Integrating Semantic Knowledge to Tackle Zero-shot Text Classification

Jingqing Zhang *
Data Science Institute
Imperial College London
London, UK

Piyawat Lertvittayakumjorn *
Department of Computing
Imperial College London
London, UK

Yike Guo
Data Science Institute
Imperial College London
London, UK

{jingqing.zhang15,pl1515,y.guo}@imperial.ac.uk

Abstract

Insufficient or even unavailable training data of emerging classes is a big challenge of many classification tasks, including text classification. Recognising text documents of classes that have never been seen in the learning stage, so-called *zero-shot text classification*, is therefore difficult and only limited previous works tackled this problem. In this paper, we propose a two-phase framework together with data augmentation and feature augmentation to solve this problem. Four kinds of semantic knowledge (word embeddings, class descriptions, class hierarchy, and a general knowledge graph) are incorporated into the proposed framework to deal with instances of unseen classes effectively. Experimental results show that each and the combination of the two phases clearly outperform baseline and recent approaches in classifying real-world texts under the zero-shot scenario.

1 Introduction

As one of the most fundamental problems in machine learning, automatic classification has been widely studied in several domains. However, many approaches, proven to be effective in traditional classification tasks, cannot catch up with a dynamic and open environment where new classes can emerge after the learning stage (Romera-Paredes and Torr, 2015). For example, the number of topics on social media is growing rapidly, and the classification models are required to recognise the text of the new topics using only general information (e.g., descriptions of the topics) since labelled training instances are unfeasible to obtain for each new topic (Lee et al., 2011). This scenario holds in many real-world domains such as object recognition and medical diagnosis (Xian et al., 2017; World Health Organization, 1996).

Zero-shot learning (ZSL) for text classification aims to classify documents of classes which are absent from the learning stage. Although it is challenging for a machine to achieve, humans are able to learn new concepts by transferring knowledge from known to unknown domains based on high-level descriptions and semantic representations (Thrun and Pratt, 1998). Therefore, without labelled data of unseen classes, a zero-shot learning framework is expected to exploit supportive semantic knowledge (e.g., class descriptions, relations among classes, and external domain knowledge) to generally infer the features of unseen classes using patterns learned from seen classes.

So far, three main types of semantic knowledge have been employed in general zero-shot scenarios (Fu et al., 2018). The most widely used one is semantic attributes of classes such as visual concepts (e.g., colours, shapes) and semantic properties (e.g., behaviours, functions) (Lampert et al., 2009; Zhao et al., 2018). The second type is concept ontology, including class hierarchy and knowledge graphs, which represents relationships among classes and features (Wang et al., 2018; Fergus et al., 2010). The third type is semantic word embeddings which capture implicit relationships between words thanks to a large training text corpus (Socher et al., 2013; Norouzi et al., 2013). Nonetheless, concerning ZSL in text classification particularly, there are few studies exploiting one of these knowledge types and none has considered the combinations of them (Pushp and Srivastava, 2017; Dauphin et al., 2013). Moreover, some previous works used different datasets to train and test, but there is similarity between classes in the training and testing set. For example, in (Dauphin et al., 2013), the class "imdb.com" in the training set naturally corresponds to the class "Movies" in

* Piyawat Lertvittayakumjorn and Jingqing Zhang contributed equally to this project.

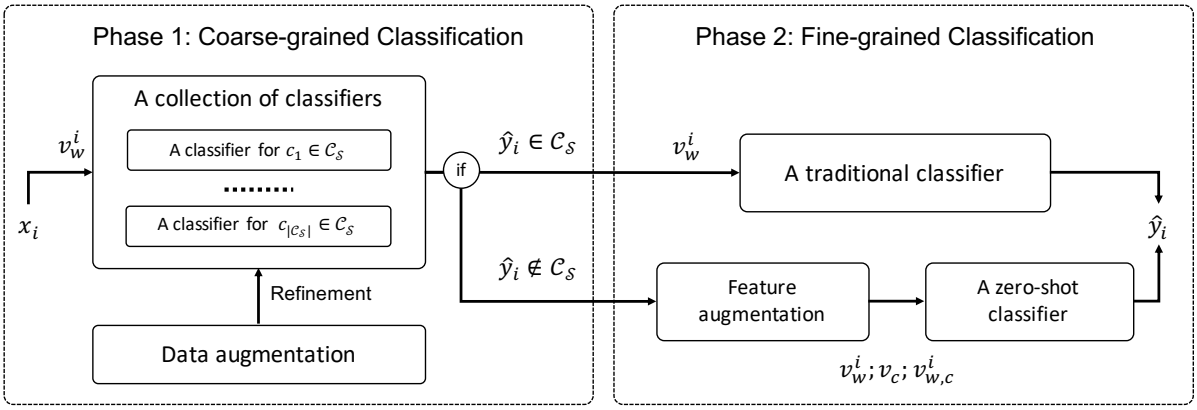

Figure 1: The overview of the proposed framework with two phases. The coarse-grained phase judges if an input document x_i comes from seen or unseen classes. The fine-grained phase finally decides the class $\hat{y}_i$. All notations are defined in section 2.1-2.2.

the testing set. Hence, these methods are not working under a strict zero-shot scenario.

