Zero-shot User Intent Detection via Capsule Neural Networks

Preprint · April 2018

DOI: 10.13140/RG.2.2.11739.46889

5 authors, including:

Congying Xia
University of Illinois at Chicago
26 PUBLICATIONS **205** CITATIONS
SEE PROFILE

Chenwei Zhang
Amazon
63 PUBLICATIONS **621** CITATIONS
SEE PROFILE

Yi Chang
Merck & Co.
177 PUBLICATIONS **3,992** CITATIONS
SEE PROFILE

Philip S. Yu
University of Illinois at Chicago
1,536 PUBLICATIONS **78,550** CITATIONS
SEE PROFILE

Zero-shot User Intent Detection via Capsule Neural Networks

Congying Xia[†*], **Chenwei Zhang**[†*], **Xiaohui Yan**[‡], **Yi Chang**[‡], **Philip S. Yu**[†]
[†]Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607 USA
[‡]Huawei Technologies, San Jose, CA 95050 USA
{cxia8,czhang99,psyu}@uic.edu, {yanxiaohui2,yi.chang}@huawei.com

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 Echo, Google Home and Cortana from Microsoft 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; Zhang et al., 2017) 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 GetWeather and PlayMusic 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 for these intents. Moreover, it's labor-intensive and time-consuming to obtain utterance labels for these 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.

[*]Indicates Equal Contribution

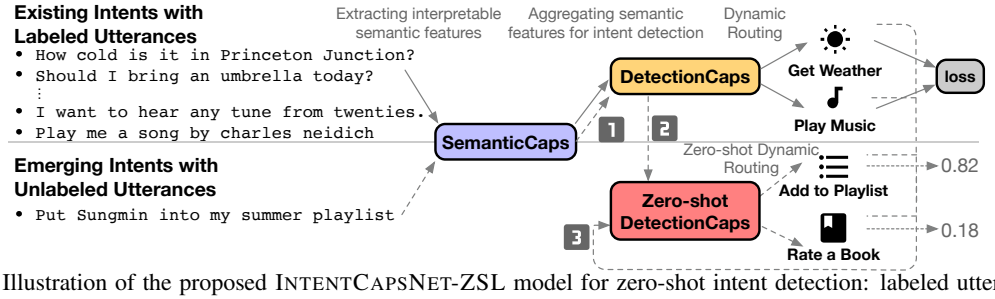

Figure 1: Illustration of the proposed INTENTCAPSNET-ZSL model for zero-shot intent detection: labeled utterances with existing intents like GetWeather and PlayMusic are used to train an intent detection classifier among existing intents, in which SemanticCaps extract intepretable semantic features and 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.

Such resources require extra efforts for annotation; it is usually hard to find a universal schema (e.g. PlayMusic may have music name as an attribute, which GetWeather does not have). Compatibility-based methods for zero-shot intent detection (Chen et al., 2016; Kumar et al., 2017) assume the capability 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. Upadhyay et al. (2018) shows that multilingual corpus takes a synergistic effect in zero-shot natural language understanding.

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 face is likely to be where the nose is in the center). The vector representation not only empowers flexibility (noses/faces with similar properties have similar orientations), but also makes the capsule model efficient in encoding the viewpoint variance (noses/faces from different viewpoints share a nose-face geometrical relation).

The aforementioned properties of capsule models are appealing for intent modeling: low-level semantic features such as the get_action, time and city_name contribute to a more abstract intents (GetWeather) collectively. A semantic feature, which includes quite different expressions e.g. "How"/"What...looks like"/"Is it going to ..." for the get_action in GetWeather, may contribute more to GetWeather than other semantic features like play_action/musician_name do. 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 intent detection 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: (i) expanding capsule neural networks to text modeling, specifically, to intent modeling by extracting and aggregating semantics from utterances in a hierarchical manner; (ii) proposing a novel and effective capsule-based model for zero-shot intent detection; (iii) 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 the 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 for simplicity. However, utterances with multiple intents can be handled by segmenting them into single-intent snippets using sequential tagging tools such as 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 dataset $\{(x, y)\}$ where each $y \in Y$, the zero-shot intent detection task aims to detect an utterance $x_{emerging}$ which belongs to one of L emerging intents $Z = \{z_1, z_2, ..., z_L\}$ 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 intents for intent detection; IntentCapsNet-ZSL that gives zero-shot learning ability to IntentCapsNet for discriminating emerging intents. 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 that utilize semantic features extracted by SemanticCaps, the existing routing information on existing intents as well as similarities between existing and emerging intent label embeddings to discriminate emerging intents.

