Reconstructing Capsule Networks for Zero-shot Intent Classification

Han Liu[1*] **Xiaotong Zhang**[1*] **Lu Fan**[1*] **Xuandi Fu**[1]
Qimai Li[1] **Xiao-Ming Wu**[1†] **Albert Y.S. Lam**[2]
Department of Computing, The Hong Kong Polytechnic University, Hong Kong S.A.R.[1]
Fano Labs, Hong Kong S.A.R.[2]
{cshliu,csxtzhang,cslfan}@comp.polyu.edu.hk
{csxfu,csqmli,csxmwu}@comp.polyu.edu.hk, albert@fano.ai

Abstract

Intent classification is an important building block of dialogue systems. With the burgeoning of conversational AI, existing systems are not capable of handling numerous fast-emerging intents, which motivates zero-shot intent classification. Nevertheless, research on this problem is still in the incipient stage and few methods are available. A recently proposed zero-shot intent classification method, IntentCapsNet, has been shown to achieve state-of-the-art performance. However, it has two unaddressed limitations: (1) it cannot deal with polysemy when extracting semantic capsules; (2) it hardly recognizes the utterances of unseen intents in the generalized zero-shot intent classification setting. To overcome these limitations, we propose to reconstruct capsule networks for zero-shot intent classification. First, we introduce a dimensional attention mechanism to fight against polysemy. Second, we reconstruct the transformation matrices for unseen intents by utilizing abundant latent information of the labeled utterances, which significantly improves the model generalization ability. Experimental results on two task-oriented dialogue datasets in different languages show that our proposed method outperforms IntentCapsNet and other strong baselines.

1 Introduction

With the advent of conversational AI, task-oriented spoken dialogue systems are becoming ubiquitous, e.g., chatbots deployed on different applications, or modules integrated in the popular virtual personal assistants like Apple Siri or Microsoft Cortana (Chen et al., 2017). To improve business effectiveness and user satisfaction, accurately identifying the intents behind user utterances is indispensable. However, it is extremely challenging not only because user queries are sometimes short and expressed diversely, but also because it may continuously encounter new or unacquainted intents popped up quickly from various domains. Conventional intent classification methods (Hu et al., 2009; Tur et al., 2012; Xu and Sarikaya, 2013; Ravuri and Stolcke, 2015; Liu and Lane, 2016; Nam et al., 2016) typically train a supervised learning model on large amounts of labeled data, and are not effective in recognizing emerging unseen intents.

Several zero-shot learning approaches attempted to address the challenges for classifying intents whose instances are not present during training. One common idea is to utilize some external resources (Ferreira et al., 2015a,b; Yazdani and Henderson, 2015; Kumar et al., 2017; Zhang et al., 2019) such as label ontologies or manually defined attributes. However, such external resources are usually unavailable, as they require substantial extra time and expensive human labour to produce. To implement zero-shot intent classification more easily and intelligently, recent works rely more on the word embeddings of intent labels, which can be easily pretrained on text corpus. Methods proposed by Chen et al. (2016) and Kumar et al. (2017) utilize neural networks to project intent labels and data samples to the same semantic space and then measure their similarity. However, learning a good projection function is usually difficult due to the diversity of user expressions, especially in some complex domains such as medical queries (Zhang et al., 2016).

Unlike previous models, IntentCapsNet (Xia et al., 2018) employs capsule networks to extract high-level semantic features and then transfers the prediction vectors for seen intents to unseen intents. Although IntentCapsNet has achieved impressive performance in some zero-shot intent

[*] Equal contribution.
[†] Corresponding author.

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 4799–4809,
Hong Kong, China, November 3–7, 2019.

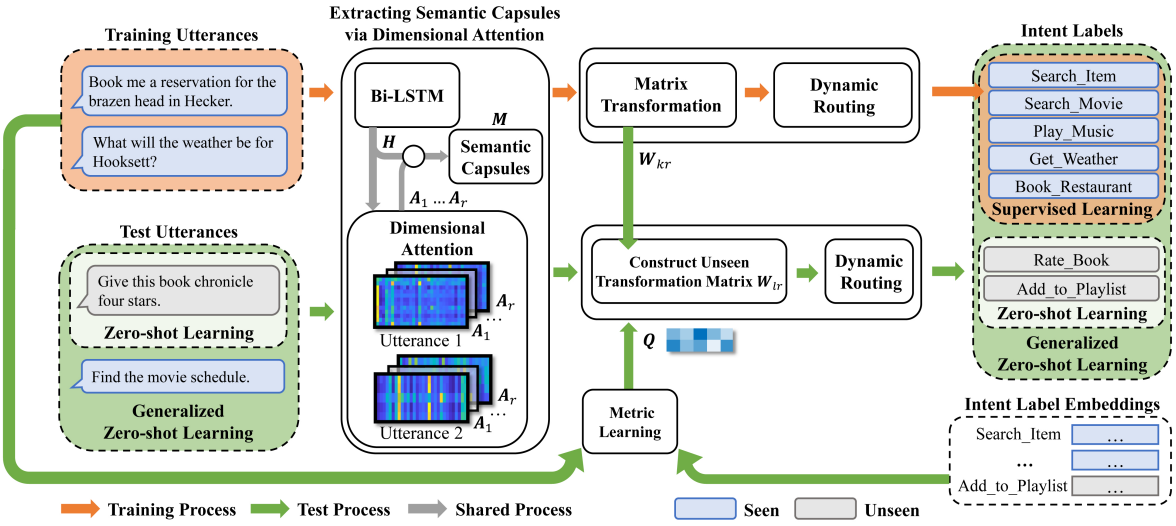

Figure 1: Illustration of our framework ReCapsNet-ZS. **In the training process**, labeled utterances are first encoded by Bi-LSTM. Then, a set of semantic capsules are extracted via the dimensional attention module. Finally, these semantic capsules are fed to a capsule network to train a model for predicting the seen intents. **In the testing process**, to predict the unseen intents, a metric learning method is trained on labeled utterances and intent label embeddings to learn the similarities between the unseen and seen intents. Then, the learned similarities and the transformation matrices for the seen intents trained by capsule networks are used to construct the transformation matrices for the unseen intents. When a test utterance arrives, it is first encoded into semantic capsules by the trained Bi-LSTM and dimensional attention module. There are two settings for intent classification. (1) Zero-shot intent classification: only utterances of the unseen intents participate in testing, so each utterance needs to be classified to one of the unseen intents. In this case, only the transformation matrices for the unseen intents are used for prediction. (2) Generalized zero-shot intent classification: test utterances may come from both the seen and unseen intents, so each utterance needs to be classified to either a seen or an unseen intent. In this case, the transformation matrices for the seen and unseen intents are all used for prediction.