To tackle the zero-shot text classification problem, this paper proposes a novel two-phase framework together with data augmentation and feature augmentation (Figure 1). In addition, four kinds of semantic knowledge including word embeddings, class descriptions, class hierarchy, and a general knowledge graph (ConceptNet) are exploited in the framework to effectively learn the unseen classes. Both of the two phases are based on convolutional neural networks (Kim, 2014). The first phase called **coarse-grained classification** judges if a document is from seen or unseen classes. Then, the second phase, named **fine-grained classification**, finally decides its class. Note that all the classifiers in this framework are trained using labelled data of seen classes (and augmented text data) only. None of the steps learns from the labelled data of unseen classes.

The contributions of our work can be summarised as follows.

- We propose a novel deep learning based two-phase framework, including coarse-grained and fine-grained classification, to tackle the zero-shot text classification problem. Unlike some previous works, our framework does not require semantic correspondence between classes in a training stage and classes in an inference stage. In other words, the seen and unseen classes can be clearly different.
- We propose a novel data augmentation technique called topic translation to strengthen the capability of our framework to detect documents from unseen classes effectively.
- We propose a method to perform feature augmentation by using integrated semantic knowledge to transfer the knowledge learned from seen to unseen classes in the zero-shot scenario.

In the remainder of this paper, we firstly explain our proposed zero-shot text classification framework in section 2. Experiments and results, which demonstrate the performance of our framework, are presented in section 3. Related works are discussed in section 4. Finally, section 5 concludes our work and mentions possible future work.

2 Methodology

2.1 Problem Formulation

Let $\mathcal{C}_S$ and $\mathcal{C}_U$ be disjoint sets of seen and unseen classes of the classification respectively. In the learning stage, a training set $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ is given where x_i is the i-th document containing a sequence of words $[w_1^i, w_2^i, \ldots, w_t^i]$ and $y_i \in \mathcal{C}_S$ is the class of x_i. In the inference stage, the goal is to predict the class of each document, $\hat{y}_i$, in a testing set which has the same data format as the training set except that y_i comes from $\mathcal{C}_S \cup \mathcal{C}_U$. Note that *(i)* every class comes with a class label and a class description (Figure 2a); *(ii)* a class hierarchy showing superclass-subclass relationships is also provided (Figure 2b); *(iii)* the documents from unseen classes cannot be observed to train the framework.

2.2 Overview and Notations

As discussed in the Introduction, our proposed classification framework consists of two phases

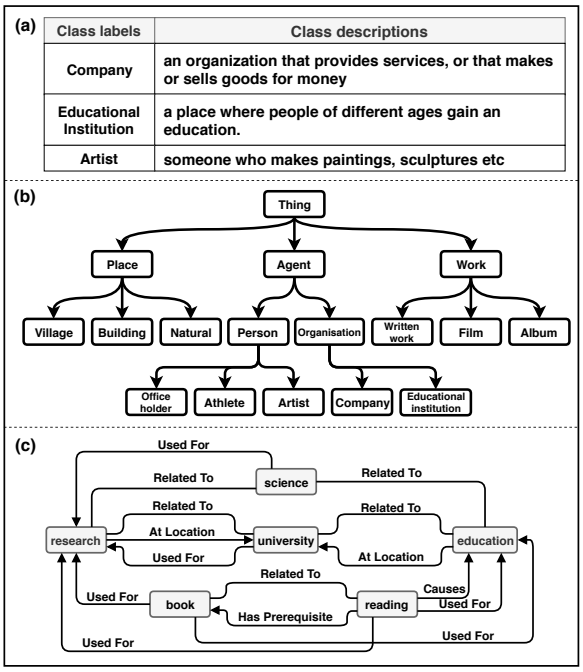

Figure 2: Illustrations of semantic knowledge integrated into our framework: (a) class labels and class descriptions (b) class hierarchy and (c) a subgraph of the general knowledge graph (ConceptNet).

(Figure 1). The first phase, coarse-grained classification, predicts whether an input document comes from seen or unseen classes. We also apply a data augmentation technique in this phase to help the classifiers be aware of the existence of unseen classes without accessing their real data. Then the second phase, fine-grained classification, finally specifies the class of the input document. It uses either a traditional classifier or a zero-shot classifier depending on the coarse-grained prediction given by Phase 1. Also, feature augmentation based on semantic knowledge is used to provide additional information which relates the document and the unseen classes to generalise the zero-shot reasoning.

We use the following notations in Figure 1 and throughout this paper.

- The list of embeddings of each word in the document x_i is denoted by $v^i_w = [v^i_{w_1}, v^i_{w_2}, \ldots, v^i_{w_t}]$.
- The embedding of each class label c is denoted by $v_c, \forall c \in \mathcal{C_S} \cup \mathcal{C_U}$. It is assumed that each class has a one-word class label. If the class label has more than one word, a similar one-word class label is provided to find v_c.
- As augmented features, the relationship vector $v^i_{w_j,c}$ shows the degree of relatedness between the word w_j and the class c according to semantic knowledge. Hence, the list of relationship vectors between each word in x_i and each class $c \in \mathcal{C_S} \cup \mathcal{C_U}$ is denoted by $v^i_{w,c} = [v^i_{w_1,c}, v^i_{w_2,c}, \ldots, v^i_{w_t,c}]$. We will explain the construction method in section 2.4.1.