3.1 SemanticCaps

In (Sabour et al., 2017), convolution-based PrimaryCaps are introduced as the first layer of capsules to obtain different vectorized features from the raw input image (e.g. detecting nose/eyes in vector forms from the raw image of a human face). While

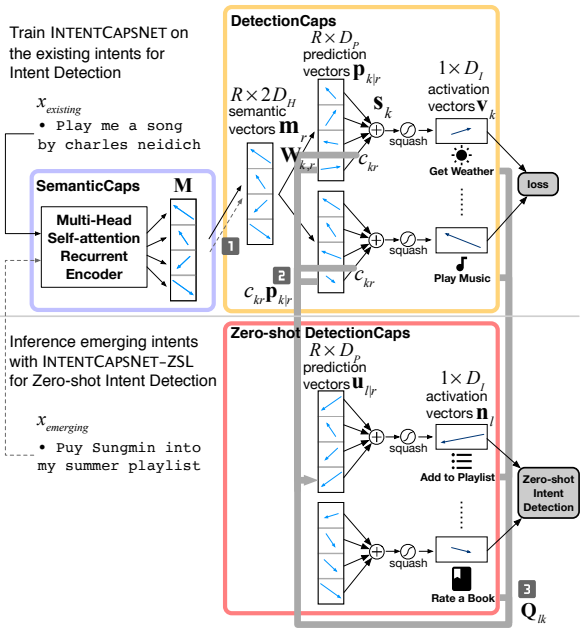

Figure 2: The architecture of IntentCapsNet and IntentCapsNet-ZSL. During training, utterances with existing intents are fed into the SemanticCaps which output vectorized semantic features, i.e. semantic vectors. Then DetectionCaps combines these features into higher-level prediction vectors and outputs 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.

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 extracts a semantic feature that is not necessarily proximate in the utterance.

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

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

For each word $\mathbf{w_t}$, we concatenate each forward hidden state $\vec{\mathbf{h}}_t$ 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 attention head is encouraged to be attentive to a specific semantic feature of the utterance, such as certain sets of keywords or phrases of 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 calculated 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:

$$\mathbf{M} = \mathbf{AH}, \quad (3)$$

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.

When trained with proper regularization that encourages discrepancies among different self-attention heads (introduced in Equation (8)), each semantic vector will have a distinguishable orientation as its corresponding attention is attentive to one unique semantic feature of the utterance. The norm of the vector indicates the probability of the existence of a semantic feature in the utterance. 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, traditional neural network methods usually concatenate the vectors together or use max/mean-pooling strategy to keep the most salient scalar feature. As capsule models are good at learning a global linear manifold between low-level vectorized representations and high-level ones via an unsupervised dynamic routing-by-agreement mechanism, 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. The routing-by-agreement mechanism gives a large scalar output only when the SemanticCaps send an extracted semantic feature to DetectionCaps with a strong agreement of their predictions to high-level intent.

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

$$\mathbf{p}_{k|r} = \mathbf{m}_r \mathbf{W}_{k,r}, \quad (4)$$

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 the prediction vectors:

$$\mathbf{s}_k = \sum_r^R c_{kr} \mathbf{p}_{k|r}, \quad (5)$$

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.