classification tasks, it has two unaddressed limitations. (1) The self-attention module of IntentCapsNet cannot handle polysemy, which weakens the representation capacity of semantic capsules. (2) For the generalized zero-shot classification setting where newly arrived utterances come from both seen and unseen intents, the method of IntentCapsNet for constructing the prediction vectors can easily cause the model to completely fail in detecting unseen intents, which is clearly undesirable and inadequate for real dialogue systems.

In this paper, we propose to reconstruct capsule networks for zero-shot intent classification (ReCapsNet-ZS), which effectively addresses the limitations of IntentCapsNet and adapts well to the generalized zero-shot intent classification tasks. As illustrated in Figure 1, ReCapsNet-ZS consists of two components. First, it introduces a dimensional attention module to alleviate the polysemy problem, which helps to extract semantic features for capsule networks. Second, it computes the similarities between unseen and seen intents by utilizing the rich latent information of labeled utterances, and then constructs the transformation matrices for unseen intents with the computed similarities and the trained transformation matrices for seen intents, which greatly improves the generalization ability to unseen intents.

To verify the effectiveness of the proposed ReCapsNet-ZS for zero-shot intent classification, we conduct extensive experiments on two real task-oriented dialogue datasets in English and Chinese respectively. The empirical study validates our proposals and shows promising results of ReCapsNet-ZS, which are significantly better than state-of-the-art methods, especially on the generalized zero-shot intent classification tasks.

2 Related Works

Zero-shot Intent Classification. Zero-shot learning (Larochelle et al., 2008; Palatucci et al., 2009) aims to use the knowledge learned from seen classes, of which abundant labeled samples are typically available for training, to recognize unseen classes, of which no labeled samples are provided. It has been widely studied in computer

vision (Ba et al., 2015; Xian et al., 2016) and natural language processing (Sappadla et al., 2016; Zhang et al., 2019).

Zero-shot intent classification is an important and challenging task for many natural language understanding applications (Hu et al., 2009; Liu and Lane, 2016; Nam et al., 2016; Xu and Sarikaya, 2013), in which new intents emerge constantly and they cannot be easily recognized. Several methods have been proposed to tackle this problem. Ferreira et al. (2015a,b) and Yazdani and Henderson (2015) utilize external resources such as label ontologies or manually defined attributes to find the relationship between seen and unseen intent labels. However, the external resources are usually difficult to obtain, as collecting them is labor intensive and time consuming. Chen et al. (2016) and Kumar et al. (2017) project the utterances and intent labels to a same semantic space and then compute the similarities between utterances and intent labels (Chen et al., 2016; Kumar et al., 2017). However, diverse user expressions may make it difficult to learn a good projection function and thus affect the classification performance. Recently, Xia et al. (2018) extend capsule networks for zero-shot intent classification by transferring the prediction vectors from seen classes to unseen classes. However, there are some key issues left to be resolved, including how to deal with polysemy in word embeddings and how to improve the model generalization ability to unseen intents in the generalized zero-shot intent classification setting.

Capsule Networks. Capsule Networks (Sabour et al., 2017) were first proposed to address the shortcomings of convolutional neural networks (CNN) in the domain of computer vision. It allows the networks to learn part-whole invariant relationships consecutively. Recently, some studies have attempted to apply capsule networks in the domain of natural language processing (Yang et al., 2018; Geng et al., 2019; Xia et al., 2018) and obtained promising results. Yang et al. (2018) first extend capsule networks for text classification. Geng et al. (2019) successfully combine the dynamic routing algorithm with some meta-learning framework for few-shot text classification. However, their model still requires some labeled samples for each class. Xia et al. (2018) propose a model based on capsule networks for zero-shot intent classification and has achieved state-of-the-art performance, but as mentioned above, their model has some intrinsic limitations remained to be addressed.

3 Preliminaries

3.1 Problem Formulation

Given the set of all intent labels $Y = Y^s \bigcup Y^u$, where $Y^s = \{y_1^s, y_2^s, \ldots, y_K^s\}$ and $Y^u = \{y_1^u, y_2^u, \ldots, y_L^u\}$ are the sets of seen and unseen intent labels respectively. There is no overlap between Y^s and Y^u, i.e., $Y^s \bigcap Y^u = \emptyset$, and K and L are the numbers of seen and unseen intent labels respectively. The embeddings of the seen and unseen intent labels are denoted by $E^s = \{\boldsymbol{e}_1^s, \boldsymbol{e}_2^s, \ldots, \boldsymbol{e}_K^s\}$ and $E^u = \{\boldsymbol{e}_1^u, \boldsymbol{e}_2^u, \ldots, \boldsymbol{e}_L^u\}$ respectively. Each embedding is a d-dimensional vector. For all the seen and unseen intent labels, their associated embeddings are available. The sample (utterance) sets for the seen and unseen intent labels are denoted by $X^s = \{x_1^s, x_2^s, \ldots, x_{n_s}^s\}$ and $X^u = \{x_1^u, x_2^u, \ldots, x_{n_u}^u\}$ respectively, where n_s is the number of instances of the seen labels and n_u is the number of instances of the unseen labels.

Zero-shot Intent Classification. For this setting, the training set is $X^{tr} = \{X^s, Y^s\}$, and X^u is not available for training. In the test phase, the goal is to assign an unseen intent label $y \in Y^u$ to a given utterance.

