Knowledge-aware Zero-Shot Learning: Survey and Perspective

Jiaoyan Chen\textsuperscript{1*}, Yuxia Geng\textsuperscript{2}, Zhuo Chen\textsuperscript{2}, Ian Horrocks\textsuperscript{1}, Jeff Z. Pan\textsuperscript{3} and Huajun Chen\textsuperscript{2*}

\textsuperscript{1}Department of Computer Science, University of Oxford
\textsuperscript{2}College of Computer Science & AZFT Knowledge Engine Lab, Zhejiang University
\textsuperscript{3}School of Informatics, The University of Edinburgh

Abstract

Zero-shot learning (ZSL) which aims at predicting classes that have never appeared during the training using external knowledge (a.k.a. side information) has been widely investigated. In this paper we present a literature review towards ZSL in the perspective of external knowledge, where we categorize the external knowledge, review their methods and compare different external knowledge. With the literature review, we further discuss and outlook the role of symbolic knowledge in addressing ZSL and other machine learning sample shortage issues.

1 Introduction

Normal supervised machine learning (ML) classification trains a model with labeled samples and predicts the classes of subsequent samples using classes that were encountered during the training stage. Zero-shot learning (ZSL), however, aims to also predict novel classes that did not occur in the training samples. Such novel classes are known as unseen classes, while the classes occurring in training samples are known as seen classes. ZSL has been widely investigated as a means of addressing common ML issues such as emerging classes, sample shortage, etc.

ZSL was originally proposed for image classification in Computer Vision (CV) [Palatucci et al., 2009; Lampert et al., 2009]. One typical case study is recognizing animals that have no training images by exploiting their semantic relationships to animals that have training images through external knowledge (a.k.a. side information) such as text descriptions, visual annotations and the taxonomy. It has since been applied to Natural Language Processing (NLP) tasks such as text classification and relation extraction (a.k.a. relation classification), as well as to ML tasks in other domains such as Knowledge Graph (KG) [Pan et al., 2016] construction and completion. Until now hundreds of papers have been published concerning ZSL theories, methods and applications.

Although several review papers have been published, they mostly focus on categorizing the ZSL settings and the algorithm design patterns (e.g., into classifier-based and instance-based) [Fu et al., 2018; Wang et al., 2019; Xian et al., 2018], and few of them systematically analyze the external knowledge which play a key role in designing ZSL methods and in improving their performance since no samples are given for the unseen classes. In contrast, this paper reviews ZSL studies mainly from the perspective of external knowledge. We categorize the external knowledge into four kinds — text, attribute, KG and ontology & rules, according to their data structures, sources, expressivity, etc. For each kind we review its methods, case studies and benchmarks. We also compare different external knowledge, and discuss the role of symbolic knowledge representation in addressing ZSL and other sample shortage settings. Although [Fu et al., 2018] and [Wang et al., 2019] also introduce the semantic space (i.e., encoding of the external knowledge), no paper analysis is conducted for each external knowledge, and more importantly the external knowledge involved (mainly text and attributes) are quite incomplete — KG and ontology which started to be widely investigated in recent three years are not covered. [Xian et al., 2018] contributes a comprehensive evaluation to multiple ZSL methods, but these methods are limited to image classification, while we consider different tasks in multiple domains including CV, NLP, KG construction and completion, etc.

2 Overview

2.1 Problem Definition and Annotations

In ML classification, a classifier is trained to approximate a target function $f : x \rightarrow y$, where $x$ represents the input data and $y$ represents the output class. Standard ZSL aims to classify data with the candidate classes that have never appeared in the training data. We denote (i) the training samples as $D_t = \{(x, y) | x \in \mathcal{X}_s, y \in \mathcal{Y}_s\}$ where $\mathcal{X}_s$ and $\mathcal{Y}_s$ represent the training sample inputs and the seen classes, respectively; (ii) the testing (prediction) samples as $D_te = \{(x, y) | x \in \mathcal{X}_u, y \in \mathcal{Y}_u\}$ where $\mathcal{X}_u$ and $\mathcal{Y}_u$ represent the testing sample inputs and unseen classes, respectively, with $\mathcal{Y}_u \cap \mathcal{Y}_s = \emptyset$; and (iii) the external knowledge (side information) as a class semantic encoding function $h : y \rightarrow z$ where $z$ represents the semantic vector of the class $y$. Note the external knowledge are originally represented as symbolic data such as class names, class textual descriptions and inter-class relationships. To be involved in ZSL, they are transformed into sub-symbolic representations (i.e., vectors). The ZSL problem is to predict the classes of $\mathcal{X}_u$ as correctly as possible.
sible. Specially, when the candidate classes in $D_{tr}$ are set to $\mathcal{Y}_s \cup \mathcal{Y}_u$, the problem is known as generalized ZSL.

