Dynamic Semantic Matching and Aggregation Network for Few-shot Intent Detection

Hoang Nguyen[1], **Chenwei Zhang**[2], **Congying Xia**[1], **Philip S. Yu**[1]
[1] Department of Computer Science, University of Illinois at Chicago, Chicago, IL, USA
[2] Amazon, Seattle, WA, USA
{hnguy7,cxia8,psyu}@uic.edu, cwzhang@amazon.com

Abstract

Few-shot Intent Detection is challenging due to the scarcity of available annotated utterances. Although recent works demonstrate that multi-level matching plays an important role in transferring learned knowledge from seen training classes to novel testing classes, they rely on a static similarity measure and overly fine-grained matching components. These limitations inhibit generalizing capability towards Generalized Few-shot Learning settings where both seen and novel classes are co-existent. In this paper, we propose a novel Semantic Matching and Aggregation Network where semantic components are distilled from utterances via multi-head self-attention with additional dynamic regularization constraints. These semantic components capture high-level information, resulting in more effective matching between instances. Our multi-perspective matching method provides a comprehensive matching measure to enhance representations of both labeled and unlabeled instances. We also propose a more challenging evaluation setting that considers classification on the joint all-class label space. Extensive experimental results demonstrate the effectiveness of our method. Our code and data are publicly available [1] .

1 Introduction

Intent Detection (ID) is a crucial task in natural language understanding, whose objective is to extract underlying intents behind the given utterances. The extracted intents could provide further contexts for further downstream Natural Language Processing tasks such as dialogue state tracking or question answering. Unlike traditional text classification, ID is challenging for two main reasons (1) Utterances are usually short and diversely expressed, (2) Emerging intents occur continuously, especially across different domains (Liu et al., 2019a).

Despite recent advances, state-of-the-art ID methods (Haihong et al., 2019; Goo et al., 2018) require a large amount of annotated data to achieve competitive performance. This requirement inhibits models' capability in generalizing to newly emerging intents with no or limited annotations during inference. Re-training or fine-tuning large models on few samples of emerging classes could easily lead to overfitting problems.

Motivated by human capability in correctly categorizing new classes with only a few examples (Lake et al., 2011; Gidaris and Komodakis, 2018), few-shot learning (FSL) paradigms are adopted to tackle the scarcity problems of emerging classes. FSL methods take advantage of a small set of labeled examples (support set) to learn how to discriminate unlabeled samples (query samples) between classes, even those not seen during training.

Recent works in FSL (Sun et al., 2019; Ye and Ling, 2019) focus on learning the matching information between the labeled samples (support) and the unlabeled samples (query) to provide additional contextual information for instance-level representations, leading to effective prototype representation. However, these methods only extract similarity based on fine-grained word semantics, failing to capture the diverse expressions of users' utterances. This problem could further lead to overfitting either to seen intents or novel intents, especially in the challenging Generalized Few-shot Intent Detection (GFSID) setting where both seen and novel intents are existent in a joint label space during inference. Instead, matching support and query samples on coarser-grained semantic components could provide additional informative contexts beyond word levels. For instance, two utterances *"i need to get a table at a pub with southeastern cuisine"* and *"book a spot for six friends"* share a sim-

[1] https://github.com/nhhoang96/Semantic_Matching

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 1209–1218
November 16 - 20, 2020.

ilar intent label *"Book Restaurant"*. While word-level semantics might find similar action words as *"get"* and *"book"*, these words do not necessarily contribute to the correct intent findings. Instead, coarser-grained semantics such as *"get a table"* and *"book a spot"* could provide further hints to identify *"Book Restaurant"* intent.

As semantic components (SC) could be effectively extracted from multi-head self-attention, matching these SC between support and query can enhance both query and support representations, leading to improvements in generalization from seen training classes to unseen testing classes. To further enhance the dynamics of extracted SC across various domains and diversely expressed utterances, we introduce additional head regularizations. In addition, to overcome the insufficiency of a single similarity measure for matching sentences with diverse semantics, a more comprehensive matching method is further explored.

Our main contribution is summarized as follows:

- We propose a Semantic Matching and Aggregation Network that automatically extracts multiple semantic components from support and query sentences via multi-head self-attention. Additional regularizations are introduced to (1) encourage extracted heads to attend to all words of utterances and (2) encourage semantic alignment between utterances with similar intent labels.

- Comprehensive multi-perspective matching is proposed to reduce reliance on a single fixed similarity measure and enhance generalizability towards Generalized Few-shot Learning setting (GFSL).

- We also propose a more challenging but realistic FSL and GFSL evaluation setting.

2 Related Work

Few-shot Learning Few-shot learning refers to problems where classifiers are required to generalize to unseen classes with only a few training examples per class (Chen et al., 2019). To overcome challenges of potential overfitting, most FSL methods adopt meta-learning approach where knowledge is extracted and transferred across multiple tasks. There are two major approaches towards FSL: (1) metric-based approach whose goal is to learn feature extractor that extract and generalize to emerging classes (Vinyals et al., 2016; Snell et al., 2017; Sung et al., 2018), and (2) optimization-based approach that aims to optimize model parameters from few samples (Santoro et al., 2016; Finn et al., 2017; Ravi and Larochelle, 2017; Mishra et al., 2018). In this work, we focus mostly on metric-based learning approach. Specifically, we extend Prototypical Network (PN) (Snell et al., 2017) in which prototypes are not only represented by support samples but also matching information between support and query samples.