Generalized Zero-shot Intent Classification. For this setting, the training procedure is the same as above, while the difference is in the test phase, where the goal is to assign an intent label $y \in Y^s \bigcup Y^u$ to a given utterance.

In this paper, we aim to reconstruct capsule networks for handling both of the two settings of zero-shot intent classification.

3.2 Limitations of IntentCapsNet

IntentCapsNet (Xia et al., 2018) is the first work to employ capsule networks for zero-shot intent classification. It exploits the self-attention mechanism to extract semantic features (capsules) of an utterance. For zero-shot intent classification, it utilizes the vote vectors of seen intents and the similarities between seen and unseen intents based on Euclidean distance to make predictions for unseen intents. Although IntentCapsNet has demonstrated strong performance, it has two fundamental limitations.

Limitation 1. The self-attention module of IntentCapsNet cannot handle the polysemy problem, which limits the representation capacity of semantic capsules.

Typically, a word is represented by a multi-dimensional embedding. Since a word can have different meanings in different contexts, some interesting recent studies (Shen et al., 2018; Şenel et al., 2018) suggest that different dimensions of a word embedding may tend to represent different semantic meanings. For example, the word "book" has different meanings in the two utterances: "Book a restaurant in Michigan for 4 people" and "Give 4 out of 6 points to this book". For the embedding of the word "book", it is hypothesized that some dimensions may be more indicative for the first meaning – "reserve", while some other dimensions may be more indicative for the second meaning. Apparently, the self-attention mechanism cannot pay more attentions to the dimensions that best describe the specific meaning of a word in a given context, as it assigns the same attention score for all the dimensions, which significantly limits the representation capacity of semantic capsules and undermines the performance of capsule networks.

Limitation 2. For the generalized zero-shot classification setting, the method of IntentCapsNet for constructing the prediction vectors is highly likely to cause the model to lose generalization ability to unseen intents.

Here, we provide an analysis of IntentCapsNet for predicting an unseen intent in the generalized zero-shot classification setting. In IntentCapsNet, the probability of a test utterance x belonging to a seen intent label k is computed as:

$$P_k = \|\sum_{r=1}^{R} c_{kr}\boldsymbol{p}_{k|r}\| = \|\sum_{r=1}^{R}\boldsymbol{g}_{k,r}\|, \quad (1)$$

where $\|\cdot\|$ is the L2-norm of a vector, R is the number of semantic capsules, $\boldsymbol{p}_{k|r}$ is the prediction vector for the r-th semantic capsule with respect to the seen intent k, and c_{kr} is the weight of the r-th semantic capsule with respect to the seen intent k, which is computed by the dynamic routing algorithm of capsule networks. $\boldsymbol{g}_{k,r} = c_{kr}\boldsymbol{p}_{k|r}$ is called the r-th vote vector for the seen intent k. By Eq. (1), we have a tight upper bound for P_k:

$$P_k \leqslant \sum_{r=1}^{R}\|\boldsymbol{g}_{k,r}\|. \quad (2)$$

IntentCapsNet computes the probability of x belonging to an unseen intent label l as:

$$P_l = \|\sum_{r=1}^{R} c_{lr}\boldsymbol{u}_{l|r}\| = \|\sum_{r=1}^{R} c_{lr}\sum_{k=1}^{K} q_{lk}\boldsymbol{g}_{k,r}\|, \quad (3)$$

where $\boldsymbol{u}_{l|r}$ is the prediction vector for the r-th semantic capsule with respect to the unseen intent l, and c_{lr} is the weight of the r-th semantic capsule with respect to the unseen intent l, which is determined by the dynamic routing algorithm. $\boldsymbol{u}_{l|r} = \sum_{k=1}^{K} q_{lk}\boldsymbol{g}_{k,r}$, where K is the number of seen intents, $\boldsymbol{g}_{k,r}$ is the r-th vote vector for the seen intent k, and q_{lk} is the similarity between an unseen intent $y_l^u \in Y^u$ and a seen intent $y_k^s \in Y^s$. $q_{lk} = \frac{exp(-d(\boldsymbol{e}_k^s, \boldsymbol{e}_l^u))}{\sum_{k=1}^{K} exp(-d(\boldsymbol{e}_k^s, \boldsymbol{e}_l^u))}$, where $\boldsymbol{e}_k^s$ and $\boldsymbol{e}_l^u$ are the embeddings of the seen and unseen intents respectively, and $d(\boldsymbol{e}_k^s, \boldsymbol{e}_l^u)$ is the scaled squared Euclidean distance between $\boldsymbol{e}_k^s$ and $\boldsymbol{e}_l^u$. Since $q_{lk} \in (0,1)$, $c_{lr} \in (0,1)$, $\sum_{k=1}^{K} q_{lk} = 1$ and $\sum_{r=1}^{R} c_{lr} = 1$, we have a tight upper bound for P_l:

$$P_l \leqslant \sum_{r=1}^{R} c_{lr}\sum_{k=1}^{K} q_{lk}\|\boldsymbol{g}_{k,r}\| \leqslant \|\boldsymbol{g}_{k,r}\|_{\max}, \quad (4)$$

where $\|\boldsymbol{g}_{k,r}\|_{\max}$ is the maximum among $\|\boldsymbol{g}_{k,r}\|$, $\forall r \in \{1, 2, \ldots, R\}$ and $\forall k \in \{1, 2, \ldots, K\}$.

By Eq. (2) & (4), it can be seen that the upper bound of P_k is much larger than P_l, indicating that for any utterance x, it is highly likely that $P(y \in Y^s|x)$ is larger than $P(y \in Y^u|x)$. Hence, for generalized zero-shot classification, with high probability IntentCapsNet will classify a test utterance to the seen intents, which is also verified by our experiments in section 5.

4 The Proposed Approach

To overcome the limitations of IntentCapsNet, we propose to reconstruct capsule networks for zero-shot intent classification. In particular, we introduce two modules to capsule networks: (1) a dimensional attention module that helps to extract more representative semantic capsules and (2) a new method for constructing the transformation matrices to improve the model generalization ability to unseen intents.