Note in addressing some tasks such as KG link prediction, the original modeling function $f$ is often transformed into a scoring function by moving $y$ to the input, denoted as $f' : (x, y) \rightarrow s$ where $s$ is a score indicating the truth degree of $y$. With $f'$ the class of a testing sample $x$ in $\mathcal{X}_s$ is predicted as the class in $\mathcal{Y}_s$ (or $\mathcal{Y}_s \cap \mathcal{Y}_u$) that maximizes $s$. Considering a link prediction task with two given entities as the input $x$ and one relation as the class $y$ to predict, the score $s$ quantifies whether the relation matches the two entities’ relationship.

2.2 Technical Solutions

With the external knowledge and the class semantic encoding function $h$ (cf. details in Section 3), a few machin learning methods can be applied to address the above ZSL problem. We divide the majority of them into three kinds.

Mapping Function Based

Given the training samples $D_{tr}$, the mapping function based methods learn a mapping function from the input space to the class semantic encoding space, denoted as $m : x \rightarrow h(y)$. In prediction, they adopt the nearest neighbour searching in the class semantic encoding space. Namely, for a testing sample $x'$ in $D_{te}$, its predicted class is calculated as $\arg\min_{y \in \mathcal{Y}_s} d(m(x'), h(y))$, where $d$ is a metric that calculates the distance between two vectors, such as the Euclidean distance. Some methods such as [Frome et al., 2013] learns a linear mapping function, while some other methods prefer to non-linear mapping functions (e.g., [Socher et al., 2013] uses a two-layer neural network).

It is worth noting that some methods such as [Shigeto et al., 2015] and [Zhang et al., 2017] learn an inverse mapping function $m^{-1}$ from the class semantic vector to the input. The class of $x'$ is then searched in the input space, i.e., calculating $\arg\min_{y \in \mathcal{Y}_s} d(m^{-1}(h(y)))$. This is believed to be able to release the hubness problem encountered in nearest neighbor search. Besides, there are some methods that project the class semantic vector and the input to an intermediate space [Yang and Hospedales, 2015; Lei Ba et al., 2015].

Generative Model Based

These methods leverage a generative model (e.g., Generative Adversarial Network (GAN) [Goodfellow et al., 2014]) to synthesize training data for unseen classes conditioned on their semantic vectors. Considering those methods using GAN, given the training data $D_{tr}$, they learn a conditional generator $G : (h(y), v) \rightarrow \hat{x}$ which takes random Gaussian noise $v$ and the class semantic vector $h(y)$ as its input and outputs the synthetic sample $\hat{x}$ of class $y$, and at the same time, a discriminator $D : (x, \hat{x}) \rightarrow [0, 1]$ is trained to distinguish the generated sample from the real sample. Once the generator $G$ is trained to be able to synthesize plausible samples for seen classes (i.e., the synthesized and real samples cannot be distinguished by $D$), it is used to generate samples $\hat{x}_n$ for each unseen class $y_u$ in $\mathcal{Y}_u$ via its semantic vector $h(y_u)$. This transforms the (generalized) ZSL problem into a normal supervised learning problem.