Traditionally, FSL methods are evaluated in episodic procedure due to the major principle that test and train conditions must match (Vinyals et al., 2016). Each episode represents a meta-learning task in which the models explicitly "learn to learn" minimize the loss on an unlabeled/ query set given the support/ labeled set. However, we claim that this evaluation is lack of practicality for two main reasons. First, evaluation on random samples could not help us understand the strengths or weaknesses of the model. For instance, if the trained model overfits a subset of novel classes, it is impossible to pinpoint the overfitting classes with episodic evaluation. Secondly, in realistic applications, there is a need to categorize unlabeled samples into one of the novel/joint classes, rather than a set of sampled classes. Episodic testing does not provide an end-to-end systematic evaluation. Therefore, in our work, we propose a more challenging but realistic non-episodic evaluation setting where unlabeled samples are only inferred once with a probablility distribution over a fixed set of classes in novel or joint label space.

Sentence Matching Recent FSL works adopt multi-level matching and aggregation methods to improve FSL performance (Gao et al., 2019; Sun et al., 2019; Ye and Ling, 2019). Instead of constructing prototypes purely from support samples, recent works integrate matching information between support and query samples on multiple levels. Gao et al. (2019) introduces feature-level and instance-level attention. Sun et al. (2019) introduces additional word-level attention and proposes more advanced multi-cross attention on instance-level. On the other hand, Ye and Ling (2019) adopts soft matching between support and query samples to build local context representation for both support and query samples. These methods have been proven effective in few-shot relation classification tasks. However, they rely on overly fine-grained

level matching which potentially causes overfitting problems towards either seen or unseen set of classes. Our work mainly differs in two aspects: (1) Comprehensive multi-perspective matching for information matching and (2) Matching on coarser-grained semantic-component levels that are extracted dynamically for effective knowledge transfer, especially in GFSL settings.

3 Problem Formulation

In this section, we provide definitions for both Few-shot Intent Detection (FSID) and GFSID task. Traditional FSL task is defined as C-way K-shot classification task in which classifier performs a series of tasks during both training and inference, which involves C randomly chosen classes with only K labeled samples from each class ($K \leq 5$). These $C \cdot K$ samples are named as support samples. This series of tasks are repeated via episodes (Vinyals et al., 2016). In each episode, the objective is to correctly classify unlabeled samples (query samples) by using only the support samples.

We denote seen label space as Y_s, novel label space as Y_n, and $Y_s \cap Y_n = \varnothing$. Given the seen labels (Y_s), we define $D_s = \{(x_1, y_1), (x_2, y_2), ...(x_{N_s}, y_{N_s})\}$, where N_s denotes the total number of seen samples and (x, y) denotes a pair of utterance and intent label. Similarly, $D_n = \{(x_1, y_1), (x_2, y_2), ...(x_{N_n}, y_{N_n})\}$.

Given an unlabeled utterance x, the objective of FSID is to maximize correct prediction for x within the novel label subspace Y_n as summarized in (1).

$$\hat{y} = \underset{y \in Y_n}{argmax}\ p(y|x, D_n) \tag{1}$$

For GFSID, there exists an additional joint label space $Y_j = Y_s \cup Y_n$. Unlike FSID, GFSID is more challenging as the test samples could come from either seen or novel sample space. The objective function is modified as follows.

$$\hat{y} = \underset{y \in Y_j}{argmax}\ p(y|x, D_j) \tag{2}$$

4 Methodology

In this section, we introduce our proposed architecture. Specifically, we divide the framework into 3 main components: Semantic Encoder, Semantic Matching & Aggregation, Instance Aggregation & Class Matching as illustrated in Figure 1.

4.1 Semantic Encoder

The objective of Semantic Encoder (SE) is to extract semantic components from the given support or query instances. Given an input support or query instance $\mathbf{x} = [x_1, x_2, ..., x_T]$ with T words, SE first maps each word into a d_w dimensional word embedding. Pre-trained embedding such as Glove (Pennington et al., 2014), or even contextualized embedding BERT (Devlin et al., 2018) could be elevated. In our work, we adopt pre-trained FastText embedding (Bojanowski et al., 2017).

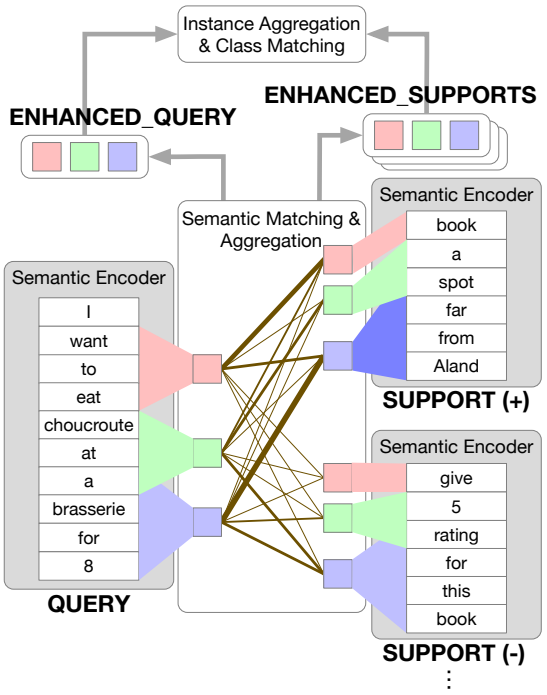

Figure 1: Illustration of the proposed Semantic Matching and Aggregation Model for few-shot intent detection. Semantic Components extracted from Semantic Encoder capture high-level semantics beyond word semantic level. Matching these components is more effective than word-by-word matching as contextual phrases are further taken into consideration and non-essential words do not distract the matching functions.