4.1 Dimensional Attention Capsule Networks

Pre-processing. An utterance with T words can be represented as $x = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_T\}$, where

$\boldsymbol{w}_t \in \mathbb{R}^{d_w}$ is the word embedding of the t-th word and can be pretrained by the skip-gram model (Mikolov et al., 2013). Each word can be further encoded sequentially using a recurrent neural network such as bidirectional LSTM (Hochreiter and Schmidhuber, 1997), i.e.,

$$\begin{aligned} \overrightarrow{\boldsymbol{h}}_t &= \text{LSTM}_{fw}(\boldsymbol{w}_t, \overrightarrow{\boldsymbol{h}}_{t-1}), \\ \overleftarrow{\boldsymbol{h}}_t &= \text{LSTM}_{bw}(\boldsymbol{w}_t, \overleftarrow{\boldsymbol{h}}_{t+1}), \end{aligned} \tag{5}$$

where LSTM_{fw} and LSTM_{bw} denote the forward and backward LSTM respectively, and $\overrightarrow{\boldsymbol{h}}_t \in \mathbb{R}^{d_h}$ and $\overleftarrow{\boldsymbol{h}}_t \in \mathbb{R}^{d_h}$ are the hidden states of the word $\boldsymbol{w}_t$ learned from LSTM_{fw} and LSTM_{bw} respectively. The entire hidden state of $\boldsymbol{w}_t$ is represented by concatenating $\overrightarrow{\boldsymbol{h}}_t$ and $\overleftarrow{\boldsymbol{h}}_t$, i.e., $\boldsymbol{h}_t = [\overrightarrow{\boldsymbol{h}}_t, \overleftarrow{\boldsymbol{h}}_t]$, and the hidden state matrix of the utterance is $\boldsymbol{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_T] \in \mathbb{R}^{2d_h \times T}$.

4.1.1 Extracting Semantic Capsules with Dimensional Attention

In general, an utterance is composed of multiple semantic features, and these semantic features collectively contribute to a more abstract intent label. For example, an utterance "I want to know the temperature of Hong Kong" is composed by multiple semantic features such as get_action (want to know), weather (temperature), and city_name (Hong Kong), and these semantic features collectively reflect the intent label "Get_Weather". Capsule networks provide a hierarchical reasoning structure for modeling semantic features for intent classification. First, the primary capsules in capsule networks can properly match multiple semantic features of an utterance. Second, the dynamic routing mechanism of capsule networks can be used to automatically learn the importance weight of each semantic feature and aggregate them into a high-level intent label.

It is assumed that a high-level semantic feature of an utterance is largely generated by some of its words that have similar semantic meaning (Xia et al., 2018). To extract the semantic features of an utterance, the key problem is to learn the importance weight of each word for a semantic feature. IntentCapsNet (Xia et al., 2018) utilizes the self-attention mechanism to extract the semantic features (capsules) of each utterance. However, self-attention cannot effectively deal with polysemy. Inspired by the work of Shen et al. (2018), we propose to use the dimensional attention mechanism to alleviate the polysemy problem in extracting semantic features. Dimensional attention can automatically assign different attention scores to different dimensions of a word embedding, which not only helps to solve the polysemy problem to some extent, but also expands the search space of the attention parameters, thus improving model flexibility and effectiveness.

Assume each utterance has R semantic features. We propose to learn a dimensional attention matrix $\boldsymbol{A}_r \in \mathbb{R}^{2d_h \times T}$ that encodes the dimensional attentions of the T words with respect to the r-th semantic feature by:

$$\boldsymbol{A}_r = \text{softmax}\left(\boldsymbol{F}_2 \, \text{ReLU}\left(\boldsymbol{F}_1 \boldsymbol{H}\right)\right), \tag{6}$$

where $\boldsymbol{F}_1 \in \mathbb{R}^{d_a \times 2d_h}$ and $\boldsymbol{F}_2 \in \mathbb{R}^{2d_h \times d_a}$ are the trainable parameters, and $\boldsymbol{A}_r(i, j)$ (the element of $\boldsymbol{A}_r$ in the i-th row and j-th column) means the importance weight of the i-th dimension of the j-th word embedding to the r-th semantic feature. Compared with self-attention, dimensional attention can help to choose the appropriate dimensions of a word embedding that can best express the specific meaning of the word in a given context.

After obtaining $\boldsymbol{A}_r$, the r-th semantic feature $\boldsymbol{m}_r \in \mathbb{R}^{2d_h}$ is computed by:

$$\boldsymbol{m}_r = \sum\nolimits_{\text{row}} \left(\boldsymbol{A}_r \odot \boldsymbol{H}\right), \tag{7}$$

where $\odot$ is element-wise multiplication, and $\sum_{\text{row}}$ is an operator that sums up elements of each row. The entire semantic features for each utterance is $\boldsymbol{M} = [\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_R] \in \mathbb{R}^{2d_h \times R}$.

4.1.2 Improved Max-margin Loss

The semantic features of the utterance can then be fed into a capsule network to learn the intent. First, we transform each semantic feature $\boldsymbol{m}_r$ of the utterance to a prediction vector with respect to each intent as:

$$\boldsymbol{p}_{k|r} = \boldsymbol{W}_{kr} \boldsymbol{m}_r, \tag{8}$$

where $\boldsymbol{p}_{k|r} \in \mathbb{R}^{d_p}$ is the prediction vector of the r-th semantic feature with respect to the k-th intent, and $\boldsymbol{W}_{kr} \in \mathbb{R}^{d_p \times 2d_h}$ is the associated transformation matrix.