To improve the quality of generated samples, some works encourage the generator to generate samples that statistically match the distribution of real samples; for example, [Qin et al., 2020] proposes a pivot regularization to enforce the mean of features of the generated samples to be that of the real samples. Some other works encourage the generated samples to preserve the inter-class relationship indicated by the class semantic vectors; for example, [Felix et al., 2018] develops a cycle consistency loss to enforce the generated samples to reconstruct their classes’ semantic vectors. Besides, more GAN variants such as StackGAN [Pandey et al., 2020], other generative models such as Variational Autoencoder (VAE) [Kingma and Welling, 2013], and the combination of different generative models such as VAE and GAN [Xian et al., 2019] have been explored.

Graph Neural Network Based

These methods are mainly developed for graph-structured external knowledge where each class is aligned with one graph node and each inter-class relationship is represented by one graph edge. Given the external knowledge graph, its nodes’ states (semantic vectors) are initialized by e.g., word embeddings and multi-hot encodings, and a Graph Neural Network (GNN) is then applied to learn the nodes’ semantic vectors with their relationships to the neighbours encoded. Briefly, in every layer of the GNN, a propagation function is learned to update the state of each node by aggregating the features from its adjacent nodes in the graph, denoted as $p : h^l_u \rightarrow h^{l+1}_u$, where $h^l_u$ denotes the hidden state of node $u$ at the $l^{th}$ layer. In the training, the classifiers of seen classes learned from $D_{tr}$ are used to train the GNN, while in prediction, the GNN is used to calculate the classifiers of unseen classes.

Some methods implement the information propagation via convolutional operations [Kipf and Welling, 2017]; for example, [Wang et al., 2018] uses a Graph Convolutional Neural Network for the external knowledge from WordNet [Miller, 1995], while [Kampffmeyer et al., 2019] uses fewer convolutional layers but one additional dense connection layer to propagate features towards distant nodes for the same graph. To enhance the information propagation, [Geng et al., 2020] propose weighted aggregation to emphasize those more important adjacent nodes; [Lee et al., 2018] designs multiple relation-specific functions for the Graph Gated Neural Network to learn the propagation weights which control the information exchange between nodes.

In the above methods, GNNs are directly used to predict the classifiers of unseen classes. Actually GNNs can also be used to embed the graph semantics (inter-class relationships) and calculate the semantic vectors $h(y)$ [Geng et al., 2021; Roy et al., 2020]. With these class semantic vectors, different ZSL methods including the generative model based and the mapping function based can be further applied. It is worth noting that for each class, multiple semantic vectors which may be calculated from different external knowledge resources, can be considered by simple concatenation or weighted combination [Roy et al., 2020].

3 External Knowledge

Different kinds of external knowledge have been explored to build the relationship between the seen and unseen classes.
In this survey we divide them into four categories: text, attribute, KG, rule & ontology, according to their characteristics, expressivity and the semantic encoding approaches. See Table 1 for a brief summary. In the remainder of this section, we will review the ZSL work for each category. Note some work use more than one kind external knowledge. This is as expected because different external knowledge often have different semantics and thus are complementary to each other.

3.1 Text

Text external knowledge refer to unstructured textual information of the classes, such as their names, definitions and descriptions. They vary from words and phrases to long text such as sentences and documents. Here are some typical examples in different ZSL tasks. In image classification, [Norouzi et al., 2014], [Socher et al., 2013] and [Frome et al., 2013] utilize the class names and their word embeddings to address the unseen classes; [Elhoseiny et al., 2013] and [Qiao et al., 2016] prefer to existing class sentence descriptions from encyclopedia entries (articles); [Reed et al., 2016] collects more fine-grained and compact visual description sentences via crowdsourcing by the Amazon Mechanical Turk (AMT) platform. In KG link prediction, [Qin et al., 2020] utilizes relation sentence descriptions from the KG itself for addressing the unseen relations. In entity extraction (a.k.a entity linking), [Logeswaran et al., 2019] deals with the unseen entities in a new specialized domain by using the entities’ encyclopedia documents.

To encode the semantics of a class name, one approach is directly using its words’ vectors by a language model or word embedding model that has been trained by a general purpose or domain specific corpus. However, this makes the two tasks – class semantic encoding and prediction model training detached with no interaction between them. A coupled approach is jointly learning the prediction model and the class semantic encoding. Note both can be pre-trained independently for high efficiency. One representative method is DeViSE where a skip-gram word embedding model and an image classifier are fine-tuned jointly [Frome et al., 2013].