The dynamic routing-by-agreement mechanism, initially introduced in (Sabour et al., 2017), is briefly recalled in Algorithm 1. As shown in this algorithm, b_{kr} is the initial logit representing the log prior probabilities that semantic capsule r coupled to intent capsule k. 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:

$$\mathbf{v}_k = \frac{\|\mathbf{s}_k\|^2}{1 + \|\mathbf{s}_k\|^2} \frac{\mathbf{s}_k}{\|\mathbf{s}_k\|}, \quad (6)$$

where the orientation of the activation vector $\mathbf{v}_k$ represents intent properties while its norm indi-

Algorithm 1 Dynamic routing algorithm

1: **procedure** DYNAMIC ROUTING($\mathbf{p}_{k|r}$, $iter$)
2: for all semantic capsule r and intent capsule k: $b_{kr} \leftarrow 0$.
3: **for** $iter$ iterations **do**
4: for all SemanticCaps r: $c_r \leftarrow \mathrm{softmax}(\mathbf{b}_r)$
5: for all DetectionCaps k: $s_k \leftarrow \Sigma_r c_{kr} \mathbf{p}_{k|r}$
6: for all DetectionCaps k: $v_k = \mathrm{squash}(\mathbf{s}_k)$
7: for all SemanticCaps r and DetectionCaps k: $b_{kr} \leftarrow b_{kr} + \mathbf{p}_{k|r} \cdot \mathbf{v}_k$
8: **end for**
9: Return $\mathbf{v}_k$
10: **end procedure**

cates the probability that an intent exists. 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 an appropriate subsequent DetectionCaps.

Max-margin loss for existing intent existence A max-margin loss function is used during the training phase on each labeled utterance:

$$\mathcal{L}_{existing} = \sum_{k=1}^{K} \{ [\![y = y_k]\!] \cdot \max(0, m^+ - \|\mathbf{v}_k\|)^2 + \lambda [\![y \neq y_k]\!] \cdot \max(0, \|\mathbf{v}_k\| - m^-)^2 \}, \quad (7)$$

where $[\![]\!]$ is an indicator function, y is the ground truth intent label for the utterance x, λ is a down-weighting coefficient, m_k^+ and m_k^- are margins.

The final objective for training considers both the max-margin loss as the first term, as well as a regularization term that encourages that each self-attention head to be attentive to a different semantic feature of the utterance:

$$\mathcal{L}_{train} = \mathcal{L}_{existing} + \alpha ||\mathbf{A}\mathbf{A}^T - I||_F^2, \quad (8)$$

where 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 utilize the following components for knowledge transfer from existing intents to emerging intents:

- Semantic Exaction Behavior: as SemanticCaps are trained to extract semantic features from utterances with various existing intents, a self-attention head which has similar extraction behavior may help transfer knowledge. For example, a self-attention head that extracts the "play" action mentioned by `play/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/include/I want to have` in the beginning of an utterance with an emerging intent AddtoPlaylist.
- Routing information: the coupling coefficient c_{kr} learned by DetectionCaps in a totally unsupervised fashion embodies 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 is helpful in routing the word `add` to AddtoPlaylist.
- Intent Label Similarity: 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:

$$\mathbf{g}_{k,r} = c_{kr} \mathbf{p}_{k|r}, \quad (9)$$

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 $\mathbf{Q} \in \mathbb{R}^{L \times K}$. The similarity between an emerging intent $z_l \in Z$ and an existing intent $y_k \in Y$ is computed as:

$$q_{lk} = \frac{exp\{-d(\mathbf{e}_{z_l}, \mathbf{e}_{y_k})\}}{\sum_{k=1}^{K} exp\{-d(\mathbf{e}_{z_l}, \mathbf{e}_{y_k})\}}, \quad (10)$$

where

$$d\left(\mathbf{e}_{z_l}, \mathbf{e}_{y_k}\right) = \left(\mathbf{e}_{z_l} - \mathbf{e}_{y_k}\right)^T \Sigma^{-1} \left(\mathbf{e}_{z_l} - \mathbf{e}_{y_k}\right). \tag{11}$$

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

$$\mathbf{u}_{l|r} = \sum_{k=1}^{K} q_{mk} \mathbf{g}_{k,r}. \tag{12}$$

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 $\mathbf{n}_l$. Thanks to the vector representation of activity vectors, we can easily classify the utterance of emerging intents by choosing the activation vector with the largest norm $\hat{y} = \arg\max_{z_l \in Z} \|\mathbf{n}_l\|$.