2.3 Phase 1: Coarse-grained Classification

Given a document x_i, Phase 1 performs a binary classification to decide whether $\hat{y}_i \in \mathcal{C_S}$ or $\hat{y}_i \notin \mathcal{C_S}$. In this phase, each seen class $c_s \in \mathcal{C_S}$ has its own CNN classifier (with a subsequent dense layer and a sigmoid output) to predict the confidence that x_i comes from the class c_s, i.e., $p(\hat{y}_i = c_s|x_i)$. The classifier uses v^i_w as an input and it is trained using a binary cross entropy loss with all documents of its class in the training set as positive examples and the rest as negative examples.

For a test document x_i, this phase computes $p(\hat{y}_i = c_s|x_i)$ for every seen class c_s in $\mathcal{C_S}$. If there exists a class c_s such that $p(\hat{y}_i = c_s|x_i) > \tau_s$, it predicts $\hat{y}_i \in \mathcal{C_S}$; otherwise, $\hat{y}_i \notin \mathcal{C_S}$. τ_s is a classification threshold for the class c_s, calculated based on the threshold adaptation method from (Shu et al., 2017).

2.3.1 Data Augmentation

During the learning stage, the classifiers in Phase 1 use negative examples solely from seen classes, so they may not be able to differentiate the positive class from unseen classes. Hence, when the names of unseen classes are known in the inference stage, we try to introduce them to the classifiers in Phase 1 via augmented data so they can learn to reject the instances likely from unseen classes. We do data augmentation by translating a document from its original seen class to a new unseen class using analogy. We call this process *topic translation*.

In the word level, we translate a word w in a document of class c to a corresponding word w' in the context of a target class c' by solving an analogy question "c:w :: c':?". For example, solving the analogy "company:firm :: village:?" via word embeddings (Mikolov et al., 2013), we know that the word "firm" in a document of class "company" can be translated into the word "hamlet" in the context of class "village". Our framework adopts the 3COSMUL method by Levy and Goldberg (2014) to solve the analogy question and find

candidates of w':

$$w' = \underset{x \in V}{\text{argmax}} \frac{\cos(x, c') \cos(x, w)}{\cos(x, c) + \epsilon}$$

where V is a vocabulary set and $\cos(a, b)$ is a cosine similarity score between the vectors of word a and word b. Also, ϵ is a small number (i.e., 0.001) added to prevent division by zero.

In the document level, we follow Algorithm 1 to translate a document of class c into the topic of another class c'. To explain, we translate all nouns, verbs, adjectives, and adverbs in the given document to the target class, word-by-word, using the word-level analogy. The word to replace must have the same part of speech as the original word and all the replacements in one document are 1-to-1 relations, enforced by replace_dict in Algorithm 1. With this idea, we can create augmented documents for the unseen classes by topic-translation from the documents of seen classes in the training dataset. After that, we can use the augmented documents as additional negative examples for all the CNNs in Phase 1 to make them aware of the tone of unseen classes.

Algorithm 1: Document-level topic translation

Input : a document $x_i = [w_1^i, w_2^i, \ldots, w_t^i]$, an original class label c, a target class label c'
Output: a translated document x_i'

replace_dict = *dict*(); x_i' = [];
foreach $w \in x_i$ **do**
  if is_valid_pos(w) **then**
    if $w \notin$ replace_dict **then**
      cands = solve_analogy(w, c, c', top_k=20);
      for j = 0 **to** len(cands)-1 **do**
        if cands[j] $\notin$ replace_dict.values() $\wedge$ pos_of(w) $\in$ pos_list(cands[j]) **then**
          replace_dict[w] = cands[j];
          break;
      if j == len(cands) **then**
        x_i'.append(w);
        continue;
    x_i'.append(replace_dict[w]);
  else
    x_i'.append(w);
return x_i'

2.4 Phase 2: Fine-grained Classification

Phase 2 decides the most appropriate class $\hat{y}_i$ for x_i using two CNN classifiers: a traditional classifier and a zero-shot classifier as shown in Figure 1. If $\hat{y}_i \in \mathcal{C}_\mathcal{S}$ predicted by Phase 1, the traditional classifier will finally select a class $c_s \in \mathcal{C}_\mathcal{S}$ as $\hat{y}_i$. Otherwise, if $\hat{y}_i \notin \mathcal{C}_\mathcal{S}$, the zero-shot classifier will be used to select a class $c_u \in \mathcal{C}_\mathcal{U}$ as $\hat{y}_i$.

The traditional classifier and the zero-shot classifier have an identical CNN-based structure followed by two dense layers but their inputs and outputs are different. The traditional classifier is a multi-class classifier ($|\mathcal{C}_\mathcal{S}|$ classes) with a softmax output, so it requires only the word embeddings v_w^i as an input. This classifier is trained using a cross entropy loss with a training dataset whose examples are from seen classes only.