Different from class names, textual class descriptions such as sentences and documents contain more yet noisy information. To suppress the noise, some additional feature learning and selection over the text (or text embedding) has been considered. Among the aforementioned works, [Elhoseiny et al., 2013] and [Qin et al., 2020] extract features from the text by the TF-IDF algorithm through which the vectors of some critical words get more weights; [Qiao et al., 2016] initially encode the class descriptions into simple bag-of-words vectors, and then jointly learns text features and the image classifier; [Reed et al., 2016] also jointly learns text features and the image classifier, but considers both word-level and character-level text features.

In summary, the text external knowledge are easy to access for common ZSL tasks. They can be extracted from not only the data of the ZSL tasks themselves but also encyclopedias, Web pages and other online resources. However, they are often noisy with irrelevant words and the words are always ambiguous. They have limited expressivity on the semantics and cannot accurately express those fine-grained, logical or quantified inter-class relationships.

3.2 Attribute

Attribute external knowledge refer to those class properties with categorical, boolean or real values, which are often organized as key-value pairs. They were originally explored in zero-shot object recognition, where the attributes are used to annotate visual characteristics such as object colours and shapes. The simplest attributes are those binary annotations; for example, “furry” and “striped” indicate whether an animal looks furry and striped, respectively, in animal recognition [Lampert et al., 2009; Lampert et al., 2013]; while the annotation “has wheel” is used in recognizing vehicles [Farhadi et al., 2009]. Relative attributes which enable comparing the degree of each attribute between classes (e.g., “bears are furrier than giraffes”) [Parikh and Grauman, 2011] are more expressive. Another kinds of more expressive visual attributes are those associated with real values for quantifying the degree. One typical example is the animal image classification benchmark named Animals with Attributes (AWA) [Lampert et al., 2013; Xian et al., 2018]. Besides the visual tasks, the attribute external knowledge can also be applied in zero-shot graph link prediction. [Hao et al., 2020] utilizes the node attributes with categorical values to address the prediction involving unseen nodes which have no connection to the graph.

All the binary attributes of each class can be encoded into a multi-hot vector by associating one slot to one attribute and setting it to 1 if the attribute is positive and to 0 otherwise. This can be extended to attributes with numeric values by filling their slots with their values. For attributes with categorical values, each categorical value can be transformed into an integer. In utilizing the attributes in image classification, the attribute vector can be directly used as the class semantic vector (i.e., $h(y)$), and then the mapping function based on the generative model based methods can be applied. This kind of methods are also known as indirect attribute prediction according to [Lampert et al., 2013]. In contrast, the other kind of methods are called direct attribute prediction where the attributes of each testing sample are directly predicted. The predicted attributes of a testing sample are then used to determine the sample’s class as the candidate that has the most similar attributes (e.g., [Parikh and Grauman, 2011]).

In comparison with the text, the attribute external knowledge have higher expressivity with less noise and ambiguity. They can even indirectly represent some quantified relationships between classes. However, high quality attributes such as image annotations are often not available for a new task, while human annotations, which are often voted by multiple volunteers or even exerts, are very costly. Therefore, in ZSL tasks beyond CV, they have been rarely explored. In KG link prediction, entities and relations do not always have attributes while the existing attributes are often sparse.

3.3 Knowledge Graph

Another form of external knowledge for ZSL tasks is graph knowledge represented as facts in RDF triple1, where seen and unseen classes are usually represented by KG entities.