To capture semantic and syntactic information of the given instance, we adopt self-attentive semantic encoder inspired by multi-head self-attention in (Lin et al., 2017). Specifically, we first use Bi-Directional Long short-term Memory (Bi-LSTM) to capture contextual information between words within a sentence.

$$\begin{aligned} \overrightarrow{\mathbf{h}_t} &= \overrightarrow{LSTM}(\mathbf{w}_t, \overrightarrow{\mathbf{h}_{t-1}}) \\ \overleftarrow{\mathbf{h}_t} &= \overleftarrow{LSTM}(\mathbf{w}_t, \overleftarrow{\mathbf{h}_{t+1}}) \end{aligned} \tag{3}$$

The hidden representation of $\mathbf{x}$ (denoted as $\mathbf{H} \in \mathbb{R}^{T \times 2d_h}$) is a concatenation of both forward and backward hidden states where d_h is the hidden size.

$$\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_T] \tag{4}$$

To capture more fine-grained signals other than sentence vector representation, self-attention mechanism is adopted to extract important semantic components of the sentence. Each semantic component, denoted as "head", is learned from the hidden state $\mathbf{H}$ via multi-layer perceptrons (MLP).

$$\mathbf{A} = softmax(\mathbf{W}_{s2} tanh(\mathbf{W}_{s1}\mathbf{H}^T)) \quad (5)$$

where $\mathbf{W}_{s1}$, $\mathbf{W}_{s2}$ are the learning weights with dimension of $\mathbb{R}^{d_a \times 2d_h}$ and $\mathbb{R}^{r \times d_a}$. d_a and r can be simply seen as the hidden size and output size of the embedded feed-forward network. r represents the number of heads or important features that the network extracts from the given sentence. The r-head representation $\mathbf{M} \in \mathbb{R}^{r \times 2d_h}$ is a product of attention matrix and the obtained hidden states $\mathbf{M} = \mathbf{AH}$.

Additional regularization terms are introduced to enforce (1) Each head focuses on different aspects of a sentence, (2) All words in an utterance are covered by the extracted heads, (3) Head distribution between query and support with the same intent labels should be similar to one another. These regularized terms are optimized together with the query classification loss ($\mathcal{L}_{class}$) to further improve the model's performance. In summary, our training loss is summarized as follows.

$$\mathcal{L} = \mathcal{L}_{class} + \alpha\mathcal{L}_{self\_attn} + \beta\mathcal{L}_{uniform} + \gamma\mathcal{L}_{discr} \quad (6)$$

where α, β, γ are hyperparameters.

Self-attention regularization Additional regularization term is needed to enforce that each attention head focuses on different semantic components of the utterance. The most intuitive approach is to minimize the number of "attended" tokens for each head, forcing each head vector to attend to a single aspect of the given sentence (Lin et al., 2017).

$$\mathcal{L}_{self\_attn} = ||(\mathbf{A}\mathbf{A}^T - \mathbf{I})||_F^2 \quad (7)$$

where $\mathbf{A}$ denotes the obtained attention matrix from SE and $||\bullet||_F^2$ denotes Frobenius matrix norm.

Head uniform regularization To ensure that all words of a given utterance are covered by at least one head obtained by multi-head self-attention, we minimize the Kullback-Leiber (KL) divergence between the word probability distribution over all heads ($\sum_{i=1}^{r} \mathbf{A}_i$) and a uniform distribution U.

$$\mathcal{L}_{uniform} = D_{KL}(p(\sum_{i=1}^{r} \mathbf{A}_i)||U) \quad (8)$$

Head uniform regularization is introduced to increase robustness and dynamic of extraction behavior by covering even rare words that are not widely used in utterances.

Head distribution regularization To encourage semantic alignment between support and query samples of the same intent, we minimize the KL divergence in terms of head distributions among those with similar intents while maximizing KL divergence among those that are different.

$$\mathcal{L}_{discr} = (\hat{Y}_Q = Y_S) D_{KL}(p(\sum_{i=1}^{L_Q} \mathbf{A}_Q)||p(\sum_{j=1}^{L_S} \mathbf{A}_S)) - (\hat{Y}_Q \neq Y_S) D_{KL}(p(\sum_{i=1}^{L_Q} \mathbf{A}_Q)||p(\sum_{j=1}^{L_S} \mathbf{A}_S)) \quad (9)$$

L_Q and L_S denote the lengths of query and support sentences respectively. $\hat{Y}_Q$ and Y_S denote predicted query label and ground truth support label respectively. This regularization allows for dynamic multi-head self-attention extraction behavior by incorporating query predicted label from downstream task into the objective function.

4.2 Semantic Matching & Aggregation

In order to enrich representations for both support and query instances, given SCs extracted from Semantic Encoder, we introduce Semantic Matching & Aggregation module to capture and aggregate matching local contexts between support and query via SCs. Specifically, our module is made up of two components: (1) Multi-perspective Semantic Matching and (2) Semantic Aggregation.