In contrast, the zero-shot classifier is a binary classifier with a sigmoid output. Specifically, it takes a text document x_i and a class c as inputs and predicts the confidence $p(\hat{y}_i = c|x_i)$. However, in practice, we utilise v_w^i to represent x_i, v_c to represent the class c, and also augmented features $v_{w,c}^i$ to provide more information on how intimate the connections between words and the class c are. Altogether, for each word w_j, the classifier receives the concatenation of three vectors (i.e., $[v_{w_j}^i; v_c; v_{w_j,c}^i]$) as an input. This classifier is trained using a binary cross entropy loss with a training data from seen classes only, but we expect this classifier to work well on unseen classes thanks to the distinctive patterns of $v_{w,c}^i$ in positive examples of every class. This is how we transfer knowledge from seen to unseen classes in ZSL.

2.4.1 Feature Augmentation

The relationship vector $v_{w_j,c}$ contains augmented features we input to the zero-shot classifier. $v_{w_j,c}$ shows how the word w_j and the class c are related considering the relations in a general knowledge graph. In this work, we use ConceptNet providing general knowledge of natural language words and phrases (Speer and Havasi, 2013). A subgraph of ConceptNet is shown in Figure 2c as an illustration. Nodes in ConceptNet are words or phrases, while edges connecting two nodes show how they are related either syntactically or semantically.

We firstly represent a class c as three sets of nodes in ConceptNet by processing the class hierarchy, class label, and class description of c. (1) ***the_class_nodes*** is a set of nodes of the class label c and any tokens inside c if c has more than one word. (2) ***superclass_nodes*** is a set of nodes of all the superclasses of c according to the class hierarchy. (3) ***description_nodes*** is a set of nodes of all nouns in the description of the class c. For example, if c is the class "Educational Institution", according to Figure 2a-2b, the three sets of Con-

ceptNet nodes for this class are:
(1) educational_institution, educational, institution
(2) organization, agent
(3) place, people, ages, education.

To construct $v_{w_j,c}$, we consider whether the word w_j is connected to the members of the three sets above within K hops by particular types of relations or not[1]. For each of the three sets, we construct a vector with $3K+1$ dimensions.

- $v[0] = 1$ if w_j is a node in that set; otherwise, $v[0] = 0$.
- for $k = 0, \ldots, K-1$:
  - $v[3k+1] = 1$ if there is a node in the set whose shortest path to w_j is $k+1$. Otherwise, $v[3k+1] = 0$.
  - $v[3k+2]$ is the number of nodes in the set whose shortest path to w_j is $k+1$.
  - $v[3k+3]$ is $v[3k+2]$ divided by the total number of nodes in the set.

Thus, the vector associated to each set shows how w_j is semantically close to that set. Finally, we concatenate the constructed vectors from the three sets to become $v_{w_j,c}$ with $3\times(3K+1)$ dimensions.

3 Experiments

3.1 Datasets

We used two textual datasets for the experiments. The vocabulary size of each dataset was limited by 20,000 most frequent words and all numbers were excluded. (1) **DBpedia** ontology dataset (Zhang et al., 2015) includes 14 non-overlapping classes and textual data collected from Wikipedia. Each class has 40,000 training and 5,000 testing samples. (2) The **20newsgroups** dataset [2] has 20 topics each of which has approximately 1,000 documents. 70% of the documents of each class were randomly selected for training, and the remaining 30% were used as a testing set.

3.2 Implementation Details [3]

In our experiments, two different rates of unseen classes, 50% and 25%, were chosen and the corresponding sizes of $\mathcal{C}_\mathcal{S}$ and $\mathcal{C}_\mathcal{U}$ are shown in Table 1. For each dataset and each unseen rate, the random selection of ($\mathcal{C}_\mathcal{S}$, $\mathcal{C}_\mathcal{U}$) were repeated ten times and these ten groups were used by all the experiments with this setting for a fair comparison. All documents from $\mathcal{C}_\mathcal{U}$ were removed from the training set accordingly. Finally, the results from all the ten groups were averaged.

In Phase 1, the structure of each classifier was identical. The CNN layer had three filter sizes [3, 4, 5] with 400 filters for each filter size and the subsequent dense layer had 300 units. For data augmentation, we used gensim with an implementation of 3CosMul (Řehůřek and Sojka, 2010) to solve the word-level analogy (line 5 in Algorithm 1). Also, the numbers of augmented text documents per unseen class for every setting (if used) are indicated in Table 1. These numbers were set empirically considering the number of available training documents to be translated.

In Phase 2, the traditional classifier and the zero-shot classifier had the same structure, in which the CNN layer had three filter sizes [2, 4, 8] with 600 filters for each filter size and the two intermediate dense layers had 400 and 100 units respectively. For feature augmentation, the maximum path length K in ConceptNet was set to 3 to create the relationship vectors[4]. The DBpedia ontology[5] was used to construct a class hierarchy of the DBpedia dataset. The class hierarchy of the 20newsgroups dataset was constructed based on the namespaces initially provided by the dataset. Meanwhile, the classes descriptions of both datasets were picked from Macmillan Dictionary[6] as appropriate.

For both phases, we used 200-dim GloVe vectors[7] for word embeddings v_w and v_c (Pennington et al., 2014). All the deep neural networks were implemented with TensorLayer (Dong et al., 2017a) and TensorFlow (Abadi et al., 2016).