1https://www.w3.org/TR/rdf11-concepts/
Considering animal image classification, a KG can express kinds of semantics for the inter-animal relationship, such as the animal taxonomy (e.g., *cheetah* and *jaguar* are species of *big cats*) and the animal habitats (e.g., *jaguar* lives in *forest* or *swamp*; *cheetah* lives in *open area*). The exploration of KGs now mostly lies in ZSL tasks in CV, and most studies prefer to existing KGs that are open online. [Wang et al., 2018], [Lee et al., 2018] and [Kampffmeyer et al., 2019] adopt WordNet which includes semantic relationships such as synonyms and hyponyms. [Wang et al., 2018] also studies NELL (a general KG extracted from the Web [Carlson et al., 2010]) for ZSL image classification. [Roy et al., 2020] adopts the common sense KG named ConceptNet [Speer et al., 2017]. [Geng et al., 2020] constructs task-specific KGs using knowledge from both WordNet and DBpedia (a KG extracted from Wikipedia [Auer et al., 2007]) while [Gao et al., 2019] further considers ConceptNet besides WordNet and DBpedia.

As some KGs, such as WordNet and ConceptNet, are often very large with much irrelevant knowledge, a task-specific KG is often constructed by knowledge extraction and integration. To extract relevant knowledge, the task-specific data such as the class names are matched with KG entities either using some existing associations (e.g., ImageNet classes are matched with WordNet entities [Deng et al., 2009]) or by some mapping methods such as string matching. With the constructed KG, the class semantic vectors can then be learned by a KG embedding method which can be either GNN variants such as GCN by [Wang et al., 2018], Relational GCN by [Roy et al., 2020] and Attentive GCN by [Geng et al., 2020], or translation based or factorization based KG embedding models such as TransE and DistMult. See [Wang et al., 2020] for a survey on KG embedding.

KGs are more expressive than both text and attribute. Besides the relational graph, a KG can represent and incorporate the text and attribute external knowledge as well. Text and attributes with real values can be represented by literals with data properties (such as *obo:abstract*, *rdfs:label* and *dbo:populationMetro* in DBpedia), while attributes with binary or categorical values can be either represented as literals with (new) data properties or transformed into relational facts by creating new relations and new entities for the values [Geng et al., 2020]. A KG and its literals can be jointly embedded by some literal-aware KG embedding methods such as DKRL which supports entity descriptions [Xie et al., 2016]. See [Gesese et al., 2020] for a survey.

On the other hand, KGs with suitable knowledge are not always available for a new real word ZSL task. Although many public KGs such as ConceptNet and WordNet are very large, their contents are still incomplete and biased with a limited coverage for some specific domains. Thus extracting, matching and curating knowledge from external resources and the task itself becomes a key challenge for KG-based ZSL.

### 3.4 Rule & Ontology

Rule and ontology can express additional logical relationships between seen and unseen classes. [Li et al., 2020] addresses the zero-shot relation extraction problem by using Horn rules to describe an unseen relation with some seen relations. For example, the unseen relation *nominated_for* is defined as the composition of two seen relations, namely, *nominated_for*(x, z) ⇐ *award_received*(x, y) ∧ *winner*(y, z). Briefly this work matches all the candidate relations with Wikidata relations, construct a KG from Wikidata, learns the KG embeddings, and finally computes each unseen relation’s semantic vector with the KG embeddings of its own and its compositional relations. [Rocktäschel et al., 2015] injects background knowledge on the relations in form of first-order formulae (e.g., ∀x, y : *daughter_of*(x, y) ⇒ *has_parents*(x, y)) into relation embeddings to augment relation extraction and address unseen relations. Three methods are considered to inject the rules: symbolic reasoning for ground atoms before or after learning the embeddings, jointly learning the embeddings with a loss term for the formulae.

An ontology is to represent and exchange knowledge such as names and definitions of the concepts and relations, their relationships, annotations, properties, etc. The most common inter-concept or inter-relation relationship is the subsumption through which an ontology can represent the taxonomy. On-