In training, there are K output capsules, corresponding to K seen intents. The k-th output capsule $\boldsymbol{o}_k$ is the weighted sum of all the prediction vectors $\boldsymbol{p}_{k|r}$ ($r \in \{1, \ldots, R\}$),

$$\boldsymbol{o}_k = \sum_{r=1}^{R} c_{kr} \boldsymbol{p}_{k|r}, \tag{9}$$

Algorithm 1 Dynamic Routing Algorithm

Procedure Dynamic Routing($\boldsymbol{p}_{k|r}, n_{\text{route}}$)
  for all semantic capsule r and intent capsule k: $b_{kr} \leftarrow 0$.
  for n_{route} iterations **do**
    for all semantic capsule r: $\boldsymbol{c}_r \leftarrow \text{softmax}(\boldsymbol{b}_r)$.
    for all intent capsule k: $\boldsymbol{o}_k \leftarrow \sum_{r=1}^{R} c_{kr}\boldsymbol{p}_{k|r}$.
    for all intent capsule k: $\boldsymbol{v}_k \leftarrow \text{squash}(\boldsymbol{o}_k)$.
    for all semantic capsule r and intent capsule k:
      $b_{kr} \leftarrow b_{kr} + \boldsymbol{p}_{k|r} \cdot \boldsymbol{v}_k$.
  end for
  return $\boldsymbol{v}_k$.

where c_{kr} is the coupling coefficient representing the contribution degree of the r-th semantic feature to the k-th intent, which can be computed by the dynamic routing algorithm (Algorithm 1).

Then, a squashing function $\text{squash}(\cdot)$ is applied on $\boldsymbol{o}_k$, and the final output capsule of the k-th intent is:

$$\boldsymbol{v}_k = \text{squash}(\boldsymbol{o}_k) = \frac{\|\boldsymbol{o}_k\|^2}{1+\|\boldsymbol{o}_k\|^2}\frac{\boldsymbol{o}_k}{\|\boldsymbol{o}_k\|}. \quad (10)$$

Now, the probability of the existence of the k-th intent can be represented as the length of the output capsule $\boldsymbol{v}_k$. The computation procedure of $\boldsymbol{v}_k$ is shown in Algorithm 1, where $\boldsymbol{p}_{k|r} \cdot \boldsymbol{v}_k$ denotes the inner product between $\boldsymbol{p}_{k|r}$ and $\boldsymbol{v}_k$.

To train the dimensional attention capsule network, we propose an improved max-margin loss function consisting of two parts.

The first part is the max-margin loss on each labeled utterance, which is the original loss function of capsule networks (Sabour et al., 2017):

$$\begin{aligned} L_1 = \sum_{k=1}^{K} \{ & y_k \max(0, m^+ - \|\boldsymbol{v}_k\|)^2 \\ & + \lambda(1-y_k)\max(0, \|\boldsymbol{v}_k\| - m^-)^2\}, \end{aligned} \quad (11)$$

where $y_k = 1$ if the utterance is of intent label k and $y_k = 0$ otherwise, λ is a down-weighting parameter, and m^+ and m^- are the margins.

The second part is to ensure the diversity of the semantic capsules, i.e., different semantic capsules are likely to be generated by different words in an utterance. The importance weight of each word to the r-th semantic capsule can be represented by the average value of each column of the dimensional attention matrix $\boldsymbol{A}_r$, i.e.,

$$\boldsymbol{s}_r = \frac{1}{2d_h}\sum\nolimits_{\text{col}} \boldsymbol{A}_r, \quad (12)$$

where $\boldsymbol{s}_r \in \mathbb{R}^{1\times T}$ and $\sum_{\text{col}}$ is an operator that sums up elements of each column. Denote by $\boldsymbol{S} = [\boldsymbol{s}_1^\top, \boldsymbol{s}_2^\top, \ldots, \boldsymbol{s}_R^\top] \in \mathbb{R}^{T\times R}$ the importance weight matrix of each word to all the R semantic capsules. To ensure the diversity of the semantic capsules, a natural idea is to constrain the columns of $\boldsymbol{S}$ to be orthogonal with the following loss function:

$$L_2 = \|\boldsymbol{S}^\top\boldsymbol{S} - \boldsymbol{I}\|_F^2, \quad (13)$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix.

Combining Eq. (11) and Eq. (13), the overall loss function of the proposed dimensional attention capsule network is:

$$L_{total} = L_1 + \beta L_2, \quad (14)$$

where β is a trade-off parameter. By minimizing L_{total} with gradient descent methods, all model parameters including $\boldsymbol{F}_1$, $\boldsymbol{F}_2$ and $\boldsymbol{W}_{kr}$ can be learned.

4.2 Zero-shot Intent Classification

To solve zero-shot intent classification with capsule networks, two key problems need to be addressed. (1) How to find the relationship between unseen and seen intents? (2) How to make predictions for unseen intents?

Measuring Intent Relations. To tackle the first problem, we propose to learn a Mahalanobis distance metric to measure the relationship between unseen and seen intents. Specifically, given the embeddings of an unseen intent l and a seen intent k, their squared Mahalanobis distance is given by:

$$d_M(\boldsymbol{e}_l^u, \boldsymbol{e}_k^s) = (\boldsymbol{e}_l^u - \boldsymbol{e}_k^s)^\top \boldsymbol{\Omega}^{-1}(\boldsymbol{e}_l^u - \boldsymbol{e}_k^s), \quad (15)$$

where $\boldsymbol{\Omega}$ is a learnable covariance matrix which models the correlation between dimensions of the embedding. Note that IntentCapsNet (Xia et al., 2018) also tries to use Eq. (15) to model the relationship between unseen and seen intents, but it ignores the correlation between dimensions and simply sets $\boldsymbol{\Omega} = \sigma^2\boldsymbol{I}$ (σ is a scaling hyper-parameter), which is actually a scaled squared Euclidean distance.