Table 1: The rates of unseen classes and the numbers of augmented documents (per unseen class) in the experiments

Dataset	Unseen rate	$\lvert\mathcal{C}_\mathcal{S}\rvert$	$\lvert\mathcal{C}_\mathcal{U}\rvert$	#Augmented docs per c_u
DBpedia (14 classes)	25%	11	3	12,000
	50%	7	7	8,000
20news (20 classes)	25%	15	5	4,000
	50%	10	10	3,000

[1] In this paper, we only consider the most common types of positive relations which are *RelatedTo*, *IsA*, *PartOf*, and *AtLocation*. They cover ∼60% of all edges in ConceptNet.

[2] http://qwone.com/∼jason/20Newsgroups/

[3] Code: https://github.com/JingqingZ/KG4ZeroShotText.

[4] Based on our observation, most of the related words stay within 3 hops from the class nodes in ConceptNet.

[5] http://mappings.dbpedia.org/server/ontology/classes/

[6] https://www.macmillandictionary.com/

[7] glove6B.zip in https://nlp.stanford.edu/projects/glove/

3.3 Baselines and Evaluation Metrics

We compared each phase and the overall framework with the following approaches and settings.

Phase 1: Proposed by (Shu et al., 2017), **DOC** is a state-of-the-art open-world text classification approach which classifies a new sample into a seen class or "reject" if the sample does not belong to any seen classes. The DOC uses a single CNN and a 1-vs-rest sigmoid output layer with threshold adjustment. Unlike DOC, the classifiers in the proposed Phase 1 work individually. However, for a fair comparison, we used DOC only as a binary classifier in this phase ($\hat{y}_i \in \mathcal{C}_\mathcal{S}$ or $\hat{y}_i \notin \mathcal{C}_\mathcal{S}$).

Phase 2: To see how well the augmented feature $v_{w,c}$ work in ZSL, we ran the zero-shot classifier with **different combinations of inputs**. Particularly, five combinations of v_w, v_c, and $v_{w,c}$ were tested with documents from unseen classes only (traditional ZSL).

The whole framework: (1) **Count-based model** selected the class whose label appears most frequently in the document as $\hat{y}_i$. (2) **Label similarity** (Sappadla et al., 2016) is an unsupervised approach which calculates the cosine similarity between the sum of word embeddings of each class label and the sum of word embeddings of every n-gram ($n = 1, 2, 3$) in the document. We adopted this approach to do single-label classification by predicting the class that got the highest similarity score among all classes. (3) **RNN AutoEncoder** was built based on a Seq2Seq model with LSTM (512 hidden units), and it was trained to encode documents and class labels onto the same latent space. The cosine similarity was applied to select a class label closest to the document on the latent space. (4) **RNN+FC** refers to the architecture 2 proposed in (Pushp and Srivastava, 2017). It used an RNN layer with LSTM (512 hidden units) followed by two dense layers with 400 and 100 units respectively. (5) **CNN+FC** replaced the RNN in the previous model with a CNN, which has the identical structure as the zero-shot classifier in Phase 2. Both RNN+FC and CNN+FC predicted the confidence $p(\hat{y}_i = c|x_i)$ given v_w and v_c. The class with the highest confidence was selected as $\hat{y}_i$.

For Phase 1, we used the accuracy for binary classification ($y, \hat{y}_i \in \mathcal{C}_\mathcal{S}$ or $y, \hat{y}_i \notin \mathcal{C}_\mathcal{S}$) as an evaluation metric. In contrast, for Phase 2 and the whole framework, we used the multi-class classification accuracy ($\hat{y}_i = y_i$) as a metric.

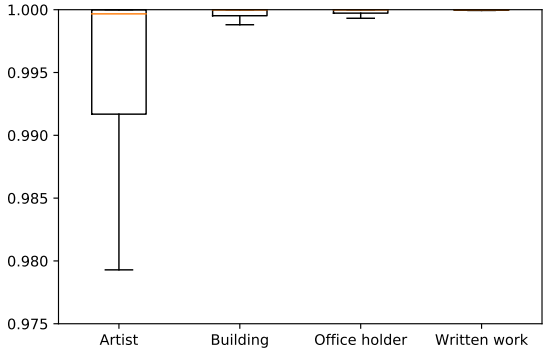

Figure 3: The distributions of confidence scores of positive examples from four seen classes of DBpedia in Phase 1.

3.4 Results and Discussion

The evaluation of Phase 1 (coarse-grained classification) checks if each x_i was correctly delivered to the right classifier in Phase 2. Table 3 shows the performance of Phase 1 with and without augmented data compared with DOC. Considering test documents from seen classes only, our framework outperformed DOC on both datasets. In addition, the augmented data improved the accuracy of detecting documents from unseen classes clearly and led to higher overall accuracy in every setting. Despite no real labelled data from unseen classes, the augmented data generated by topic translation helped Phase 1 better detect documents from unseen classes. Table 4 shows some examples of augmented data from the DBpedia dataset. Even if they are not completely understandable, they contain the tone of the target classes.