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
<th>Embedding</th>
<th>Semantic Richness</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>Unstructured text that describe the classes, such as class names, phrases, sentences and documents</td>
<td>Word embedding, text feature learning</td>
<td>Weak</td>
<td>Very easy to access; words are often ambiguous; long text is usually noisy</td>
</tr>
<tr>
<td>Attribute</td>
<td>Semi-structured class properties with categorical, boolean or real values, such as annotations that describe object visual characteristics</td>
<td>Vectors with binary or numeric values</td>
<td>Medium</td>
<td>Attributes by manual annotation are accurate but very costly</td>
</tr>
<tr>
<td>KG</td>
<td>Multi-relation graph composed of entities aligned with the classes, other entities and their relationships such as the subsumption and the relational facts</td>
<td>Decomposition-based, translation-based, GNNs</td>
<td>High</td>
<td>KGs can also encompass the text and attribute external knowledge; some open KGs can be used</td>
</tr>
<tr>
<td>Ontology &amp; Rule</td>
<td>Logical relationships between the classes (and other concepts), such as the subsumption, the quantification constraints and the composition</td>
<td>Ontology embedding, materialization embedding</td>
<td>Very high</td>
<td>Ontologies include KGs (as the fact parts) and can encompass the text and attributes; construction of logics relies on domain knowledge</td>
</tr>
</tbody>
</table>

Table 1: A Brief Summary of The External Knowledge
ontologies can also define constraints and meta data (schema) of the concepts and relations such as the concept existen-
tial quantifier and the relation domain and range. Besides, ontologies by Web Ontology Language (OWL)\(^3\) are able to repre-
sent quite a few logical relationships. Considering the above rule about *nominated for,* an OWL ontology can represent it by defining a complex relation with atom relations, i.e., \(\text{nominated for} \equiv \text{award received} \circ \text{winner}\) (a.k.a. role composition). [Geng et al., 2021] studies zero-shot animal image classification and KG link prediction for unseen relations with two ontologies of RDF Schema (RDFS), respectively. The former ontology encompasses an animal taxonomy, annotations and text descriptions; while the later ontology encompasses the relations, concepts, their hierarchies (e.g., \(\text{radiostation in city} \subseteq \text{office in city}\)), the relation domain and range (e.g., \(\text{radiostation in city}\) should be followed by instances that belong to \(\text{city}\)). The semantic vectors of the animal classes (which are aligned to ontology concepts) and the relations are then learned by literal aware KG embedding (RDFS ontologies can be directly transformed into RDF KGs), and a generative model based ZSL method is eventually applied. [Chen et al., 2020b] explores OWL ontology for augmenting ZSL with animal image classification. The ontology includes an animal taxonomy and complex concepts defined by the composition of atom concepts and existen-
tial quantifiers (e.g., \(\text{Killer\_Whale} \equiv \text{Toothed\_Whale} \land \exists \text{has\_Texture\_Patches}\)). Following the EL fragment of OWL 2 which can be embedded by geometric methods such as [Kulmanov et al., 2019] or word embedding based meth-
ods such as OWL2Vec* [Chen et al., 2020a].

4 Application and Benchmark

This section introduces popular ZSL applications. Table 2 briefly summarizes some open benchmarks with novel ex-
ternal knowledge. Those normal supervised learning bench-
marks that are used to evaluate ZSL by new splits are ignored.

ZSL has been widely investigated for image classification for both general purpose tasks such as those using ImageNet and domain-specific tasks such as mammal classification (cf. the AWA1 and AWA2 benchmarks [Lampert et al., 2013; Xian et al., 2018]) and fine-grained bird classification (cf. the CUB benchmark [Welinder et al., 1]). Almost all these tasks allow to use class names while more textual information sometimes is accessed online. Manual visual annotations (attributes) are widely studied, but are mostly limited to domain-
specific tasks with a small to medium number of classes. KGs and ontologies are often studied for tasks whose classes have been aligned to entities of some KG; for example, ImageNet classes are aligned to WordNet entities. Another CV task is visual question answering (VQA) which aims to predict the answer of a natural language question according to the con-
tent of an image. ZSL in VQA has several different definiti-
ons. [Chen et al., 2020b] defines ZSL as a setting where some testing answers have never appeared in training, while [Ramakrishnan et al., 2017] defines ZSL as a setting where the image or the question contains some novel object (e.g., ‘Is the dog black?’ where ‘dog’ has never appeared in train-
ing). [Teny and Hengel, 2016] has a more tolerant definition which regards a question as unseen if there is at least one novel word in this question or in its answer.

