Exploiting the Matching Information in the Support Set for Few Shot Event Classification

Viet Dac Lai[1], Thien Huu Nguyen[1], Frank Dernoncourt[2]

[1]University of Oregon
[2]Adobe Research

PAKDD 2020

Event Classification

- A task in Information Extraction
- To classify event in text
- Applications: Knowledge Base Population, Question Answering

Examples

The police shot the drug dealer.
$\Rightarrow$ **Attack**
The undercover cop was fired.
? **End-position, Attack, Die**

Extending event classification to new domains

Extending:

- Dealing with **unseen** event types
- Supervised learning fails to transfer knowledge

Supervised learning setting:

- Predefined set of classes
- Trained to classify among this set

Extending supervised learning model to unseen event types:

- Annotate more data expensive!
- Transfer knowledge to new event types low performance!

$\Rightarrow$ Reformulate training setting is the key to open to new event types.

Few-shot Learning (FSL) for Unseen Classes

Low cost, high performance setting

- Train on existing data with know event types
- Test on unseen class at **high accuracy**
- **Few** (5,10) examples per class are required

$\Rightarrow$ Extending to new classes at low cost with FSL

Few-Shot Learning vs Supervised Learning

Supervised learning: Given a query instance x, predict the label of x:

$$P(\hat{y} = t_i|x) \quad \text{where } t_i \in T$$

Few shot learning: Given a support set S and a query instance x, predict the label of x

$$P(\hat{y} = t_i|x, S, t_i \in T(S))$$

such that:

$$S = \{(x, y)|y \in T(S)\}$$

Common FSL setting: N-way K-shot

$$|T(S)| = N$$

$$|\{(x_j^i, y^i)|(x_j^i, y^i) \in S\}| = K$$

Training a FSL Model

Data splits:

- Training set: $D^{train} = \{(x, t)|t \in T^{train}\}$
- Testing set: $D^{test} = \{(x, t)|t \in T^{test}\}$

such that $\boxed{T^{train} \cap T^{test} = \emptyset}$

In each training iteration, sample:

- Subset of classes $T' \in T^{train}$
- Support set $S = \{(x, t)|t \in T'\} \subset D^{train}$
- Query set $Q = \{(x, t)|t \in T'\} \subset D^{train}$

such that $\boxed{S \cap Q = \emptyset}$

Prototypical Network

- Feature extraction $v_i^j = f(s_i^j, a_i^j)$
- Class prototype

$$\mathbf{c}_i = \frac{1}{K} \sum_{(s_i^j, a_i^j, t_i) \in S} f(s_i^j, a_i^j)$$

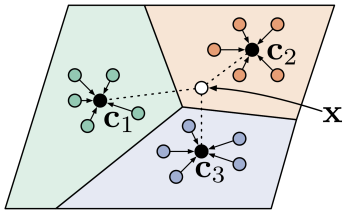


Figure: Prototypical Network. (Snell et al. (2017))

- Distribution using distance function **d**

$$P(y = t_i | (q, p), S) = \frac{\exp(-d(f(q, p), \mathbf{c}_i))}{\sum_{j=1}^{N} \exp(-d(f(s_i^j, a_i^j), \mathbf{c}_j))}$$

- Minimize negative log-likelihood

$$L(Q, S) = - \sum_{(q,p,t) \in Q} \log P(y = t | x, S) \quad (1)$$

Problem with Prototypical Network

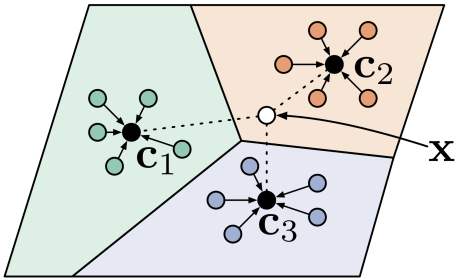
What have been done:

- Intuitively method based on mixtrure model
- Constraint between **support set** and **query set**

What have not been exploited:

- Matching information between **examples within the support set**

Domain matching



Our idea:

- Samples in the same class should be similar
- Samples in the different class should be different

Implementation

- Split the support set S into two smaller sets S^s, S^q
- Formulate an auxiliary FSL problem with S^s, S^q
- Jointly optimize both original and auxiliary FSL problem.