Extracted head representations from SE (matrix $\mathbf{M}$) for both support and query samples are used in this module.We denote representations of k-th support sample as $\mathbf{S}_k = [\mathbf{M}_{s_k}^1, \mathbf{M}_{s_k}^2, ..., \mathbf{M}_{s_k}^r]$ and query sample as $\mathbf{Q} = [\mathbf{M}_q^1, \mathbf{M}_q^2, ..., \mathbf{M}_q^r]$ respectively , where r denotes the number of extracted heads from SE. This module is applied to both support and query samples to build an enhanced instance representation $\hat{\mathbf{S}}_k$ and $\hat{\mathbf{Q}}$. For simplicity, we only define the one-way matching ($\mathbf{S}_k \rightarrow \mathbf{Q}$).

4.2.1 Multi-perspective Semantic Matching

Following (Wang et al., 2017), we define the multi-perspective matching function f_m between two vectors as $\mathbf{m} = f_m(\mathbf{v}_1, \mathbf{v}_2; \mathbf{W})$ where $\mathbf{W} \in \mathbb{R}^{l \times d}$ is a trainable weight parameter. l is a hyperparameter defining the number of perspectives. Each

perspective in vector $\mathbf{m}$ is a cosine similarity between weighted vectors $\mathbf{v}_1$ and $\mathbf{v}_2$. In other words, $m_k = cosine(\mathbf{W}_k \circ \mathbf{v}_1, \mathbf{W}_k \circ \mathbf{v}_2)$ where $\circ$ defines element-wise multiplication.

We define four different components of multi-perspective matching method as follows.

Head-wise Matching Each head's forward and backward contextualized embedding of $\mathbf{S}_k$ are compared with the corresponding head's forward and backward contextual embedding of $\mathbf{Q}$.

$$\begin{aligned}\overrightarrow{\mathbf{m}_i^{head\_wise}} &= f_m(\overrightarrow{\mathbf{M}_{s_k}^i}, \overrightarrow{\mathbf{M}_q^i}; \mathbf{W}^1) \\ \overleftarrow{\mathbf{m}_i^{head\_wise}} &= f_m(\overleftarrow{\mathbf{M}_{s_k}^i}, \overleftarrow{\mathbf{M}_q^i}; \mathbf{W}^2)\end{aligned} \tag{10}$$

Max-pooling Matching Each head's forward and backward contextualized embedding of $\mathbf{S}_k$ is compared with all heads' forward and backward contextual embedding of $\mathbf{Q}$. However, only the maximum value in each dimension is extracted and retained in the matching vector.

$$\begin{aligned}\overrightarrow{\mathbf{m}_i^{max}} &= \underset{j\in(1..r)}{max}\, f_m(\overrightarrow{\mathbf{M}_{s_k}^i}, \overrightarrow{\mathbf{M}_q^j}; \mathbf{W}^3) \\ \overleftarrow{\mathbf{m}_i^{max}} &= \underset{j\in(1..r)}{max}\, f_m(\overleftarrow{\mathbf{M}_{s_k}^i}, \overleftarrow{\mathbf{M}_q^j}; \mathbf{W}^4)\end{aligned} \tag{11}$$

Attentive Matching Unlike Max-Pooling matching, Attentive Matching is divided into two steps (1) Head representative is aggregated via similarity scores between different heads of each support and query sample (2) Matching head representative and the support heads. For similarity measure, cosine function is utilized.

$$\begin{aligned}\overrightarrow{\beta_{i,j}} &= cosine(\overrightarrow{\mathbf{M}_{s_k}^i}, \overrightarrow{\mathbf{M}_q^j}) \\ \overleftarrow{\beta_{i,j}} &= cosine(\overleftarrow{\mathbf{M}_{s_k}^i}, \overleftarrow{\mathbf{M}_q^j})\end{aligned} \tag{12}$$

Head representative is defined as a weighted sum of all query heads.

$$\begin{aligned}\overrightarrow{\mathbf{M}_i^{rep}} &= \frac{\sum_{j=1}^r \overrightarrow{\beta_{i,j}} \cdot \overrightarrow{\mathbf{M}_q^j}}{\sum_{j=1}^r \overrightarrow{\beta_{i,j}}} \\ \overleftarrow{\mathbf{M}_i^{rep}} &= \frac{\sum_{j=1}^r \overleftarrow{\beta_{i,j}} \cdot \overleftarrow{\mathbf{M}_q^j}}{\sum_{j=1}^r \overleftarrow{\beta_{i,j}}}\end{aligned} \tag{13}$$

The computed head representative is compared with each head's contextualized embedding of $\mathbf{S}_k$.

$$\begin{aligned}\overrightarrow{\mathbf{m}_i^{attn}} &= f_m(\overrightarrow{\mathbf{M}_{s_k}^i}, \overrightarrow{\mathbf{M}_i^{rep}}; \mathbf{W}^5) \\ \overleftarrow{\mathbf{m}_i^{attn}} &= f_m(\overleftarrow{\mathbf{M}_{s_k}^i}, \overleftarrow{\mathbf{M}_i^{rep}}; \mathbf{W}^6)\end{aligned} \tag{14}$$

Max-Attentive Matching Similar to Attentive Matching, Max-Attentive extracts head representative in Equation (13). Instead of doing the pairwise matching, Max-Attentive conducts max-pooling between $\mathbf{M}_j^{rep}$ and $\mathbf{M}_{s_k}^i$.