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 1: Dataset statistics.

4 Experiment Setup

To demonstrate the effectiveness of our proposed models, we apply the proposed 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 datasets: SNIPS Natural Language Understanding benchmark (SNIPS-NLU) and a Commercial Voice Assistant (CVA) dataset. The statistical information on two datasets are summarized in Table 1.

SNIPS-NLU[1] is an English natural language corpus collected in a crowdsourced fashion to benchmark the performance of voice assistants. Seven different intents are included in this dataset: SearchCreativeWork, GetWeather, BookRestaurant, PlayMusic, AddToPlaylist, RateBook and SearchScreeningEvent.

CVA is a Chinese natural language corpus collected anonymously from a commercial voice assistant on smart phones. There are 100 types of intents in this dataset, such as TurnOn4g, ModifyRingtones, and IncreaseScreenBrightness.

Baselines We first compare the proposed capsule-based model INTENTCAPSNET with other 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 baselines that adopts different zero-shot learning strategies: 1) DeViSE (Frome et al., 2013) learns a linear compatibility function between the utterance space and the intent space, so as to find the most compatible emerging intent label for an utterance; 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., 2016) uses CNN to extract character-level sentences 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.

Implementation Details We pre-trained two word embedding models, an English one with D_W =

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

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 2: 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.

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., 2016)	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-ttention	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 3: 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.

300 for SNIPS-NLU and an Chinese one with D_W = 200 for CVA. We use three fold cross-validation to choose the following hyperparameters: the forward/backward hidden states of the Bi-LSTM in SemanticCaps have a hidden size D_H of 32 for SNIPS-NLU and 200 for CVA. We set the self-attention head number R and its hidden size D_A to 3 and 20 in SNIPS-NLU, while 8 and 100 are used for CVA. 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. The scale hyper-parameter σ for the similarity is set to 4 for SNIPS-NLU and 1 for CVA. 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 regularization coefficient α is set as 0.0001 for SNIPS-NLU and 0.01 for CVA. 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 2, 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. These results demonstrate the novelty of the proposed capsule-based model INTENTCAPSNET in intent modeling, where the SemanticCaps extract semantic features and DetectionCaps aggregate the extracted semantics via the dynamic routing-by-agreement mechanism.

Also, we report results on zero-shot intention detection task in Table 3, where our model INTENTCAPSNET-ZSL outperforms other baselines that adopt different zero-shot learning strategies: DeVise, CMT are compatibility models that learn the mapping from utterances to intents directly. CDSSM and Zero-shot DNN translate both the utterance and the intent to a new space to learn a scoring function. While 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 derives intent representations explicitly to detect emerging intents without labeled utterances. 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 3. "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 3 we can see that all modules (self-attention, Bi-LSTM, and the self-attention regularizer) 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.

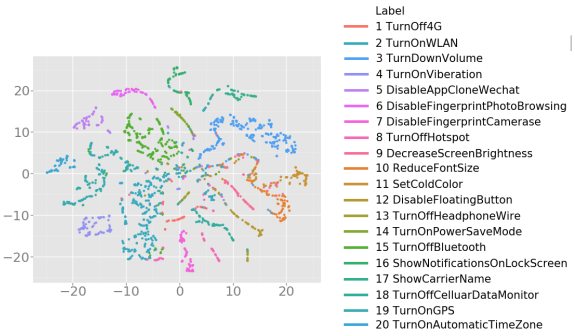

Figure 3: t-SNE visualization of normal activation vectors of utterances with 20 emerging intents in CVA.

Discriminative Emerging Intent Representation Besides quantitative evidences supporting the effectiveness of the INTENTCAPSNET-ZSL, we visualize activation vectors of emerging intents to show that INTENTCAPSNET-ZSL learns discriminative intent representations for each emerging intents. The visualization result of 20 emerging intents on CVA datasets is presented in Figure 3.