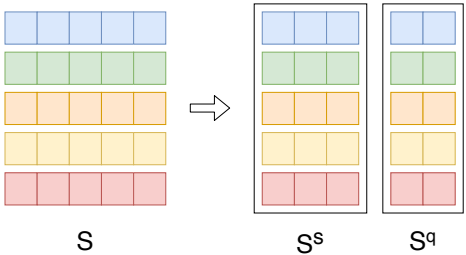


Figure: Split support set into smaller sets

Leave-out-Loss

Compute the loss for the smaller problem

$$L_{LoLoss}(S) = L(S^q, S^s) \tag{2}$$

Final objective function:

$$L = L(Q, S) + \lambda L(S^q, S^s) \tag{3}$$

Observations:

- LoLoss is an identical function to the main objective function

Models

FSL models

- **Proto**: Euclidean distance, averaging prototype [Snell et al.2017]
- **Proto+Att**: Euclidean distance, weighted sum prototype [Gao et al.2019]
- **Matching**: Cosine similarity, averaging prototype [Vinyals et al.2016]
- **Relation**: Learnable distance, averaging prototype [Sung et al.2018]

Encoder:

- CNN Encoder
- Transformer encoder

FSL Training settings

- Dataset: ACE-2005, TAC-KBP
- N-way K-shot few shot learning:
  - N classes (5, 10)
  - K samples per class (5, 10)

  $\Rightarrow$ Support set: $N \times K$ examples
- Query set: $N \times Q$ queries
- Metric: Accuracy

Result: ACE with CNN

FSL Setting	**5 way 5 shot**	**5 way 10 shot**	**10 way 5 shot**	**10 way 10 shot**
Matching	45.81	49.01	30.41	35.66
Matching+LoLoss	**51.78**	**52.64**	**32.48**	**39.15**
Proto	70.92	74.40	57.59	62.67
Proto+LoLoss	**76.98**	**82.19**	**66.92**	**73.63**
Proto+Att	72.26	74.22	57.28	64.36
Proto+Att+LoLoss	**76.93**	**75.59**	**67.54**	**66.70**
Relation	36.33	33.75	24.21	18.04
Relation+LoLoss	**37.86**	**38.52**	**25.99**	**23.47**

Table: Accuracy of event classification with CNN encoder on ACE-2005 dataset . **+LoLoss** indicates the use of the proposed loss.

Result: KBP

Model	5 way 5 shot	10 way 10 shot
Matching	72.78	65.55
Matching+LoLoss	**75.58**	**68.53**
Proto	78.08	73.23
Proto+LoLoss	**78.88**	**74.82**
Proto+Att	75.35	71.28
Proto+Att+LoLoss	**79.93**	**76.37**
Relation	50.97	34.91
Relation+LoLoss	**51.65**	**35.13**

Table: Accuracy of the models with the Transformer encoder on the TAC-KBP test dataset. **+LoLoss** indicates the use of the proposed loss.

Result: Loss

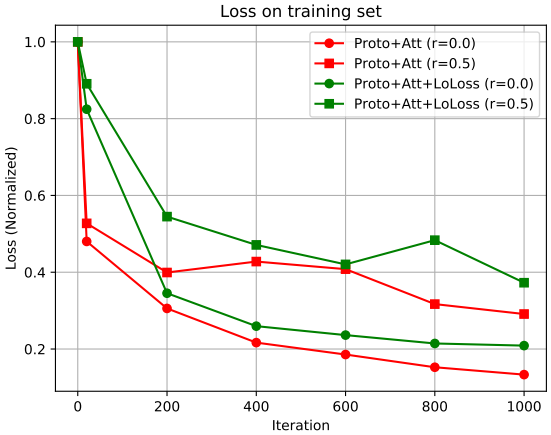


Figure: Comparing the loss on the training data of the models using the Transformer encoder with and without noise.

Conclusion

Extending event classification to new event types:

- Formulate event classification as a few-shot learning problem
- Provide a baseline for FSL event classification
- Propose a novel training signal

Question?