Zero-shot User Intent Detection via Capsule Neural Networks

Congying Xia[1]*, **Chenwei Zhang**[1]*, **Xiaohui Yan**[2], **Yi Chang**[3,4,5], **Philip S. Yu**[1]
[1]Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607 USA
[2]Huawei Technologies, San Jose, CA 95050 USA
[3]College of Artificial Intelligence, Jilin University, Changchun, China
[4]College of Computer Science and Technology, Jilin University, Changchun, China
[5]Key Laboratory of Symbolic Computation and
Knowledge Engineering of Ministry of Education, China
{cxia8,czhang99,psyu}@uic.edu,
yanxiaohui2@huawei.com, yichang@acm.org

Abstract

User intent detection plays a critical role in question-answering and dialog systems. Most previous works treat intent detection as a classification problem where utterances are labeled with predefined intents. However, it is labor-intensive and time-consuming to label users' utterances as intents are diversely expressed and novel intents will continually be involved. Instead, we study the zero-shot intent detection problem, which aims to detect emerging user intents where no labeled utterances are currently available. We propose two capsule-based architectures: IntentCapsNet that extracts semantic features from utterances and aggregates them to discriminate existing intents, and IntentCapsNet-ZSL which gives IntentCapsNet the zero-shot learning ability to discriminate emerging intents via knowledge transfer from existing intents. Experiments on two real-world datasets show that our model not only can better discriminate diversely expressed existing intents, but is also able to discriminate emerging intents when no labeled utterances are available.

1 Introduction

With the increasing complexity and accuracy of speech recognition technology, companies are striving to deliver intelligent conversation understanding systems as people interact with software agents that run on speaker devices or smart phones via natural language interface (Hoy, 2018). Products like Apple's Siri, Amazon's Alexa and Google Assistant are able to interpret human speech and respond them via synthesized voices.

With recent developments in deep neural networks, user intent detection models (Hu et al., 2009; Xu and Sarikaya, 2013; Zhang et al., 2016; Liu and Lane, 2016; Chen et al., 2016b) are proposed to classify user intents given their diversely expressed utterances in the natural language. The decent performances on intent detection usually come with deep neural network classifiers optimized on large-scale utterances which are human-labeled among existing predefined user intents.

As more features and skills are being added to devices which expand their capabilities to new programs, it is common for voice assistants to encounter the scenario where no labeled utterance of an emerging user intent is available in the training data, as illustrated in Figure 1. Current intent detection methods train classifiers in a supervised fashion and they are good at discriminating existing intents such as Get Weather and Play Music whose labeled utterances are already available. However, these models, by the nature of designs, are incapable to detect utterances of emerging intents like AddToPlaylist and RateABook, since no labeled utterances are available. Moreover, it's labor-intensive and time-consuming to annotate utterances of emerging intents and retrain the whole intent detection model.

Thus, it is imperative to develop intent detection models with the zero-shot learning (ZSL) ability (Lampert et al., 2014; Socher et al., 2013; Changpinyo et al., 2016): the ability to expand classifiers and the intent detection space beyond the existing intents, of which we have labeled utterances during training, to emerging intents, of which no labeled utterances are available.

The research on zero-shot intent detection is still in its infancy. Previous zero-shot learning methods for intent detection utilize external resources such as label ontologies (Ferreira et al., 2015a,b) or manually defined attributes that describe intents (Yazdani and Henderson, 2015) to associate existing and emerging intents, which require extra annotation. Compatibility-based methods for zero-shot intent detection (Chen et al., 2016a; Kumar et al., 2017) assume the capability

*Indicates Equal Contribution

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3090–3099
Brussels, Belgium, October 31 - November 4, 2018.

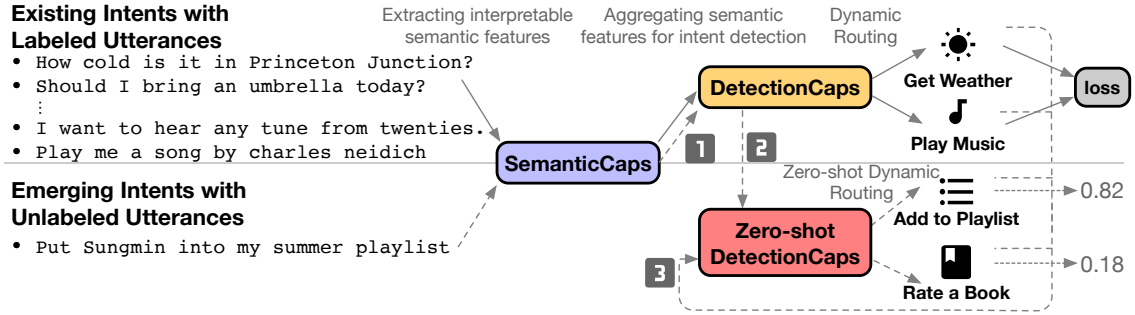

Figure 1: Illustration of the proposed INTENTCAPSNET-ZSL model for zero-shot intent detection: labeled utterances with existing intents like GetWeather and PlayMusic are used to train an intent detection classifier among existing intents, in which SemanticCaps extract intepretable semantic features and DetectionCaps dynamically aggregate semantic features for intent detection using a novel routing-by-agreement mechanism. For emerging intents, INTENTCAPSNET-ZSL builds zero-shot DetectionCaps that utilize the (1) outputs of SemanticCaps, (2) the routing information on existing intents from DetectionCaps, and (3) similarities of the emerging intent label to existing intent labels to discriminate emerging intents like AddToPlayist from RateABook. Solid lines indicate the training process and dash lines indicate the zero-shot inference process.

of learning a high-quality mapping from the utterance to its intent directly, so that such mapping can be further capitalized to measure the compatibility of an utterance with emerging intents. However, the diverse semantic expressions may impede the learning of such mapping.

In this work, we make the very first attempt to tackle the zero-shot intent detection problem with a capsule-based (Hinton et al., 2011; Sabour et al., 2017) model. A capsule houses a vector representation of a group of neurons, and the orientation of the vector encodes properties of an object (like the shape/color of a face), while the length of the vector reflects its probability of existence (how likely a face with certain properties exists). The capsule model learns a hierarchy of feature detectors via a routing-by-agreement mechanism: capsules for detecting low-level features (like nose/eyes) send their outputs to high-level capsules (such as faces) only when there is a strong agreement of their predictions to high-level capsules.

The aforementioned properties of capsule models could be quite appealing for text modeling, specifically in this case, modeling the user utterance for intent detection: low-level semantic features such as the get_action, time and city_name contribute to a more abstract intent (GetWeather) collectively. A semantic feature, which may be expressed quite differently among users, can contribute more to one intent than others. The dynamic routing-by-agreement mechanism can be used to dynamically assign a proper contribution of each semantic and aggregate them to get an intent representation.

More importantly, we discover the potential of zero-shot learning ability on the capsule model, which is not yet widely recognized. It makes the capsule model even more suitable for text modeling when no labeled utterances are available for emerging intents. The ability to neglect the disagreed output of low-level semantics for certain intents during routing-by-agreement encourages the learning of generalizable semantic features that can be adapted to emerging intents. For each emerging intent with no labeled utterances, a Zero-shot DetectionCaps is constructed explicitly by using not only semantic features SemanticCaps extracted, but also existing routing agreements from DetectionCaps and similarities of an emerging intent label to existing intent labels.

In summary, the contributions of this work are:

• Expanding capsule neural networks to text modeling, by extracting and aggregating semantics from utterances in a hierarchical manner;

• Proposing a novel and effective capsule-based model for zero-shot intent detection;

• Showing and interpreting the effectiveness of our model on two real-world datasets.

2 Problem Formulation

In this section, we first define related concepts, and formally state the problem.

Intent. An intent is a purpose, or a goal that underlies a user-generated utterance (Watson Assistant, 2017). An utterance can be associated with one or multiple intents. We only consider the basic case that an utterance is with a single intent. However, utterances with multiple intents can be handled by segmenting them into single-intent snippets using sequential tagging tools like CRF (Lafferty et al.,

2001), which we leave for future works.

Intent Detection. Given a labeled training dataset where each sample has the following format: (x, y) where x is an utterance and y is its intent label, each training example is associated with one of K existing intents $y \in Y = \{y_1, y_2, ..., y_K\}$. The intent detection task tries to associate an utterance $x_{existing}$ with its correct intent category in the existing intent classes Y.

Zero-shot Intent Detection. Given the labeled training set $\{(x, y)\}$ where $y \in Y$, the zero-shot intent detection task aims to detect an utterance $x_{emerging}$ which belongs to one of L emerging intents $z \in Z = \{z_1, z_2, ..., z_L\}$ where $Y \cap Z = \varnothing$.

3 Approach

We propose two architectures based on capsule models: INTENTCAPSNET that is trained to discriminate among utterances with existing labels, e.g. existing intents for intent detection; INTENTCAPSNET-ZSL that gives zero-shot learning ability to INTENTCAPSNET for discriminating unseen labels, i.e. emerging intents in this case. As shown in Figure 2, the cores of the proposed architectures are three types of capsules: SemanticCaps that extract interpretable semantic features from the utterance, DetectionCaps that aggregate semantic features for intent detection, and Zero-shot DetectionCaps which discriminate emerging intents.

3.1 SemanticCaps

In the original capsule model (Sabour et al., 2017), convolution-based PrimaryCaps are introduced as the first layer to obtain different vectorized features from the raw input image. While in this work, an intrinsically similar motivation is adopted to extract different semantic features from the raw utterance by a new type of capsule named SemanticCaps. Unlike the PrimaryCaps which use convolution operators with a large reception field to extract spacial-proximate features, the SemanticCaps is based on a bi-direction recurrent neural network with multiple self-attention heads, where each self-attention head focuses on certain part of the utterance and extracts a semantic feature that may not be expressed by words in proximity.

Given an input utterance $x = (\mathbf{w_1}, \mathbf{w_2}, ..., \mathbf{w_T})$ of T words, each word is represented by a vector of dimension D_W that can be pre-trained using a skip-gram language model (Mikolov et al., 2013).

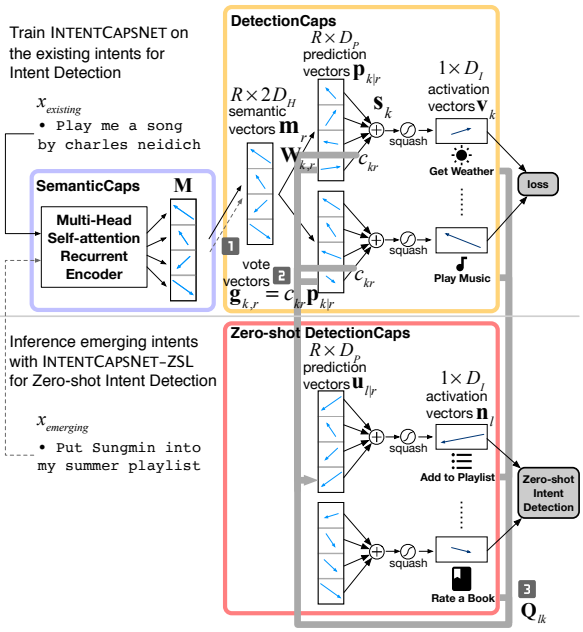

Figure 2: The architecture of INTENTCAPSNET and INTENTCAPSNET-ZSL. During training, utterances with existing intents are fed into the SemanticCaps which output vectorized semantic features, i.e. semantic vectors. Then DetectionCaps combine these features into higher-level prediction vectors and output an activation vector for intent detection on each existing intent. During inference, emerging utterances take advantages of the SemanticCaps trained in INTENTCAPSNET to extract semantic features from the utterance (shown in 1), then the vote vectors on the existing intents are transferred to emerging intents (shown in 2) using similarities between existing and emerging intents (shown in 3). The obtained activation vectors for emerging intents are used for zero-shot intent detection.

A recurrent neural network such as a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) is applied to sequentially encode the utterance into hidden states:

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

For each word $\mathbf{w_t}$, we concatenate each forward hidden state $\overrightarrow{\mathbf{h}}_t$ obtained from the forward LSTM_{fw} with a backward hidden state $\overleftarrow{\mathbf{h}}_t$ from LSTM_{bw} to obtain a hidden state $\mathbf{h_t}$ for the word $\mathbf{w_t}$. The whole hidden state matrix can be defined as $\mathbf{H} = (\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_T) \in \mathbb{R}^{T \times 2D_H}$, where D_H is the number of hidden units in each LSTM.

Inspired by the success of self-attention mechanisms (Vaswani et al., 2017; Lin et al., 2017) for sentence embedding, we adopt a multi-head self-attention framework where each self-attention head is encouraged to be attentive to a specific semantic feature of the utterance, such as certain sets

of keywords or phrases in the utterance: one self-attention may be attentive for the "get" action in GetWeather, while another one may be attentive to city_name in GetWeather: it decides for itself what semantics to be attentive to.

A self-attention weight matrix $\mathbf{A}$ is computed as:

$$\mathbf{A} = \text{softmax}\left(\mathbf{W}_{s2}\text{tanh}\left(\mathbf{W}_{s1}\mathbf{H}^T\right)\right), \quad (2)$$

where $\mathbf{W}_{s1} \in \mathbb{R}^{D_A \times 2D_H}$ and $\mathbf{W}_{s2} \in \mathbb{R}^{R \times D_A}$ are weight matrices for the self-attention. D_A is the hidden unit number of self-attention and R is the number of self-attention heads. The softmax function makes sure for each self-attention head, the attentive scores on all the words sum to one.

A total number of R semantic features are extracted from the input utterance, each from a separate self-attention head: $\mathbf{M} = \mathbf{AH}$, where $\mathbf{M} = (\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_R) \in \mathbb{R}^{R \times 2D_H}$. Each $\mathbf{m}_r$ is a $2D_H-$dimensional semantic vector.

Each semantic vector will have a distinguishable orientation when the objective is properly regularized (details in Equation 6), as we want each attention to be attentive to a unique semantic feature of the utterance. The vector representation adopted in capsules is suitable to portray the low-level semantic properties as well as high-level intents of the utterance, where the orientation of a vector represents semantic/intent properties that may slightly vary depending on the expressions. The capsule encourages the learning of generalizable semantic vectors: less informative semantic properties for one intent may not be penalized by their orientations: they simply possess small norms as they are less likely to exist.

3.2 DetectionCaps

The output of SemanticCaps are low-level vector representations of R different semantic features extracted from the utterances. To combine these features into higher-level representations, we build DetectionCaps that choose different semantic features dynamically so as to form an intent representation for each intent via an unsupervised routing-by-agreement mechanism.

As a semantic feature may contribute differently in detecting different intents, the DetectionCaps first encode semantic features with respect to each intent:

$$\mathbf{p}_{k|r} = \mathbf{m}_r \mathbf{W}_{k,r}, \quad (3)$$

where $k \in \{1, 2, ..., K\}$, $r \in \{1, 2, ..., R\}$. $\mathbf{W}_{k,r} \in \mathbb{R}^{2D_H \times D_P}$ is the weight matrix of the DetectionCaps, $\mathbf{p}_{k|r}$ is the prediction vector of the r-th semantic feature of an existing intent k, and D_P is the dimension of the prediction vector.

Dynamic Routing-by-agreement. The prediction vectors obtained from SemanticCaps route dynamically to DetectionCaps. The DetectionCaps computes a weighted sum over all prediction vectors:

$$\mathbf{s}_k = \sum_r^R c_{kr} \mathbf{p}_{k|r}, \quad (4)$$

where c_{kr} is the coupling coefficient that determines how informative, or how much contribution the r-th semantic feature is to the intent y_k. c_{kr} is calculated by an unsupervised, iterative dynamic routing-by-agreement algorithm (Sabour et al., 2017), which is briefly recalled in Algorithm 1. As shown in this algorithm, b_{kr} is the initial logit representing the log prior probability that a SemanticCap r is coupled to an DetectionCap k.

Algorithm 1 Dynamic routing algorithm

```
procedure DYNAMIC ROUTING(p_{k|r}, iter)
    for all semantic capsule r and intent capsule k:
  b_kr ← 0.
    for iter iterations do
        for all SemanticCaps r: c_r ← softmax(b_r)
        for all DetectionCaps k: s_k ← Σ_r c_kr p_{k|r}
        for all DetectionCaps k: v_k = squash(s_k)
        for all SemanticCaps r and DetectionCaps k:
            b_kr ← b_kr + p_{k|r} · v_k
    end for
    Return v_k
end procedure
```

The squashing function $\text{squash}(\cdot)$ is applied on $\mathbf{s}_k$ to get an activation vector $\mathbf{v}_k$ for each existing intent class k:

$$\mathbf{v}_k = \frac{\|\mathbf{s}_k\|^2}{1 + \|\mathbf{s}_k\|^2} \frac{\mathbf{s}_k}{\|\mathbf{s}_k\|}, \quad (5)$$

where the orientation of the activation vector $\mathbf{v}_k$ represents intent properties while its norm indicates the activation probability. The dynamic routing-by-agreement mechanism assigns low c_{kr} when there is inconsistency between $p_{k|r}$ and v_k, which ensures the outputs of the SemanticCaps get sent to appropriate subsequent DetectionCaps.

Max-margin Loss for Existing Intents. The loss function considers both the max-margin loss on each labeled utterance, as well as a regularization term that encourages each self-attention head to be

attentive to a different semantic feature of the utterance:

$$\begin{aligned}\mathcal{L} = \sum_{k=1}^{K} \{ & [\![y = y_k]\!] \cdot \max(0, m^+ - \|\mathbf{v}_k\|)^2 \\ & + \lambda [\![y \neq y_k]\!] \cdot \max(0, \|\mathbf{v}_k\| - m^-)^2 \} \\ & + \alpha ||\mathbf{A}\mathbf{A}^T - I||_F^2,\end{aligned} \tag{6}$$

where $[\![]\!]$ is an indicator function, y is the ground truth intent label for the utterance x, λ is a down-weighting coefficient, m^+ and m^- are margins. α is a non-negative trade-off coefficient that encourages the discrepancies among different attention heads.

3.3 Zero-shot DetectionCaps

To detect emerging intents effectively, Zero-shot DetectionCaps are designed to transfer knowledge from existing intents to emerging intents.

Knowledge Transfer Strategies. As SemanticCaps are trained to extract semantic features from utterances with various existing intents, a self-attention head which has similar extraction behavior among existing and emerging intents may help transfer knowledge. For example, a self-attention head that extracts the "play" action mentioned by `turn on/I want to hear` in the beginning of an utterance for PlayMusic is helpful if it is also attentive to expressions for the "add" action like `add/I want to have` in the beginning of an utterance with an emerging intent AddtoPlaylist.

The coupling coefficient c_{kr} learned by DetectionCaps in a totally unsupervised fashion embodies rich knowledge of how informative r-th semantic is to the existing intent k. We can capitalize on the existing routing information for emerging intents. For example, how the word `play` routes to GetWeather can be helpful in routing the word `add` to AddtoPlaylist.

The intent labels also contain knowledge of how two intents are similar with each other. For example, an emerging intent AddtoPlaylist can be closer to one existing intent PlayMusic than GetWeather due to the proximity of the embedding of Playlist to Play or Music, than Weather.

Build Vote Vectors. As the routing information and the semantic extraction behavior are strongly coupled (c_{kr} is calculated by $\mathbf{p}_{k|r}$ iteratively in Line 4-6 of Algorithm 1) and their products are summarized to get the activation vector v_k for intent k (Line 5-6 of Algorithm 1), we denote vectors before summation as vote vectors:

$$\mathbf{g}_{k,r} = c_{kr}\mathbf{p}_{k|r}, \tag{7}$$

where $\mathbf{g}_{k,r}$ is the r-th vote vector for an existing intent k.

Zero-shot Dynamic Routing. The zero-shot dynamic routing utilizes vote vectors from existing intents to build intent representations for emerging intents via a similarity metric between existing intents and emerging intents.

Since there are K existing intents and L emerging intents, the similarities between existing and emerging intents form a matrix $\mathbf{Q} \in \mathbb{R}^{L \times K}$. Specifically, the similarity between an emerging intent $z_l \in Z$ and an existing intent $y_k \in Y$ is computed as:

$$q_{lk} = \frac{exp\left\{-d\left(\mathbf{e}_{z_l}, \mathbf{e}_{y_k}\right)\right\}}{\sum_{k=1}^{K} exp\left\{-d\left(\mathbf{e}_{z_l}, \mathbf{e}_{y_k}\right)\right\}}, \tag{8}$$

where

$$d\left(\mathbf{e}_{z_l}, \mathbf{e}_{y_k}\right) = \left(\mathbf{e}_{z_l} - \mathbf{e}_{y_k}\right)^T \Sigma^{-1} \left(\mathbf{e}_{z_l} - \mathbf{e}_{y_k}\right). \tag{9}$$

$\mathbf{e}_{z_l}, \mathbf{e}_{y_k} \in \mathbb{R}^{D_I \times 1}$ are intent embeddings computed by the sum of word embeddings of the intent label. Σ models the correlations among intent embedding dimensions and we use $\Sigma = \sigma^2 I$. σ is a hyper-parameter for scaling. The prediction vectors for emerging intents are thus computed as:

$$\mathbf{u}_{l|r} = \sum_{k=1}^{K} q_{lk}\mathbf{g}_{k,r}. \tag{10}$$

We feed the prediction vector $\mathbf{n}_l$ to Algorithm 1 and derive activation vectors $\mathbf{n}_l$ on emerging intents as the output. The final intent representation $\mathbf{n}_l$ for each emerging intent is updated toward the direction where it coincides with representative votes vectors.

We can easily classify the utterance of emerging intents by choosing the activation vector with the largest norm $\hat{z} = \arg\max_{z_l \in Z} \|\mathbf{n}_l\|$.

4 Experiment Setup

To demonstrate the effectiveness of our proposed models, we apply INTENTCAPSNET to detect existing intents in an intent detection task, and use INTENTCAPSNET-ZSL to detect emerging intents in a zero-shot intent detection task.

Datasets. For each task, we evaluate our proposed models by applying it on two real-word

Model	SNIPS-NLU (on 5 existing intents)				CVA (on 80 existing intents)			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
TFIDF-LR	0.9546	0.9551	0.9546	0.9545	0.7979	0.8104	0.7979	0.7933
TFIDF-SVM	0.9584	0.9586	0.9584	0.9581	0.7989	0.8111	0.7989	0.7942
CNN	0.9595	0.9596	0.9595	0.9595	0.8223	0.8288	0.8223	0.8210
RNN	0.9516	0.9522	0.9516	0.9518	0.8286	0.8330	0.8286	0.8275
GRU	0.9535	0.9535	0.9535	0.9534	0.8239	0.8281	0.8239	0.8216
LSTM	0.9569	0.9573	0.9569	0.9569	0.8319	0.8387	0.8319	0.8306
Bi-LSTM	0.9501	0.9502	0.9501	0.9502	0.8428	0.8479	0.8428	0.8419
Self-attention Bi-LSTM	0.9524	0.9522	0.9524	0.9522	0.8521	0.8590	0.8521	0.8513
INTENTCAPSNET	**0.9621**	**0.9620**	**0.9621**	**0.9620**	**0.9088**	**0.9160**	**0.9088**	**0.9023**

Table 1: Intention detection results using INTENTCAPSNET on two datasets. All the metrics (Accuray, Precision, Recall and F1) are reported using the average value weighted by their support on per class.

datasets: SNIPS Natural Language Understanding benchmark (SNIPS-NLU) and a Commercial Voice Assistant (CVA) dataset. The statistical information on two datasets are shown in Table 2. SNIPS-NLU[1] is an English natural language corpus collected in a crowdsourced fashion to benchmark the performance of voice assistants. CVA is a Chinese natural language corpus collected anonymously from a commercial voice assistant on smart phones.

Dataset	SNIPS-NLU	CVA
Vocab Size	10,896	1,709
Number of Samples	13,802	9,992
Average Sentence Length	9.05	4
Number of Existing Intents	5	80
Number of Emerging Intents	2	20

Table 2: Dataset statistics.

Baselines. We first compare the proposed capsule-based model INTENTCAPSNET with other text classification alternatives on the detection of existing intents: 1) TFIDF-LR/TFIDF-SVM: we use TF-IDF to represent the utterance and use logistic regression/support vector machine as classifiers. 2) CNN: a convolutional neural network (Kim, 2014) that uses convolution and pooling operations, which is popular for text classification. 3) RNN/GRU/LSTM/BiLSTM: we adopt different types of recurrent neural networks: the vanilla recurrent neural network (RNN), gated recurrent unit (GRU) (Tang et al., 2015), long short-term memory networks (LSTM) (Hochreiter and Schmidhuber, 1997), and bi-directional long short-term memory (Bi-LSTM) (Schuster and Paliwal, 1997). Their last hidden states are used for classification. 4) Self-Attention Bi-LSTM: we apply a Bi-LSTM model with self-attention mechanism (Lin et al., 2017) and the output sentence embedding is used for classification.

We also compare our proposed model INTENTCAPSNET-ZSL with different zero-shot learning strategies: 1) DeViSE (Frome et al., 2013) finds the most compatible emerging intent label for an utterance by learning a linear compatibility function between utterances and intents; 2) CMT (Socher et al., 2013) introduces non-linearity in the compatibility function; CMT and DeViSE are originally designed for zero-shot image classification based on pretrained CNN features. We use LSTM to encode the utterance and adopt their zero-shot learning strategies in our task; 3) CDSSM (Chen et al., 2016a) uses CNN to extract character-level sentence features, where the utterance encoder shares the weights with the label encoder; 4) Zero-shot DNN (Kumar et al., 2017) further improves the performance of CDSSM by using separate encoders for utterances and intent. The proposed model INTENTCAPSNET-ZSL can be seen as a hybrid model: it has the advantages of the compatibility models to model the correlations between utterances and intents directly; it also explicitly derives intent representations for emerging intents without labeled utterances.

Dataset	D_W	D_H	D_A	R	σ	α
SNIPS-NLU	300	32	20	3	4	0.0001
CVA	200	200	100	8	1	0.01

Table 3: Hyperparameter settings.

Implementation Details. The hyperparameters used for experiments are shown in Table 3. We use three fold cross-validation to choose hyperpa-

[1] https://github.com/snipsco/nlu-benchmark/

Model	SNIPS-NLU (on 2 emerging intents)				CVA (on 20 emerging intents)			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
DeViSE (Frome et al., 2013)	0.7447	0.7448	0.7447	0.7446	0.7809	0.8060	0.7809	0.7617
CMT (Socher et al., 2013)	0.7396	**0.8266**	0.7396	0.7206	0.7721	0.7728	0.7721	0.7445
CDSSM (Chen et al., 2016a)	0.7588	0.7625	0.7588	0.7580	0.2140	0.4072	0.2140	0.1667
Zero-shot DNN (Kumar et al., 2017)	0.7165	0.7330	0.7165	0.7116	0.7903	0.8240	0.7903	0.7774
IntentCapsNet-ZSL w/o Self-attention	0.7587	0.7764	0.7588	0.7547	0.8103	0.8512	0.8103	0.8115
IntentCapsNet-ZSL w/o Bi-LSTM	0.7619	0.7631	0.7619	0.7616	0.8366	**0.8770**	0.8366	0.8403
IntentCapsNet-ZSL w/o Regularizer	0.7675	0.7676	0.7675	0.7675	0.8544	0.8730	0.8544	0.8553
IntentCapsNet-ZSL	**0.7752**	0.7762	**0.7752**	**0.7750**	**0.8628**	0.8751	**0.8629**	**0.8635**

Table 4: Zero-shot intention detection results using IntentCapsNet-ZSL on two datasets. All the metrics (Accuray, Precision, Recall and F1) are reported using the average value weighted by their support on per class.

rameters. The dimension of the prediction vector D_P is 10 for both datasets. $D_I = D_W$ because we use the averaged word embeddings contained in the intent label as the intent embedding. An additional input dropout layer with a dropout keep rate 0.8 is applied to the SNIPS-NLU dataset. In the loss function, the down-weighting coefficient λ is 0.5, margins m_k^+ and m_k^- are set to 0.9 and 0.1 for all the existing intents. The iteration number $iter$ used in the dynamic routing algorithm is 3. Adam optimizer (Kingma and Ba, 2014) is used to minimize the loss.

5 Results

Quantitative Evaluation. The intention detection results on two datasets are reported in Table 1, where the proposed capsule-based model IntentCapsNet performs consistently better than bag-of-word classifiers using TF-IDF, as well as various neural network models designed for text classification. These results demonstrate the novelty and effectiveness of the proposed capsule-based model IntentCapsNet in modeling text for intent detection.

Also, we report results on zero-shot intention detection task in Table 4, where our model IntentCapsNet-ZSL outperforms other baselines that adopt different zero-shot learning strategies. CMT has higher precision but low accuracy and recall on the SNIPS-NLU dataset. CDSSM fails on CVA dataset, probably because the character-level model is suitable for English corpus but not for CVA, which is in Chinese.

Ablation Study. To study the contribution of different modules of IntentCapsNet-ZSL for zero-shot intent detection, we also report ablation test results in Table 4. "w/o Self-attention" is the model without self-attention: the last forward/backward hidden states of the bi-LSTM recurrent encoder are used; "w/o Bi-LSTM" uses the LSTM with only a forward pass; "w/o Regularizer" does not encourage discrepancies among different self-attention heads: it adopts $\alpha = 0$ in the loss function. Generally, from the lower part of Table 4 we can see that all modules contribute to the effectiveness of the model. On the SNIPS-NLU dataset, each of the three modules has a comparable contribution to the whole model (around 2-3% improvement in F1 score). While on the CVA dataset, the self-attention plays the most important role, which gives the model a 5.2% improvement in F1 score.

Discriminative Emerging Intent Representations. Besides quantitative evidences supporting the effectiveness of the IntentCapsNet-ZSL, we visualize activation vectors of emerging intents in Figure 3. Since the activation vectors of utterances with emerging intents are of high dimension and we are interested in their orientations which indicate their intent properties, t-SNE is applied on the normal vector of the activation vectors to reduce the dimension to 2. We color the utterances according to their ground-truth emerging intent labels.

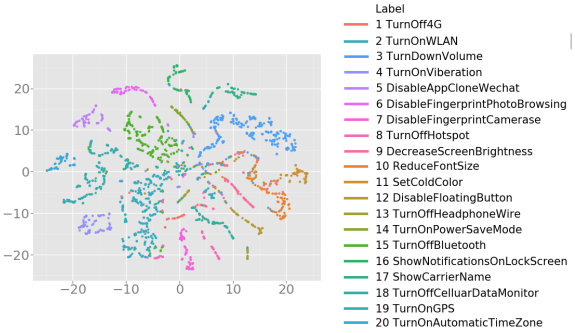

Figure 3: t-SNE visualization of normal activation vectors of utterances with 20 emerging intents in CVA.

As illustrated in Figure 3, IntentCapsNet-ZSL has the ability to learn discriminative intent representations for emerging intents in zero-shot

DetectionCaps, so that utterances with different intents naturally have different orientations. In the meanwhile, utterances of the same emerging intent but with nuances in expressions result in their proximity in the t-SNE space. However, we do observe less satisfied cases where the model mistake an emerging intent DecreaseScreenBrightness (No. 9) with ReduceFontSize (No. 10) and SetColdColor (No. 11). When we check activation vectors of intents in Figure 3 we also find that these three intents tend to have similar representations around the area (15, -5). We think it is due to their inherent similarity as these three intents all try to tune display configurations.

6 Interpretability

Capsule models try to bring more interpretability when compared with traditional deep neural networks. We provide case studies here toward the intepretability of the proposed model in 1) extracting meaningful semantic features and 2) transferring knowledge from existing intents to emerging intents.

Extracting Meaningful Semantic Features. To show that SemanticCaps have the ability to extract meaningful semantic features from the utterance, we study the self-attention matrix **A** within the SemanticCaps and visualize the attention scores of utterances on both existing and emerging intents.

Existing Intent: PlayMusic
• Play Action
play music by charlie adams from
i want to hear any tune from twenties
open up music on last fm
• Musician Name
i want to hear music by madeleine peyroux from on youtube
play me a song by charles neidich
use itunes to play artist ringo shiina track in heaven
Existing Intent: SearchCreativeWork
• Search Action
find fields of sacrifice movie
i m looking for music of nashville season saga
show me television show children in need rocks
• Creative Work Type
please find me platinum box ii song ?
show me a picture called heart like a hurricane
where can i buy a photograph called feel love ?

Table 5: Attentions on utterances with existing intents on SNIPS-NLU.

From Table 5 we can see that each self-attention head almost always focuses on one unique semantic feature of the utterance. For example, in the intent of PlayMusic one self-attention head always focuses on the "play" action while another attention focuses on musician names. We also observe that the learned attention adopts well to diverse expressions. For example, the self-attention head in PlayMusic is attentive to various mentions of musician names when they are followed by words like `by`, `play` and `artist`, even when named entities are not tagged and given to the model. The self-attention head that extracts the "search" action in SearchCreativeWork is able to be attentive to various expressions such as `find`, `looking for` and `show`.

Extraction-behavior Transfer by SemanticCaps. More importantly, we observe appealing extraction behaviors of SemanticCaps on utterances of emerging intents as well, even if they are not trained to perform semantic extraction on utterances of emerging intents.

Emerging Intent: RateBook
• Rate Action
i d rate this novel a five
add the rating for this current series a four out of points
i give ruled britannia a rating of five out of
• Book Name
give the televised morality series a one
i want to give the coming of the terraphiles a rating of
the chronicle charlie peace earns stars from me
• Rating Score
rate the grisly wife three points out of five
i would give this current chronicle three points
this saga deserves a score of four
Emerging Intent: AddToPlaylist
• Song/Artist Name
add star light star bright to my jazz classics playlist
i want a song by john schlitt in the bajo las estrellas playlist
put sungmin into my summer playlist
• Playlist Name
add an album to my list la mejor msica dance
can you add danny carey to my masters of metal playlist
i want to put a copy of this tune into skatepark punks

Table 6: Attentions on utterances with emerging intents on SNIPS-NLU.

From Table 6 we observe that the same self-attention head that extracts "play" action in the existing intent PlayMusic is also attentive to words or phrases referring to the "rate" action in an emerging intent RateABook: like `rate`, `add the rating`, and `give`. Other self-attention heads are almost always focusing on other aspects of the utterances such as the book name or the actual rating score.

Such behavior not only shows that SemanticCaps have the capacity to learn an intent-independent semantic feature extractor, which extracts generalizable semantic features that either existing or emerging intent representations are built upon, but also indicates that SemanticCaps has the ability to transfer extraction behaviors among utterances of different intents.

Knowledge Transfer via Intent Similarity. Beside extracting semantic features and utilizing existing routing information, we use similarities between intent embeddings to help trans-

fer vote vectors from INTENTCAPSNET to INTENTCAPSNET-ZSL. We study the similarity distribution of each emerging intents to all existing intents in Figure 4.

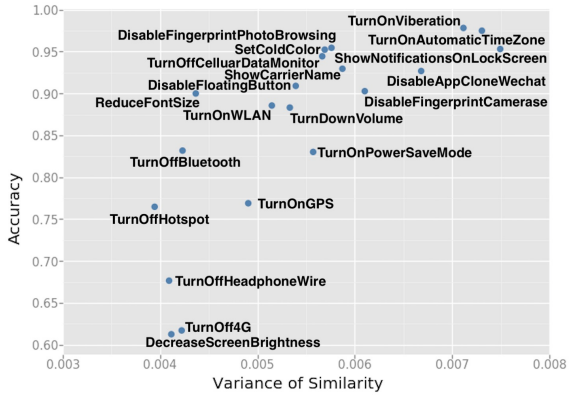

Figure 4: Accuracy vs. variance of the similarity distribution for 20 emerging intents in CVA dataset.

The y axis is the zero-shot detection accuracy on each emerging intent in the CVA dataset. The x axis measures $\text{var}(\mathbf{q}_l)$, the variance of the similarity distribution of each emerging intent l to all the existing intents. If an emerging intent has a high variance in the similarity distribution, it means that some existing intents have higher similarities with this emerging intent than others: the model is more certain about which existing intent to transfer the similarity knowledge from, based on intent label similarities. In this case, 13 out of 20 emerging intents with high variances where $\text{var}(\mathbf{q}_l) > 0.005$ always have a decent performance (Accuracy$\geqslant$0.83). While a low variance does not necessarily always lead to less satisfied performances as some intents can rely on existing intents more evenly together, but with less confidence on each, for knowledge transfer.

7 Conclusions

In this paper, a capsule-based model, namely INTENTCAPSNET, is first introduced to harness the advantages of capsule models for text modeling in a hierarchical manner: semantic features are extracted from the utterances with self-attention, and aggregated via the dynamic routing-by-agreement mechanism to obtain utterance-level intent representations. We believe that the inductive biases subsumed in such capsule-based hierarchical learning schema have broader applicability on various text modeling tasks, besides its evidenced performance on the intent detection task we studied in this paper. The proposed INTENTCAPSNET-ZSL model further introduces zero-shot learning ability to the capsule model via various means of knowledge transfer from existing intents for discriminating emerging intents where no labeled utterances or excessive external resources are available. Experiments on two real-world datasets show the effectiveness and intepretability of the proposed models.

Acknowledgments

We thank the reviewers for their valuable comments. This work is supported in part by NSF through grants IIS-1526499, IIS-1763325, and CNS-1626432, and NSFC 61672313. Xiaohui Yan's work is funded by the National Natural Science Foundation of China (NSFC) under Grant No. 61502447.

References

Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. 2016. Synthesized classifiers for zero-shot learning. In *CVPR*, pages 5327–5336.

Yun-Nung Chen, Dilek Hakkani-Tür, and Xiaodong He. 2016a. Zero-shot learning of intent embeddings for expansion by convolutional deep structured semantic models. In *ICASSP*, pages 6045–6049.

Yun-Nung Chen, Dilek Hakkani-Tür, Gökhan Tür, Jianfeng Gao, and Li Deng. 2016b. End-to-end memory networks with knowledge carryover for multi-turn spoken language understanding. In *INTERSPEECH*, pages 3245–3249.

Emmanuel Ferreira, Bassam Jabaian, and Fabrice Lefevre. 2015a. Online adaptative zero-shot learning spoken language understanding using word-embedding. In *ICASSP*, pages 5321–5325.

Emmanuel Ferreira, Bassam Jabaian, and Fabrice Lefèvre. 2015b. Zero-shot semantic parser for spoken language understanding. In *INTERSPEECH*, pages 1403–1407.

Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Tomas Mikolov, et al. 2013. Devise: A deep visual-semantic embedding model. In *NIPS*, pages 2121–2129.

Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. 2011. Transforming auto-encoders. In *ICANN*, pages 44–51.

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

Matthew B Hoy. 2018. Alexa, siri, cortana, and more: An introduction to voice assistants. *Medical reference services quarterly*, 37(1):81–88.

Jian Hu, Gang Wang, Fred Lochovsky, Jian-tao Sun, and Zheng Chen. 2009. Understanding user's query intent with wikipedia. In *WWW*, pages 471–480.

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

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Anjishnu Kumar, Pavankumar Reddy Muddireddy, Markus Dreyer, and Björn Hoffmeister. 2017. Zero-shot learning across heterogeneous overlapping domains. In *INTERSPEECH*, volume 2017, pages 2914–2918.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289.

Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. 2014. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. In *ICLR*.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. In *INTERSPEECH*, pages 685–689.

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

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. In *NIPS*, pages 3859–3869.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. 2013. Zero-shot learning through cross-modal transfer. In *NIPS*, pages 935–943.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *EMNLP*, pages 1422–1432.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 6000–6010.

IBM Watson Assistant. 2017. Defining intents. In https://console.bluemix.net/docs/services/conversation/intents.html#defining-intents.

Puyang Xu and Ruhi Sarikaya. 2013. Convolutional neural network based triangular crf for joint intent detection and slot filling. In *ASRU*, pages 78–83.

Majid Yazdani and James Henderson. 2015. A model of zero-shot learning of spoken language understanding. In *EMNLP*, pages 244–249.

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 *WWW*, pages 1373–1384.