Since the activation vectors of utterances with emerging intents are of high dimension and we are interested in studying 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.

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

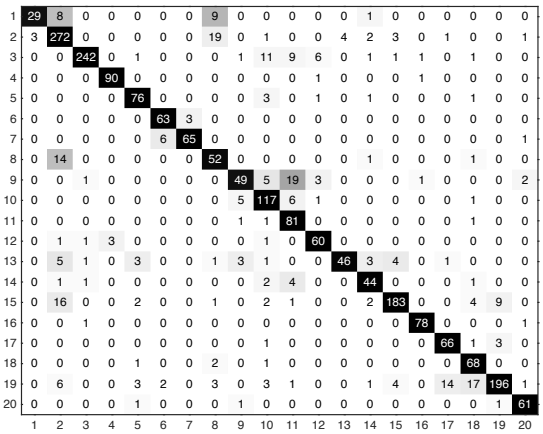

Figure 4: Confusion matrix of emerging intents on CVA.

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.

The confusion matrix for CVA is shown in Figure 4, which shows that most emerging intents are well classified by our proposed model INTENTCAPSNET-ZSL. 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.

From Table 4 we can see that each self-attention head almost always focuses on one unique semantic feature of the utterance. For example, in the in-

Existing Intent: PlayMusic
• Play Action
open up music on last fm
play music by charlie adams from
i want to hear any tune from twenties
open up music on last fm
use spotify to play greatest songs from kailash kher
• Musician Name
play a ballad by owen pallett from seventies on slacker
play kurt cobain ballad tunes
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 a novel called best hits live save children speed live
find fields of sacrifice movie
i m looking for music of nashville season saga
show me television show children in need rocks
i want to read book lion sleeps tonight
• Creative Work Name
i m looking for music of nashville season saga
please find me platinum box ii song ?
show me a picture called heart like a hurricane
i m looking to find suryavanshi
where can i buy a photograph called feel love ?

Table 4: Attentions on utterances with existing intents on SNIPS-NLU.

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 liked ports of call i d give it a out of
i give ruled britannia a rating of five out of
• Book Name
give the televised morality series a one
i give the previous novel one out of stars
i would give dead man falling points and a best rating of
i want to give the coming of the terraphiles a rating of
the chronicle charlie peace earns stars from me
• Rating Score
rate the book strumpet city zero for
rate the grisly wife three points out of five
i would give this current chronicle three points
this saga deserves a score of four
the current essay gets four points
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 the broken wave on lo ms dance
put sungmin into my summer playlist
i want tanya stephens in my soul classics to playlist
add new wave blues to my push button funk playlist
• Playlist Name
add an album to my list la mejor msica dance
add exorcising ghosts to joys thrash attack playlist
can you add danny carey to my masters of metal playlist
can you put this tune onto latin dance cardio ?
i want to put a copy of this tune into skatepark punks

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

tent 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 no named entities are 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`, `show`, and `want to`.

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. From Table 5 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 with existing and emerging intents.

Knowledge transfer via intent similarity Beside extracting semantic features and utilizing existing routing information for knowledge transfer, we use similarities between existing intents and emerging intents to transfer vote vectors from IntentCapsNet to IntentCapsNet-ZSL. We study the distribution of similarities of each emerging intents to existing intents in Figure 5.

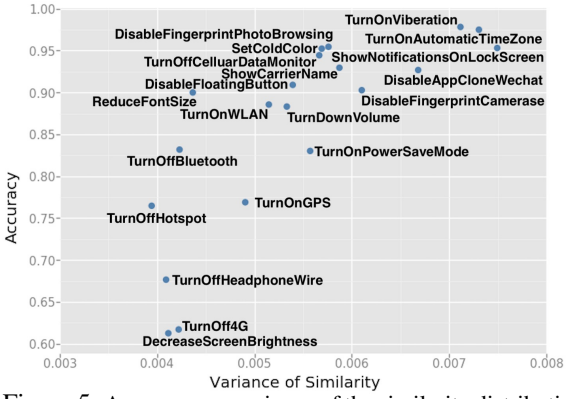

Figure 5: Accuracy vs. variance of the similarity distribution for 20 emerging intents in CVA dataset.