Although Phase 1 provided confidence scores for all seen classes, we could not use them to predict $\hat{y}_i$ directly since the distribution of scores of positive examples from different CNNs are different. Figure 3 shows that the distribution of confidence scores of the class "Artist" had a noticeably larger variance and was clearly different from the class "Building". Hence, even if $p(\hat{y}_i =$ "Building"$|x_i) > p(\hat{y}_i =$ "Artist"$|x_i)$, we cannot conclude that x_i is more likely to come from the class "Building". This is why **a traditional classifier in Phase 2 is necessary**.

Regarding Phase 2, fine-grained classification is in charge of predicting $\hat{y}_i$ and it employs two classifiers which were tested separately. Assuming Phase 1 is perfect, the classifiers in Phase 2 should be able to find the right class. The purpose of Table 5 is to show that **the traditional CNN classifier in Phase 2** was highly accurate.

Table 2: The accuracy of the whole framework compared with the baselines.

Dataset	Unseen rate	y_i	Count-based	Label Similarity (Sappadla et al., 2016)	RNN Autoencoder	RNN + FC (Pushp and Srivastava, 2017)	CNN + FC	Ours
DBpedia	25%	seen	0.322	0.377	0.250	0.895	**0.985**	0.977
		unseen	0.372	0.426	0.267	0.046	0.204	**0.540**
		overall	0.334	0.386	0.254	0.713	0.818	**0.883**
	50%	seen	0.358	0.401	0.202	0.960	**0.991**	0.984
		unseen	0.304	**0.369**	0.259	0.044	0.069	0.267
		overall	0.333	0.386	0.230	0.502	0.530	**0.625**
20news	25%	seen	0.205	0.279	0.263	0.614	**0.792**	0.782
		unseen	0.201	0.287	0.149	0.065	0.134	**0.360**
		overall	0.204	0.280	0.236	0.482	0.633	**0.681**
	50%	seen	0.219	0.293	0.275	0.709	0.684	**0.785**
		unseen	0.196	**0.266**	0.126	0.052	0.126	0.208
		overall	0.207	0.280	0.200	0.381	0.405	**0.497**

Table 3: The accuracy of Phase 1 with and without augmented data compared with DOC .

Dataset Unseen rate	y_i	DOC	Ours w/o aug.	Ours w/ aug.
DBpedia 25%	seen	0.980	**0.982**	**0.982**
	unseen	0.471	0.388	**0.536**
	overall	0.871	0.855	**0.886**
DBpedia 50%	seen	0.983	0.986	**0.987**
	unseen	0.384	0.345	**0.512**
	overall	0.684	0.666	**0.749**
20news 25%	seen	0.800	**0.838**	0.831
	unseen	0.573	0.431	**0.577**
	overall	0.745	0.754	**0.770**
20news 50%	seen	0.824	**0.856**	0.843
	unseen	0.562	0.419	**0.603**
	overall	0.694	0.639	**0.724**

Table 4: Examples of augmented data translated from a document of the original class "Animal" into two target classes "Plant" and "Athlete".

Animal (Original)	Mitra perdulca is a species of sea snail a marine gastropod mollusk in the family Mitridae the miters or miter snails.
Animal → Plant	Arecaceae perdulca is a flowering of port aster a naval mollusk gastropod in the fabaceae Clusiaceae the tiliaceae or rockery amaryllis.
Animal → Athlete	Mira perdulca is a swimmer of sailing sprinter an Olympian limpets gastropod in the basketball Middy the miters or miter skater.

Besides, given test documents from unseen classes only, the performance of **the zero-shot classifier in Phase 2** is shown in Table 6. Based on the construction method, $v_{w,c}$ quantified the relatedness between words and the class but, unlike v_w and v_c, it did not include detailed semantic meaning. Thus, the classifier using $v_{w,c}$ only could not find out the correct unseen class and neither $[v_w; v_{w,c}]$ and $[v_c; v_{w,c}]$ could do. On the other hand, the combination of $[v_w; v_c]$, which included semantic embeddings of both words and the class label, increased the accuracy of predicting unseen classes clearly. However, the zero-shot classifier fed by the combination of all three types of inputs $[v_w; v_c; v_{w,c}]$ achieved the highest accuracy in all settings. It asserts that the integration of semantic knowledge we proposed is an effective means for knowledge transfer from seen to unseen classes in the zero-shot scenario.

Table 5: The accuracy of the traditional classifier in Phase 2 given documents from seen classes only.

Dataset	DBpedia		20news	
Input \ Unseen rate	50%	25%	50%	25%
v_w	0.993	0.990	0.884	0.897

Table 6: The accuracy of the zero-shot classifier in Phase 2 given documents from unseen classes only.