As the number of intents is limited, it is difficult to learn a desirable covariance matrix $\boldsymbol{\Omega}$ with the intent embeddings only. Fortunately, we can leverage the word embeddings of the utterances, which come from the same semantic space as the intent embeddings (pre-trained by the same skip-gram model). Hence, we propose to learn the covariance matrix $\boldsymbol{\Omega}$ with the labeled utterances in the

training set. Inspired by the work of Ying and Li (2012), we propose to learn the Mahalanobis distance metric by optimizing the objective:

$$\max_{\mathbf{\Omega}} \min_{(i,j)\in\mathcal{D}} d_M(\boldsymbol{z}_i^s, \boldsymbol{z}_j^s), \quad s.t. \sum_{(i,j)\in\mathcal{S}} d_M(\boldsymbol{z}_i^s, \boldsymbol{z}_j^s) \leqslant 1, \tag{16}$$

where $\mathcal{D}$ and $\mathcal{S}$ respectively denote the pair sets in which utterances belong to different classes and the same class. For an utterance i, $\boldsymbol{z}_i^s$ denotes the average sum of all the word embeddings. As shown by Ying and Li (2012), optimizing Eq. (16) with respect to $\mathbf{\Omega}$ is equivalent to solving an efficient eigenvalue optimization problem. With the learned metric $\mathbf{\Omega}$, we can have the relationship between any unseen and seen intents by substituting it into Eq. (15). Furthermore, we can compute the similarity between them by $q_{lk} = \exp\left(-\alpha \cdot d_M(\boldsymbol{e}_l^u, \boldsymbol{e}_k^s)\right)$, where α is a scaling parameter.

Constructing Transformation Matrices. Intuitively, if an unseen intent is similar to a seen intent, their corresponding transformation matrices should also be similar. Based on this, to solve the second problem, we propose to derive the transformation matrices of unseen intents using the transformation matrices of seen intents and the similarities between unseen and seen intents, and then make predictions for unseen intents with the transformation matrices. Specifically, given a matrix $\boldsymbol{Q} \in \mathbb{R}^{L\times K}$ that encodes the similarities between unseen and seen intents, for an unseen intent l, we propose to construct the transformation matrix $\boldsymbol{W}_{lr}$ for the r-th semantic capsule with respect to the l-th unseen intent by:

$$\boldsymbol{W}_{lr} = \sum_{k=1}^{K} q_{lk} \boldsymbol{W}_{kr}, \tag{17}$$

where q_{lk} is the element in the l-th row and the k-th column of $\boldsymbol{Q}$, $\boldsymbol{W}_{kr}$ is the transformation matrix for the r-th semantic capsule with respect to the k-th seen intent. By Eq. (17), the transformation matrices for all unseen intents can be obtained. When a test utterance arrives, it can be directly fed into the trained dimensional attention capsule network for intent prediction.

Dataset	**SNIPS-NLU**	**SMP-2018**
Vocab Size	11641	2682
Number of Samples	13802	2460
Average Sentence Length	9.05	4.86
Number of Seen Intents	5	24
Number of Unseen Intents	2	6

Table 1: Dataset statistics.

Dataset	d_w	d_h	d_a	d_p	R	n_{route}
SNIPS-NLU	300	16	10	10	3	3
SMP-2018	300	32	30	10	8	3

Table 2: Network structure hyperparameters.

5 Experiments

5.1 Datasets

We evaluate our model on two real task-oriented dialogue datasets in different languages. Table 1 summarizes the dataset statistics.

SNIPS-NLU. Following (Xia et al., 2018), we use the SNIPS-NLU (SNIPS Natural Language Understanding) benchmark dataset (Coucke et al., 2018). SNIPS-NLU is an open-source single-turn English corpus, which contains crowdsourced queries evenly distributed in 7 intents.

SMP-2018. It is a real Chinese dialogue corpus released in SMP 2018 (The China National Conference on Social Media Processing) for user intent classification tasks in Chinese (Zhang et al., 2017). The dataset is provided by the iFLYTEK Corporation, and it can be divided into two parts: chit-chat dialogues and task-oriented dialogues. Here, we only use the task-oriented dialogues.

Dataset Splitting. For zero-shot intent classification, we take all the samples of seen intents as the training set, and all the samples of unseen intents as the test set. For generalized zero-shot intent classification, we randomly take 70% samples of each seen intent as the training set, and the remaining 30% samples of each seen intent and all the samples of unseen intents as the test set.

5.2 Baselines

We compare ReCapsNet-ZS with the following state-of-the-art zero-shot learning methods: DeViSE (Frome et al., 2013), CMT (Socher et al., 2013), CDSSM (Chen et al., 2016), Zero-shot DNN (Kumar et al., 2017) and IntentCapsNet (Xia et al., 2018). To make DeViSE and CMT suit-

Method	SNIP-NLU						SMP-2018					
	Seen		Unseen		Overall		Seen		Unseen		Overall	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
DeViSE	0.9481	0.6536	0.0211	0.0398	0.4215	0.3049	0.8040	0.6740	0.0270	0.0310	0.5030	0.4250
CMT	**0.9755**	0.6648	0.0397	0.0704	0.4438	0.3271	0.8314	0.7221	0.0798	0.1069	0.5398	0.4834
CDSSM	0.9549	**0.7033**	0.0111	0.0218	0.4234	0.3194	0.6653	0.5540	0.1436	0.1200	0.4864	0.4052
Zero-shot DNN	0.9432	0.6679	0.0682	0.1041	0.4488	0.3493	0.7323	0.6116	0.0590	0.0869	0.5013	0.4316
IntentCapsNet	0.9741	0.6517	0.0000	0.0000	0.4200	0.2810	**0.8850**	0.7281	0.0000	0.0000	0.5375	0.4423
ReCapsNet-ZS-Dim	**0.9743**	0.6580	0.0000	0.0000	0.4201	0.2837	**0.8952**	0.7360	0.0000	0.0000	0.5437	0.4470
ReCapsNet-ZS-TM	0.9663	**0.6845**	**0.0981**	**0.1534**	**0.4724**	**0.3824**	0.7863	**0.7390**	**0.1896**	**0.1537**	**0.5521**	**0.5092**
ReCapsNet-ZS	0.9664	0.6743	**0.1121**	**0.1764**	**0.4805**	**0.3911**	0.8230	**0.7450**	**0.1720**	**0.1526**	**0.5674**	**0.5124**

Table 4: Results of generalized zero-shot intent classification. "Seen", "Unseen" and "Overall" respectively denote the performance on the utterances from seen intents, unseen intents, and both seen and unseen intents.

Method	SNIPS-NLU		SMP-2018	
	Acc	F1	Acc	F1
DeViSE	0.7447	0.7446	**0.5456**	0.3875
CMT	0.7396	0.7206	0.4452	0.4245
CDSSM	0.7588	0.7580	0.4308	0.3765
Zero-shot DNN	0.7165	0.7116	0.4615	0.3897
IntentCapsNet	0.7752	0.7750	0.4864	0.4227
ReCapsNet-ZS-Dim	**0.7868**	**0.7859**	0.5005	0.4501
ReCapsNet-ZS-TM	0.7860	0.7837	0.5315	**0.4630**
ReCapsNet-ZS	**0.7996**	**0.7980**	**0.5418**	**0.4769**

Table 3: Results of zero-shot intent classification.

able for intent classification, we use a multi-head self-attention Bi-LSTM model to encode the utterances and then feed the final hidden states to their zero-shot learning models. In addition, we conduct ablation study to evaluate the contribution of each module of our ReCapsNet-ZS. "ReCapsNet-ZS-Dim" refers to the model that only uses the dimensional attention mechanism, and "ReCapsNet-ZS-TM" refers to the one that only uses the proposed transformation matrix construction method.

5.3 Implementation Details

Parameter Settings. For SNIPS-NLU, we use 300-dim embeddings pre-trained on English Wikipedia (Bojanowski et al., 2017). For SMP-2018, we use 300-dim Chinese word embeddings pre-trained by Li et al. (2018). The main network structure hyperparameters are shown in Table 2. In addition, for the zero-shot classification setting, we set α to 1 for SNIPS-NLU and 10 for SMP-2018 respectively. For the generalized zero-shot classification setting, we set α to 1 for SNIPS-NLU and 5 for SMP-2018 respectively. To avoid overfitting, we use dropout with 0.5 dropout rate on the input of the attention layer. For the loss function, we set $\lambda = 0.5$, $m^+ = 0.9$, $m^- = 0.1$, $\beta = 0.001$, and use the Adam optimizer (Kingma and Ba, 2015) with initial learning rate 0.01.

Evaluation Metrics. We adopt two widely used metrics: accuracy (Acc) and micro-average F1-measure (F1) to evaluate the classification performance. Both metrics are computed with the average value weighted by the support of each class, where the support means the sample ratio of the corresponding class.

5.4 Result Analysis

Zero-shot Intent Classification. Table 3 summarizes the average results over 10 runs, where the top 2 results are highlighted in bold. The baseline results on SNIPS-NLU are taken from Xia et al. (2018). The results show that ReCapsNet-ZS outperforms all the baselines, demonstrating its superiority in tackling zero-shot intent classification. We can also see that ReCapsNet-ZS performs better than either ReCapsNet-ZS-Dim or ReCapsNet-ZS-TM, which shows the effectiveness of both of the dimensional attention mechanism and the transformation matrix construction method.

Generalized Zero-shot Intent Classification. Table 4 shows the average results over 10 runs, where the top 2 results are highlighted in bold. We can make the following observations. 1) The performances look much worse than the standard zero-shot setting, but it is not surprising since the seen intent labels are included in the test phase and it makes the problem harder. 2) ReCapsNet-ZS sometimes performs slightly worse than the baselines in detecting seen intents, which is because some baselines tend to classify the test utterances as seen intents, which on the other hand explain-

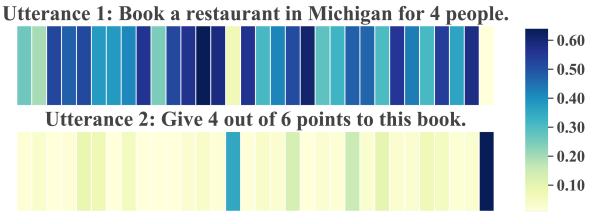

Figure 2: Comparison of the dimensional attentions of the word "book" in different contexts.

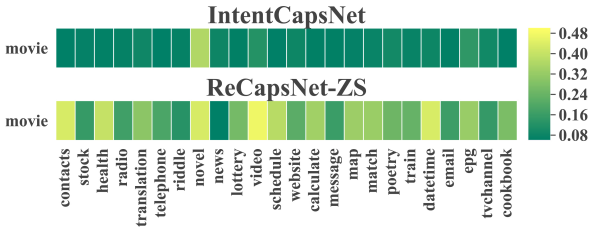

Figure 3: Comparison of the similarity scores between the unseen intent "movie" and the seen intents learned by IntentCapsNet and ReCapsNet-ZS (ours).

s why they perform much worse in detecting unseen intents. 3) ReCapsNet-ZS and ReCapsNet-ZS-TM perform much better than others in detecting unseen intents, whereas IntentCapsNet and ReCapsNet-ZS-Dim both have 0% Acc and F1. This verifies that the proposed transformation matrix construction method has much better generalization ability in detecting unseen intents. 4) Overall, ReCapsNet-ZS consistently performs the best, which further demonstrates the superiority of ReCapsNet-ZS on the generalized zero-shot intent classification tasks.

5.5 Visualization

Dimensional Attentions. Figure 2 visualizes the attention score for each dimension of the same word "book" in two different utterances (contexts) by heatmaps. The utterances are "Book a restaurant in Michigan for 4 people" and "Give 4 out of 6 points to this book", which are taken from SNIPS-NLU. It can be seen that for the two utterances the attention values of the word "book" exhibit completely different patterns, which makes sense as it contains different meanings in different contexts. Furthermore, for each utterance, the attention scores of "book" on different dimensions are also quite different. This shows that the dimensional attention mechanism can effectively capture the semantic differences of the same word in different contexts and encode more useful information than the traditional self-attention method, and thus helps to alleviate the polysemy problem.

