(\mathbf{w}_j\right)\right)\right] \tag{23}$$

3.3. Discussions

Existing works reviewed in this section are further classified into two sub-categories: *Learning from scratch* and *Post-processing*. Most of the methods under these two sub-categories make use of auxiliary objective functions which attempt to exploit linguistic constraints from external knowledge. As such, NLP models can produce powerful and knowledgeable word embeddings. However, there are some obvious limitations with the learning from scratch methods: 1) each method under this sub-category is limited to a certain word embedding technique, i.e. it specifies an underlying distributional objective which is not adaptable enough; and 2) it is computationally expensive, especially when the corpus size is large. On the contrary, post-processing methods are more flexible by refining the pre-trained embeddings with knowledge from linguistic resources while preserving the original topology.

A common problem in the studies [19], [65] and [66] is that all relations are assigned with the same weight. This is not desirable since different lexical relations often imply different degrees of relatedness among words. The works in [62] and [22] use associative weights between words based on their relations or strength of associations. However, selection of optimal parameters becomes a challenging issue, which relies heavily on empirical evidence for different NLP tasks. The studies in [21], [63], [23] and [67] only take two lexical relations (synonyms and antonyms) in linguistic knowledge resources into account. They learn word embeddings in a contrastive way to cluster synonyms and push away antonyms in the semantic space. The works in [20] and [64] define customised rules to construct linguistic constraints. Overall, the design of these methods is intuitive; however, the results of several downstream tasks suggest that more fine-grained linguistic information should also be included for semantic specialisation to deal with sophisticated languages.

4. Bridging Knowledge and Language

Unlike linguistic resources such as WordNet, some KGs contain common sense and factual data knowledge regarding the real world. During knowledge integration, the alignment between texts and entities from such external knowledge resources is an essential process. Existing methods can be categorised into two groups: 1) Alignment with distant supervision: language and knowledge representations, e.g. word embeddings and KG embeddings, are first computed separately, and then alignment is performed to map representations to a unified space with distant supervision; and 2) Encoding with deep learning models: deep learning models are used to directly encode texts into the KG embedding space. Bridging the gap between knowledge and language representations can better link and resolve the mentioned real-world entities or relations in documents with accurate references to external knowledge resources. This is useful in some text mining tasks, such as named entity recognition, mention disambiguation and relation extraction.

4.1. Alignment with Distant Supervision

As presented in Section 2.2 and 2.3, there are several common approaches to obtaining word embeddings based on co-occurrence from the corpora and entity/relation embeddings based on the knowledge resources. However, these two kinds of embeddings are represented in separate semantic spaces. To connect them for relation extraction, Weston *et al.*[70] learned a mapping function $f(\cdot)$ to project a mention from text into the KG embedding space. For each mention-relation pair (m_i, r_i) and an irrelevant relation r'_i, they proposed a constraint with 1 as the margin, i.e.

$$\forall(m_i,r_i)\ f\left(m_i\right)^\top\mathbf{r}_i>1+f\left(m_i\right)^\top\mathbf{r}'_i \tag{24}$$

where $\mathbf{r}_i$ is the relation representation for r_i. In this way, the inner product of each mention is transformed by the mapping function $f(\cdot)$, and its corresponding relevant relation can always be larger than the one with an irrelevant relation. According to this constraint, the ranking loss is employed to optimise the function $f(\cdot)$ and word embeddings.

Wang *et al.*[25] proposed two alignment techniques: The former is to employ the Wikipedia Anchors, an external tool that can connect words or phrases in an English Wikipedia page to their corresponding entities in Freebase KG. The loss function of the probabilistic-based alignment model can be written as follows:

$$\sum_{(w_i,w_j)\in C} \log p(w_i|e_{w_j}) \tag{25}$$

where (w_j, e_{w_j}) is the connection between a word w_j and entity e_{w_j} from the Wikipedia Anchor; w_i and w_j are from the same context, denoted as C, in one of the Wikipedia pages. This method assumes that all words from the content in the Wikipedia pages of an entity should be strongly-related to the entity. Due to the unambiguity and completeness of this tool, a similar idea is also used in [26] and [71] for the named entity and mention disambiguation.

The latter is alignment by entity names. For a triple (h, r, t), where h, t, r are the head entity, tail entity and their corresponding relation, they generated some new triples by replacing the entity with one or several words according to the entity name. The alignment process is formulated as follows:

$$\sum_{(h,r,t)\in\mathcal{S}} [f_r(w_h, w_t) + f_r(h, w_t) + f_r(w_h, t)] \tag{26}$$

where $\mathcal{S}$ is the set of all triples extracted from a KG; w_h and w_t are names of the head and tail entities, respectively. Compared to the alignment model with Wikipedia Anchors, this method does not depend on any external tools, which is more straightforward and adaptable.

While many methods focus on representation learning from symbolic triples, supplementary textual data, such as entity descriptions [72] and text corpora [73], which usually provide much more semantic information about an entity, can also be well utilised for alignment between knowledge and language. This textual data can effectively solve the sparsity issue [74] in most KGs. Therefore, Zhong *et al.*[27] proposed an alignment model based on entity descriptions. Let $D_e = \{w_1, w_2, ..., w_n\}$ stands for a set containing all words appearing in the description of an entity e_i; they defined the loss function of alignment model as follows:

$$\sum_{e_i} \sum_{w\in D_{e_i}} [\log p(w|e_i) + \log p(e_i|w)] \tag{27}$$

This method mutually updates the entity and word embeddings to align them by maximising the probabilities of an entity given the word from the entity description and, reversely, the word from the entity description given the entity. Newman-Griffis *et al.*[75] proposed another method to align words and entities with distant supervision based on an unannotated text corpus. For each entity e_i, there is a corresponding word w_i with an observed context in the corpus, denoted as $\mathcal{W}_i^c$, where c is the context window size. Moreover, a set of negative context words, denoted as $\mathcal{W}_i^{c\prime}$ is also used. The objective function is defined as follows (σ is the Sigmoid function).

$$\sum_{w\in\mathcal{W}_i^c} \log\sigma(\mathbf{w}^\top\mathbf{e}_i) + \sum_{w'\in\mathcal{W}_i^{c\prime}} \log\sigma(-\mathbf{w}'^\top\mathbf{e}_i) \tag{28}$$

where $\mathbf{e}_i$ is the embedding for entity e_i. The idea behind this approach is to enforce an entity and context words of this entity in the corpus to have similar representations while making the words, not from the context, have dissimilar representations. A similar approach is also adopted in [76] to make sure the consistency of language and knowledge representations.

4.2. Encoding with Deep Learning Models

With the advancement of deep learning, a considerable amount of neural network models have been proposed to encode language for automatic analysis and better representation in sentence-level [77], [78], [79]. Another solution to bridge the gap between language and knowledge is to encode textual data with deep learning models into the KG embedding space.

Toutanova *et al.*[28] pre-processed sentences using dependency parsing. They employed a neural network model to extract relations from annotated textual patterns. However, according to Han *et al.*[34], the linguistic analysis is relatively complicated and may lead to parsing errors, especially when the textual data is noisy. To address this problem, they utilised the CNN-based model proposed by Zeng *et al.*[80] to encode the plain text that contains two entities. The corresponding representations of the relation encoded by the deep CNN model and obtained using TransE [15] are denoted as $\bar{\mathbf{r}}$ and $\mathbf{r}$, respectively. They used a scoring function to minimise the distance between $\bar{r}$ and r in the unified semantic space, i.e. $\|\bar{\mathbf{r}} - \mathbf{r}\|_2$.