$$\begin{aligned}\overrightarrow{\mathbf{m}_i^{max\_attn}} &= \underset{j\in(1..r)}{max}\, f_m(\overrightarrow{\mathbf{M}_{s_k}^i}, \overrightarrow{\mathbf{M}_j^{rep}}; \mathbf{W}^7) \\ \overleftarrow{\mathbf{m}_i^{max\_attn}} &= \underset{j\in(1..r)}{max}\, f_m(\overleftarrow{\mathbf{M}_{s_k}^i}, \overleftarrow{\mathbf{M}_j^{rep}}; \mathbf{W}^8)\end{aligned} \tag{15}$$

4.2.2 Semantic Aggregation

In order to aggregate the matched representation into a single instance representation, we use another Bi-LSTM whose input is a concatenation of matched representation in previous sections.

$$\begin{aligned}\overrightarrow{\hat{\mathbf{S}}_k} &= LSTM(\overrightarrow{\mathbf{m}_i^{head\_wise}} \oplus \overrightarrow{\mathbf{m}_i^{max\_attn}} \oplus \overrightarrow{\mathbf{m}_i^{attn}} \oplus \overrightarrow{\mathbf{m}_i^{max}}) \\ \overleftarrow{\hat{\mathbf{S}}_k} &= LSTM(\overleftarrow{\mathbf{m}_i^{head\_wise}} \oplus \overleftarrow{\mathbf{m}_i^{max\_attn}} \oplus \overleftarrow{\mathbf{m}_i^{attn}} \oplus \overleftarrow{\mathbf{m}_i^{max}})\end{aligned} \tag{16}$$

where $\oplus$ denotes concatenation operation.

Similarly, we obtain the final representation of query with reverse matching ($\mathbf{Q} \rightarrow \mathbf{S}_k$) where $\{\hat{\mathbf{Q}}, \hat{\mathbf{S}}_k\} \in \mathbb{R}^{2d_h}$.

4.3 Instance Aggregation & Class Matching

As indicated in previous works, when class label covers diverse semantics, each support instance contributes differently to the class prototype given the query instance. Therefore, we replace the mean operation over all support instances of PN with attentive aggregation. Attention weight for each support instance $\hat{\mathbf{S}}_k$ is learned via a MLP.

$$\alpha_k = \mathbf{W}_9^T(ReLU(\mathbf{W}_{10}[\hat{\mathbf{S}}_k \oplus \hat{\mathbf{Q}}])) \tag{17}$$

Support prototype ($\hat{\mathbf{S}}$) is computed as a weighted sum aggregation via support attention weight and each k-th support instance representation.

$$\hat{\mathbf{S}} = \sum_{k=1}^{K} softmax(\alpha_k)\hat{\mathbf{S}}_k \tag{18}$$

Another MLP is used as class matching function by using support prototype and query representation.

$$\hat{Y} = \mathbf{W}_9^T(ReLU(\mathbf{W}_{10}[\hat{\mathbf{S}} \oplus \hat{\mathbf{Q}}])) \tag{19}$$

Weights $\mathbf{W}_9 \in \mathbb{R}^{d_h}$ and $\mathbf{W}_{10} \in \mathbb{R}^{d_h \times 4d_h}$ are shared between instance aggregation (Equation (17)) and class matching (Equation (19)) for optimal performance (Ye and Ling, 2019).

Table 1: Details of SNIPS and NLUE (Fold 1) datasets.

	SNIPS	NLUE
# Seen classes ($\|Y_s\|$)	5	48
# Novel classes ($\|Y_n\|$)	2	16
# Seen samples (N_s)	7887	6393
# Novel samples (N_n)	769	274
# Joint samples (N_j)	2688	1873
# Seen samples per class ($\bar{N}_s$)	1577.4	133.2
# Novel samples per class ($\bar{N}_n$)	384.5	17.1
# Joint samples per class ($\bar{N}_j$)	384.0	29.3

Table 2: Hyperparameters for both datasets.

	d_a	d_h	r	L	α	β	γ
SNIPS	20	64	4	5	0.0001	1e-5	0.01
NLUE	20	64	4	5	1e-5	1e-5	0.001

5 Experiments

5.1 Dataset

We evaluate our proposed model on two real-world datasets for the GFSID task: SNIPS-NLU (SNIPS) and NLU-Evaluation Dataset (NLUE). Both datasets are widely as benchmarks for Natural Language Understanding tasks. Statistics of both datasets are summarized in Table 1.

For each dataset, we define Seen-Novel-Joint datasets. To build a joint dataset (D_j), we aggregate 20% of seen intent utterances with novel intent utterances. The remaining seen intent utterances (80%) are used as training data (reported N_s in Table 1). The support samples (1 or 5 shots) are randomly sampled in advance and not counted in either N_s, N_n or N_j.

SNIPS-NLU: Following (Xia et al., 2018),we select two intents (RateBook and AddToPlaylist) as novel/ emerging intents and the other five intents as seen intents.

NLUE: Following (Liu et al., 2019b), we utilize a subset of utterances covering 64 intents. We randomly choose 16 intents as unseen intents while the remaining 48 intents are considered seen.

5.2 Baselines

We compare our model with several traditional FSL models, and specifically metric-based network models. For fair comparison and consistency, we implement our SE proposed in Section 4.1 for all considered baselines. Final instance embedding is obtained as a mean operation over all heads. The only exception is HAPN and MLMAN as they require local matching (i.e. word matching) modules. In that case, we use output of Bi-LSTM (in Equation (4)) and enhance it with the head regularization term (Section 4.1) during training.