Similarity Scores. Figure 3 visualizes the similarity scores between the unseen intent "movie" and the seen intents learned via metric learning by IntentCapsNet and ReCapsNet-ZS respectively on SMP-2018. It can be seen that IntentCapsNet can only discover few connections between the unseen and seen intents. In contrast, ReCapsNet-ZS can detect a lot more connections between them. Further, though the similarity scores of ReCapsNet-ZS may introduce some noise (e.g., "contacts"), they can capture many more positive relations between "movie" and the seen intents (e.g., "novel", "video", "schedule", and "datetime"), which is beneficial for detecting unseen intents.

6 Conclusion

In this paper, we have proposed a novel framework to reconstruct capsule networks for zero-shot intent classification and demonstrated empirically that it compares favourably with existing methods on some real dialogue datasets. The performance gains of our method come from two aspects: the introduction of a new dimensional attention module to capsule networks for feature extraction and the proposal of a new transformation scheme for detecting unseen intents.

There are several directions of the future works. One is to customize our model for few-shot intent classification. Another is to extend our framework to deal with multiple-intent classification. We also plan to apply our model in dialogue systems for low-resource languages such as Cantonese.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments towards improving the manuscript. This research was supported by the grant HK ITF UIM/377.

References

Lei Jimmy Ba, Kevin Swersky, Sanja Fidler, and Ruslan Salakhutdinov. 2015. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4247–4255.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Associa-*

tion for Computational Linguistics (TACL), 5:135–146.

Yun-Nung Chen, Asli Çelikyilmaz, and Dilek Hakkani-Tür. 2017. Deep learning for dialogue systems. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8–14.

Yun-Nung Chen, Dilek Z. Hakkani-Tür, and Xiaodong He. 2016. Zero-shot learning of intent embeddings for expansion by convolutional deep structured semantic models. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6045–6049.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv*, abs/1805.10190.

Emmanuel Ferreira, Bassam Jabaian, and Fabrice Lefèvre. 2015a. Online adaptative zero-shot learning spoken language understanding using word-embedding. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5321–5325.

Emmanuel Ferreira, Bassam Jabaian, and Fabrice Lefèvre. 2015b. Zero-shot semantic parser for spoken language understanding. In *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1403–1407.

Andrea Frome, Gregory S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. 2013. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2121–2129.

Ruiying Geng, Binhua Li, Yongbin Li, Yuxiao Ye, Ping Jian, and Jian Sun. 2019. Few-shot text classification with induction network. *arXiv*, abs/1902.10482.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Jian Hu, Gang Wang, Frederick H. Lochovsky, Jian-Tao Sun, and Zheng Chen. 2009. Understanding user's query intent with wikipedia. In *International Conference on World Wide Web (WWW)*, pages 471–480.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Anjishnu Kumar, Pavankumar Reddy Muddireddy, Markus Dreyer, and Björn Hoffmeister. 2017. Zero-shot learning across heterogeneous overlapping domains. In *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 2914–2918.

Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio. 2008. Zero-data learning of new tasks. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 646–651.

Shen Li, Zhe Zhao, Renfen Hu, Wensi Li, Tao Liu, and Xiaoyong Du. 2018. Analogical reasoning on chinese morphological and semantic relations. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 138–143.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. In *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 685–689.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations (ICLR)*.

Jinseok Nam, Eneldo Loza Menc'ia, and Johannes Fürnkranz. 2016. All-in text: Learning document, label, and word representations jointly. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 1948–1954.

Mark Palatucci, Dean Pomerleau, Geoffrey E. Hinton, and Tom M. Mitchell. 2009. Zero-shot learning with semantic output codes. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1410–1418.

Suman Ravuri and Andreas Stolcke. 2015. Recurrent neural network and lstm models for lexical utterance classification. In *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 135–139.

Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3859–3869.

Prateek Veeranna Sappadla, Jinseok Nam, Eneldo Loza Menc'ia, and Johannes Fürnkranz. 2016. Using semantic similarity for multi-label zero-shot classification of text documents. In *European Symposium on Artificial Neural Networks (ESANN)*.

Lütfi Kerem Şenel, Ihsan Utlu, Veysel Yücesoy, Aykut Koc, and Tolga Cukur. 2018. Semantic structure and interpretability of word embeddings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(10):1769–1779.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2018. Disan: Directional self-attention network for rnn/cnn-free language understanding. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 5446–5455.

Richard Socher, Milind Ganjoo, Christopher D. Manning, and Andrew Y. Ng. 2013. Zero-shot learning through cross-modal transfer. In *Advances in Neural Information Processing Systems (NIPS)*, pages 935–943.

Gokhan Tur, Li Deng, Dilek Hakkani-Tr, and Xiaodong He. 2012. Towards deeper understanding: Deep convex networks for semantic utterance classification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5045–5048.

Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip S. Yu. 2018. Zero-shot user intent detection via capsule neural networks. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3090–3099.

Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh N. Nguyen, Matthias Hein, and Bernt Schiele. 2016. Latent embeddings for zero-shot classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 69–77.

Puyang Xu and Ruhi Sarikaya. 2013. Convolutional neural network based triangular CRF for joint intent detection and slot filling. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU Workshop)*, pages 78–83.

Min Yang, Wei Zhao, Jianbo Ye, Zeyang Lei, Zhou Zhao, and Soufei Zhang. 2018. Investigating capsule networks with dynamic routing for text classification. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3110–3119.

Majid Yazdani and James Henderson. 2015. A model of zero-shot learning of spoken language understanding. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 244–249.

Yiming Ying and Peng Li. 2012. Distance metric learning with eigenvalue optimization. *Journal of Machine Learning Research*, 13:1–26.

Chenwei Zhang, Wei Fan, Nan Du, and Philip S. Yu. 2016. Mining user intentions from medical queries: A neural network based heterogeneous jointly modeling approach. In *International Conference on World Wide Web (WWW)*, pages 1373–1384.

Jingqing Zhang, Piyawat Lertvittayakumjorn, and Yike Guo. 2019. Integrating semantic knowledge to tackle zero-shot text classification. *arXiv*, abs/1903.12626.

Weinan Zhang, Zhigang Chen, Wanxiang Che, Guoping Hu, and Ting Liu. 2017. The first evaluation of chinese human-computer dialogue technology. *arXiv*, abs/1709.10217.