The y axis is the zero-shot detection accuracy on each emerging intent in the CVA dataset. The x axis measures $\text{var}(\mathbf{q}_l)$, which is 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. 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). For example, ShowNotificationsOnLockScreen (No.16) has a high variance of 0.0073 and Accuracy=0.95. Table 6 further shows that ShowNotificationsOnLockScreen (No.16)'s similarity distribution on the top-5 existing intents, shown as red bars, is highly skewed towards its top-1 choice. By transferring knowledge more from more similar intents, such property contributes to its highest accuracy among all the emerging intents.

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. For example, ReduceFontSize (No.10) has a relative low variances (0.0044) but it is still able to achieve Accuracy=0.90. In Table 6, we find the blue bars have less significant, but still a skewed distribution towards the top-1 intent. However, for 5 out of 20 intents with $\text{var}(\mathbf{q}_l) < 0.005$, our model achieves less satisfied results: DecreaseScreenBrightness (No. 9) has a low variance of 0.0041, which is similar as 0.0044 for ReduceFontSize (No.10), but its distribution (yellow bars in Table 6) is more evenly distributed when compared with the distributions of other two intents aforementioned. This partially results in its low accuracy (0.61).

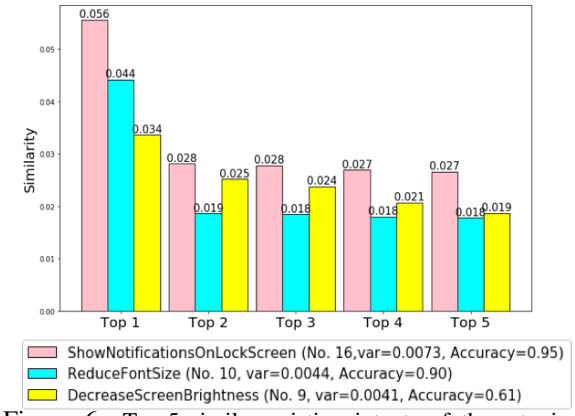

Figure 6: Top-5 similar existing intents of three typical emerging intents.

Generally, we find that relying on a less number of existing intents, but with more confidence, is still a preferred knowledge transfer strategy that contributes to the good performance of INTENTCAPSNET-ZSL.

7 Conclusions

In this paper, a capsule-based model, namely INTENTCAPSNET, is first introduced to harness the advantages of capsule models for intent modeling in a hierarchical manner: semantic features are extracted from the utterances and aggregated to obtain intent representations via the dynamic routing-by-agreement mechanism. The proposed INTENTCAPSNET-ZSL model further introduces zero-shot learning ability to the capsule model via various means of knowledge transfer from existing intents to emerging intents. INTENTCAPSNET-ZSL has the ability to discriminate 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.

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. 2016. Zero-shot learning of intent embeddings for expansion by convolutional deep structured semantic models. In *ICASSP*, pages 6045–6049. IEEE.

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

Emmanuel Ferreira, Bassam Jabaian, and Fabrice Lefèvre. 2015b. Zero-shot semantic parser for spoken language understanding. In *Interspeech*.

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 *International Conference on Artificial Neural Networks*, pages 44–51. Springer.

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. ACM.

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*, ICML 2001, 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. *ICLR*.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. *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.

Shyam Upadhyay, Manaal Faruqui, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck. 2018. (almost) zero-shot cross-lingual spoken language understanding. In *ICASSP*.

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. IEEE.

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

Chenwei Zhang, Nan Du, Wei Fan, Yaliang Li, Chun-Ta Lu, and Philip S Yu. 2017. Bringing semantic structures to user intent detection in online medical queries. In *IEEE Big Data*, pages 1019–1026.

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.