ZSL has also been studied in NLP for e.g., text classification and open information extraction. The most common ex-
ternal knowledge for text classification are class names (with word embeddings pre-trained by an external corpus) and text descriptions [Srivastava et al., 2018]. The class hierarchy and KG are found to be investigated by one study, i.e., [Zhang et al., 2019]. Most current evaluation re-uses the existing benchmarks for e.g., topic categorization and emotion detec-
tion with new seen and unseen splits [Yin et al., 2019]. In information extraction, ZSL aims at addressing unseen rela-
tions and entities. One popular direction is exploring kinds of textual information such as entity descriptions [Logeswaran et al., 2019], relation web pages [Lockard et al., 2020] and question template-based relation explanations [Levy et al., 2017]. KGs and rules have been explored (e.g., [Li et al., 2020]) but not widely, with no open benchmarks. Different from CV tasks, ZSL text classification and open information extraction have rarely considered attributes.

Recently ZSL has been applied to address emerging enti-
ties and relations in KG embedding and link prediction. Note we exclude those new entities or relations that have a few links to the existing KG as they lead to some training sam-
ple. There are only a small number of relevant papers, but the external knowledge of text descriptions and ontological schemas have been explored (cf. [Qin et al., 2020], [Xie et al., 2016] and [Geng et al., 2021], respectively). Besides, ZSL has also been applied in other prediction tasks involving KGs such as question answering [Banerjee and Baral, 2020].

5 Conclusion and Discussion

This paper presents a literature review for ZSL mainly from the perspective of external knowledge. We briefly reviewed the ZSL definitions, technical solutions, applications and open benchmarks. We divided the external knowledge into text, attribute, KG, and ontology & rule. For each category, we introduced the characteristics, presented how they are uti-
ized and summarized the advantages and disadvantages. We analyzed those typical papers for the text and attribute ex-
ternal knowledge due to the space limitation, while giving a rather complete review to those ZSL works using KGs, rules and ontologies as external knowledge.

Trend Analysis. After ZSL was proposed in around 2009, it was widely investigated for image classification at the begin-
ing with attributes (e.g., [Lampert et al., 2009; Farhadi et al., 2009; Parikh and Grauman, 2011]) and then with kinds of text and text-based latent embeddings (e.g., DeViSE [Frome et al., 2013] and SAE [Kodirov et al., 2017]). KG for ZSL can be traced back to [Palatucci et al., 2009] which uses two small KGs containing word attributes and word co-
ocurrence relationships respectively. However, KG and on-
tology for ZSL were then not studied until in recent four years when general purpose KGs, such as NELL and Wikidata, became popular (e.g., [Lee et al., 2018; Wang et al., 2018; Geng et al., 2020]). Due to higher expressivity, compatibility

---

\(^3\)https://www.w3.org/TR/owl-features/
Table 2: A Summary of Some Open ZSL Benchmarks with Novel External Knowledge