Dataset	DBpedia		20news	
Inputs \ Unseen rate	50%	25%	50%	25%
Random guess	0.143	0.333	0.100	0.200
$v_{w,c}$	0.154	0.443	0.104	0.210
$[v_c; v_{w,c}]$	0.163	0.400	0.099	0.215
$[v_w; v_{w,c}]$	0.266	0.460	0.122	0.307
$[v_w; v_c]$	0.381	0.711	0.274	0.431
$[v_w; v_c; v_{w,c}]$	**0.418**	**0.754**	**0.302**	**0.500**

Last but most importantly, we compared **the whole framework** with four baselines as shown in Table 2. First, the count-based model is a rule-based model so it failed to predict documents from seen classes accurately and resulted in unpleasant overall results. This was similar to the label similarity approach even though it had higher degree of flexibility. Next, the RNN Autoencoder was trained without any supervision since $\hat{y}_i$ was predicted based on the cosine similarity. We believe

the implicit semantic relatedness between classes caused the failure of the RNN Autoencoder. Besides, the CNN+FC and RNN+FC had same inputs and outputs and it was clear that CNN+FC performed better than RNN+FC in the experiment. However, neither CNN+FC nor RNN+FC was able to transfer the knowledge learned from seen to unseen classes. Finally, our two-phase framework can boost up the prediction accuracy on unseen classes while maintaining the accuracy on seen classes. This made it achieve the highest overall accuracy on both datasets and both unseen rates. In conclusion, by using integrated semantic knowledge, the proposed two-phase framework with data and feature augmentation is a promising step to tackle this challenging zero-shot problem.

Furthermore, another benefit of the framework is high flexibility. As the modules in Figure 1 has less coupling to one another, it is flexible to improve or customise each of them. For example, we can deploy an advanced language understanding model, e.g., BERT (Devlin et al., 2018), as a traditional classifier. Moreover, we may replace ConceptNet with a domain-specific knowledge graph to deal with medical texts.

4 Related Work

4.1 Zero-shot Text Classification

There are a few more related works to discuss besides recent approaches we compared with in the experiments (explained in section 3.3). Dauphin et al. (2013) predicted semantic utterance of texts by mapping class labels and text samples into the same semantic space and classifying each sample to the closest class label. Nam et al. (2016) learned the embeddings of classes, documents, and words jointly in the learning stage. Hence, it can perform well in domain-specific classification, but this is possible only with a large amount of training data. Overall, most of the previous works exploited semantic relationships between classes and documents via embeddings. In contrast, our proposed framework leverages not only the word embeddings but also other semantic knowledge. While word embeddings are used to solve analogy for data augmentation in Phase 1, the other semantic knowledge sources (in Figure 2) are integrated into relationship vectors and used as augmented features in Phase 2. Furthermore, our framework does not require any semantic correspondences between seen and unseen classes.

4.2 Data Augmentation in NLP

In the face of insufficient data, data augmentation has been widely used to improve generalisation of deep neural networks especially in computer vision (Krizhevsky et al., 2012) and multimodality (Dong et al., 2017b), but it is still not a common practice in natural language processing. Recent works have explored data augmentation in NLP tasks such as machine translation and text classification (Saito et al., 2017; Fadaee et al., 2017; Kobayashi, 2018), and the algorithms were designed to preserve semantic meaning of an original document by using synonyms (Zhang and LeCun, 2015) or adding noises (Xie et al., 2017), for example. In contrast, our proposed data augmentation technique translates a document from one meaning (its original class) to another meaning (an unseen class) by analogy in order to substitute unavailable labelled data of the unseen class.

4.3 Feature Augmentation in NLP

Apart from improving classification accuracy, feature augmentation is also used in domain adaptation to transfer knowledge between a source and a target domain (Pan et al., 2010b; Fang and Chiang, 2018; Chen et al., 2018). An early research paper applying feature augmentation in NLP is Daume III (2007) which targeted domain adaptation on sequence labelling tasks. After that, feature augmentation was used in several NLP tasks such as cross-domain sentiment classification (Pan et al., 2010a), multi-domain machine translation (Clark et al., 2012), semantic argument classification (Batubara et al., 2018), etc. Our work is different from previous works not only that we applied this technique to zero-shot text classification but also that we integrated many types of semantic knowledge to create the augmented features.

5 Conclusion and Future Work

To tackle zero-shot text classification, we proposed a novel CNN-based two-phase framework together with data augmentation and feature augmentation. The experiments show that data augmentation by topic translation improved the accuracy in detecting instances from unseen classes, while feature augmentation enabled knowledge transfer from seen to unseen classes for zero-shot learning. Thanks to the framework and the integrated semantic knowledge, our work outperformed all the existing methods in terms of over-

all accuracy. In the future, we plan to extend our framework to do multi-label classification with a larger amount of data, and also study how semantic units defined by linguists can be used in the zero-shot scenario.

Acknowledgments

We would like to thank Douglas McIlwraith, Nontawat Charoenphakdee, and three anonymous reviewers for helpful suggestions. Jingqing and Piyawat would also like to thank the support from the LexisNexis® Risk Solutions HPCC Systems® academic program and Anandamahidol Foundation, respectively.

References

Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283.

Dina Khaira Batubara, Moch Arif Bijaksana, and Adiwijaya. 2018. On feature augmentation for semantic argument classification of the quran english translation using support vector machine. *Journal of Physics: Conference Series*, 971(1):012043.

Zitian Chen, Yanwei Fu, Yinda Zhang, Yu-Gang Jiang, Xiangyang Xue, and Leonid Sigal. 2018. Semantic feature augmentation in few-shot learning. *CoRR*, abs/1804.05298.

Jonathan H Clark, Alon Lavie, and Chris Dyer. 2012. One system, many domains: Open-domain statistical machine translation via feature augmentation. In *Proceedings of the Conference of the Association for Machine Translation in the Americas*.