- **Matching Network (MN)** (Vinyals et al., 2016): few-shot learning paradigm mapping samples to labels via attention mechanism.
- **Prototypical Network (PN)** (Snell et al., 2017): few-shot method categorizing samples via Euclidean distance from class prototypes.
- **Relation Network (RN)** (Sung et al., 2018) few-shot model that uses neural network to learn deep metric known as relation scores.
- **Hybrid Attention-based Prototypical Network (HATT)** (Gao et al., 2019): initial few-shot learning model that integrates feature-level attention and instance-level attention between support and query samples.
- **Hierarchical Prototypical Network (HAPN)** (Sun et al., 2019): few-shot learning paradigm that extracts similarity on all feature, word and instance levels.
- **Multi-level Matching and Aggregation Network (MLMAN)** (Ye and Ling, 2019): multi-level matching approach exploiting both fusion and dot product similarity on local/ word level to enhance instance representation.

5.3 Implementation Details

We use 3-fold cross-validation to tune all of the hyperparameters based on S-J accuracy on SNIPS and Fold 1 of NLUE datasets as summarized in Table 2. Pre-trained FastText word embedding is used to initialize word embedding and stays fixed during both training and testing for fair comparison between our proposed model and baselines. We train each model over 1000 randomly sampled episodes with learning rate of 0.0001. The number of query samples (N_Q) for each episode is 20.

Following (Shi et al., 2019), we evaluate our models on overall Seen-Joint (S-J) and Seen-Novel (S-N) accuracy. Reported S-J accuracy denotes GFSID evaluation result while S-N indicates traditional FSID results. Reported h-accuracy is a harmonic mean between S-J and S-N accuracy to evaluate the stability of the overall model in both GFSID and FSID settings.

Episodic Evaluation Traditional FSL methods are evaluated in episodes due to the major principle that test and train conditions (C-way K-shot) must match (Vinyals et al., 2016). On SNIPS dataset,we

Table 3: Experimental result on SNIPS dataset.

	1-shot						5-shot					
	Non-episodic (noneps)			Episodic (eps)			Non-episodic (noneps)			Episodic (eps)		
Model	S-J	S-N	h_acc	S-J	S-N	h_acc	S-J	S-N	h_acc	S-J	S-N	h_acc
MN	73.5	86.99	79.68	82.67	85.97	84.29	77.31	90.12	83.22	84.6	90.12	87.27
PN	71.61	94.67	81.54	87.04	89.91	88.45	85.31	93.11	89.04	91.05	92.96	92.00
RN	74.94	88.14	81.01	85.63	87.63	86.62	64.09	87.99	74.16	79.25	83.86	81.49
HATT	71.54	**93.76**	81.16	84.51	**93.55**	88.80	**86.53**	94.15	**90.18**	**91.85**	93.98	**92.90**
MLMAN	**78.61**	94.41	**85.79**	**87.77**	92.48	**90.06**	79.58	**95.06**	86.64	89.27	94.13	91.64
HAPN	74.33	91.42	81.99	85.37	91.52	88.34	86.19	92.85	89.40	89.4	**94.32**	91.79
Ours	**81.85**	**95.84**	**88.29**	**88.1**	**95.48**	**91.64**	**87.87**	**97.01**	**92.21**	**93.18**	**96.81**	**94.96**

Table 4: Experimental result on NLUE dataset.

	1-shot						5-shot					
	Non-episodic (noneps)			Episodic (eps)			Non-episodic (noneps)			Episodic (eps)		
Model	S-J	S-N	h_acc	S-J	S-N	h_acc	S-J	S-N	h_acc	S-J	S-N	h_acc
MN	62.3	35.4	45.15	76.21	58.16	65.97	56.27	52.55	54.35	78.85	73.69	76.18
PN	62.63	36.86	46.41	80.78	58.44	67.82	66.2	59.49	62.67	85.13	79.39	82.16
RN	56.75	27.74	37.26	73.57	49.47	59.16	46.5	34.31	39.49	75.23	62.15	68.07
HATT	**64.01**	34.67	44.98	81.39	58.47	68.05	67.86	61.15	64.33	78.41	74.74	76.53
MLMAN	63.12	41.61	**51.60**	**82.65**	60.64	69.95	60.7	59.49	60.09	84.45	76.7	80.39
HAPN	60.44	**41.78**	49.41	82.00	**62.39**	**70.86**	**68.34**	**64.6**	**66.42**	**84.75**	**80.11**	**82.36**
Ours	**66.1**	**44.11**	**52.91**	**89.54**	**62.81**	**73.83**	**72.18**	**66.96**	**69.47**	**87.76**	**81.12**	**84.31**

conduct experiments with $K = \{1, 5\}$ and $C = 2$ with 5 random seed initialization and report average accuracy in Table 3. For NLUE dataset, we average accuracy over 10 Folds with similar K and $C = 5$. The sampling procedure for GFSL is conducted in a similar way as (Shi et al., 2019).

Non-episodic Evaluation As mentioned in Section 2, Episodic Evaluation is lack of practicality and does not provide an end-to-end system evaluation. Therefore, we also evaluate the models on our proposed non-episodic procedure where unlabeled samples are only inferred once and the predicted probability distribution is over all Y_n or Y_j label space.

5.4 Experimental Results

As we observe from Table 3 and 4, our proposed model outperforms the previous baselines by a large margin in both episodic and non-episodic evaluations on both datasets. Our model also observes a consistent stability between FSID and GFSID tasks across both datasets.

All of the models observe a major decrease in accuracy when evaluated on our challenging non-episodic evaluation as compared to the traditional episodic procedure. Specifically, GFSID tasks are mostly affected by non-episodic evaluation (around 10% S-J accuracy drop in both datasets). On SNIPS dataset, since both non-episodic and episodic evaluations on S-N are conducted as 2-way 1-shot or 2-way 5-shot, the reported accuracy is almost similar. However, on the other hand, as C and $|Y_n|$ or $|Y_j|$ are different (5 vs 16 or 64) on NLUE dataset, we observe significant differences in reported S-N accuracy across all models.

On NLUE dataset, S-N accuracy is consistently lower than S-J accuracy across all models. This is mainly because the hyperparameter N_Q is higher than the $\bar{N}_n$ on NLUE ($20 > 17.1$), affecting the training and evaluation on D_n.

5.5 Ablation Study

Multi-perspective Matching To evaluate the effectiveness of our Semantic Matching Module, we conduct further studies on individual components of our head matching. Table 5 shows that using only a single matching function is not sufficient to capture matching information between query and support samples. By aggregating all four matching methods, we observe a consistent improvement in both FSL and GFSL evaluations.

Head Matching vs Word Matching As introduced in Section 4, each head aims to extract a SC that covers a different aspect of a given sentence. To evaluate the effectiveness of head matching, we compare it with its corresponding word matching. In word matching, the hidden state embedding ($\mathbf{h}_i$) from Bi-LSTM is used for comparison rather than

Table 5: H-acc comparison on individual components of Semantic Matching module on SNIPS dataset.

	1-shot		5-shot	
	noneps	**eps**	**noneps**	**eps**
Head-wise	85.40	88.84	90.86	93.63
Max-pooling	85.54	88.87	90.79	93.63
Attentive	87.06	90.85	92.09	94.04
Max-attentive	87.37	90.87	92.18	94.37
Full Model	**88.29**	**91.64**	**92.21**	**94.96**

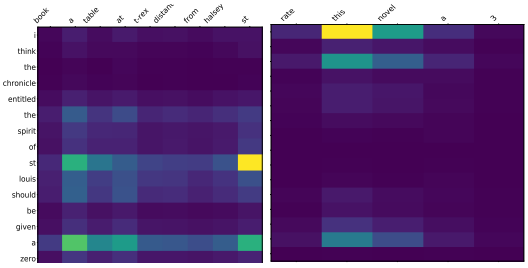

Figure 2: Word level matching. Y-axis denotes words of a sample query utterance *"i think the chronicle entitled the spirit of st louis should be given a zero rating"* and X-axis (left) denote words of negative support utterance *"book a table at t-rex distant from halsey st"* and X-axis (right) denotes positive support *"rate this novel a 3"*. The label for query and positive support is "Rate Book" and the negative support's label is "Book Restaurant". The lighter color implies higher attention score.

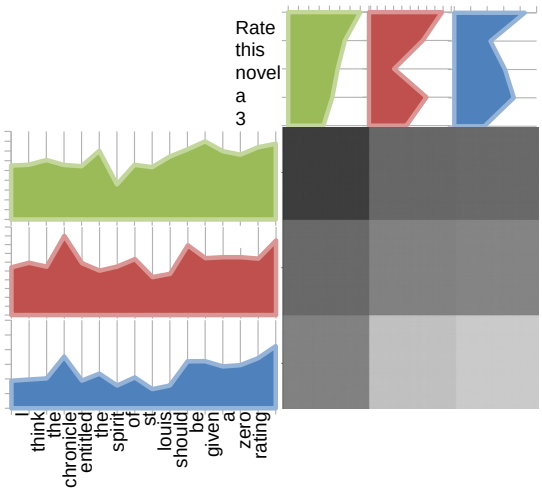

Figure 3: Head level matching between the same query and positive support utterance. Y-axis denotes 3 heads extracted from query utterance labeled with the word distribution of each head. X-axis denotes 3 heads extracted from positive support utterance with similar label technique. Different curve colors are used to denote head indexes. The lighter color of each cell in 3x3 square matrix denotes the higher attention score.

the head representation ($\mathbf{M}_i$). In addition, instead of head-wise matching, we compare each word forward and backward embedding of sentence $\mathbf{S}_k$ with the last (forward) and first (backward) embedding of sentence $\mathbf{Q}$ where T_q denotes the last word in sentence $\mathbf{Q}$.

$$\begin{aligned}\overrightarrow{\mathbf{m}_i^{word\_wise}} &= f_m(\overrightarrow{\mathbf{h}_{s_k}^i}, \overrightarrow{\mathbf{h}_q^{T_q}}; \mathbf{W}^1) \\ \overleftarrow{\mathbf{m}_i^{word\_wise}} &= f_m(\overleftarrow{\mathbf{h}_{s_k}^i}, \overleftarrow{\mathbf{h}_q^1}; \mathbf{W}^2)\end{aligned} \tag{20}$$

Figure 2 illustrates an example when overly fine-grained matching sends the wrong matching signal, causing mis-classification for a query sample. Although *"st"* exists in both query and negative support sample, it contains different meanings depending on contexts (*"street"* vs *"saint"*) and does not contribute to the correct intent *"Rate Book"*. However, word matching assigns high matching score, leading to mis-classification of query sample as *"Book Restaurant"* intent. As shown in the right part of Figure 2 word matching fails to identify indicative matching information with positive support sample (i.e. *"rate"* vs *"rating"*). This observation indicates that matching on the overly fine-grained word level semantics could lead to overfitting problems as only query samples of high word overlaps with support samples could yield high matching score. As utterances are diversely expressed, word-level semantic is insufficient to capture similarity between different utterances of the same intent.

On the other hand, when we use extracted heads for matching, as observed from Figure 3, the importance of *"st"* is significantly downplayed. Instead, query heads focus on extracting different aspects of the query: verb *"should"*, *"be"* (head 1), object target *"chronicle"* (head 2), rating-related information *"ratings"* (head 3). These key components are also captured in the positive support: target object(*"novel"*) and rating keyword (*"rate"*). As clearly indicated in Figure 3, the head with color blue of query and positive support sample that both extract important rating-related keywords (*"rating"* vs *"rate"*) achieve high matching score.

This observation confirms our intuitions (1) Each SC extracts essential high-level semantics of a given utterance, (2) Without sharing word-level similarity, essential keywords for intent label of query samples are extracted and matched with those from support samples (i.e. *"rating"* vs *"rate"*) via intermediate semantic component level. Further qualitative results in Table 6 validate the effectiveness of head-vs-head matching as it outperforms its word matching counterpart in all evaluation scenarios. This is mainly because the semantic

components extracted from SE effectively capture the most important words in the given utterances as observed in a sample query utterance, reducing the necessity to focus on matching irrelevant words.

Table 6: H-accuracy evaluation on head matching vs word matching and regularization terms effectiveness on SNIPS dataset.

	1-shot		5-shot	
	noneps	**eps**	**noneps**	**eps**
Word Match	85.94	90.87	90.11	93.22
No $\mathcal{L}_{cross}$	87.58	91.2	92.06	94.84
No $\mathcal{L}_{self}$	87.96	91.60	92.10	94.89
No $\mathcal{L}_{uniform}$	87.65	91.17	92.09	94.92
Full Model	**88.29**	**91.64**	**92.21**	**94.96**

Head Matching Regularization As observed from Table 6, adding each additional regularization term boosts both GFSL and FSL performance. $\mathcal{L}_{cross}$ contributes most to the overall performance improvement. It is mainly due to its ability to align head distribution of samples with the same class label. Therefore, each extracted head could focus more on an indicative signal of the intent label.

6 Conclusions

In this paper, we propose an effective Semantic Matching and Aggregation Network for few-shot intent detection. Semantic components extracted from multi-head self-attention capture higher level contextual information beyond the word level, enhancing model's generalizability towards both seen and novel intents, especially when utterances are diversely expressed. Comprehensive multi-perspective matching method thoroughly exploits the similarity between query and support samples for further robust representations. In this work, we also propose a more challenging but realistic non-episodic evaluation for both FSL and GFSL beyond traditional setting. Our model achieves the state-of-the-art performance in both evaluation settings for SNIPS and NLUE benchmark datasets. Further studies of more dynamic semantic extraction and effectively synthesized matching techniques are our desired future work.

Acknowledgments

We thank you reviewers for insightful feedback. This work is supported in part by NSF under grants III-1763325, III-1909323, and SaTC-1930941.

We would like to acknowledge the use of the facilities of the High Performance Computing Division and High Performance Research and Development Group at the National Center for Atmospheric Research and the use of computational resources (doi:10.5065/D6RX99HX) at the NCAR-Wyoming Supercomputing Center provided by the National Science Foundation and the State of Wyoming, and supported by NCAR's Computational and Information Systems Laboratory.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Wang, and Jia-Bin Huang. 2019. A closer look at few-shot classification. In *International Conference on Learning Representations*.

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

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org.

Tianyu Gao, Xu Han, Zhiyuan Liu, and Maosong Sun. 2019. Hybrid attention-based prototypical networks for noisy few-shot relation classification. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence,(AAAI-19), New York, USA*.

Spyros Gidaris and Nikos Komodakis. 2018. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4367–4375.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757.

E Haihong, Peiqing Niu, Zhongfu Chen, and Meina Song. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5467–5471.

Brenden Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua Tenenbaum. 2011. One shot learning of simple visual concepts. In *Proceedings of the annual meeting of the cognitive science society*, volume 33.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

Han Liu, Xiaotong Zhang, Lu Fan, Xuandi Fu, Qimai Li, Xiao-Ming Wu, and Albert YS Lam. 2019a. Reconstructing capsule networks for zero-shot intent classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4801–4811.

Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2019b. Benchmarking natural language understanding services for building conversational agents. In *10th International Workshop on Spoken Dialogue Systems Technology 2019*.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. 2018. A simple neural attentive meta-learner. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

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

Sachin Ravi and Hugo Larochelle. 2017. Optimization as a model for few-shot learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. 2016. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850.

Xiahan Shi, Leonard Salewski, Martin Schiegg, Zeynep Akata, and Max Welling. 2019. Relational generalized few-shot learning. *arXiv preprint arXiv:1907.09557*.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087.

Shengli Sun, Qingfeng Sun, Kevin Zhou, and Tengchao Lv. 2019. Hierarchical attention prototypical networks for few-shot text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 476–485, Hong Kong, China. Association for Computational Linguistics.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4144–4150.

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

Zhi-Xiu Ye and Zhen-Hua Ling. 2019. Multi-level matching and aggregation network for few-shot relation classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2872–2881, Florence, Italy. Association for Computational Linguistics.