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Names</th>
<th>Information (Splitting)</th>
<th>External Knowledge</th>
<th>Sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Classification</td>
<td>ImageNet</td>
<td>1K classes of ImageNet 2012 as the seen; classes that are 2/3-hops away, or all the other ImageNet classes (21K) as the unseen</td>
<td>The WordNet KG</td>
<td>Splits proposed by [Qin et al., 2020] and [Geng et al., 2021]</td>
</tr>
<tr>
<td></td>
<td>ImNet-A</td>
<td>28 seen classes (37,800 images) and 52 unseen classes (39,523 images)</td>
<td>Ontologies with animal taxonomies, names, attributes and textual descriptions</td>
<td>Images from ImageNet; splits and external knowledge by [Xia et al., 2018]</td>
</tr>
<tr>
<td></td>
<td>ImNet-O</td>
<td>10 seen classes (13,407 images) and 25 unseen classes (25,954 images)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Visual Question Answering</td>
<td>VQA dataset</td>
<td>2951/811 seen/unseen objects; a testing question has at least one unseen object; 20,472 images, 614,164 questions, 10 answers per question;</td>
<td>Object description from Wikipedia and books on the Web</td>
<td>The version with splits and object description by [Ramakrishnan et al., 2017]</td>
</tr>
<tr>
<td>Link Prediction</td>
<td>NELL-ZS</td>
<td>KG from NELL with 65,567 entities and 188,392 triples; 149/32 seen/unseen relations</td>
<td>Textual description, names and ontologies (e.g., relation hierarchy relation domain/range)</td>
<td>Original version with text by [Qin et al., 2020]; new version with ontologies by [Geng et al., 2021]</td>
</tr>
<tr>
<td></td>
<td>Wikidata-ZS</td>
<td>KG from Wikidata with 605,812 entities and 724,967 triples; 489/48 seen/unseen relations</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>FB20k</td>
<td>KG from Freebase with 1,341 relations and 14,904/5,019 seen/unseen entities</td>
<td>Textual descriptions</td>
<td>[Xie et al., 2016]</td>
</tr>
<tr>
<td></td>
<td>DBpedia50k</td>
<td>KG from DBpedia with 654 relations and 46,264/3,636 seen/unseen entities</td>
<td>Textual descriptions</td>
<td>[Shi and Weninger, 2018]</td>
</tr>
<tr>
<td>Text Classification</td>
<td>DBpedia-Wikipedia</td>
<td>14 non-overlapping DBpedia classes with 45,000 docs (samples) from Wikipedia; 11/3 (or 7/7) seen/unseen classes</td>
<td>Class names, textual descriptions, class hierarchy, the ConceptNet KG</td>
<td>The version with external knowledge and splits proposed by [Zhang et al., 2019]</td>
</tr>
<tr>
<td></td>
<td>20news-group</td>
<td>20 topics each of which has around 1,000 docs; 15/5 (or 10/10) seen/unseen classes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relation Extraction</td>
<td>Reading Comprehension</td>
<td>840,000 samples with 10 folds of train/dev/test; 84/12 train/dev relations as the seen; 12 test relations as the unseen</td>
<td>Relation names, question templates (textual explanations)</td>
<td>[Levy et al., 2017]</td>
</tr>
<tr>
<td></td>
<td>Expanded SWDE</td>
<td>21 English websites; each website has 400 to 2,000 pages; 18/14/13 relations for the Movie/NBA/University vertical; two verticals the seen, one vertical as the unseen</td>
<td>Relation names, web page text fields</td>
<td>[Lockard et al., 2020]</td>
</tr>
</tbody>
</table>

Table 2: A Summary of Some Open ZSL Benchmarks with Novel External Knowledge

to text and attribute, more and more resources and embedding techniques, we believe KGs and ontologies will continue to play an increasingly important role in ZSL. Meanwhile, ZSL has been extended from CV to other domains such as NLP, KG construction and completion in recent five years. This trend will continue as the real world contexts often violate the normal supervised learning setting with emerging classes.

**Few-shot & Transfer Learning.** ZSL is highly relevant to another two widely investigated directions: *few-shot learning* (FSL) where one or only a few labeled samples are available for some classes, and *transfer learning* (TL) which transfers data or model from one domain to another different yet related domain. Since some labeled samples are given, methods for FSL focus more on generalizing from these samples by e.g., meta learning and TL algorithms, and fewer FSL works study the usage of external knowledge [Wang et al., 2020]. However, we argue that combining the available samples and the external knowledge would be a promising direction in many tasks (e.g., [Rios and Kavuluru, 2018] utilizes the class label structure for few-shot text classification). On the other hand, TL studies have led to a few effective algorithms for sharing samples and models, most of which can be applied in FSL, ZSL and other sample shortage settings. External knowledge such as those on the domains and features can be used to select transferable models and samples, avoid negative transfer and explain the transfer (e.g., [Chen et al., 2018] uses the ontology knowledge on airlines, airports and so on to explain the transferability of neural network features for flight delay forecasting). These ZSL, FSL and TL studies which explore symbolic knowledge represented by KG, ontology and logical rule form a new and interesting neural-symbolic paradigm which injects background and human knowledge in learning for addressing ML sample shortage challenges.
References


[Lockard et al., 2020] Colin Lockard, Prashant Shiralkar, Xin Luna Dong, and Hannaneh Hajishirzi. Zeroshotceres:


