Benchmarking knowledge-driven zero-shot learning

Citation for published version:
Geng, Y, Chen, J, Zhuang, X, Chen, Z, Pan, JZ, Li, J, Yuan, Z & Chen, H 2023, 'Benchmarking knowledge-driven zero-shot learning', *Journal of Web Semantics*, vol. 75, 100757.
https://doi.org/10.1016/j.websem.2022.100757

Digital Object Identifier (DOI):
10.1016/j.websem.2022.100757

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Journal of Web Semantics

Benchmarking Knowledge-driven Zero-shot Learning

Yuxia Geng[a,b,*], Jiaoyan Chen[c], Xiang Zhuang[a,b], Zhuo Chen[a,b], Jeff Z. Pan[d], Juan Li[a,b], Zonggang Yuan[e] and Huajun Chen[a,b]

[a] College of Computer Science & HIC, Zhejiang University, 38 Zheda Rd, Hangzhou, 310012, Zhejiang, China
[b] Knowledge Engine Group, AZFT Joint Lab, 1818-2 Wenyi West Rd, Hangzhou, 311000, Zhejiang, China
[c] Department of Computer Science, University of Oxford, 15 Parks Rd, Oxford, OX1 3QD, UK
[d] School of Informatics, The University of Edinburgh, 10 Crichton St, Edinburgh, EH8 9AB, UK
[e] NAIE Product Department, Huawei Technologies Co., Ltd., 101 Software Avenue, Nanjing, 210012, China

ARTICLE INFO

Keywords:
Zero-shot Learning
Knowledge Graph
Image Classification
Relation Extraction
Knowledge Graph Completion
Ontology
Semantic Embedding

ABSTRACT

External knowledge (a.k.a. side information) plays a critical role in zero-shot learning (ZSL) which aims to predict with unseen classes that have never appeared in training data. Several kinds of external knowledge, such as text and attribute, have been widely investigated, but they alone are limited with incomplete semantics. Some very recent studies thus propose to use Knowledge Graph (KG) due to its high expressivity and compatibility for representing kinds of knowledge. However, the ZSL community is still in short of standard benchmarks for studying and comparing different external knowledge settings and different KG-based ZSL methods. In this paper, we proposed six resources covering three tasks, i.e., zero-shot image classification (ZS-IMGC), zero-shot relation extraction (ZS-RE), and zero-shot KG completion (ZS-KGC). Each resource has a normal ZSL benchmark and a KG containing semantics ranging from text to attribute, from relational knowledge to logical expressions. We have clearly presented these resources including their construction, statistics, data formats and usage cases w.r.t. different ZSL methods. More importantly, we have conducted a comprehensive benchmarking study, with two general and state-of-the-art methods, two setting-specific methods and one interpretable method. We discussed and compared different ZSL paradigms w.r.t. different external knowledge settings, and found that our resources have great potential for developing more advanced ZSL methods and more solutions for applying KGs for augmenting machine learning. All the resources are available at `https://github.com/China-UK-ZSL/Resources_for_KZSL`.

1. Introduction

Supervised learning has achieved great success in many domains such as natural language processing and computer vision. Its methods often require a large number of labeled training samples to achieve good performance, following a closed world assumption. Namely, they predict with classes that have appeared in the training stage (i.e., seen classes). However, in many real-world applications, new classes always emerge, and it often costs too much computation, human labour and time to address these new classes by collecting labeled samples and training the model from scratch. To this end, Zero-shot Learning (ZSL), which aims at predicting with classes that have no training samples (i.e., unseen classes), was proposed and has been widely investigated in the past decade [1, 2, 3].

Since no labeled samples are given for unseen classes, existing ZSL methods usually rely on *external knowledge* (a.k.a. *side information*) which describes prior semantic relationships between classes. They follow some paradigms to utilize these external knowledge to transfer data and/or models from seen classes to unseen classes. For example, one classic paradigm is mapping-based which first embeds all the classes with their external knowledge, then (jointly) maps the class embeddings and the sample features into one common space where testing samples can be matched with classes by measuring distances with metrics such as the Cosine similarity. Widely investigated external knowledge includes class textual information (e.g., names and descriptions) [4, 5] and class annotations (e.g., attributes) [6]. However, each kind of such external knowledge fails to accurately or fully express inter-class relationships.

Recently, Knowledge Graphs (KGs) [7, 8, 9] have attracted wide attention as the external knowledge of ZSL. For example, Wang et al. [10] and Kampffmeyer et al. [11] incorporate hierarchical inter-class relationships from WordNet [12]; Gao et al. [13], Zhang et al. [14], Nayak et al. [15] and Roy et al. [16] explore relational class knowledge from common sense KGs such as ConceptNet [17]. Significant performance improvement is often achieved when these KGs are well utilized. Besides, KGs can also be used to represent many other kinds of traditional external knowledge such as human annotations and textual information [18, 19], due to its high compatibility in representing and integrating different knowledge. However, although various KGs have been exploited by current methods, there is still a concern on semantics completeness especially in distinguishing fine-grained classes. Meanwhile, very few methods have been developed that can jointly utilize multiple kinds of knowledge in a KG, while other kinds of KG semantics such as logical expressions have not been investigated yet. Furthermore, existing works all build their own KGs for evaluation, and the community lacks standard and unified benchmarks for

*Corresponding author
gengyx@zju.edu.cn (Y. Geng); jiaoyan.chen@cs.ox.ac.uk (J. Chen); zhuangxiang@zju.edu.cn (X. Zhuang); zhuo.chen@zju.edu.cn (Z. Chen); j.z.pan@ed.ac.uk (J.Z. Pan); lijuan18@zju.edu.cn (J. Li); yuanzonggang@huawei.com (Z. Yuan); huajunsir@zju.edu.cn (H. Chen)
ORCID(s): 0000-0002-2461-2613 (Y. Geng)

comparing different KG-based ZSL methods under settings with ranging semantics. And more importantly, KG-based external knowledge has been attractively investigated in the computer vision but rarely covered in other domains. One critical reason for all these issues is that there is a shortage of high quality open benchmarking resources for method development and evaluation.

In this work, we constructed systemic resources for KG-based ZSL research. The resources include six benchmarks with corresponding KGs for three ZSL tasks from different domains: zero-shot image classification (ZS-IMGC), zero-shot relation extraction (ZS-RE), and zero-shot knowledge graph completion (ZS-KGC). The KGs contain different kinds of external knowledge, including not only typical external knowledge such as attribute, text and hierarchy, but also relational facts and logical expressions, with the goal of providing ranging semantic settings for investigating different KG-based ZSL methods. In the paper, we present the technical details of how these resources are constructed, their statistics, data formats, and very high usage for evaluating and developing robust and interpretable ZSL methods.

More importantly, we present an extensive benchmarking study by evaluating and comparing two representative and general ZSL methods, two setting-specific ZSL methods and one interpretable ZSL method, under different external knowledge settings that are supported by our resources. It is worth mentioning that there are currently no methods to utilize those potentially useful logical external knowledge, and we thus developed an effective ensemble-based method which combines symbolic reasoning and neural prediction for ZS-KGC. Through this benchmarking study, we analyzed the benefits of different external knowledge, and the pros and cons of different ZSL methods. We have quite a few concrete observations and future work perspectives that will benefit the ZSL community and the KG community. See details in Section 6 and 7. Here are two brief conclusions:

- Utilizing various semantics represented by KGs can often lead to higher performance and more interpretable solutions, even when they are simply embedded and fed into some methods that are originally developed for single semantics. More effective methods for fusing and injecting different kinds of external knowledge should be investigated in the future.

- ZSL methods of the generation-based paradigm often have more robust performance when both seen and unseen classes are predicted, than methods of the mapping-based paradigm, while the propagation-based paradigm can often well utilize the graph structure. In future ZSL studies, more ZSL methods of different paradigms should be tested and compared with, under different external knowledge settings.

The remainder of this paper is organized as follows. In Section 2, we set up the background of our work, including an introduction to KG and the ZSL tasks in three different domains, review the related works. In the next three sections, we introduce the resources of ZS-IMGC, ZS-RE, and ZS-KGC, respectively. In Section 6, we present the benchmarking study using these resources. Subsequently, we summarize the evaluation results, and discuss the challenges and some potential research directions in Section 7. In the end, we conclude the paper.

2. Background

2.1. Knowledge Graph

Knowledge Graph (KG) is famous for representing and managing graph structured knowledge [7, 8, 9]. It has widely applied in many domains such as search engine, recommendation system, clinic AI, personal assistant, bioinformatics, intelligent finance, software engineering and data analysis [20, 21]. A KG is often largely composed of relational facts in the form of triples of Resource Description Framework (RDF)[1] [22]. Each RDF triple is denoted as (s, r, o), where s represents a subject entity, o represents an object entity, and r represents a relation between these two entities (a.k.a. object property). All these triples compose a multi-relational graph whose nodes correspond to entities and edges are labeled by relations. A KG also contains RDF triples that represent literals and meta information such as entity attributes and textual definition via built-in or bespoke data and annotation properties such as *rdfs:label* and *rdfs:comment*.

In addition to these facts, KGs are often accompanied by an ontological schema and constraints in languages from the Semantic Web community such as RDF Schema (RDFS)[2], Web Ontology Language (OWL)[3] and SHACL[4] for richer semantics and higher quality [23, 24, 25, 26]. They often defines entities' classes (a.k.a. concepts), properties (i.e., stating the terms used as relations), concept and relation hierarchies, constraints (e.g., relation domain and range, and class disjointness), and logical expressions such as relation composition. The languages such as RDF, RDFS and OWL have defined a number of built-in vocabularies for representing these knowledge, such as *rdfs:subClassOf*, *rdf:type* and *owl:disjointWith*. It is worth mentioning that a KG, especially those equipped with schemas and constraints, can support symbolic reasoning such as consistency checking, and entailment reasoning which infers hidden knowledge according to the defined logics. Some ontology reasoners such as HermiT [27] and TrOWL [28] can be directly applied.

Many kinds of data mining and machine learning techniques can also be applied to KGs for approximate inference and knowledge discover. One typical example is KG completion (KGC) by KG embedding techniques which are to learn vector representations of KG components such that their semantics such as relationships are kept in the vector space [29, 30, 31, 32, 33]. KGC tasks often predict links between different KG components, such as between entities, between entities and classes, and between classes. Please see Section 2.4 for more details on KGC. Another example

[1] https://www.w3.org/TR/rdf11-concepts/.
[2] https://www.w3.org/TR/rdf-schema/
[3] https://www.w3.org/TR/owl2-overview/
[4] https://www.w3.org/TR/shacl/

is learning or mining concepts, rules, constraints and other ontological knowledge from KGs [34, 35, 36].

2.2. Zero-shot Image Classification

Image classification is a critical task in computer vision. Zero-shot image classification (ZS-IMGC) refers to predicting images with new classes that have no labeled training images. In the literature of ZS-IMGC, case studies range from classifying general objects [37, 38] to classifying (fine-grained) objects in specific domains such as animals [6, 2], birds [39], and flowers [40]. Please see [2] for a comprehensive survey on ZS-IMGC studies.

To address new classes, some early ZS-IMGC works employ class attributes as external knowledge, which describe objects' visual characteristics about e.g., colors and shapes, to model the relationships between classes. However, these attributes ignore the direct associations between classes, cannot represent complicated relationship and usually need human labour for annotation. Some other works adopt the word embeddings of class names [5, 41], or the sentence embeddings or textual features of class descriptions [42] to model the inter-class relationships. Although such textual information is easy to access, it cannot represent logical or quantitative semantics, and is often quite noisy containing many irrelevant words.

Recently, several methods model the inter-class relationships via KG, with promising results achieved. Wang et al. [10] and Kampffmeyer et al. [11] adopt WordNet to represent the hierarchy of classes of images from ImageNet; Gao et al. [13], Nayak et al. [15] and Roy et al. [16] propose to use common sense KG ConceptNet to introduce more relational knowledge; Geng et al. [43] extract knowledge from DBpedia as a complement of the WordNet class hierarchy. However, all these KG-based ZS-IMGC studies are still preliminary in terms of both semantic sufficiency in the methodology and benchmarking in the evaluation. To bridge the gap of benchmarking and support research in utilizing different external knowledge, in this work, we contributed three resources, each of which can support ranging external knowledge settings with a KG that has incorporated not only class hierarchy, text and attributes, but also common sense class knowledge and logical relationships between classes.

2.3. Zero-shot Relation Extraction

As an important semantic processing task in the field of natural language processing, the objective of relation extraction (a.k.a. relation classification) is to predict the semantic relation of two given entity mentions in a sentence. Since the predicted relation and the given entity mentions can compose a relational fact, RE also serves as an essential technique for KG construction with text. Similar to image classification, conventional supervised relation extraction approaches cannot address new relation types that have never appeared in the training data. To this end, the task of zero-shot relation extraction (ZS-RE), which is to predict unseen relations with given entity mentions and their sentences, was proposed and has been investigated by some studies [44, 45, 46, 47, 48, 49].

To tackle these unseen relations, some ZS-RE studies convert the original problem to another text understanding problem by utilizing the relations' descriptive information. For example, Levy et al. [44] reduce relation extraction to answering reading comprehension questions by associating one or more natural-language questions to each relation type. Obamuyide et al. [45] formulate relation extraction as a textual entailment problem with the relation descriptions, and consider the input sentence and the description as the premise and hypothesis, respectively. However, these works are labor intensive as human efforts are required to design questions or write descriptions for relations. And the transformed tasks may not be suitable for the RE problem enough.

Another attractive way is to leverage the external information that explicitly describes the semantic associations between relations, according to which unseen relations can be directly predicted by transferring features learned from seen relations. For example, Chen et al. [49] explore them from the text descriptions of relation labels. Considering that relation labels can be represented in a KG by a number of relational facts, the state-of-the-art is achieved by those who explored the semantics from KGs [47]. One representative work is [48] by our team, which builds associations between seen and unseen relations via implicit and explicit semantic representations with KG embeddings and logic rules. In comparison with hand-crafted questions or descriptions in aforementioned works, these external information contain more semantic knowledge about relations and are easier to collect such as accessing from online open resources.

Back to the external knowledge currently used, the structured knowledge by KGs is usually more accurate than the text, with less noise when incorporated in learning algorithms. Thus, to better study the impact of KG external knowledge on ZS-RE and facilitate developing more effective KG-based ZS-RE methods, we developed a new ZS-RE resource, where some data in [48] are inherited and improved. In particular, targeting the situation that the original benchmark used in [48] is mainly for the standard zero-shot setting but ignores the more realistic generalized zero-shot setting, we segment the original training set, and down-sample to extract two balanced subsets: one is taken as the new training set and the other is used for testing. Besides, in the benchmarking study, we evaluated more ZS-RE methods such as OntoZSL [18] under more settings, in comparison with the original paper [48].

2.4. Zero-shot Knowledge Graph Completion

KGs such as Wikidata and DBpedia mostly face the challenge of incompleteness, and thus KG completion (KGC), which is often defined to predict the subject, relation or object of a missing triple (fact), has been widely investigated. The KGC methods usually first embed entities and relations into vectors by e.g., geometric learning and Graph Neural Networks (GNNs), and then discover the missing facts in the vector space [50]. However, these methods can only predict entities and relations that have been associated in some training triples, but cannot address newly-added

Table 1
Comparison of our original resources used in OntoZSL [18] and the new resources in KZSL (this paper).

Types of External Knowledge	ZS-IMGC Hierarchies	ZS-IMGC Attributes	ZS-IMGC Literals	ZS-IMGC Relational Facts	ZS-IMGC Logics	ZS-RE KGs+Logic Rules	ZS-KGC Text	ZS-KGC RDFS	ZS-KGC OWL
OntoZSL	✓	✓	✓	✗	✗	✗	✓	✓	✗
KZSL	✓	✓	✓	✓	✓	✓	✓	✓	✓

(unseen) entities and relations, which are quite common as KGs are often evolving. To this end, zero-shot KGC (ZS-KGC), which is to predict triples with entities or relations that have never appeared in the training set, was recently proposed and has achieved quite much attention in recent years [51, 52, 53, 4]. As setting both entities and relations unseen makes the problem much more challenging, in this paper, we focus on those newly-added relations but keep the entity set seen. We will consider unseen entities in the future. **It is worth noting that ZS-RE and ZS-KGC here both involve completing relational facts with unseen relations, but they are totally different tasks. ZS-RE aims to predict the missing relations given entity mentions and their associated text, while ZS-KGC aims to predict new facts given the existing relational facts.**

There are relatively few ZS-KGC studies that aim at addressing unseen relations. Qin et al. [4] leveraged the features learned from relations' textual descriptions, and extracted two benchmarks from NELL and Wikidata for evaluation. As in ZS-IMGC, textual external knowledge is usually noisy, with irrelevant words and ambiguous meanings. To support further studies for developing and comparing ZS-KGC methods that can utilize different kinds of external knowledge, we propose two ZS-KGC resources, each of which is associated with one KG composed of relational facts (i.e., **data graph**) as the target for completion (fact prediction), and one ontological schema (i.e., **schema graph**) as external knowledge. For the schema graph, we adopt some vocabularies in RDFS, such as *rdfs:domain*, *rdfs:range*, *rdfs:subPropertyOf* and *rdfs:comment*, to define and describe relations with their e.g., domain and range constraints, hierarchy and text descriptions, and adopt some vocabularies in OWL, such as *owl:inverseOf*, *owl:propertyChainAxiom* and *owl:SymmetricProperty*, to define some logics such as relation inversion and composition, and some characteristics of relations.

2.5. Related Resources

There have been some open resources that can be used for KG-based ZSL. However, as already discussed above, the KGs of the existing resources usually have only one kind of semantics such as class hierarchy. This makes it hard to fairly compare different methods that use different semantics, and limits the development of more effective methods that can fuse and utilize different KG semantics. Meanwhile, the construction of these resources is usually very briefly introduced in evaluation sessions with details missing. This significantly limits their usage. In contrast, our resources cover different tasks with KGs having different kinds of semantics, and the construction of these resources are well presented with details.

A part of the proposed resources have already been very briefly introduced and used in our OntoZSL paper [18] which focuses on presenting a new ZSL method. However, these resources have been extended massively and some new resources have been added in this work. As shown in Table 1, we *i)* extended the KGs for ZS-IMGC with logical expressions and new common sense knowledge; *ii)* extended the ontological schemas for ZS-KGC with relation semantics in OWL; *(iii)* re-organized all the resources with formal knowledge representation, and higher accessibility; and *iv)* added a resource for a new task — ZS-RE which is widely investigated in natural language processing.

It is worth noting that this is more than a resource paper, but includes an extensive benchmarking study. We have evaluated different ZSL methods using all these resources for different tasks, and have analyzed the impact of different kinds of semantics of the KGs. Besides, we also present the use case of these resources for evaluating the explanations of some ZSL methods.

3. Resource Construction for ZS-IMGC

3.1. Images and Classes

We extract two benchmarks named ImNet-A and ImNet-O from ImageNet which is a large-scale image database organized according to the WordNet taxonomy [37]. Each class in ImageNet is matched to a WordNet node, and has hundreds and thousands of images. Due to a large number of hierarchical classes and a huge number of images, ImageNet is widely adopted in computer vision research as well as in ZSL research.

We focus on the class families (groups) in ImageNet, such as vehicles and dogs, and extract fine-grained classes with the following conditions: 1) seen classes are classes in the ImageNet 2012 1K subset that is often used to train CNNs e.g., ResNet [54], and unseen classes are those one-hop away according to the WordNet hierarchy; 2) the connection between seen and unseen classes are dense, e.g., a seen class has more than one neighboring unseen classes; 3) every class can be linked to a Wikipedia article such that more additional information about this class can be accessed; and 4) the total number of selected seen and unseen classes in each family is at least 5. As a result, we extracted 28 seen classes and 52 unseen classes from 11 families all about animals (e.g., *bees* and *foxes*) for a benchmark named ImNet-A, and extracted 10 seen classes and 25 unseen classes for a benchmark named ImNet-O for general object classification from 5 different class families

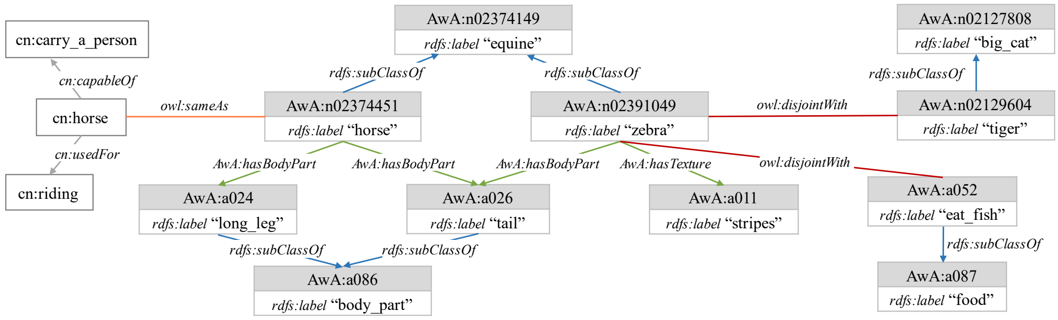

Figure 1: A snapshot of the KG of AwA. The prefixes *AwA* and *cn* are two ad-hoc namespaces of the KG.

Table 2
Statistics of ZS-IMGC benchmarks. "#Att." refers to the number of attributes. S/U denote seen/unseen classes.

Datasets	#Att.	#Classes	#Images		
				Training	Testing
		Total (S/U)	Total	S/U	S/U
ImNet-A	85	80 (28/52)	77,323	36,400/0	1,400/39,523
ImNet-O	40	35 (10/25)	39,361	12,907/0	500/25,954
AwA	85	50 (40/10)	37,322	23,527/0	5,882/7,913

(e.g., *food* and *fungi*). Table 2 shows detailed statistics of ImNet-A and ImNet-O.

In addition, we also re-use a very popular ZS-IMGC benchmark named Animals with Attributes (AwA) [2]. AwA is a coarse-grained dataset for animal classification that contains 37,322 images from 50 animal classes, all of which can be matched to WordNet nodes. The original AwA benchmark has no KG, while in this work, we build a KG as its external knowledge.

3.2. External Knowledge and KG Construction

We collect different kinds of external knowledge, including class hierarchy, attribute, text, relational fact and logical expression, for ImNet-A, ImNet-O and AwA. For each benchmark, we then construct one KG which integrates all these external knowledge. The statistics of the resulting KGs are shown in Table 3.

3.2.1. Class Hierarchy

We first extract the class hierarchy from WordNet whose class nodes are connected via the *super-subordinate* relation (a.k.a. hyponymy or hypernomy relation). The class hierarchy is used as our KG backbone, and is formally represented by the RDFS vocabulary *rdfs:subClassOf*, as Fig. 1 shows. Each class's IRI (Internationalized Resource Identifier) is created following its original WordNet id, e.g., *zebra* from AwA has the IRI *AwA:n02391049*. The prefix "*AwA*" here refers to an ad-hoc namespace of our KG. Since WordNet contains a very large taxonomy, we extract a subset that covers all the benchmark classes and all their ancestors, using the WordNet interface in the Python package NLTK[5].

[5] https://www.nltk.org/howto/wordnet.html

3.2.2. Class Attribute

Based on this structure, we then add the attribute annotations of seen and unseen classes to the graph. Before adding, there is a need to establish the hierarchy of attributes. This is because some attributes describe the same aspect of objects. For example, attributes like *black*, *white* and *red* all describe the appearance color of objects, while *head*, *tail* and *claws* all describe the body parts of animals. The categorization of attributes on the one hand models richer relationships among attributes, and on the other hand, it is helpful for defining the relations between classes and attributes. Under the guidance of WordNet hierarchy, we manually gather the attributes in the datasets into different groups. For example, we gather 17 attributes into the group of body parts and 8 attributes into the color group for the KG of AwA.

We represent these attributes and attribute groups as KG nodes, each of them also has a namespace specified by the dataset to which it belongs and a unique id defined by ourselves, as shown in Fig. 1. Then, we connect attribute nodes to their group nodes via relation *rdfs:subClassOf*, and define the relation edges from class nodes to attribute nodes according to the group to which the connected attribute belongs. For example, for class *zebra* and its one annotated attribute *tail*, a relation named *AwA:hasBodyPart* is defined.

Regarding the attribute annotation data, for AwA, we use its published class-attribute matrices annotated by experts [6], in which each AwA class has an associated binary-valued or continuous-valued attribute vector. In order to avoid the loss of semantic information, we adopt the binary-valued version and extract attributes whose corresponding vector values are 1 as annotated ones for each class. While for ImNet-A/O, we manually annotate attributes for classes as the attributes of ImageNet classes are not available. Briefly, we prepare a list of attributes that are gathered from Wikipedia pages and attribute annotations of other ZS-IMGC datasets such as AwA, divide all 115 classes into 5 parts, and invite 15 volunteers who are undergraduate from Zhejiang University for annotation. Every volunteer is asked to assign 3~6 attributes for each class with the images and Wikipedia articles of classes as references. Each class is independently reviewed by 3 volunteers and the final decision is made by voting. The statistics of attributes of these three datasets are listed in Table 2.

Table 3
Statistics of different kinds of entities, relations and triples in the KGs of ZS-IMGC. "Hie.", "Att." and "CN" are short for Hierarchy, Attribute and ConceptNet, respectively.

Datasets	# Entities				# Relations			# Triples						
	Total	Class	Att.	CN	**Total**	Att.	CN	**Total**	Hie.	Att.	Literal	sameAs	CN	disjointWith
ImNet-A	8,920	111	103	8,706	41	17	21	10,461	210	335	214	156	9,546	/
ImNet-O	3,148	59	52	3,037	31	6	22	3,990	110	110	111	93	3,566	/
AwA	9,195	100	102	8,993	42	15	23	14,112	197	1,562	202	182	10,546	1,423

3.2.3. Class Text

In addition to structured triples, we also introduce the textual information of classes and attributes. Here, we choose their English surface names considering that some classes are hierarchically related and their names are similar. For example, classes *red_fox*, *grey_fox*, *kit_fox* and their parent class *fox*. The class names can also be looked up by NLTK WordNet interface, we represent them in the graph using RDFS vocabulary *rdfs:label*, as Fig. 1 shows.

3.2.4. Relational Fact

We also access more relational class knowledge from a large scale common sense KG ConceptNet [17] whose knowledge is collected from multiple resources including WordNet, DBpedia, etc. We use its latest dump[6] and extract the English subset which contains over 3.4 million triples and around 1.8 million nodes in total. It is obvious that not all of them contain valid information about the classes in the benchmarks, we therefore choose to extract a relevant subgraph by aligning classes and attributes to the entities of ConceptNet and querying their 1-hop neighbors.

Considering that entities in ConceptNet are words and phrases of natural language, we use the literal names of classes and attributes and conduct string matching for alignment. For example, class *zebra* can be aligned to ConceptNet entity *c/en/zebra*. For some attributes that cannot be matched due to different word forms, we lemmatize them before alignment. For example, attribute *spots* is lemmatized to *spot* that can be found in ConceptNet. Besides, we also find that some ConceptNet entities refer to the same objects but have different forms, e.g., *c/en/zebra* and *c/en/zebra/n/wn/animal*. Targeting this, we merge them using a custom namespace "*cn*" and extract the union of these entities' neighborhoods. For example, the above two entities are merged as *cn:zebra*. To be unified, other entities are also represented with this namespace. Finally, for the aligned elements, we use a vocabulary *owl:sameAs* defined in OWL to relate them in the graph. From the statistics of resulting KGs shown in Table 3, we find that the ConceptNet entities of some classes or attributes are still missing, it may be because they are fined-grained concepts and have not been included in ConceptNet yet. We choose to skip them and leave the knowledge extraction of them as a future work. In addition, to reduce the noise during neighborhood query, we ignore the relations with less information, e.g., Synonym, Antonym, SymbolOf, NotCapableOf and NotHasProperty.

However, to use the extracted subgraph, there are still some issues to be addressed. One issue is that the relations extracted from ConceptNet may have the same semantics with the relations we have defined. For example, *cn:isA* and *rdfs:subClassOf* both indicate the semantic of hierarchy. For this, we unify them into *rdfs:subClassOf*. The other is that the knowledge extracted from ConceptNet may already exist. An example is (*cn:squirrel*, *cn:LocatedNear*, *cn:tree*), which is already modelled by the attribute triple: (*AwA:squirrel*, *AwA:hasHabitat*, *AwA:tree*). To solve this, we extract the subjects and objects of these triples to generate a set of tuples (s, o), and filter out ConceptNet triples with repetitive tuples.

3.2.5. Logical Expression

In ZS-IMGC, we also found that some classes that belong to different families and look greatly different have many identical attributes. For example, two animal classes *zebra* and *tiger* both have attributes *stripes*, *tail* and *muscle*. During inference, *tiger* may provide an unexpected significant contribution to the feature learning of *zebra* due to too many shared attributes between them. Although their parent classes (i.e., *equine* and *big_cat*) and literal names have been introduced in the KG to distinguish them, more direct information would benefit the model and should be investigated. One kind of semantics that can be expressed by a KG for further augmentation is the logical relationship defined using OWL vocabulary. Therefore, we define disjointness for these classes and add disjointness axioms using built-in property *owl:disjointWith*, as shown in Fig. 1. In the example above, the disjointness between *zebra* and *tiger* means that an images of *zebra* cannot simultaneously be the instance of *tiger* so that avoiding the misclassification. We also define the disjointness between classes and attributes. For example, the fact "Zebra doesn't eat fish" means the disjointness between class *zebra* and attribute *eat_fish*.

Since the overlap of attributes of ImNet-A/O classes in different families is low, we mainly set the disjoint constraints for classes and attributes in AwA. For the disjointness between different classes, we first generate a candidate set by counting the number of shared attributes. Specifically, for a pair of classes that belong to different families, if their shared attributes are more than $2/3$ of the attributes of class that has fewer attributes, we set a candidate disjoint relationship between them. Then, we invite volunteers to check these candidates with their images as references so that ensuring the correctness of extracted class disjointness.

[6] https://github.com/commonsense/conceptnet5/wiki/Downloads

For the disjointness between classes and attributes, we leverage the continuous-valued attribute vectors, and each class is disjoint with attributes whose vector values are 0.

3.3. Data Overview and Storage

In this section, we contributed two new fined-grained benchmarks ImNet-A and ImNet-O as well as their corresponding KGs. ImageNet has been widely used in the ZS-IMGC literature due to its large scale and diverse granularity, while our work is the first attempt to extract and study its fined-grained subsets. We also build a KG for the widely used coarse-grained dataset AwA. Different from the external knowledge built in previous works with limited class semantics, our constructed KGs not only represent the widely used class knowledge — class hierarchy, attributes and text in a unified graph, but also integrate a subgraph newly extracted from ConceptNet as well as some logic expressions that have not yet been investigated.

Each constructed KG is composed of RDF triples which are stored in a CSV file with three columns corresponding to subjects, relations and objects. The KG files can easily be accessed by Python libraries or be loaded into graph stores. Regarding the images, we follow previous work [2] and provide ResNet features which are stored as a matrix, whose two dimensions correspond to feature vector length and image number, respectively.

4. Resource Construction for ZS-RE

4.1. Relation Text and Types

In the previous work [48], we contributed a benchmark for ZS-RE which is constructed from a well-known relation extraction dataset named Wikipedia-Wikidata [55]. Due to the tight integration of Wikipedia and Wikidata, Wikipedia-Wikidata collects sentences from English Wikipedia corpus, identifies Wikidata entities in the sentences, and annotates relation labels by querying Wikidata relations that connect the extracted entities. As a result, a number of samples are generated, each of them is accompanied by a sentence text, a pair of entity mentions and a relation between them.

To construct the ZS-RE benchmark, all relations in the Wikipedia-Wikidata are first clustered based on their word embeddings, and then the relations in each cluster are divided into two disjoint groups — the seen group and the unseen group according to the number of their samples. The first step ensures the preliminary semantic associations (from word embeddings) between seen and unseen relations, while the second step ensures that seen relations are data-rich while unseen relations are data-poor. In [48], the relations with more than 1,200 samples are specified as seen relations while the rest are specified as unseen ones. After manual adjustments, a dataset with 70 seen relations and 30 unseen relations is finally outputted.

However, the dataset focuses on serving the standard zero-shot setting where samples of only unseen relations are tested, while ignores a more realistic generalized ZSL (GZSL) setting where samples of seen and unseen relations both appear during testing. Moreover, the training samples are not well balanced with respect to the relation — 44 out of 70 seen relations all have 2,800 samples, while the rest 26 have less than 2,800 samples (the minimum is 1,246). The imbalanced training set may have a negative impact on the quality of pre-training a deep neural network that will be used to extract the features of samples. To support the GZSL setting and release the imbalance issue, in this paper, we further split the current training set. Firstly, we down-sample the training set to 1,200 instances per relation to make the training samples balanced over different seen relations. Then, we further down-sample the rest data to the maximum of 50 sentences per relation to generate another balanced subset as the testing data of seen relations. The sample numbers of the original dataset from [48] and the new dataset are compared as Table 4 shows. We rename the new dataset as **ZeroRel**.

Table 4
Statistics of the ZS-RE benchmarks. S/U means seen/unseen relations.

Datasets	# Relations	# Sentences		
			Training	Testing
	S/U	Total	S/U	S/U
Origin Data [48]	70/30	193,867	176,717/0	0/17,150
ZeroRel (New)	70/30	104,646	84,000/0	3,496/17,150

4.2. Knowledge Graph and Logic Rule

We next present how the KG and rule external knowledge are constructured for ZeroRel. As introduced in [48], the implicit semantic association between seen and unseen relations is first mined from pre-trained KG embeddings following the assumption that the embeddings of semantically similar relations are located to each other in the embedding space [31]. For example, the similarity of vectors of relations *nationality* and *live_in_country* is higher, whereas the relation *profession* has low similarity with the former two relations. Therefore, given Wikidata entity mentions and relations in the dataset, we directly use the Wikidata knowledge graph and adopt a version of dump[7] that is often used for training KGE models. Statistically, the dump contains 20,982,733 entities, 594 relations and 68,904,773 triples in total. For convenience, we do not set additional namespaces for it. Based on such a KG, various KGE methods can be applied to pre-train semantically meaningful representations for relation labels and build the implicit semantic associations. From another point of view, the semantic associations between relations can also be modeled by this KG in a symbolic way, e.g., a batch of shared neighboring entities.

Logic rules are further constructed to represent the associations between relations. They are in the form of *body* ⇒ *head*, where *head* is a binary relation and *body* is a conjunction of binary and unary relations, and the length of a rule is determined by the number of relations in its *body*. A rule of length 2 is like $brother(x, y) \wedge father(y, z) \Rightarrow uncle(x, z)$, where x, y, z are three entity variables, showing

[7] http://openke.thunlp.org/download/wikidata

the instances of relation *uncle* can be inferred by the instances of relations *brother* and *father*, and the relation *uncle* can be viewed as a composition of relations *brother* and *father*. A rule of length 1 such as $born\_in\_country(x, y) \Rightarrow nationality(x, y)$ illustrates the semantic identity between relations *born_in_country* and *nationality*. In the ZSL setting, if a rule involves an unseen relation, its instances can be predicted based on other seen relations in the rule. Therefore, it is intuitive to incorporate with logic rules to build an explicit association between seen and unseen relations.

Logic rules could be automatically extracted from structured KGs by any KG rule mining algorithms or tools. We choose AMIE [56] for its convenience and fast-speed to extract logic rules of different lengths from the Wikidata KG proposed above. Each mined rule is associated with a PCA confidence score provided by AMIE. In particular, we limit the maximum length of rules to 2 for the efficiency of mining valid rules, and then keep those rules which include at least one relation in the dataset with a confidence threshold of 0.3. Finally, we mined 50 length-1 rules and 122 length-2 rules in total for the relations in the dataset. **It is noted that these rules can also be formulated using OWL axioms but with probability values, i.e., the length-1 rules with higher confidence values have higher possibility to mean the relation equivalence that can be represented by OWL vocabulary *owl:equivalentProperty*, and the length-2 rules with higher confidence values have higher possibility to mean the relation composition.**

4.3. Data Overview and Storage

To evaluate the zero-shot learning in relation extraction, we constructed a ZS-RE benchmark and contributed a new version that supports more ZSL settings. Based on this, we contributed two kinds of external knowledge — KG and logic rules, both of them contain richer and more accurate relation semantics than the base one from word embeddings.

We store the KG resource in a CSV file as ZS-IMGC case, with three columns corresponding to the subject entities, relations and object entities. The extracted logic rules are stored in a JSON file with "head", "body" and "pcaconf" properties specifying the *head*, *body* and PCA confidence score of a rule. As for the ZS-RE data ZeroRel, we follow previous work to provide it in a CSV file, in which each row corresponds to a sample including the sentence text, the relation label, the entity mention pairs and their indexes in the sentence.

5. Resource Construction for ZS-KGC

5.1. KGs for Completion

We employ two ZS-KGC datasets (i.e., two sub-KGs for completion) proposed in [4]. They are NELL-ZS extracted from NELL[8] and Wiki-ZS extracted from Wikidata[9]. In both datasets, relations are divided into two disjoint sets: a seen relation set $\mathcal{R}_s$ and an unseen relation set $\mathcal{R}_u$. In training, a set of facts (RDF triples) of the seen relations are used, while in testing, the model predicts the facts involving unseen relations in $\mathcal{R}_u$. A closed set of entities are considered in both datasets, which means each entity in the testing triples has already appeared in the training triples. Besides, a subset of training facts is left out as the validation set by filtering all training facts of the validation relations. The statistics of these two datasets are shown in Table 5.

Table 5
Statistics of ZS-KGC datasets. "Tr/V/Te" is short for training/validation/testing. "#Ent." denotes the number of entities.

Datasets	# Ent.	# Relations Total (Tr/V/Te)	# Triples Total (Tr/V/Te)
NELL-ZS	65,567	181 (139/10/32)	188,392 (181,053/1,856/5,483)
Wiki-ZS	605,812	537 (469/20/48)	724,928 (701,977/7,241/15710)

5.2. Ontological Schema

We build an ontological schema as external knowledge for each dataset. It mainly includes semantics expressed by RDFS (concept hierarchy, relation hierarchy, relation's domain and range), semantics expressed by OWL (relation characteristics and inter-relation relationships), and textual meta data (e.g., the names and descriptions of concepts and relations). Note concept here refers to entity type/class. In our paper, the sub-KG for completion is also named as a data graph, and its ontology schema is a schema graph. The statistics of the resulting ontological schemas of NELL-ZS and Wiki-ZS are shown in Table 7.

5.2.1. Semantics in RDFS

Semantics by RDFS vocabularies act as the backbone of the schema graph. Different from the data graph where relations act as edges between nodes, in the schema graph, relations act as nodes (i.e., subjects or objects in RDF triples). Specifically, we use the vocabularies *rdfs:subPropertyOf*, *rdfs:domain*, *rdfs:range* and *rdfs:subClassOf* to define the relation semantics and generate corresponding triples:

- (r_1, *rdfs:subPropertyOf*, r_2), subproperty triple, states the hierarchical relationships between relations, i.e., relation r_1 is a subrelation of relation r_2;
- (r, *rdfs:domain*, C_s), domain triple, summarizes the subject entity type (i.e., subject concept) C_s of relation r;
- (r, *rdfs:range*, C_o), range triple, summarizes the object entity type (i.e., object concept) C_o of relation r;
- (C_i, *rdfs:subClassOf*, C_j), subclass triple, states the hierarchical relationships between entity types C_i and C_j.

A snapshot of the schema graph of NELL-ZS is shown in Fig. 2. As we can see, the schema graph's nodes are relations and entity types of the data graph. We also add dataset-specific namespace for these nodes. Besides, the edge in the schema graph, i.e., the relationship between relations, is called as "meta-relation".

[8] http://rtw.ml.cmu.edu/rtw/
[9] https://www.wikidata.org/

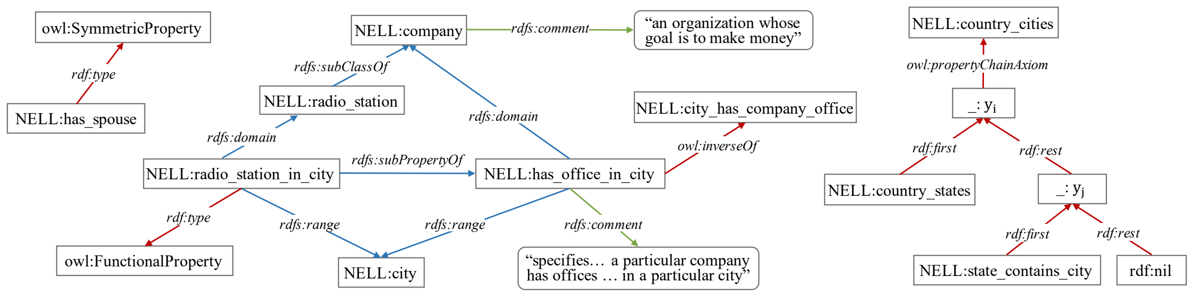

Figure 2: A snapshot of the constructed schema graph for ZS-KGC dataset NELL-ZS. $\_:y_i$ and $\_:y_j$ denote two blank nodes.

For the schema graph of NELL-ZS, the aforementioned semantics in RDFS can be extracted from NELL's ontology. The ontology is saved and published as a CSV file[10] which has three columns corresponding to subjects, predicates and objects of RDF triples. From these triples, we extract domain and range triples according to the predicates "domain" and "range", respectively, and extract subproperty and subclass triples according to the predicate "generalizations". For the schema graph of Wiki-ZS, these semantics can be accessed from Wikidata by a tookit implemented in Python[11]. Specifically, we look up a relation's super-relations by Wikidata property P1647 (subproperty of), look up a relation's domain concepts and range concepts by Wikidata property P2302 (property constraint) with constraints Q21503250 (type constraint) and Q21510865 (value-type constraint), respectively, and look up a concept's super-concepts by Wikidata property P279 (subclass of).

5.2.2. Semantics in Text

We further enrich the schema graphs with textual information of the nodes (i.e., relations and concepts), which usually act as important external knowledge in addressing ZS-KGC [4, 18, 52]. For NELL-ZS, we extract the textual descriptions of relations and concepts from NELL's ontology file by the predicate "description". For Wiki-ZS, we look up the surface names and descriptions of relations and concepts from Wikidata using properties *label* and *description*, respectively. The extracted text can be represented in the graph by RDFS vocabularies *rdfs:label* and *rdfs:comment*, leading to a literal-aware schema graph.

5.2.3. Semantics in OWL

We also introduce relation semantics in OWL including the relationships between relations and relation characteristics. We provide an overview illustration with definitions and examples in Table 6, and will next introduce the details.

Inverse Relationship. The inverse relationship between two relations is defined by *owl:inverseOf*. If r_1 is an inverse relation of r_2, when a fact (e_1, r_1, e_2) holds, the fact (e_2, r_2, e_1) also holds, and vice versa. In building the ontological schemas for ZS-KGC, we introduce the inverse relationships between seen and unseen relations, with triples in format of $(r_1, \textit{owl:inverseOf}, r_2)$. In prediction, the ZS-KGC models can utilize the unseen relations' inverse relations which have been involved in the training triples. Since the inverse relations have been removed from NELL-ZS when it is originally constructed, we only add inverse triples for relations in Wiki-ZS, which are extracted from Wikidata by its property P1696 (inverse property).

Compositional Relationship. A relation can be constructed by ordered composition of several other relations. Relation r_3 is a composition of another two relations r_1 and r_2, denoted as $r_1 \wedge r_2 \Rightarrow r_3$, if we have $(x, r_1, y) \wedge (y, r_2, z) \Rightarrow (x, r_3, z)$, where x, y and z are three entity variables. Such compositional relationships are also helpful for KGC. For example, with *brother* $\wedge$ *father* $\Rightarrow$ *uncle*, we can infer a is an uncle of c if a is a brother of b and b is a parent of c. Therefore, we add composition axioms in our ontological schemas to define some seen and unseen relations as the compositions of some seen relations. We limit the number of compositional relations in each axiom to 2. A composition axiom can be represented in the schema graph as the rightest part of Fig. 2 shows, where it has been serized as RDF triples with blank nodes according to W3C OWL to RDF graph mapping standard[12].

For NELL-ZS, we first extract a set of candidate relation compositions via checking relation's domain and range. Specifically, for any three relations r_1, r_2 and r_3, if the range of r_1 is the domain of r_2, the domain of r_1 is also the domain of r_3 and the range of r_2 is also the range of r_3, then $r_1 \wedge r_2 \Rightarrow r_3$ is regarded as a candidate composition. These candidates can be extracted according to the schema of NELL-ZS defined by RDFS. Briefly, for each relation of NELL-ZS, we traverse all seen relation pairs and check whether the domains and ranges of the two seen relations and the current relation match the above condition. Some candidate relation compositions extracted in the above step are not correct; one example is *mother_of_person* $\wedge$ *person_also_knownas* $\Rightarrow$ *wife_of*. Targeting this, we manually check these candidates. Briefly, each candidate is independently reviewed by three volunteers (including one of the authors and two of our colleagues who are familiar with KGs and ontologies) and the final decision is made by voting. It is also allowed that volunteers can look up all the information about relations such as triples and descriptions during review.

[10] http://rtw.ml.cmu.edu/resources/results/08m/NELL.08m.1115.ontology.csv.gz

[11] https://pypi.org/project/Wikidata/

[12] https://www.w3.org/TR/owl2-mapping-to-rdf/

Table 6
Illustrations and statistics of inter-relation relationships and relation characteristics of the ontological schemas of NELL-ZS and Wiki-ZS. x, y, z are entity variables. "[NELL]" and "[Wiki]" denote the example comes from NELL-ZS and Wiki-ZS, respectively.

OWL Semantics	Formula	Example	Statistics NELL-ZS	Wiki-ZS
Inversion	$(x, r_1, y) \Leftrightarrow (y, r_2, x)$	P802 (student) & P1066 (student of) [Wiki]	0	39
Composition	$(x, r_1, y) \wedge (y, r_2, z) \Rightarrow (x, r_3, z)$	countrystates ∧ statecontainscity ⇒ countrycities [NELL]	20	7
Symmetry	$(x, r, y) \Rightarrow (y, r, x)$	hasspouse [NELL]	20	25
Asymmetry	$(x, r, y) \Rightarrow \neg(y, r, x)$	subpartof [NELL]	24	11
Reflexivity	(x, r, x)	animalpreyson [NELL]	2	0
Irreflexivity	$\neg(x, r, x)$	P184 (doctoral advisor) [Wiki]	46	15
Functionality	$(x, r, y) \wedge (x, r, z) \Rightarrow y = z$	airportincity [NELL]	6	15
Inverse Functionality	$(x, r, y) \wedge (z, r, y) \Rightarrow x = z$	statecontainscity [NELL]	16	1

Table 7
Number of relations, concepts, literals, meta-relations, and different axioms in the ontological schema.

Datasets	# relations	# concepts	# literals	# meta-relations	# subproperty	# domain	# range	# subclass	# relation characteristics
NELL-ZS	894	292	1,063	9	935	894	894	332	114
Wiki-ZS	560	1,344	3,808	11	208	1,843	1,378	1,392	67

However, the method of checking relation's domain and range can not be well applied to Wiki-ZS because most Wiki-ZS relations have multiple domains and multiple ranges, which will result in too many candidates and cost too much manual assessment. Therefore, for Wiki-ZS, we use AMIE to mine compositional rules from facts. Different from mining rules from the Wikidata dump in the ZS-RE task, we mine relation compositions from the triples of Wiki-ZS dataset since some Wiki-ZS relations are not included in the dump. Moreover, to ensure the correctness of mined rules, we *i)* filter out those rules whose scores are below 0.9, and *ii)* invite volunteers to manually assess the remaining rules as for NELL-ZS. Finally, we transform the correct ones into relation composition axioms for the schema of Wiki-ZS.

Symmetry & Asymmetry. In a KG, a relation r is symmetric if we have (y, r, x) given (x, r, y). One typical example is *has_spouse*. In contrast, a relation r is defined as asymmetric if (y, r, x) is always false given (x, r, y). We add symmetric and asymmetric characteristics to some relations in our schemas of NELL-ZS and Wiki-ZS, because they could be utilized by potential methods for addressing ZS-KGC by e.g., inferring more facts for training and finding similar seen relations for an unseen relation. In our resource we add symmetry and asymmetry for relations that have identical domains and ranges.

To add symmetry and asymmetry for relations in NELL-ZS, we use the predicate "anti-symmetric" defined in the ontology file of NELL. Specifically, for symmetric relations, we first extract relations whose "anti-symmetric" values are false, and then select those with the same domain and range. Some of the resultant relations are still not symmetric, and we invite volunteers to filter out them. For asymmetric relations, they can be automatically extracted by the predicates "anti-symmetric" and "irreflexive" considering that a relation is asymmetric iff it is antisymmetric and irreflexive. For relations of Wiki-ZS, the symmetric relations can be extracted by looking up the Wikidata constraint Q21510862 (symmetric constraint) stated in the property P2302 (property constraint). While for the asymmetry, we extract relations which have identical domain and range, and manually assess them.

We use membership axioms and two OWL built-in concepts *owl:SymmetricProperty* and *owl:AsymmetricProperty* to represent relation symmetry and asymmetry in our ontological schemas. When the ontologies are represented as schema graphs, relation characteristics are transformed into RDF triples like (r, *rdf:type*, *owl:SymmetricProperty*) which means r is a symmetric relation.

Reflexivity & Irreflexivity. A relation r is regarded as reflexive if (x, r, x) holds and as irreflexive if (x, r, x) does not hold where x is an entity variable. Similar to symmetry and asymmetry, relation reflexivity and irreflexivity could be utilized in ZS-KGC with e.g., additional training samples and more information for relation similarity. We can even directly infer testing triples in form of (e, r, e) if r is reflexive.

We use the values of the predicate "anti-reflexive" defined in the NELL's ontology file to add reflexive and irreflexive characteristics for relations in NELL-ZS. For relations in Wiki-ZS, since Wikidata has no definitions towards these two characteristics, we extract relations that have identical domain and range, and manually assess their reflexivity and irreflexivity. The representation of relation reflexivity and irreflexivity in the schema graph is the same as symmetry and asymmetry but uses the built-in concepts of *owl:ReflexiveProperty* and *owl:IrreflexiveProperty*.

Functionality & Inverse Functionality. Given a functional relation r, if (x, r, y) and (x, r, z) holds, then y and z must be the same entity. Namely every entity can be related to at most one entity via a functional relation. A relation can also be defined as inverse functional when its inverse relation is functional. We add both functionality and inverse functionality to some relations. They can be potentially used for addressing ZS-KGC as the other relation characteristics discussed above. They can also constrain the searching space for new triples. For example, given a subject and a functional relation, the corresponding object must be unique.

To extract these two characteristics, we look up the whole set of triples of NELL (via its published dump[13]) and Wikidata (via its SPARQL Endpoint), and extract relations whose object entity is unique for the same subject, and relations whose subject entity is unique for the same object. They are represented in the same way as symmetry and asymmetry but use the built-in concepts of *owl:FunctionalProperty* and *owl:InverseFunctionalProperty*.

5.3. Data Overview and Storage

To the best of our knowledge, this resource is the first to incorporate with rich ontology information for tackling the ZS-KGC problems with unseen relations. In comparison with the external knowledge contained in text, ontological schema provides richer and more accurate semantics about KG relations. With our resources, various ontology-driven ZS-KGC methods are expected to develop.

Each ontological schema is saved in two formats. The first is the original ontology file ended with ".owl". It can be directly loaded and easily viewed by ontology editors such as Protege. The second is an RDF triple file to save the schema graph which is transformed from the ontology according to W3C OWL to RDF graph mapping. It is convenient for graph embedding methods e.g., GNNs and KGE algorithms to process. Note other mappings from OWL ontology to RDF graph can be considered by the user.

6. Benchmarking and Results

6.1. Evaluating ZSL Model Performance

In this section, we evaluate and compare the performance of different ZSL methods under different KG settings, using the aforementioned resources. We first introduce the ZSL methods and evaluation settings, and then separately introduce the results on ZS-IMGC, ZS-RE and ZS-KGC.

6.1.1. ZSL Methods and Evaluation Settings

As we introduce earlier, a kind of widely investigated ZSL methods are mapping-based. They usually learn a mapping between the class embedding space and the sample space via data of seen classes, and generalize the learned mapping to unseen classes for prediction which is implemented by searching for matched classes for a testing sample according to some distance metrics. According to the space where the searching is conducted, these methods can be further divided into three categories: semantic-space based which maps the sample feature to the space of class embedding, sample-space based which maps the class embedding to the space of sample feature, and common-space based which maps the sample feature and the class embedding to a common latent space. In addition to these mapping-based methods, another popular branch of methods are generation-based. They formalize ZSL as a missing data problem and learn to synthesize samples (features) for unseen classes conditioned on their class embeddings to augment data.

In this work, we adopt two representative methods as our evaluation approaches. One is a classic semantic-space based algorithm named DeViSE [5], which is widely used in various ZSL studies. The other method is a state-of-the-art generation-based method named OntoZSL [18] which leverages Generative Adversarial Network (GAN) [57] to generate data and is originally developed to utilize ontologies as external knowledge. Another reason for selecting them is that they are compatible to different external knowledge that have been embedded, through which we are able to systematically compare different knowledge settings. We benchmarked both methods for all the three ZSL tasks.

It is worth noting that we considered the performance in two ZSL settings: one is the standard ZSL which tests samples of only unseen classes, the other is the generalized ZSL (GZSL) which tests samples of both seen and unseen classes and is more challenging. We evaluated under both settings for the ZS-IMGC and ZS-RE tasks, while for the ZS-KGC task, we only considered the standard ZSL setting since in our task of predicting a triple's object entity, the subject entity and the unseen relation are given (see more details in Section 6.1.4).

6.1.2. ZS-IMGC

With the KGs of the ZS-IMGC benchmarks, we made the following four external knowledge settings which have different semantics:

- **Basic KG**: Hierarchy and attribute triples which cover the semantics of class hierarchy, class attributes and attribute hierarchy.
- **Basic KG+literals**: Basic KG plus textual information.
- **Basic KG+CN**: Basic KG plus ConceptNet subgraph.
- **Basic KG+logics**: Basic KG plus disjointness axioms.

To apply these external knowledge in ZSL, we take advantage of some semantic embedding techniques to encode them and generate a vector representation for each class. Specifically, we adopt mature and widely-used TransE [58] to encode the graph structural knowledge contained in **Basic KG**, **Basic KG+CN** and **Basic KG+logics**. For **Basic KG+literals**, we adopt a text-aware graph embedding method used in [18] to simultaneously embed the textual and graph structural knowledge.

Besides the above KG-based external knowledge settings, we also made the following simple but widely used external knowledge settings. Relevant external knowledge can be extracted from the original benchmarks, and have also been included in our new resources:

[13] http://rtw.ml.cmu.edu/resources/results/08m/NELL.08m.1115.esv.csv.gz

Table 8
*Acc*uracy (%) of DeViSE on AwA, ImNet-A and ImNet-O. The best result on each metric is marked with underline.

External Knowledge	AwA				ImNet-A				ImNet-O			
	acc	acc_s	acc_u	H	acc	acc_s	acc_u	H	acc	acc_s	acc_u	H
w2v (500)	24.22	78.42	1.05	2.08	13.52	59.71	0.63	1.25	14.21	66.40	3.93	7.43
w2v (300)	8.42	86.32	0.00	0.00	26.95	84.36	0.16	0.32	20.49	93.60	0.00	0.00
att	38.48	81.86	3.59	6.88	35.72	61.00	12.60	20.89	31.75	47.80	17.24	25.34
hie	43.50	65.25	5.60	10.32	30.94	62.07	1.67	3.25	29.25	54.60	10.85	18.10
Basic KG	43.24	86.44	6.40	11.91	34.38	25.50	28.13	26.75	24.77	34.20	22.49	27.14
Basic KG + literals	46.12	84.42	8.76	15.88	33.62	23.36	29.33	26.01	26.13	38.60	21.67	27.75
Basic KG + CN	45.56	88.85	0.38	0.76	35.11	67.71	7.46	13.43	26.72	70.40	7.23	13.11
Basic KG + logics	37.54	80.69	1.09	2.15	–	–	–	–	–	–	–	–

Table 9
*Acc*uracy (%) of OntoZSL on AwA, ImNet-A and ImNet-O. The best result on each metric is also underlined.

External Knowledge	AwA				ImNet-A				ImNet-O			
	acc	acc_s	acc_u	H	acc	acc_s	acc_u	H	acc	acc_s	acc_u	H
w2v(500)	45.39	57.83	34.53	43.24	20.94	34.50	15.62	21.50	20.00	41.20	14.33	21.27
w2v(300)	20.80	22.67	12.88	16.43	27.76	40.50	20.40	27.13	24.73	37.20	17.52	23.83
att	58.47	59.90	44.24	50.89	37.87	33.50	27.62	30.28	32.98	42.00	20.67	27.71
hie	38.89	51.08	31.38	38.88	33.32	40.93	23.06	29.50	33.17	36.80	21.13	26.85
Basic KG	62.65	59.59	50.58	54.71	38.21	45.71	23.21	30.79	32.14	44.60	18.74	26.39
Basic KG + literals	59.21	62.39	45.55	52.66	38.58	35.64	27.64	31.13	32.57	44.80	19.35	27.03
Basic KG + CN	54.61	63.31	39.19	48.41	35.24	39.86	24.97	30.71	29.39	42.20	19.64	26.80
Basic KG + logics	54.65	65.37	40.76	50.21	–	–	–	–	–	–	–	–

- **Class Attributes (att)**: 85, 85 and 40 dimensional binary-valued attribute vectors (multi-hot vectors) for classes of AwA, ImNet-A and ImNet-O, respectively.
- **Class Word Embeddings (w2v)**: Two kinds of word vectors for each class via its name. One is by [59] with a dimension of 500; the other is by a Glove model with a dimension of 300.
- **Class Hierarchy (hie)**: A 100 dimensional vector for each class, encoded by a graph auto-encoder [60] over the class hierarchy.

We evaluate these external knowledge settings and two ZSL methods by macro accuracy following the standard in the ZS-IMGC community. Macro accuracy is calculated in the following way: an accuracy, the ratio of correct predictions over all the testing samples, is first independently computed for each class, and then the accuracies of all tested classes are averaged. In the stardard ZSL setting, we compute the macro accuracy over all the unseen classes, denoted as acc. In the generalized ZSL setting, two macro accuracies are calculated over the seen classes and the unseen classes, respectively, denoted as acc_s and acc_u, and a harmonic mean $H = (2 \times acc_s \times acc_u)/(acc_s + acc_u)$ which balances the performance of predicting seen classes and unseen classes is calculated as the overall metric.

The results of DeViSE and OntoZSL on three datasets are shown in Table 8 and Table 9, respectively. From both tables, we can see that the KG-based external knowledge settings all achieve better performance than those currently widely used none-KG-based settings — **att**, **w2v** and **hie** on AwA. Although the KG-based settings do not always achieve the best performance w.r.t. all the metrics on ImNet-A and ImNet-O, the results are still comparable. All these illustrate the potential of KG-based external knowledge in ZS-IMGC. Besides, with TransE which was originally developed for KGs with relational facts alone, and a simple pipeline that stacks semantic embedding and a ZSL method, although more semantics are introduced in **Basic KG+CN** and **Basic KG+logics**, their results are not better than **Basic KG** on most metrics. This motivates the community to *i)* develop more effective ZS-IMGC techniques to utilize all these promising semantics for better performance, or *ii)* adopt more flexible strategies to take advantage of these semantics such as retrieving more refined knowledge for specific prediction tasks or datasets, and our resources provide a chance for such investigations.

We also observe the performance differences when applying the setting of w2v to different datasets, i.e., **w2v(500)** performs better than **w2v(300)** on AwA, whereas on ImNet-A/O, the situation is inverse — **w2v(300)** performs better on most metrics. For example, when experimenting with OntoZSL, on AwA, **w2v(500)** achieves 45.39% in acc and 43.24% in H, with improvements of 24.59% in acc and 26.81% in H over **w2v(300)**. While for ImNet-A, on the metrics of acc and H, **w2v(300)** inversely achieve respective 6.82% and 5.63% performance gains against **w2v(500)**. For ImNet-O, the improvements are 4.73% on acc and 2.56% on H. These two kinds of word embeddings are both pre-trained on Wikipedia corpus, but the difference is that **w2v(500)** by [59] directly learns a word vector for each class, no matter the class name contains a single word or multiple words, while **w2v(300)** generates word vectors for multiple-word classes by averaging the word vectors of terms in names. The averaged class word embeddings may be more beneficial for the fine-grained classes of ImNet-A/O as the words in the names of the classes in a family are highly overlapped.

Table 10
*Acc*uracy (%) of GCNZ on AwA, ImNet-A and ImNet-O. The better result on each metric is marked with underline.

External Knowledge	AwA				ImNet-A				ImNet-O			
	acc	acc_s	acc_u	H	acc	acc_s	acc_u	H	acc	acc_s	acc_u	H
Class Hierarchy	37.44	<u>81.45</u>	7.86	14.34	33.95	<u>48.71</u>	18.37	26.68	32.24	<u>49.00</u>	18.04	26.37
Basic KG	<u>62.98</u>	75.59	<u>20.28</u>	<u>31.98</u>	<u>36.64</u>	45.57	<u>23.92</u>	<u>31.38</u>	<u>33.98</u>	43.80	<u>21.61</u>	<u>28.94</u>

Instead, the classes in AwA are coarse-grained, the overlap of words in their names is relatively less. Correspondingly, in encoding the text in the **Basic KG+literals** with 300-dimensional word vectors from Glove (i.e., **w2v(300)**), **Basic KG+literals** gains an improved performance w.r.t most metrics against **Basic KG** on ImNet-A/O whereas even performs worse on AwA when experimenting with OntoZSL. See the fifth and sixth rows of Table 8 and Table 9 for more detailed comparisons. According to these observations, we can conclude that **w2v(300)** may be more appropriate for tasks with fine-grained classes, while **w2v(500)** may be better for tasks with coarse-grained classes. Moreover, it also inspires us to take the properties of the task and data into account when designing semantic embedding techniques for the external knowledge.

Apart from these two representative ZSL methods, we also evaluate a famous KG-based ZSL method named GCNZ [10], which uses a Graph Convolutional Network (GCN) to propagate features between class nodes and outputs a classifier for each unseen class, under two semantic settings that GCNZ can support, i.e., **Basic KG** and **Class Hierarchy**. GCNZ is a typical method of the propagation-based paradigm which has attracted wide attention for KG-based ZS-IMGC studies [61, 13, 62, 15, 43]. The results are shown in Table 10. It can be seen that along with more semantics introduced in **Basic KG** than in **Class Hierarchy**, a higher performance is obtained over all three datasets, including an improvement on predicting unseen classes in the standard ZSL and a better balance between acc_s and acc_u in the generalized ZSL. We also find a larger performance gap between **Basic KG** and **Class Hierarchy** on AwA than on ImNet-A/O. It may be because the distances between AwA classes in the class hierarchy graph are larger since the nature of coarse granularity of AwA, and this may penalize the feature propagation in the graph convolutional layer. Contrastingly, when more semantics such as class attributes are introduced in **Basic KG**, the connections between AwA classes are greatly enriched, resulting in a significant performance improvement. It is worth noting that though we ignore the relational edges in the **Basic KG** when GCNZ is applied (since GCN can only support single relation graph), we still obtain promising results. In the future, more GCN variants that can support multi-relation graphs such as R-GCN [63] and CompGCN [64] can be investigated to make full use of the semantics of the KG.

6.1.3. ZS-RE

We evaluate one baseline external knowledge setting which is based on word embeddings, and other two external knowledge settings based on our resources introduced in Section 4.2:

Table 11
*Acc*uracy (%) of DeViSE and OntoZSL on ZeroRel. The best result on each metric is underlined.

Semantics	DeViSE				OntoZSL			
	acc	acc_s	acc_u	H	acc	acc_s	acc_u	H
w2v	21.40	<u>65.73</u>	0.04	0.08	11.00	<u>56.86</u>	0.20	0.40
KG	34.43	63.64	7.47	13.37	35.81	53.06	9.21	15.69
Rule	<u>35.74</u>	61.96	<u>10.20</u>	<u>17.51</u>	<u>36.20</u>	52.20	<u>12.05</u>	<u>19.58</u>

- **w2v**: one 100-dimensional word vector for each relation by averaging the words in its name. The word embedding model is trained on the latest Wikipedia dump[14] using the word2vec algorithm [65] with a window size of 5.
- **KG**: one 100-dimensional KG embedding for each relation. The KG entity and relation embeddings are trained by the OpenKE [66] toolkit using TransE on the Wikidata dump constructed in Section 4.2.
- **Rule**: one 100-dimensional rule-guided relation embedding for each relation. The embedding method incorporated with rules are introduced bellow.

We leverage the pre-trained KG embeddings for initial relation embeddings, and then utilize the extracted rules to generate rule-guided relation embeddings. For a length-1 rule $r_1(x, y) \Rightarrow r_2(x, y)$ and a length-2 rule $r_3(x, y) \wedge r_4(y, z) \Rightarrow r_5(x, z)$, we can get $\vec{r_1} = \vec{r_2}$ and $\vec{r_3} + \vec{r_4} = \vec{r_5}$ according to the TransE KGE algorithm which assumes $\vec{s}+\vec{r} \approx \vec{o}$ for a valid triple (s, r, o). Note $\vec{r_1}, \vec{r_2}, \vec{r_3}, \vec{r_4}, \vec{r_5}, \vec{s}, \vec{r}, \vec{o}$ all represent entity and relation embeddings. Thus, for a relation associated with such rules, its embedding can be re-calculated based on the embeddings of other relations in the rules. Specifically, the embedding of relation r associated with K rules is calculated as follows:

$$E_{rl}(r) = \frac{\sum_{k=1}^{K} s_k * E_{TransE}(R_k^r)}{\sum_{k=1}^{K} s_k}$$

where R_k^r is the k-th rule of relation r, and s_k corresponds to its confidence score. $E_{TransE}(\cdot)$ represents the rule-based operation following the TransE's assumption. For example, for an unseen relation r_u with its three rules, R_1: $r_A \Rightarrow r_u$, R_2: $r_B \wedge r_C \Rightarrow r_u$, and R_3: $r_D \wedge r_u \Rightarrow r_E$, its rule-guided embedding is calculated as:

$$E_{rl}(r_u) = \frac{s_1 * \vec{r_A} + s_2 * (\vec{r_B} + \vec{r_C}) + s_3 * (\vec{r_E} - \vec{r_D})}{s_1 + s_2 + s_3}.$$

[14] https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2

Table 12
Results (*MRR* and *hit@k* (%)) of DeViSE and OntoZSL on NELL-ZS and Wiki-ZS. The best result on each metric is underlined.

External Knowledge	DeViSE								OntoZSL							
	NELL-ZS				Wiki-ZS				NELL-ZS				Wiki-ZS			
	MRR	hit@10	hit@5	hit@1	MRR	hit@10	hit@5	hit@1	MRR	hit@10	hit@5	hit@1	MRR	hit@10	hit@5	hit@1
Text	0.221	34.6	29.0	15.5	0.183	26.7	21.7	13.5	0.215	34.5	28.3	14.5	0.185	27.3	22.3	13.5
RDFS-hie	0.229	35.1	29.3	16.3	0.179	25.4	20.9	13.5	0.225	34.8	28.9	15.9	0.175	25.4	20.4	13.1
RDFS-cons	0.221	34.5	28.7	15.3	0.183	26.4	21.7	13.6	0.220	34.3	28.0	15.4	0.177	25.7	21.2	13.0
RDFS graph	0.225	35.3	29.4	15.6	0.184	27.0	21.7	13.6	0.223	35.1	29.1	15.3	0.185	27.5	22.3	13.4
RDFS+literals	0.223	35.0	29.0	15.3	0.185	27.1	22.0	13.6	0.227	35.6	29.4	15.6	0.188	28.1	22.6	13.5

Apart from the ZSL method of DeViSE which had been evaluated in [48], in this paper, we also extend to experiment with OntoZSL, i.e., learning to generate instance features (extracted by Piecewise Convolutional Neural Networks [67]) for unseen relations conditioned on their semantic embeddings. Both methods are assembled with the three external knowledge settings mentioned above. The results on our newly constructed dataset ZeroRel are shown in Table 11, where the classification accuracy is reported. Similar to ZS-IMGC, we compute the accuracy independently for each relation type, and report the averaged accuracy on unseen relations in the standard ZSL (i.e., acc) and on seen and unseen relations in the generalized ZSL (i.e., acc_s and acc_u, respectively) with their harmonic mean computed.

From Table 11, we can see that the settings of **KG** and **Rule** both achieve significant improvements over the **w2v** setting, no matter what ZSL methods are used. Especially, the **Rule** setting leads to the best performance. All of these results illustrate that the semantics in the external knowledge from KG and logic rules are richer than the semantics in the word embeddings w.r.t. the task of ZS-RE, and the rule-guided embedding method we used is effective. In the future, we plan to develop more techniques to utilize the semantics in KGs and the logic rules. We also find that the generation-based method OntoZSL performs better than the mapping-based method DeViSE, especially with respect to acc_u and H, under the knowledge settings of **KG** and **Rule** which contain richer semantics than **w2v**. For example, under the setting of **Rule**, the H score of OntoZSL is 11.8% higher than that of DeViSE.

6.1.4. ZS-KGC

For NELL-ZS and Wiki-ZS, we made one simply none KG external knowledge setting that has already been included in the original benchmark, and five KG-based settings using our new resources:

- **Text**: relations' textual descriptions which are originally proposed and used by Qin et al. [4];
- **RDFS graph**: complete relation semantics in RDFS;
- **RDFS-hie**: a part of RDFS graph, covering subproperty and subclass triples;
- **RDFS-cons**: a part of RDFS graph, covering relation domain and range constraints;
- **RDFS+literals**: RDFS graph plus textual meta data of concepts and relations;
- **RDFS+OWL**: RDFS graph plus semantics in OWL.

As in ZS-IMGC and ZS-RE, we embed the external knowledge and apply the resultant relation vectors into the ZS-KGC methods. For **RDFS graph** and its two subgraphs (**RDFS-hie** and **RDFS-cons**), we adopt TransE for embedding, and for **RDFS+literals**, we also adopt the text-aware graph embedding method used in [18]. To embed the relation's textual descriptions, we follow [4] and perform a weighted summation of the vectors of words in descriptions, where an open word embedding set[15] of dimension 300 is used. We also use the same word vectors to initialize the representation of text in **RDFS+literals**.

We first compare these five external knowledge settings using methods of DeViSE and OntoZSL, where the sample features of KG relations are learned by their associated entity pairs. The KGC task here is to predict the object entity given a subject entity and a relation, we thus rank all the candidate entities according to their likelihood to be the object. Two commonly used metrics are adopted: mean reciprocal ranking (*MRR*) which computes the average of the reciprocal predicted ranks of all the ground truths (right objects), and $hit@k$ which represents the ratio of testing samples whose ground truths are ranked in the top-k positions (k is set to 1, 5, 10) [29]. As the candidate space only involves entities, the prediction with unseen relations is independent of the prediction with seen relations, and in fact, the latter is the traditional KGC task. Therefore, we only consider the standard ZSL testing setting in ZS-KGC. The results are shown in Table 12.

From Table 12, we find that in comparison with **Text**, **RDFS graph** and **RDFS+literals** always lead to better performance, **RDFS-hie** and **RDFS-cons** that contain incomplete RDFS semantics also achieve comparable results, and even perform better on some metrics. For example, when applying the OntoZSL method on NELL-ZS, the *MRR* values of **Text**, **RDFS-hie**, **RDFS-cons**, **RDFS graph** and **RDFS+literals** are 0.215, 0.225, 0.220, 0.223 and 0.227, respectively. These results demonstrate the superiority of our proposed RDFS-based relation semantics. Besides, we also find that when changing the external knowledge from **Text** to **RDFS graph**, a higher improvement is achieved on

[15] https://github.com/mmihaltz/word2vec-GoogleNews-vectors

NELL-ZS than on Wiki-ZS. For example, when OntoZSL is applied, the value of $hit@10$ is improved by 0.6% on NELL-ZS, whereas is only improved by 0.2% on Wiki-ZS. And on Wiki-ZS, the results w.r.t other metrics are roughly the same. See the first and fourth row of Table 12 for more comparisons. These results are mainly due to that 44 out of 537 Wikidata relations have missing semantics in RDFS. When complementing the RDFS graph with relations' textual information (i.e., **RDFS+literals**), the performance on Wiki-ZS is further improved.

In Table 12, it can also be seen that **RDFS-hie** performs better than **RDFS-cons** on NELL-ZS, while **RDFS-cons** inversely achieves better performance on Wiki-ZS. For example, when experimenting with DeViSE, the *MRR* value of **RDFS-hie** on NELL-ZS is 0.229 and 3.6% relative higher than that of **RDFS-cons**, while on Wiki-ZS, the *MRR* is decreased by 0.004 when shifting the setting from **RDFS-cons** to **RDFS-hie**. This is probably because for all relations in the NELL-ZS, around 58% of them are hierarchically related, while only nearly 30% have identical domain and range constraints; while in Wiki-ZS, the constraint information are richer than the hierarchy information, i.e., 160 relations are hierarchically related, and although only 161 relations have the same domain and range constraints, over 70% of them have more than one identical domains or ranges.

The external knowledge defined by OWL is quite promising for augmenting ZSL, but there are current no systemic or robust methods. We try to validate the effectiveness of OWL semantics by testing an ensemble method, which combines symbolic reasoning and embedding-based prediction, under the setting of **RDFS+OWL**. Briefly, for a testing tuple (subject and relation), if the object can be inferred through logical expressions (cf. Section 5.2.3), we adopt the inferred object; otherwise, we use the predicted object ranking by ZSL models such as OntoZSL. Even with such a naive ensemble solution, we got some encouraging results. On Wiki-ZS, 5 unseen relations have inverse relations which are among the seen relation set, and $hit@1$ increases from 28.6% to 55.3% when logical inference with the inverse semantics is used. On NELL-ZS, 4 unseen relations are composed by 10 seen relations, and the logical inference with composition leads to 6.3% increment on $hit@1$. These results demonstrate the superiority of the OWL-based relation semantics, although the method of utilizing OWL axioms presented here is quite preliminary. With our resources, more robust methods can be investigated to better utilize such logical expressions and significantly augment ZSL performance.

Besides, the OWL semantics can also be used to validate the correctness of prediction results or improve the prediction efficiency. For example, irreflexivity constrains that the predicted object entity can not be the same with the given subject entity; and both irreflexivity and asymmetry can be used to reduce the searching space of candidate entities during prediction. In the future, more available and effective methods are expected to take advantage of these semantics for promoting ZSL.

6.2. Evaluating ZSL Model Explanation

Argumentative machine learning explanation by additional domain or background knowledge has been widely investigated [68, 69, 70]. Our ZSL resources with rich external knowledge can also be used to investigate explainable ZSL methods and evaluate ZSL methods' interpretation and prediction justification. For demonstration, we use different external knowledge settings that can be made by our resources to evaluate a knowledge augmented ZSL explanation method named X-ZSL which explains the transferability of sample features in zero-shot image classification in a human understandable manner [43]. Briefly, X-ZSL first uses an Attentive Graph Neural Network to automatically learn seen classes that are important to the feature learning of an unseen class, then explains the feature transfer between them by extracting class knowledge from KGs, and finally uses some templates to generate human understandable natural language explanations.

Fig. 3 presents and compares X-ZSL's explanations using different external knowledge. We give two examples, each of which includes one unseen class and one seen class that contributes to this unseen class, from AwA and ImNet-A — two benchmarks for ZS-IMGC. As we can see, for the feature transferability between seen and unseen classes, the knowledge from class hierarchy provides overall explanations, from the perspective of their relatedness in biology; the knowledge from class attributes provides detailed explanations, from the perspective of their relatedness in characteristics especially in visual characteristics; while the relational facts from ConceptNet provide an important supplement. In summary, different semantics in our resources all can have positive contributions to explain a ZSL model or to justify a ZSL prediction, and thus more explanation methods with different manners can be investigated and compared with using our resources.

7. Discussion and Outlook

7.1. ZSL Methods

For all the knowledge settings across all the tasks, we compared two general ZSL methods — DeViSE and OntoZSL. According to the results, we find that the generation-based method OntoZSL always has superior performance than the mapping-based method DeViSE no matter in the standard ZSL setting or in the generalized ZSL setting, especially in ZS-IMGC and ZS-RE. For example, on AwA in ZS-IMGC, DeViSE achieves average 35.89% *acc* and average 6.25% H across all the knowledge settings, whereas the average *acc* and H values of OntoZSL are 49.33% and 44.43%, respectively. This may due to the hubness problem [71] that exists in the label searching of DeViSE. That is, since DeViSE maps a number of sample features to a point in the class embedding space for a certain class, during prediction, this manner will increase the probability of irrelevant points (hubs) being the nearest neighbors (i.e., the matched classes or relations). In contrast, OntoZSL takes a different strategy which generates a number of unseen

	Unseen Classes **horse**	**Contributing Seen Classes** **zebra**	**Unseen Class** **roseate spoonbill**	**Contributing Seen Classes** **spoonbill**
Image				
General Explanations	The prediction for samples of **horse** is supported by **zebra**.		The prediction for samples of **roseate spoonbill** is supported by **spoonbill**.	
Knowledge from Class Hierarchy	**Triples:** (horse, *rdfs:subClassOf*, equine), (zebra, *rdfs:subClassOf*, equine) **Generated Explanations:** Horse and zebra both belong to equine.		**Triples:** (roseate_spoonbill, *rdfs:subClassOf*, spoonbill) **Generated Explanations:** Roseate spoonbill is a member of spoonbill	
Knowledge from Class Attributes	**Triples:** (horse, *AwA:hasBodyPart*, long leg), (zebra, *AwA:hasBodyPart*, long leg), (horse, *AwA:hasBodyPart*, tail), (zebra, *AwA:hasBodyPart*, tail), (horse, *AwA:hasTexture*, furry), (zebra, *AwA:hasTexture*, furry), (horse, *AwA:hasBehavior*, group), (zebra, *AwA:hasBehavior*, group), … **Generated Explanations:** They both have long leg and tail, are both furry, and both like to live in a group, …		**Triples:** (roseate_spoonbill, *ImNet-A:hasBodyPart*, long thin legs), (spoonbill, *ImNet-A:hasBodyPart*, long thin legs), (roseate_spoonbill, *ImNet-A:hasBill*, white long flat and wider bill), (spoonbill, *ImNet-A:hasBill*, long flat and wider bill), (roseate_spoonbill, *ImNet-A:hasHabitat*, water), (spoonbill, *ImNet-A:hasHabitat*, water), … **Generated Explanations:** They both have long thin legs, are similar in bill, and both live in water, ….	
Knowledge from ConceptNet	**Triples:** (cn:horse, *rdfs:subClassOf*, cn:herd_animal), (cn:zebra, *rdfs:subClassOf*, cn:herd_animal), (cn:horse, *cn:hasA*, cn:eyes), (cn:zebra, *cn:hasA*, cn:eyes) **Generated Explanations:** They are both herd animals, and both have eyes.		**Triples:** (cn:roseate_spoonbill, *rdfs:subClassOf*, cn:bird), (cn:spoonbill, *rdfs:subClassOf*, cn:bird) **Generated Explanations:** They are both a type of bird.	

Figure 3: Examples for explaining why features of seen classes *zebra* and *spoonbill* are transferred to unseen classes *horse* and *roseate spoonbill*, respectively. Note we have replaced human unreadable entity ids by entity names.

samples (features) so that classifiers can be trained to classify the unseen testing samples. However, in the ZS-KGC task, DeViSE performs better than OntoZSL on some metrics. This is because we conduct an inverse mapping in the ZS-KGC case, i.e., mapping one semantic representation to a number of sample features for a relation, as the relation semantics have higher dimensions. This can suppress the hubness problem to some extent.

In particular, we also find that in the generalized ZSL, the performance gap between DeViSE and OntoZSL is larger than in the standard ZSL. This is because the mapping-based methods are trained only by the samples of seen classes, and thus have a strong bias towards seen classes during the prediction of generalized ZSL, while for the generation-based methods, they convert the ZSL problem to a standard supervised learning problem and thus the bias toward unseen classes in prediction is avoided.

Although OntoZSL usually achieves better performance, there is an issue that can not be overlooked – OntoZSL contains more parameters compared with DeViSE, and thus it takes much more effort to search for appropriate parameters and hyper parameters such as the dimension of random noise vectors and the number of synthesized samples.

Apart from these two ZSL methods that are applicable to all the tasks and all the external knowledge settings, we also evaluated a ZSL method of the propagation-based paradigm named GCNZ in ZS-IMGC to exploit the single relation graph of the KG. In addition to GCN, other multi-relation graph neural networks such as R-GCN [63] and CompGCN [64] can also been applied and evaluated. Actually, methods of the propagation-based paradigm have been rarely applied and compared in tasks beyond ZS-IMGC, such as ZS-RE and ZS-KGC. Our resources make this possible, and we regard this as an important future work for the ZSL community. There are also some recent works leveraging pre-trained language models (PLMs) such as BERT [72] to tackle the zero-shot problems in ZS-RE [49] and ZS-KGC [73, 74]. Coupled with semantics from large amount of free text data, these models can easily generalize to unseen elements with the textual external knowledge. Recently, there are also a variety of studies trying to integrate structured knowledge such as KGs into the current language models to leverage both structured and unstructured semantics, this provides us an opportunity to combine the ZSL methods of the PLM-based paradigm with our constructed KG resources. Moreover, impressive multi-modal pre-training techniques can also be experimented for the ZS-IMGC tasks.

Reviewing all the experiment results, it can also be observed that although our proposed external knowledge settings achieve great balance between acc_s and acc_u metrics in the generalized ZSL, they do not always work well on acc_s. This motivates us to explore more robust ZSL methods to maintain the high prediction accuracy on seen labels in the generalized ZSL setting.

7.2. External Knowledge

In Section 6, we evaluated different external knowledge settings including none KG-based ones such as **w2v** in ZS-IMGC and ZS-RE and KG-based ones with varying semantics. We find KG-based settings always achieve better prediction performance, especially in ZS-RE and ZS-KGC where KG-based external knowledge have been rarely studied. Moreover, KG settings with richer semantics usually have better performance. For example, **Basic KG+literals** has better performance than **Basic KG** in ZS-IMGC, and **Rule** has better performance than **KG** in ZS-RE. All these validate our motivations of investigating various external knowledge via KGs.

Although promising performance has been achieved, there are still some open problems w.r.t. utilizing the external knowledge. First, more advanced semantic embedding techniques are required to jointly embed different kinds of KG semantics. We adopted some simple semantic embedding techniques, such as TransE, and developed some new techniques for text-aware and logic-aware KG embedding.

Although they are quite effective when combined with OntoZSL and DeViSE, more advanced methods could be developed for better performance and for addressing some knowledge settings that the current embedding methods cannot address, such as **Basic KG+CN** and **Basic KG+logics** of ZS-IMGC and **OWL** of ZS-KGC. We expect that some more complicated semantic embedding methods, such as multi-relation graph embedding [63, 64], ontology embedding [33] and multi-modal KG embedding [32], can be evaluated in combination with different ZSL methods.

Second, more adaptive solutions are required for some specific KG-based knowledge settings, besides the current pipeline of first embedding the KG and then applying the class embedding in an existing ZSL method. This is motivated by the observation that sometimes some specific knowledge settings have different performance on different datasets; for example, **RDFS-hie** has better performance than **RDFS-cons** on NELL-ZS, but has worse performance than **RDFS-cons** on Wiki-ZS. It seems that different datasets have different knowledge preferences, and it is necessary to take the properties of datasets into account when utilizing the same external knowledge. One promising solution is first automatically retrieving dataset-relevant knowledge from the KG and then utilizing these knowledge in ZSL. Some learnable knowledge retrieval solutions could be considered in the future [75].

Logical expressions are all considered in three ZSL tasks, i.e., disjointness axioms in ZS-IMGC, logic rules in ZS-RE and OWL axioms in ZS-KGC. Their incorporation in different ZSL datasets and different ZSL methods also needs more adpative methods. Compared with ZS-IMGC, ZS-RE and ZS-KGC seem to benefit more from these logic expressions. This may be because the latter two are both symbolic inference tasks, i.e., inferring the semantic relations between two entity mentions and inferring the object entities given the subjects and relations. For the ZS-IMGC task, there exists a gap between the symbolic class knowledge and class instances (i.e., images), and it is more challenging to utilize the logic expressions.

Third, high quality and large scale KG-based resources covering more tasks and more knowledge settings are required. We plan to continuously extend our resources in the following aspects: *i)* more external knowledge will be added for the existing ZSL tasks, such as more logical expressions for ImNet-A/O and AwA; *ii)* resources of more tasks, such as zero-shot visual question answering that we are investigating [76], will be added; *iii)* more ZSL settings, such as ZSL with incremental unseen classes [77], will be considered; and *iv)* high quality documents and Python libraries will be made and added for easier access of the existing and new resources.

8. Conclusion

External knowledge plays a critical role in ZSL, while KGs have shown their great superiority for representing different kinds of external knowledge for augmenting ZSL. To address the issue of semantic insufficiency in existing ZSL resources and the issue of lacking standard benchmarks to investigate and fairly compare KG-based ZSL methods, we created systematic resources for KG-based research on ZS-IMGC, ZS-RE and ZS-KGC, including six standard ZSL datasets and their corresponding KGs that can support different settings with ranging semantics. For ZS-IMGC, we integrate not only typical external knowledge such as class hierarchy, attributes and text, but also common sense relational facts from ConceptNet and some logical expressions such as class disjointness. For ZS-RE, we contributed KGs equipped with logic rules as the external knowledge. For ZS-KGC, we build ontological schemas with semantics defined by RDFS and OWL, such as relation hierarchy, relation's domain and range, concepts, relation characteristics, and relation and concept textual meta data, for a NELL KG and a Wikidata KG that are to be completed. Based on these resources, we conducted an extensive benchmarking study on different ZSL methods under different external knowledge settings, which illustrate the effectiveness and great potential usage of our proposed resources. We also discussed the strongness and weakness of different KG-based methods of different paradigms, and analyzed potential solutions for addressing some specific knowledge settings, for adaption to different datasets across tasks, and for better exploiting different kinds of external knowledge.

References

[1] W. Wang, V. W. Zheng, H. Yu, C. Miao, A survey of zero-shot learning: settings, methods, and applications, ACM Transactions on Intelligent Systems and Technology (TIST) 10 (2) (2019) 1–37.

[2] Y. Xian, C. H. Lampert, B. Schiele, Z. Akata, Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly, IEEE Transactions on Pattern Analysis and Machine Intelligence 41 (9) (2018) 2251–2265.

[3] J. Chen, Y. Geng, Z. Chen, I. Horrocks, J. Z. Pan, H. Chen, Knowledge-aware zero-shot learning: survey and perspective, in: IJCAI, ijcai.org, 2021, pp. 4366–4373.

[4] P. Qin, X. Wang, W. Chen, C. Zhang, W. Xu, W. Y. Wang, Generative adversarial zero-shot relational learning for knowledge graphs, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 8673–8680.

[5] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al., Devise: A deep visual-semantic embedding model, in: Advances in Neural Information Processing Systems, 2013, pp. 2121–2129.

[6] C. H. Lampert, H. Nickisch, S. Harmeling, Attribute-based classification for zero-shot visual object categorization, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (3) (2013) 453–465.

[7] J. Pan, G. Vetere, J. Gomez-Perez, H. Wu, Exploiting linked data and knowledge graphs for large organisations, Springer, 2016.

[8] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutierrez, J. E. L. Gayo, S. Kirrane, S. Neumaier, A. Polleres, et al., Knowledge graphs, arXiv preprint arXiv:2003.02320 (2020).

[9] L. Ehrlinger, W. Wöß, Towards a definition of knowledge graphs., SEMANTiCS (Posters, Demos, SuCCESS) 48 (1-4) (2016) 2.

[10] X. Wang, Y. Ye, A. Gupta, Zero-shot recognition via semantic embeddings and knowledge graphs, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6857–6866.

[11] M. Kampffmeyer, Y. Chen, X. Liang, H. Wang, Y. Zhang, E. P. Xing, Rethinking knowledge graph propagation for zero-shot learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 11487–11496.

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

[13] J. Gao, T. Zhang, C. Xu, I know the relationships: Zero-shot action recognition via two-stream graph convolutional networks and knowledge graphs, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 33, 2019, pp. 8303–8311.

[14] J. Zhang, P. Lertvittayakumjorn, Y. Guo, Integrating semantic knowledge to tackle zero-shot text classification, in: NAACL-HLT (1), Association for Computational Linguistics, 2019, pp. 1031–1040.

[15] N. V. Nayak, S. H. Bach, Zero-shot learning with common sense knowledge graphs, arXiv preprint arXiv:2006.10713 (2020).

[16] A. Roy, D. Ghosal, E. Cambria, N. Majumder, R. Mihalcea, S. Poria, Improving zero shot learning baselines with commonsense knowledge, arXiv preprint arXiv:2012.06236 (2020).

[17] R. Speer, J. Chin, C. Havasi, Conceptnet 5.5: An open multilingual graph of general knowledge, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 31, 2017.

[18] Y. Geng, J. Chen, Z. Chen, J. Z. Pan, Z. Ye, Z. Yuan, Y. Jia, H. Chen, OntoZSL: Ontology-enhanced zero-shot learning, in: The Web Conference (WWW), 2021.

[19] C.-W. Lee, W. Fang, C.-K. Yeh, Y.-C. F. Wang, Multi-label zero-shot learning with structured knowledge graphs, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1576–1585.

[20] J. Z. Pan, S. Staab, U. Aßmann, J. Ebert, Y. Zhao (Eds.), Ontology-driven software development, Springer, 2013.

[21] X. Zou, A survey on application of knowledge graph, in: Journal of Physics: Conference Series, Vol. 1487, IOP Publishing, 2020, p. 012016.

[22] J. Z. Pan, Resource Description Framework, in: S. Staab, R. Studer (Eds.), Handbook on ontologies, Springer, 2009, pp. 71–90.

[23] I. Horrocks, Ontologies and the semantic web, Communications of the ACM 51 (12) (2008) 58–67.

[24] F. Baader, I. Horrocks, U. Sattler, Description logics as ontology languages for the semantic web, in: Mechanizing Mathematical Reasoning, Springer, 2005, pp. 228–248.

[25] P. Hitzler, A review of the semantic web field, Communications of the ACM 64 (2) (2021) 76–83.

[26] H. Paulheim, A. Gangemi, Serving dbpedia with dolce–more than just adding a cherry on top, in: International Semantic Web Conference, Springer, 2015, pp. 180–196.

[27] R. Shearer, B. Motik, I. Horrocks, Hermit: A highly-efficient OWL reasoner, in: Proceedings of the Fifth OWLED Workshop on OWL: Experiences and Directions, collocated with the 7th International Semantic Web Conference (ISWC-2008), Karlsruhe, Germany, October 26-27, 2008, 2008.

[28] E. Thomas, J. Z. Pan, Y. Ren, TrOWL: Tractable OWL 2 reasoning infrastructure, in: ESWC, 2010.

[29] 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.

[30] Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion, in: B. Bonet, S. Koenig (Eds.), Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, pp. 2181–2187.

[31] B. Yang, W.-t. Yih, X. He, J. Gao, L. Deng, Embedding entities and relations for learning and inference in knowledge bases, arXiv preprint arXiv:1412.6575 (2014).

[32] G. A. Gesese, R. Biswas, M. Alam, H. Sack, A survey on knowledge graph embeddings with literals: Which model links better literal-ly?, Semantic Web (Preprint) (2019) 1–31.

[33] J. Chen, P. Hu, E. Jimenez-Ruiz, O. M. Holter, D. Antonyrajah, I. Horrocks, Owl2vec*: Embedding of owl ontologies, Machine Learning (2021).

[34] L. Bühmann, J. Lehmann, P. Westphal, Dl-learner—a framework for inductive learning on the semantic web, Journal of Web Semantics 39 (2016) 15–24.

[35] W. Zhang, B. Paudel, L. Wang, J. Chen, H. Zhu, W. Zhang, A. Bernstein, H. Chen, Iteratively learning embeddings and rules for knowledge graph reasoning, in: The World Wide Web Conference, 2019, pp. 2366–2377.

[36] J. Völker, M. Niepert, Statistical schema induction, in: Extended Semantic Web Conference, Springer, 2011, pp. 124–138.

[37] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, Ieee, 2009, pp. 248–255.

[38] A. Farhadi, I. Endres, D. Hoiem, D. Forsyth, Describing objects by their attributes, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 1778–1785.

[39] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, P. Perona, Caltech-ucsd birds 200 (2010).

[40] M.-E. Nilsback, A. Zisserman, Automated flower classification over a large number of classes, in: 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing, IEEE, 2008, pp. 722–729.

[41] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, J. Dean, Zero-shot learning by convex combination of semantic embeddings, arXiv preprint arXiv:1312.5650 (2013).

[42] S. Reed, Z. Akata, H. Lee, B. Schiele, Learning deep representations of fine-grained visual descriptions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 49–58.

[43] Y. Geng, J. Chen, Z. Ye, W. Zhang, H. Chen, Explainable zero-shot learning via attentive graph convolutional network and knowledge graphs, Semantic Web (2020).

[44] O. Levy, M. Seo, E. Choi, L. Zettlemoyer, Zero-shot relation extraction via reading comprehension, in: CoNLL, Association for Computational Linguistics, 2017, pp. 333–342.

[45] A. Obamuyide, A. Vlachos, Zero-shot relation classification as textual entailment, in: Proceedings of the First Workshop on Fact Extraction and VERification (FEVER), 2018, pp. 72–78.

[46] C. Lockard, P. Shiralkar, X. L. Dong, H. Hajishirzi, Zeroshotceres: Zero-shot relation extraction from semi-structured webpages, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 8105–8117.

[47] W. Imrattanatrai, M. P. Kato, M. Yoshikawa, Identifying entity properties from text with zero-shot learning, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 195–204.

[48] J. Li, R. Wang, N. Zhang, W. Zhang, F. Yang, H. Chen, Logic-guided semantic representation learning for zero-shot relation classification, arXiv preprint arXiv:2010.16068 (2020).

[49] C.-Y. Chen, C.-T. Li, Zs-bert: Towards zero-shot relation extraction with attribute representation learning, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 3470–3479.

[50] A. Rossi, D. Barbosa, D. Firmani, A. Matinata, P. Merialdo, Knowledge graph embedding for link prediction: A comparative analysis, ACM Transactions on Knowledge Discovery from Data (TKDD) 15 (2) (2021) 1–49.

[51] T. Hamaguchi, H. Oiwa, M. Shimbo, Y. Matsumoto, Knowledge transfer for out-of-knowledge-base entities : A graph neural network approach, in: IJCAI, ijcai.org, 2017, pp. 1802–1808.

[52] H. Shah, J. Villmow, A. Ulges, U. Schwanecke, F. Shafait, An open-world extension to knowledge graph completion models, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 3044–3051.

[53] K. K. Teru, E. Denis, W. L. Hamilton, Inductive relation prediction by subgraph reasoning, arXiv (2019) arXiv–1911.

[54] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[55] D. Sorokin, I. Gurevych, Context-aware representations for knowledge base relation extraction, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 1784–1789.

[56] L. A. Galárraga, C. Teflioudi, K. Hose, F. Suchanek, Amie: association rule mining under incomplete evidence in ontological knowledge bases, in: Proceedings of the 22nd international conference on World Wide Web, 2013, pp. 413–422.

[57] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672–2680.

[58] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: Neural Information Processing Systems (NIPS), 2013, pp. 1–9.

[59] S. Changpinyo, W.-L. Chao, B. Gong, F. Sha, Synthesized classifiers for zero-shot learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5327–5336.

[60] T. N. Kipf, M. Welling, Variational graph auto-encoders, arXiv preprint arXiv:1611.07308 (2016).

[61] J. Chen, L. Pan, Z. Wei, X. Wang, C.-W. Ngo, T.-S. Chua, Zero-shot ingredient recognition by multi-relational graph convolutional network, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 10542–10550.

[62] J. Wei, Y. Yang, J. Li, L. Zhu, L. Zuo, H. T. Shen, Residual graph convolutional networks for zero-shot learning, in: Proceedings of the ACM Multimedia Asia, 2019, pp. 1–6.

[63] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, M. Welling, Modeling relational data with graph convolutional networks, in: European semantic web conference, Springer, 2018, pp. 593–607.

[64] S. Vashishth, S. Sanyal, V. Nitin, P. P. Talukdar, Composition-based multi-relational graph convolutional networks, in: ICLR, OpenReview.net, 2020.

[65] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: NIPS, 2013, pp. 3111–3119.

[66] X. Han, S. Cao, X. Lv, Y. Lin, Z. Liu, M. Sun, J. Li, Openke: An open toolkit for knowledge embedding, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2018, pp. 139–144.

[67] D. Zeng, K. Liu, Y. Chen, J. Zhao, Distant supervision for relation extraction via piecewise convolutional neural networks, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 1753–1762.

[68] K. Čyras, A. Rago, E. Albini, P. Baroni, F. Toni, Argumentative xai: A survey, arXiv preprint arXiv:2105.11266 (2021).

[69] J. Chen, F. Lécué, J. Z. Pan, I. Horrocks, H. Chen, Knowledge-based transfer learning explanation, in: KR, AAAI Press, 2018, pp. 349–358.

[70] R. Confalonieri, T. Weyde, T. R. Besold, F. M. del Prado Martín, Using ontologies to enhance human understandability of global post-hoc explanations of black-box models, Artificial Intelligence 296 (2021) 103471.

[71] Y. Shigeto, I. Suzuki, K. Hara, M. Shimbo, Y. Matsumoto, Ridge regression, hubness, and zero-shot learning, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2015, pp. 135–151.

[72] 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.

[73] L. Yao, C. Mao, Y. Luo, Kg-bert: Bert for knowledge graph completion, arXiv preprint arXiv:1909.03193 (2019).

[74] B. Wang, T. Shen, G. Long, T. Zhou, Y. Wang, Y. Chang, Structure-augmented text representation learning for efficient knowledge graph completion, in: Proceedings of the Web Conference 2021, 2021, pp. 1737–1748.

[75] G. Izacard, E. Grave, Distilling knowledge from reader to retriever for question answering, in: ICLR, OpenReview.net, 2021.

[76] Z. Chen, J. Chen, Y. Geng, J. Z. Pan, Z. Yuan, H. Chen, Zero-shot visual question answering using knowledge graph, in: International Semantic Web Conference, Springer, 2021, pp. 146–162.

[77] K. Wei, C. Deng, X. Yang, D. Tao, Incremental zero-shot learning, IEEE Transactions on Cybernetics (2021).