Research in [30], [81], [82], [83] concentrates on encoding descriptions, reference sentences or other textual information regarding entities with deep learning techniques into the KG embedding space. The overall procedure can be summarised in Figure 3, where $\mathbf{h}, \mathbf{r}$ and $\mathbf{t}$ are representations of the head entity, relation and tail entity, respectively, based on the structural knowledge from the KG. $\bar{\mathbf{h}}$ and $\bar{\mathbf{t}}$ are corresponding representations constructed from the descriptive knowledge (e.g. textual descriptions about entities).

Xie *et al.*[82] used a two-layered CNN model to encode descriptions of head and tail entities, and defined the scoring function based on TransE [15], i.e.

$$\begin{aligned} f_r(h,t) = &\left\|\bar{\mathbf{h}} + \mathbf{r} - \bar{\mathbf{t}}\right\|_{1/2} + \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{1/2} \\ &+ \left\|\bar{\mathbf{h}} + \mathbf{r} - \mathbf{t}\right\|_{1/2} + \left\|\mathbf{h} + \mathbf{r} - \bar{\mathbf{t}}\right\|_{1/2} \end{aligned} \tag{29}$$

However, Wu *et al.*[30] argued that this method is subject to the quality of the entity description. They suggested another way to find auxiliary textual information about an entity, which is to consider the sentences, including the entity name from a

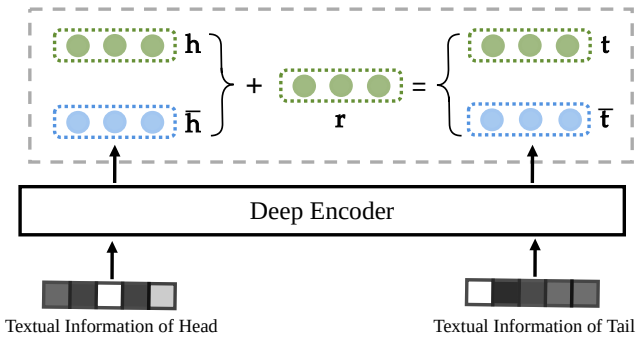

Figure 3: Encoding entity descriptions into KG embedding space. This figure is adapted from [30]

specific corpus. These selected sentences are regarded as reference sentences of an entity. Then, an attention-based LSTM [84] model is utilised for encoding all reference sentences. Xu *et al.*[81] used Bidirectional LSTM (Bi-LSTM) [85], which processes sequential text in both forward and backward directions, to encode textual descriptions of entities. Instead of separating the scoring function based on structural and descriptive knowledge, they proposed a gated unit to integrate these two different types of representations. The integration of the head entity is shown as follows:

$$\mathbf{g}_h \odot \bar{\mathbf{h}} + (1 - \mathbf{g}_h) \odot \mathbf{h} \tag{30}$$

where $\odot$ is the element-wise multiplication and $\mathbf{g}_h$ is a gate of the head entity. A similar way applies to the integration of the tail entity.

4.3. Discussion

Existing studies to bridge the gap between knowledge and language in this section are classified into two sub-categories: *Alignment with distant supervision* and *Encoding with deep learning models*.

Alignment methods with distant supervision exploit semantic similarity and relatedness information from external knowledge, and process textual data at the word-level. Tools and resources for alignment include symbolic triplets in the KG [70], [25], Wikipedia anchors [25] and other supplementary sequential data [27], [75]. The major problem with employing symbolic triplets is that they may break down while dealing with polysemy (e.g. the word "*bank*" may refer to a river bank or financial bank). In this case, the KG embedding is likely to be corrupted after the alignment. Moreover, different mentions in the text may refer to the same relation or entity in the triple, and it is infeasible to consider all possible mentions in the vocabulary. Alignment with Wikipedia anchors can mitigate this issue to a certain extent, while still suffering from the limitation of the number of word-entity pairs in this entity-linking system. The supplementary sequential data can provide the context or description of the entity. However, the comprehension and interpretation at the sentence-level are more significant due to the context-dependent nature of languages [86], [87], especially for the text mining and analysis tasks such as named entity recognition.

Sentence-level encoding with deep learning models has the potential to better resolve word sense ambiguity and vagueness, and to handle descriptive knowledge. However, it still has some obvious drawbacks, such as restrictions on the extent of words and entity vocabularies, as well as expensive model training.

5. Injecting Knowledge into Recently-developed Language Representation Models

Recently, large-scale pre-trained language models such as OpenAI GPT [36], BERT [37] and Roberta [38], have delivered state-of-the-art results in a variety of NLP problems. These models are pre-trained over unsupervised tasks with extraordinarily large-scale corpora containing general world knowledge. They can be further fine-tuned with extra labelled data in a plethora of downstream tasks and applications. In order to improve the performance or address the discrepancy between the source and target domain, it is necessary to equip them with more domain-specific knowledge [40]. Other than methods mentioned in Section 4.2, which implicitly learn and explore the representations in the KG embedding space, structured knowledge can also be explicitly injected into pre-trained models. Therefore, the process of knowledge integration can be more explainable and reasonable.

5.1. Information Fusion

One of the most intuitive ways to integrate knowledge from external resources is information fusion. This can be done at either the input or output of the pre-trained language representation models.

5.1.1. Fusion at Input

Fusion at the input can directly integrate external knowledge into models and generate contextualised global representation with auxiliary knowledge. Ke *et al.*[41] proposed SentiLARE, a BERT-based model for a number of downstream tasks in sentiment analysis. For each word in a sentence, corresponding linguistic knowledge related to the task from SentiWordNet [111] is acquired. At the token representation level, the model performs element-wise addition of the original BERT embeddings, word-level polarity embeddings and part-of-speech (POS) embeddings as the final input. Levine *et al.*[88] proposed SenseBERT to incorporate the senses of each word based on WordNet. They introduced a linear mapping between words and senses and merged the two representations to fit the transformer encoder. In this way, the input embeddings have an awareness of the senses of words, which significantly enhances the lexical understanding. Poerner *et al.*[89] employed a similar way to acquire the entity embeddings from token embeddings in the aligned Wikipedia-WordPiece vector space [26]. The token embeddings of the BERT model are concatenated with the corresponding entity embeddings to form the knowledge-enabled embeddings as the input. To help language models better understand the words with low frequency in the corpus, Wu *et*

Table 3: Summary of methods to inject knowledge into pre-trained language representation models

Categories	Methods	References	Remarks*
Information Fusion	Input-based	[41]	POS embeddings + polarity embeddings
		[88], [89]	Entity embeddings
		[90]	Entity type embeddings
		[91]	Note embeddings for rare words
		[40], [92]	Transformed triples
		[93]	Entity descriptions + Triples
	Output-based	[42], [94], [95]	Entity embeddings
		[96], [97]	Contextualised entity embeddings
		[98]	Entity embeddings + Metadata features
Knowledgeable Pre-training	Extended MLM	[99], [43]	MLM on synthetic knowledge-based corpus
		[100]	MLM at entity-level
		[45]	Entity replacement prediction
		[101]	MLM on knowledge
		[92], [102]	MLM on language and knowledge
	Multi-task	[103]	MLM + NSP + LRC
		[104]	MLM + NSP + EL
		[105]	MLM + KE
		[106]	MLM + MIM + DD
	Tasks over KG	[107]	Triple plausibility prediction
		[46]	Relation classification
Adaptable Architecture	KAR-based	[104]	A component inside BERT model with knowledge enhanced for re-contextualisation
	Adapter-based	[103], [46]	Inject different knowledge with separated parameter-efficient adapters
	Memory-based	[108], [109], [110]	Knowledge retrieval with interpretable memory access via memory layers

* For the first category, the required auxiliary information is provided in the remark column. For knowledgeable pre-training, the names of the corresponding tasks are listed.

al.[91] employed linguistic knowledge resources to construct a note dictionary. The note embedding will then be added to the embedding layer to enrich the semantic information.

Apart from this, Yamada *et al.*[90] suggested adding entity type embeddings for better relationship modelling. Different from the above-mentioned methods, Liu *et al.*[40] converted the original input sentence to a knowledge-integrated sentence tree after querying the KG with all entity names. The basic structure of the sentence tree is shown in Figure 4, which is rearranged and transformed to derive the input embeddings. They also proposed the mask-self attention, a variant of the attention mechanism proposed in [39], based on a visible matrix and soft-position embedding scheme during knowledge incorporation. Although providing transformed triples extracted from KG in the input has a positive impact on understanding the relations between entities, Xu *et al.*[93] argued that each entity in triples only contains surface names, which lacks necessary descriptive knowledge. Without sufficient entity descriptions, language models may fail to fully comprehend the actual meanings of entities. Therefore, in their work, descriptions of the entities from an online dictionary, i.e. Wiktionary, are encoded along with the text and corresponding triples by the language model.

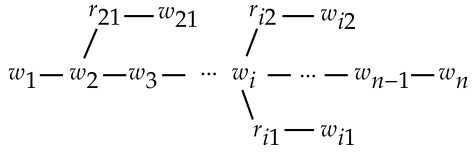

Figure 4: The basic structure of sentence tree converted from the input sentence. This figure is adapted from [40]

5.1.2. Fusion at Output

Language representations and knowledge from KG or other external knowledge resources can also be fused at the output. Since language and knowledge provide heterogeneous information, existing works prefer to adopt separate mechanisms to encode the textual data and required knowledge, or, alternatively, generate language representations in the deep semantic space first and then dynamically extract the corresponding knowledge from knowledge resources. All the information will be aggregated with the proposed mechanisms or in the neural layers.

Zhang *et al.*[42] utilised the multi-head transformer as the textual encoder to extract the semantic meanings from the input sentences. The output of the textual encoder and the aligned entities from the KG are then fed into the knowledgeable encoder for integration but with separate attention mechanisms, i.e.

$$\mathbf{o}_i = \begin{cases} \sigma\left(\mathbf{w}_i^l \bar{\mathbf{M}}^l + \mathbf{e}_j^l \mathbf{M}^l + \mathbf{b}^l\right) & \exists e_j \\ \sigma\left(\mathbf{w}_i^l \bar{\mathbf{M}}^l + \mathbf{b}^l\right) & \nexists e_j \end{cases} \tag{31}$$

where $\mathbf{o}_i$ is the output in the i-th position of the fused entity and word; $\sigma(\cdot)$ is the activation function in the information fusion layer; e_j is the corresponding entity for a word w_i; $\bar{\mathbf{M}}^l$ and $\mathbf{M}^l$ are the weight matrices of word and entity representations at the l-th layer of the knowledgeable encoder, respectively. For words without the corresponding entities, the knowledge integration step will be simply skipped.

He *et al.*[94] used the same approach to integrate bio-medical knowledge for domain-specific tasks with sub-graph construction from UMLS. However, one issue regarding the KG embedding generation for output-based information fusion in [42] has been reported in several recent studies [96], [97]. The extracted knowledge, according to the entity mention, can not appropriately match the textual context. To address this issue, they refined the KG embedding with dynamic information selection from KG based on the textual context output by the pre-trained language models.

Yu *et al.*[95] employed graph attention network [112], a GNN-based model, to encode both structural and descriptive knowledge of mentioned entities in the given sentences, which is then fused with the language representations from the BERT model. For better document classification performance, Ostendorff *et al.*[98] incorporated task-specific metadata from the external knowledge base, e.g. the number of authors, academic title, and author embeddings which are generated from the Wikidata KG. At the output layer of the BERT model, they concatenated BERT embeddings, metadata features and author embeddings as the knowledge-enabled representations for classification.

5.2. Knowledgeable Pre-training

Pre-training is a crucial stage for models to gain essential language understanding from large general domain corpora. According to [113], unsupervised pre-training tasks can perform better generalisation and speed up model convergence during the fine-tuning stage. Therefore, BERT adopts MLM and NSP as its pre-training tasks, which has been introduced in Section 2.3.

To learn common sense knowledge in the pre-training stage, Guan *et al.*[99] and Bosselut *et al.*[43] transformed triples from a KG to synthetic sentences based on the templates provided by Levy *et al.*[114] as the pre-training corpus. However, Sun *et al.*[100] identified a problem by using MLM to pre-train BERT: it cannot capture the essential high-level semantic information since each word in the input sentence is treated as the basic language unit. Therefore, they presented an extension of the MLM task at the entity-level. Instead of randomly masking single words in each sentence, they masked the entire named entities for the prediction task. Their results indicate that the model pre-trained with this approach on heterogeneous data can derive better language representations. Xiong *et al.*[45] proposed a replacement strategy at the entity-level in the pre-training stage, in which some entities in the input sentence are replaced, and the model predicts whether those entities have been replaced.

Even though the above-mentioned extended MLM tasks can help the pre-trained language models to be conscious of entities, they fail to fully extract and model useful knowledge from general plain texts for logical thinking [115]. To address this issue, Sun *et al.*[92] constructed a word graph from the input sentences and a sub-graph from knowledge resources as input. They performed the MLM tasks on words from the word graph

and relations and entities from the knowledge sub-graph. In this way, both language and knowledge representations are context-dependent. Kim *et al.*[102] formatted a symbolic triple as "head entity + head description + relation + tail + tail description" as input. Similar to Sun *et al.*[92], in the pre-training stage, they randomly masked words in entities, relations and descriptions.

Banerjee *et al.*[101] encoded names of symbolic triple (h, r, t) from KG using the transformer, with [SEP] as the separator, and then performed random masking on one of three items from the triple. Three functions $f_h(\cdot)$, $f_r(\cdot)$ and $f_t(\cdot)$ with the same transformer encoder are learned to predict the masked item, i.e.

$$f_t(h, r) \Rightarrow t, \quad f_h(r, t) \Rightarrow h, \quad f_r(h, t) \Rightarrow r \tag{32}$$

In addition to MLM and NSP, another line of research attempts to integrate knowledge into the language models by employing knowledge-intensive tasks to further pre-train the models. Lauscher *et al.*[116] proposed a new pre-training task, called lexical relation classification (LRC), to incorporate lexical information from WordNet and Thesaurus. They built a fully-connected layer on top of the final hidden state of the [CLS] token as the classifier for this task. Given a word pair, the model is required to predict if these two words have a particular lexical relation, e.g. synonym and meronym. The loss function is formalised as follows.

$$-\sum_{i,j}\Big[y \log p(y|w_i, w_j) + (1-y)\log\left(1 - p(y|w_i, w_j)\right)\Big] \tag{33}$$

where $y \in \{0, 1\}$ is the true value; $p(y|w_i, w_j)$ is the predicted output of the lexical relation classifier.

Peters *et al.*[104] proposed KnowBERT, which is jointly pre-trained with MLM, NSP and entity linking (EL) tasks to align BERT with entity embeddings. Similarly, Wang *et al.*[105] combined MLM and knowledge embedding (KE) loss to optimise the model in the pre-training stage. For the knowledge embedding loss, they encoded the descriptions of entities into BERT embedding space and employed the loss function of TransE [15]. Yu *et al.*[106] utilised a linguistic resource to co-train the BERT model with MLM and two proposed self-supervised tasks: mutual information maximisation (MIM) and definition discrimination (DD). The input sentence in their work is supplemented with two definitions of a rare word appearing in the text and a corrupted word. The goal of MIM is to maximise the mutual information between the contextualised representation of the rare word in the input text and its definition, while DD identifies which definition is the correct one for the rare word. Yao *et al.*[107] and Wang *et al.*[46] adopted triple plausibility prediction and relation classification as pre-training tasks, respectively, which require more inference and reasoning capability.

5.3. Adaptable and Interpretable Architectures

Although information fusion and pre-training can help models achieve better results on some specific NLP tasks, these approaches still have difficulties in facilitating the construction of more flexible models with the injection of knowledge from multiple sources [117]. In particular, previously learned knowledge may be abruptly lost when new knowledge is injected. This phenomenon is typically evidenced in human cognition while learning new knowledge and gradually forgetting the earlier one. In the field of machine learning, this is called catastrophic forgetting [118], a common issue associated with most pre-trained language models.

To make the architectures of these models more adaptable, they need to be extended with auxiliary components for knowledge injection. Peters *et al.*[104] proposed the Knowledge Attention and Re-contextualisation (KAR) component inside BERT to incorporate knowledge of different kinds. It performs entity linking and computes the pooled contextualised span representations for each mention with knowledge enhanced for re-contextualisation. When computing the entity embedding for each knowledge resource, network parameters that are not relevant to the entity linking task are frozen. Following [119], Lauscher *et al.*[103] investigated an adapter-based architecture to infuse knowledge into pre-trained language models. Each adapter layer consists of two fully-connected layers, which are inserted into the transformer layer, i.e.

$$\text{Adapter}(\mathbf{W}^l) = \mathbf{W}^l + f\left(\mathbf{W}^l\mathbf{M}_d + \mathbf{b}_d\right)\mathbf{M}_u + \mathbf{b}_u \tag{34}$$

where $\mathbf{W}^l$ is the output of the l-th transformer layer; down-projection weight matrix $\mathbf{M}_d$ (with bias $\mathbf{b}_d$) and up-projection weight matrix $\mathbf{M}_u$ (with bias $\mathbf{b}_u$) are parameters of the first and second fully-connected layers in the adapter, respectively. The output of the adapter, Adapter($\mathbf{W}^l$), will then be sent to the $(l+1)$-th transformer layer.

During the pre-training with knowledge from a specific resource, the parameters in the pre-trained language models remain unchanged, and only those in the adapters are adjusted. Since the number of parameters in the adapters is much fewer than those in the transformers, the efficiency of the pre-training can be well guaranteed.

Wang *et al.*[46] proposed another adapter-based architecture called K-adapter. Instead of adding the adapter layers inside the transformers [103], K-adapter treats adapters as a separate mechanism from the Roberta model. They pre-trained two independent adapters with different knowledge resources, i.e. a linguistic adapter and a factual adapter. The overall architecture is shown in Figure 5. To this extent, knowledge of multiple kinds can be injected continually into the language representation models without knowledge forgetting.

Human accumulated knowledge is constantly evolving, which inevitably necessitates data modification and additional model training. To avoid expensive computation, Verga *et al.*[109] proposed the Facts-as-Experts model, a transformer-based language model with an entity memory module [108] and a fact memory module, which is related to memory neural networks [120]. The entity memory module contains the learned entity embeddings. In the fact memory module, for each triple extracted from a KG, the head entity and relation are stored as the key, while the tail entity is stored as the corresponding value. The architecture of this model is shown in Figure 6, in

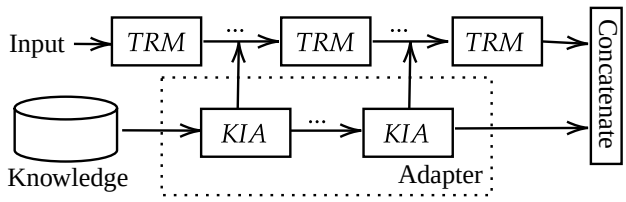

Figure 5: The adapter-based architecture, where TRM is the transformer layer, and KIA is the adapter layer. The figure is adapted from [46].

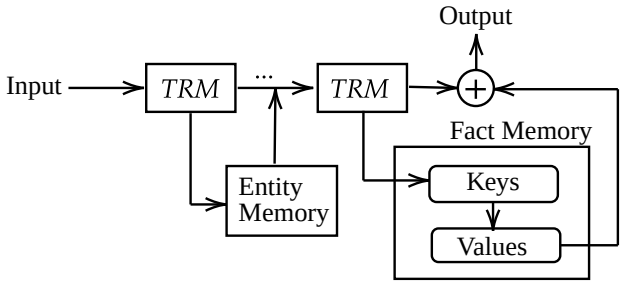

Figure 6: Pre-trained language model with memory modules. The figure is adapted from [109]

which the transformer is enriched with the entities from the entity memory and then utilises the contextualised representation to query the fact memory for the final prediction. With the supervised memory access, the architecture can still work well on factoid question-answering tasks even if the factual knowledge is modified. De Jong *et al.*[110] also emphasised the importance of internal memory enhancement for knowledge integration, especially when information assimilation and retrieval from multiple knowledge resources are required. In their work, each entity mentioned in the textual context interacts with the memory that consists of dense mention representations created from a large textual corpus with linked entities, e.g. Wikipedia.

5.4. Discussion

This section reviews the research regarding knowledge injection into recently-developed language representation models. Existing studies are further classified into three sub-categories: *information fusion*, *knowledgeable pre-training* and *adaptable architecture*.

Information fusion can be either input-based or output-based. Fusion at input normally supplements the textual input with the additional information, e.g. POS [41], transformed triples from KG [92] or the corresponding pre-trained entity embeddings [88]. Due to the fact that many knowledge resources, especially common sense KGs, are not purely tailored for the NLP models, directly adding such information into the textual input unavoidably introduces some noise, posing a negative impact on the performance of language encoding. As for the output-based fusion, the output of the language representations by the contextualised model is aggregated with the extracted knowledge. Studies such as [42] and [94] that utilised two separate mechanisms for encoding language and knowledge may cause the problem that the encoded knowledge may not highly match the textual context. This can be largely solved by the dynamic knowledge selection according to the output of the contextualised representations [96], [97].

Knowledgeable pre-training can also help models gain awareness of external knowledge. Recent studies under this sub-category propose some extensions to the original MLM, adding new tasks in addition to MLM, or employing pre-training tasks in the scope of KGs. Such extended MLM tasks greatly improve the quality of the original MLM. However, entity-oriented masking scheme [100], [45] ignores the essential relational information between entities in the KGs. In this case, MLM on the given knowledge, e.g. triples [101] appears to be a more effective approach to incorporate knowledge. Pre-trained language models can also learn language representations and understand the corresponding required knowledge simultaneously via multi-task learning. However, the significant discrepancy in the convergence time for each task makes it difficult to determine the actual time of training. Another common issue of knowledgeable pre-training is that, once the model is pre-trained with a specific knowledge resource, it is usually extremely difficult to be adapted to other domains that require different knowledge. Moreover, pre-training with these tasks more or less has the risk of causing the acquisition bottleneck of knowledge.

The adaptable design of transformer-based architectures for knowledge fusion has been gaining popularity very recently. Injecting multiple types of knowledge in adapter-based architectures is efficient since each adapter is designed for a specific type of knowledge and can be separated from the transformer blocks; however, it lacks sufficient interpretability. Memory-based architectures introduce interpretable memory access via memory layers, yet they often suffer from the over-parameterisation issue.

6. Challenges and Future Directions

Research in recent years has shown that it is possible and effective to bring different kinds of knowledge, e.g. common sense, factual and linguistic, from external knowledge resources into deep neural network-based models for NLP applications. However, there are still challenges and unsolved issues according to the research trends in recent years. In this section, we suggest and discuss a number of notable future directions.

6.1. Extreme Zero-shot Learning

Many language representation models require a significant amount of training data to integrate external knowledge for desirable performance on NLP tasks [121], while little attention has been given to the cases where only a small proportion of or even no labelled data is available. To address this resource-intensive issue, there has been some research such as [122] and [123] that applies low-resource learning, e.g. zero-shot and few-shot learning, with knowledge from external knowledge resources for textual data analysis. However, these methods may

fail if highly related information cannot be obtained in the training phase since the performance on unseen data is heavily dependent on prior knowledge [124]. Therefore, inspired by the idea of Yin *et al.*[125], a more challenging yet realistic scenario needs to be considered: without any explicit model training, the knowledge can be directly leveraged for problem-solving for many NLP tasks.

6.2. Model Ability and Robustness

Computational capacity has been improving rapidly in recent years, and more focus has been paid to model ability and robustness. For example, recent research has shown the vulnerability of the state-of-the-art contextualised language models to adversarial attacks by inconspicuous modifications of the original textual input [126]. Adversarial learning is seen as one of the empirically successful solutions to address this issue [127]. Due to the semantic constraint and discrete nature, crafting adversarial examples for text is much more challenging in comparison with continuous data, e.g. images. Fortunately, with external knowledge resources, adversarial examples that guarantee similar or the same semantic meanings can be effectively and automatically generated. Nevertheless, the construction of large-scale knowledge bases is often based on automated algorithms for knowledge construction with a limited amount of human intervention; there unavoidably exists low-quality and noisy information [128] in which consistency, completeness and accuracy of the constructed knowledge can not be fully assured. In such situations, the performance of the language models on some common NLP tasks, such as reading comprehension, is likely to degenerate. Therefore, it is expected that more research on automated detection of low-quality knowledge will be needed to further improve the ability of knowledge-injected models.

6.3. Knowledgeable Prompt-based Learning

Prompt-based learning, which aims to close the gap in the standard "pre-training and fine-tuning" paradigm [129], has gained popularity recently for many NLP applications. Specifically, each input is wrapped into a task-oriented template to predict the masked word, which is then projected to the label space by a verbaliser [130]. In this way, downstream tasks can be reduced to the MLM problem, and consequently, no extra parameters are needed. Hu *et al.*[131] presented an approach that employs knowledge bases to construct the knowledgeable verbaliser. However, the pre-trained language model may not be able to select the most suitable label from the expanded label set since the model itself is not equipped with any domain-specific knowledge. Therefore, the proposed knowledgeable prompt-based learning approach can only deal with data in general domains and needs to be enhanced to tackle domain-specific tasks. We foresee that there is a great potential to combine prompt-based learning with external knowledge resources to fully exploit the effectiveness of this technique.

6.4. Knowledge Reusability and Transferability

According to [132], transfer learning is the next driver of machine learning success after supervised learning. Analogical to human behaviour of thinking, language representation models with injected knowledge can be reused to solve related problems. For example, with profound knowledge in mathematics, one may be able to answer questions on physics without much effort in additional learning since these two disciplines share much knowledge in common and mathematics is an intrinsic foundation of physics. Existing research on transfer learning mainly focuses on transferring features [133] or instances [134] from the source to target domains. However, to the best of our knowledge, the transferring of the injected knowledge has not been studied. This is inherently a very challenging issue due to the fact that some source domain-specific knowledge may have a negative impact on the target domain during transferring. Therefore, understanding the internal working of knowledge transferring is of great significance. Both theoretical and empirical studies on the reusability and transferability of injected knowledge are needed.

6.5. Continual Knowledge Fusion

So far, existing research on integrating knowledge into language models has not paid enough attention to the dynamic nature of world knowledge, e.g. existing knowledge evolved, new knowledge created or previously unseen knowledge brought into the existing applications. The problem is that most of the existing methods just build static models whose behaviour is difficult or impossible to be adapted to the change in knowledge. Re-training models from scratch every time when new knowledge is injected is obviously intractable. Continual learning aims to learn from an infinite stream of data and to solve the catastrophic forgetting problem, with the goal of gradually extending the previously acquired knowledge for future learning use [135]. Section 5.3 mentioned some of the newly designed models to improve the interpretability and adaptability of knowledge injection. However, the amount and scope of knowledge infused in these existing models are rather limited. Moreover, the individual components for different knowledge fusions are similar, reflecting the lack of consideration for heterogeneity in knowledge resources and difficulty level in pre-training tasks for different kinds of knowledge. This problem is closely related to the so-called stability-plasticity dilemma [135], in which plasticity refers to the ability to integrate new knowledge and stability for retaining previously learned knowledge. Therefore, there is still a long way to go for continual knowledge fusion in deep learning-based language representation models.

6.6. Neurosymbolic Learning

Neurosymbolic learning is an emerging research field that attempts to draw on the strength of both cognitive learning and symbolic manipulation [136]. Unlike conventional pure deep learning-based approaches, it aims to integrate widely used, auxiliary symbolic logics, such as probabilistic logic [137] and fuzzy logic [138] for high-level cognitive tasks. In this way, the learning and inference process can be made more transparent, straightforwardly interpretable and easily tracked by human beings. Apart from traditional rule-based logics, we believe that

knowledge bases which are largely built on traditional description logics, can also be better integrated and utilised as important resources for reasoning and decision-making in many NLP tasks. Neurosymbolic learning stands for an exciting research direction and would undoubtedly contribute to the grand vision of explainable AI.

7. Conclusion

Accumulated quality knowledge can extensively benefit language understanding to tackle many downstream NLP tasks. With the recent emergence of representation learning of knowledge and language, a plethora of studies has investigated how to fuse external knowledge into NLP applications to further improve their performance. We first briefly introduced the types of knowledge resources, common language representation models and KG embedding techniques. Then, we provided a taxonomy for the related research, i.e. integrating linguistic information into static word embeddings, bridging the gap between knowledge and language, and injecting knowledge into contextualised language models. We extensively reviewed the representative studies published at top journals and conferences relating to NLP and deep learning, with an emphasis on the theoretical formulation and optimisation methods. In addition, we identified the limitations of the current state-of-the-art models and discussed the possible future directions based on the focused review. We hope that our work provided a valuable overview of the status of the research and could motivate researchers to further explore and investigate this interesting and challenging topic.

Acknowledgments

We would like to thank all the reviewers for their valuable and helpful comments to help us improve the quality and presentation of the paper. This research is funded by the Postgraduate Research Scholarship (PGRS) at Xi'an Jiaotong-Liverpool University, contract number PGRS2006013, and partially supported by 2022 Jiangsu Science and Technology Programme (General Programme), contract number BK20221260.

References

[1] G. A. Miller, W. G. Charles, Contextual correlates of semantic similarity, Language and Cognitive Processes 6 (1) (1991) 1–28.

[2] Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin, A neural probabilistic language model, Journal of Machine Learning Research 3 (2003) 1137–1155.

[3] A. Mnih, G. E. Hinton, A scalable hierarchical distributed language model, Advances in Neural Information Processing Systems 21 (2008).

[4] G. A. Miller, Wordnet: a lexical database for english, Communications of the ACM 38 (11) (1995) 39–41.

[5] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, Dbpedia: A nucleus for a web of open data, in: The semantic web, Springer, 2007, pp. 722–735.

[6] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor, Freebase: a collaboratively created graph database for structuring human knowledge, in: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, 2008, pp. 1247–1250.

[7] P. Wang, J. Hu, H.-J. Zeng, Z. Chen, Using wikipedia knowledge to improve text classification, Knowledge and Information Systems 19 (3) (2009) 265–281.

[8] S. Kiefer, Case: Explaining text classifications by fusion of local surrogate explanation models with contextual and semantic knowledge, Information Fusion 77 (2022) 184–195. doi:https://doi.org/10.1016/j.inffus.2021.07.014.

[9] Z. Wang, L. Li, D. Zeng, Knowledge-enhanced natural language inference based on knowledge graphs, in: Proceedings of the 28th International Conference on Computational Linguistics, 2020, pp. 6498–6508.

[10] P. Wu, Q. Zhou, Z. Lei, W. Qiu, X. Li, Template oriented text summarization via knowledge graph, in: 2018 International Conference on Audio, Language and Image Processing (ICALIP), 2018, pp. 79–83. doi:10.1109/ICALIP.2018.8455241.

[11] J. Chen, Y. Geng, Z. Chen, I. Horrocks, J. Z. Pan, H. Chen, Knowledge-aware zero-shot learning: Survey and perspective, in: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21) Survey Track, 2021, pp. 4366–4373.

[12] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781 (2013).

[13] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems, 2013, pp. 3111–3119.

[14] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.

[15] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, Advances in Neural Information Processing Systems 26 (2013).

[16] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph embedding by translating on hyperplanes, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 28, 2014.

[17] G. Ji, S. He, L. Xu, K. Liu, J. Zhao, Knowledge graph embedding via dynamic mapping matrix, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2015, pp. 687–696.

[18] Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion, in: Twenty-ninth AAAI conference on artificial intelligence, 2015.

[19] M. Yu, M. Dredze, Improving lexical embeddings with semantic knowledge, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2014, pp. 545–550.

[20] C. Xu, Y. Bai, J. Bian, B. Gao, G. Wang, X. Liu, T.-Y. Liu, Rc-net: A general framework for incorporating knowledge into word representations, in: Proceedings of the 23rd ACM International Conference on Information and Knowledge Management, 2014, pp. 1219–1228.

[21] M. Ono, M. Miwa, Y. Sasaki, Word embedding-based antonym detection using thesauri and distributional information, in: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2015, pp. 984–989.

[22] M. Faruqui, J. Dodge, S. K. Jauhar, C. Dyer, E. Hovy, N. A. Smith, Retrofitting word vectors to semantic lexicons, in: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2015, pp. 1606–1615.

[23] N. Mrkšić, D. Ó. Séaghdha, B. Thomson, M. Gasic, L. M. R. Barahona, P.-H. Su, D. Vandyke, T.-H. Wen, S. Young, Counter-fitting word vectors to linguistic constraints, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 142–148.

[24] S. Rothe, H. Schütze, Autoextend: Extending word embeddings to embeddings for synsets and lexemes, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2015, pp. 1793–1803.

[25] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph and text jointly embedding, in: Proceedings of the 2014 Conference on Empirical Meth-

ods in Natural Language Processing (EMNLP), 2014, pp. 1591–1601.
[26] I. Yamada, H. Shindo, H. Takeda, Y. Takefuji, Joint learning of the embedding of words and entities for named entity disambiguation, in: Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, 2016, pp. 250–259.
[27] H. Zhong, J. Zhang, Z. Wang, H. Wan, Z. Chen, Aligning knowledge and text embeddings by entity descriptions, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 267–272.
[28] K. Toutanova, D. Chen, P. Pantel, H. Poon, P. Choudhury, M. Gamon, Representing text for joint embedding of text and knowledge bases, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 1499–1509.
[29] Y. Cao, L. Huang, H. Ji, X. Chen, J. Li, Bridging text and knowledge by learning multi-prototype entity mention embedding, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 1623–1633.
[30] J. Wu, R. Xie, Z. Liu, M. Sun, Knowledge representation via joint learning of sequential text and knowledge graphs, ArXiv abs/1609.07075 (2016).
[31] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.
[32] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324.
[33] S. Riedel, L. Yao, A. McCallum, B. M. Marlin, Relation extraction with matrix factorization and universal schemas, in: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2013, pp. 74–84.
[34] X. Han, Z. Liu, M. Sun, Joint representation learning of text and knowledge for knowledge graph completion, arXiv preprint arXiv:1611.04125 (2016).
[35] W. Fang, J. Zhang, D. Wang, Z. Chen, M. Li, Entity disambiguation by knowledge and text jointly embedding, in: Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, 2016, pp. 260–269.
[36] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding with unsupervised learning (2018).
[37] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, 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), 2019, pp. 4171–4186.
[38] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).
[39] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.
[40] W. Liu, P. Zhou, Z. Zhao, Z. Wang, Q. Ju, H. Deng, P. Wang, K-bert: Enabling language representation with knowledge graph, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 2901–2908.
[41] P. Ke, H. Ji, S. Liu, X. Zhu, M. Huang, Sentilare: Sentiment-aware language representation learning with linguistic knowledge, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 6975–6988.
[42] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, Q. Liu, Ernie: Enhanced language representation with informative entities, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 1441–1451.
[43] A. Bosselut, H. Rashkin, M. Sap, C. Malaviya, A. Celikyilmaz, Y. Choi, Comet: Commonsense transformers for automatic knowledge graph construction, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 4762–4779.
[44] Z. Zhang, Y. Wu, H. Zhao, Z. Li, S. Zhang, X. Zhou, X. Zhou, Semantics-aware bert for language understanding, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 9628–9635.
[45] W. Xiong, J. Du, W. Y. Wang, V. Stoyanov, Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model, in: International Conference on Learning Representations, 2019.
[46] R. Wang, D. Tang, N. Duan, Z. Wei, X.-J. Huang, J. Ji, G. Cao, D. Jiang, M. Zhou, K-adapter: Infusing knowledge into pre-trained models with adapters, in: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, 2021, pp. 1405–1418.
[47] Q. Wang, Z. Mao, B. Wang, L. Guo, Knowledge graph embedding: A survey of approaches and applications, IEEE Transactions on Knowledge and Data Engineering 29 (12) (2017) 2724–2743.
[48] D. W. Otter, J. R. Medina, J. K. Kalita, A survey of the usages of deep learning for natural language processing, IEEE Transactions on Neural Networks and Learning Systems 32 (2) (2021) 604–624. doi:10.1109/TNNLS.2020.2979670.
[49] S. Sun, C. Luo, J. Chen, A review of natural language processing techniques for opinion mining systems, Information Fusion 36 (2017) 10–25. doi:https://doi.org/10.1016/j.inffus.2016.10.004.
[50] B. A. Kipfer, Roget's 21st century thesaurus in dictionary form: the essential reference for home, school, or office, Laurel, 1993.
[51] J. Ganitkevitch, B. Van Durme, C. Callison-Burch, Ppdb: The paraphrase database, in: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2013, pp. 758–764.
[52] D. Vrandečić, M. Krötzsch, Wikidata: a free collaborative knowledgebase, Communications of the ACM 57 (10) (2014) 78–85.
[53] R. Speer, J. Chin, C. Havasi, Conceptnet 5.5: An open multilingual graph of general knowledge, in: Thirty-first AAAI conference on artificial intelligence, 2017.
[54] H. Cai, V. W. Zheng, K. C.-C. Chang, A comprehensive survey of graph embedding: Problems, techniques, and applications, IEEE Transactions on Knowledge and Data Engineering 30 (9) (2018) 1616–1637.
[55] M. Welling, T. N. Kipf, Semi-supervised classification with graph convolutional networks, in: International Conference on Learning Representations (ICLR 2017), 2017.
[56] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. v. d. Berg, I. Titov, M. Welling, Modeling relational data with graph convolutional networks, in: European Semantic Web Conference, Springer, 2018, pp. 593–607.
[57] L. Cai, B. Yan, G. Mai, K. Janowicz, R. Zhu, Transgcn: Coupling transformation assumptions with graph convolutional networks for link prediction, in: Proceedings of the 10th International Conference on Knowledge Capture, 2019, pp. 131–138.
[58] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, G. E. Dahl, Neural message passing for quantum chemistry, in: International Conference on Machine Learning, PMLR, 2017, pp. 1263–1272.
[59] A. Roy, D. Ghosal, E. Cambria, N. Majumder, R. Mihalcea, S. Poria, Improving zero-shot learning baselines with commonsense knowledge, Cognitive Computation (2022) 1–11.
[60] U. Naseem, I. Razzak, S. K. Khan, M. Prasad, A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models, Transactions on Asian and Low-Resource Language Information Processing 20 (5) (2021) 1–35.
[61] W. L. Taylor, "cloze procedure": A new tool for measuring readability, Journalism quarterly 30 (4) (1953) 415–433.
[62] J. Bian, B. Gao, T.-Y. Liu, Knowledge-powered deep learning for word embedding, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2014, pp. 132–148.
[63] K. A. Nguyen, S. S. im Walde, N. T. Vu, Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2016, pp. 454–459.
[64] Q. Liu, H. Jiang, S. Wei, Z.-H. Ling, Y. Hu, Learning semantic word embeddings based on ordinal knowledge constraints, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2015, pp. 1501–1511.
[65] D. Bollegala, M. Alsuhaibani, T. Maehara, K.-i. Kawarabayashi, Joint word representation learning using a corpus and a semantic lexicon, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 30, 2016.
[66] D. Kiela, F. Hill, S. Clark, Specializing word embeddings for similarity or relatedness, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 2044–2048.

[67] N. Mrkšić, I. Vulić, D. Ó. Séaghdha, I. Leviant, R. Reichart, M. Gašić, A. Korhonen, S. Young, Semantic specialization of distributional word vector spaces using monolingual and cross-lingual constraints, Transactions of the Association for Computational Linguistics 5 (2017) 309–324.

[68] G. Glavaš, I. Vulić, Explicit retrofitting of distributional word vectors, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018, pp. 34–45.

[69] J. Wieting, M. Bansal, K. Gimpel, K. Livescu, From paraphrase database to compositional paraphrase model and back, Transactions of the Association for Computational Linguistics 3 (2015) 345–358.

[70] J. Weston, A. Bordes, O. Yakhnenko, N. Usunier, Connecting language and knowledge bases with embedding models for relation extraction, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013, pp. 1366–1371.

[71] I. Yamada, H. Shindo, H. Takeda, Y. Takefuji, Learning distributed representations of texts and entities from knowledge base, Transactions of the Association for Computational Linguistics 5 (2017) 397–411.

[72] H. Xiao, M. Huang, L. Meng, X. Zhu, Ssp: Semantic space projection for knowledge graph embedding with text descriptions, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 31, 2017.

[73] L. Hu, L. Zhang, C. Shi, L. Nie, W. Guan, C. Yang, Improving distantly-supervised relation extraction with joint label embedding, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 3821–3829.

[74] G. Ji, K. Liu, S. He, J. Zhao, Knowledge graph completion with adaptive sparse transfer matrix, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016, pp. 985–991.

[75] D. Newman-Griffis, A. M. Lai, E. Fosler-Lussier, Jointly embedding entities and text with distant supervision, in: Proceedings of The Third Workshop on Representation Learning for NLP, 2018, pp. 195–206.

[76] D. Zhang, B. Yuan, D. Wang, R. Liu, Joint semantic relevance learning with text data and graph knowledge, in: Proceedings of the 3rd workshop on Continuous Vector Space Models and their Compositionality, 2015, pp. 32–40.

[77] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, E. Hovy, Hierarchical attention networks for document classification, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 1480–1489.

[78] Y. Gong, Q. Zhang, Hashtag recommendation using attention-based convolutional neural network, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, 2016, pp. 2782–2788.

[79] Q. Zhang, J. Wang, H. Huang, X. Huang, Y. Gong, Hashtag recommendation for multimodal microblog using co-attention network, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, 2017, pp. 3420–3426.

[80] D. Zeng, K. Liu, S. Lai, G. Zhou, J. Zhao, Relation classification via convolutional deep neural network, in: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: technical papers, 2014, pp. 2335–2344.

[81] J. Xu, X. Qiu, K. Chen, X. Huang, Knowledge graph representation with jointly structural and textual encoding, in: Proceedings of the Twenty-sixth International Joint Conference on Artificial Intelligence, 2017.

[82] R. Xie, Z. Liu, J. Jia, H. Luan, M. Sun, Representation learning of knowledge graphs with entity descriptions, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 30, 2016.

[83] W. Gao, Y. Fang, F. Zhang, Z. Yang, Representation learning of knowledge graphs using convolutional neural networks, Neural Network World 30 (3) (2020) 145.

[84] D. Bahdanau, K. H. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: 3rd International Conference on Learning Representations, ICLR 2015, 2015.

[85] A. Graves, Generating sequences with recurrent neural networks, arXiv preprint arXiv:1308.0850 (2013).

[86] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, X. Huang, Pre-trained models for natural language processing: A survey, Science China Technological Sciences 63 (10) (2020) 1872–1897.

[87] S. Wang, J. Zhang, C. Zong, Learning sentence representation with guidance of human attention, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, 2017, pp. 4137–4143.

[88] Y. Levine, B. Lenz, O. Dagan, O. Ram, D. Padnos, O. Sharir, S. Shalev-Shwartz, A. Shashua, Y. Shoham, Sensebert: Driving some sense into bert, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 4656–4667.

[89] N. Poerner, U. Waltinger, H. Schütze, E-bert: Efficient-yet-effective entity embeddings for bert, in: Findings of the Association for Computational Linguistics: EMNLP 2020, 2020, pp. 803–818.

[90] I. Yamada, A. Asai, H. Shindo, H. Takeda, Y. Matsumoto, Luke: Deep contextualized entity representations with entity-aware self-attention, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 6442–6454.

[91] Q. Wu, C. Xing, Y. Li, G. Ke, D. He, T.-Y. Liu, Taking notes on the fly helps bert pre-training, in: International Conference on Learning Representations, ICLR 2021, 2021.

[92] T. Sun, Y. Shao, X. Qiu, Q. Guo, Y. Hu, X.-J. Huang, Z. Zhang, Colake: Contextualized language and knowledge embedding, in: Proceedings of the 28th International Conference on Computational Linguistics, 2020, pp. 3660–3670.

[93] Y. Xu, C. Zhu, R. Xu, Y. Liu, M. Zeng, X. Huang, Fusing context into knowledge graph for commonsense question answering, in: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, 2021, pp. 1201–1207.

[94] B. He, D. Zhou, J. Xiao, X. Jiang, Q. Liu, N. J. Yuan, T. Xu, Integrating graph contextualized knowledge into pre-trained language models, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, 2020, pp. 2281–2290.

[95] D. Yu, C. Zhu, Y. Yang, M. Zeng, Jaket: Joint pre-training of knowledge graph and language understanding, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, 2022, pp. 11630–11638.

[96] Y. Su, X. Han, Z. Zhang, Y. Lin, P. Li, Z. Liu, J. Zhou, M. Sun, Cokebert: Contextual knowledge selection and embedding towards enhanced pre-trained language models, AI Open 2 (2021) 127–134.

[97] Y. Lu, H. Lu, G. Fu, Q. Liu, Kelm: Knowledge enhanced pre-trained language representations with message passing on hierarchical relational graphs, in: International Conference on Learning Representations, ICLR 2022, 2022.

[98] M. Ostendorff, P. Bourgonje, M. Berger, J. Moreno-Schneider, G. Rehm, B. Gipp, Enriching bert with knowledge graph embeddings for document classification, arXiv preprint arXiv:1909.08402 (2019).

[99] J. Guan, F. Huang, Z. Zhao, X. Zhu, M. Huang, A knowledge-enhanced pretraining model for commonsense story generation, Transactions of the Association for Computational Linguistics 8 (2020) 93–108.

[100] Y. Sun, S. Wang, Y. Li, S. Feng, X. Chen, H. Zhang, X. Tian, D. Zhu, H. Tian, H. Wu, Ernie: Enhanced representation through knowledge integration, arXiv preprint arXiv:1904.09223 (2019).

[101] P. Banerjee, C. Baral, Self-supervised knowledge triplet learning for zero-shot question answering, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 151–162.

[102] B. Kim, T. Hong, Y. Ko, J. Seo, Multi-task learning for knowledge graph completion with pre-trained language models, in: Proceedings of the 28th International Conference on Computational Linguistics, 2020, pp. 1737–1743.

[103] A. Lauscher, O. Majewska, L. F. Ribeiro, I. Gurevych, N. Rozanov, G. Glavaš, Common sense or world knowledge? investigating adapter-based knowledge injection into pretrained transformers, in: Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, 2020, pp. 43–49.

[104] M. E. Peters, M. Neumann, R. Logan, R. Schwartz, V. Joshi, S. Singh, N. A. Smith, Knowledge enhanced contextual word representations, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 43–54.

[105] X. Wang, T. Gao, Z. Zhu, Z. Zhang, Z. Liu, J. Li, J. Tang, Kepler: A unified model for knowledge embedding and pre-trained language representation, Transactions of the Association for Computational Linguistics 9 (2021) 176–194.

[106] W. Yu, C. Zhu, Y. Fang, D. Yu, S. Wang, Y. Xu, M. Zeng, M. Jiang, Dict-bert: Enhancing language model pre-training with dictionary, in:

Findings of the Association for Computational Linguistics: ACL 2022, 2022, pp. 1907–1918.
[107] L. Yao, C. Mao, Y. Luo, Kg-bert: Bert for knowledge graph completion, arXiv preprint arXiv:1909.03193 (2019).
[108] T. Févry, L. B. Soares, N. Fitzgerald, E. Choi, T. Kwiatkowski, Entities as experts: Sparse memory access with entity supervision, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 4937–4951.
[109] P. Verga, H. Sun, L. B. Soares, W. Cohen, Adaptable and interpretable neural memoryover symbolic knowledge, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 3678–3691.
[110] M. de Jong, Y. Zemlyanskiy, N. FitzGerald, F. Sha, W. Cohen, Mention memory: incorporating textual knowledge into transformers through entity mention attention, arXiv preprint arXiv:2110.06176 (2021).
[111] S. Baccianella, A. Esuli, F. Sebastiani, Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining, in: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), 2010.
[112] P. Veličkovič, G. Cucurull, A. Casanova, A. Romero, P. Lió, Y. Bengio, Graph attention networks, in: International Conference on Learning Representations, 2018.
[113] D. Erhan, A. Courville, Y. Bengio, P. Vincent, Why does unsupervised pre-training help deep learning?, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, 2010, pp. 201–208.
[114] O. Levy, M. Seo, E. Choi, L. Zettlemoyer, Zero-shot relation extraction via reading comprehension, in: Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), 2017, pp. 333–342.
[115] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, A. Miller, Language models as knowledge bases?, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 2463–2473.
[116] A. Lauscher, I. Vulić, E. M. Ponti, A. Korhonen, G. Glavaš, Specializing unsupervised pretraining models for word-level semantic similarity, in: Proceedings of the 28th International Conference on Computational Linguistics, 2020, pp. 1371–1383.
[117] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al., Overcoming catastrophic forgetting in neural networks, Proceedings of the National Academy of Sciences 114 (13) (2017) 3521–3526.
[118] R. M. French, Catastrophic forgetting in connectionist networks, Trends in Cognitive Sciences 3 (4) (1999) 128–135.
[119] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, S. Gelly, Parameter-efficient transfer learning for nlp, in: International Conference on Machine Learning, PMLR, 2019, pp. 2790–2799.
[120] J. Weston, S. Chopra, A. Bordes, Memory networks, in: 3rd International Conference on Learning Representations, ICLR 2015, 2015.
[121] H. Dong, W. Wang, K. Huang, F. Coenen, Automated social text annotation with joint multilabel attention networks, IEEE Transactions on Neural Networks and Learning Systems 32 (5) (2020) 2224–2238.
[122] N. V. Nayak, S. H. Bach, Zero-shot learning with common sense knowledge graphs, Transactions on Machine Learning Research (TMLR) (2022).
[123] Y. Wang, W. Wang, Q. Chen, K. Huang, A. Nguyen, S. De, Generalised zero-shot learning for entailment-based text classification with external knowledge, in: 2022 IEEE International Conference on Smart Computing (SMARTCOMP), IEEE, 2022, pp. 19–25.
[124] M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych, B. Schiele, What helps where–and why? semantic relatedness for knowledge transfer, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 910–917.
[125] W. Yin, J. Hay, D. Roth, Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 3914–3923.
[126] D. Jin, Z. Jin, J. T. Zhou, P. Szolovits, Is bert really robust? a strong baseline for natural language attack on text classification and entailment, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 34, 2020, pp. 8018–8025.
[127] X. Wang, Y. Yang, Y. Deng, K. He, Adversarial training with fast gradient projection method against synonym substitution based text attacks, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 13997–14005.
[128] X. Wang, L. Chen, T. Ban, M. Usman, Y. Guan, S. Liu, T. Wu, H. Chen, Knowledge graph quality control: a survey, Fundamental Research (2021).
[129] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in Neural Information Processing Systems 33 (2020) 1877–1901.
[130] T. Schick, H. Schütze, Exploiting cloze-questions for few-shot text classification and natural language inference, in: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, 2021, pp. 255–269.
[131] S. Hu, N. Ding, H. Wang, Z. Liu, J. Wang, J. Li, W. Wu, M. Sun, Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2022, pp. 2225–2240.
[132] A. Ng, Nuts and bolts of building ai applications using deep learning, NIPS Keynote Talk (2016).
[133] H. Daumé III, Frustratingly easy domain adaptation, in: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, 2007, pp. 256–263.
[134] R. Chattopadhyay, Q. Sun, W. Fan, I. Davidson, S. Panchanathan, J. Ye, Multisource domain adaptation and its application to early detection of fatigue, ACM Transactions on Knowledge Discovery from Data (TKDD) 6 (4) (2012) 1–26.
[135] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, T. Tuytelaars, A continual learning survey: Defying forgetting in classification tasks, IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (7) (2021) 3366–3385.
[136] N. Díaz-Rodríguez, A. Lamas, J. Sanchez, G. Franchi, I. Donadello, S. Tabik, D. Filliat, P. Cruz, R. Montes, F. Herrera, Explainable neural-symbolic learning (x-nesyl) methodology to fuse deep learning representations with expert knowledge graphs: The monumai cultural heritage use case, Information Fusion 79 (2022) 58–83. doi:https://doi.org/10.1016/j.inffus.2021.09.022.
[137] L. De Raedt, A. Kimmig, Probabilistic (logic) programming concepts, Machine Learning 100 (1) (2015) 5–47.
[138] W. Wang, S. J. Pan, Integrating deep learning with logic fusion for information extraction, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 9225–9232.