Hal Daume III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263. Association for Computational Linguistics.

Yann N Dauphin, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck. 2013. Zero-shot learning for semantic utterance classification. *arXiv preprint arXiv:1401.0509*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Hao Dong, Akara Supratak, Luo Mai, Fangde Liu, Axel Oehmichen, Simiao Yu, and Yike Guo. 2017a. Tensorlayer: a versatile library for efficient deep learning development. In *Proceedings of the 2017 ACM on Multimedia Conference*, pages 1201–1204. ACM.

Hao Dong, Jingqing Zhang, Douglas McIlwraith, and Yike Guo. 2017b. I2t2i: Learning text to image synthesis with textual data augmentation. In *Image Processing (ICIP), 2017 IEEE International Conference on*, pages 2015–2019. IEEE.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. *arXiv preprint arXiv:1705.00440*.

Wen-Chieh Fang and Yi-Ting Chiang. 2018. A discriminative feature mapping approach to heterogeneous domain adaptation. *Pattern Recognition Letters*, 106:13–19.

Rob Fergus, Hector Bernal, Yair Weiss, and Antonio Torralba. 2010. Semantic label sharing for learning with many categories. In *Computer Vision – ECCV 2010*, pages 762–775, Berlin, Heidelberg. Springer Berlin Heidelberg.

Y. Fu, T. Xiang, Y. Jiang, X. Xue, L. Sigal, and S. Gong. 2018. Recent advances in zero-shot recognition: Toward data-efficient understanding of visual content. *IEEE Signal Processing Magazine*, 35(1):112–125.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457. Association for Computational Linguistics.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. 2009. Learning to detect unseen object classes by between-class attribute transfer. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 951–958. IEEE.

Kathy Lee, Diana Palsetia, Ramanathan Narayanan, Md Mostofa Ali Patwary, Ankit Agrawal, and Alok Choudhary. 2011. Twitter trending topic classification. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*, pages 251–258. IEEE.

Omer Levy and Yoav Goldberg. 2014. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the eighteenth conference on computational natural language learning*, pages 171–180.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Jinseok Nam, Eneldo Loza Mencía, and Johannes Fürnkranz. 2016. All-in text: Learning document, label, and word representations jointly. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg S Corrado, and Jeffrey Dean. 2013. Zero-shot learning by convex combination of semantic embeddings. *arXiv preprint arXiv:1312.5650*.

Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. 2010a. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th international conference on World wide web*, pages 751–760. ACM.

Sinno Jialin Pan, Qiang Yang, et al. 2010b. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Pushpankar Kumar Pushp and Muktabh Mayank Srivastava. 2017. Train once, test anywhere: Zero-shot learning for text classification. *arXiv preprint arXiv:1712.05972*.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA.

Bernardino Romera-Paredes and Philip Torr. 2015. An embarrassingly simple approach to zero-shot learning. In *International Conference on Machine Learning*, pages 2152–2161.

Itsumi Saito, Jun Suzuki, Kyosuke Nishida, Kugatsu Sadamitsu, Satoshi Kobashikawa, Ryo Masumura, Yuji Matsumoto, and Junji Tomita. 2017. Improving neural text normalization with data augmentation at character-and morphological levels. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 257–262.

Prateek Veeranna Sappadla, Jinseok Nam, Eneldo Loza Mencía, and Johannes Fürnkranz. 2016. Using semantic similarity for multi-label zero-shot classification of text documents. In *Proc. European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*.

Lei Shu, Hu Xu, and Bing Liu. 2017. Doc: Deep open classification of text documents. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2911–2916. Association for Computational Linguistics.

Richard Socher, Milind Ganjoo, Christopher D. Manning, and Andrew Y. Ng. 2013. Zero-shot learning through cross-modal transfer. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'13, pages 935–943, USA. Curran Associates Inc.

Robert Speer and Catherine Havasi. 2013. Conceptnet 5: A large semantic network for relational knowledge. In *The Peoples Web Meets NLP*, pages 161–176. Springer.

Sebastian Thrun and Lorien Pratt. 1998. Learning to learn: Introduction and overview. In *Learning to learn*, pages 3–17. Springer.

Xiaolong Wang, Yufei Ye, and Abhinav Gupta. 2018. Zero-shot recognition via semantic embeddings and knowledge graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6857–6866.

World Health Organization. 1996. Infectious diseases kill over 17 million people a year. *Malaria Weekly*, 3:11–16.

Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. 2017. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *arXiv preprint arXiv:1707.00600*.

Ziang Xie, Sida I Wang, Jiwei Li, Daniel Lévy, Aiming Nie, Dan Jurafsky, and Andrew Y Ng. 2017. Data noising as smoothing in neural network language models. *arXiv preprint arXiv:1703.02573*.

Xiang Zhang and Yann LeCun. 2015. Text understanding from scratch. *arXiv preprint arXiv:1502.01710*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

Bo Zhao, Yanwei Fu, Rui Liang, Jiahong Wu, Yonggang Wang, and Yizhou Wang. 2018. A large-scale attribute dataset for zero-shot learning. *arXiv preprint arXiv:1804.04314*.