MGIMN: Multi-Grained Interactive Matching Network for Few-shot Text Classification

Jianhai Zhang[1], **Mieradlijiang Maimaiti** [1], Xing Gao[1], Yuanhang Zheng [2], and Ji Zhang [1]

[1]Alibaba DAMO Academy

[2]Department of Computer Science and Technology, Tsinghua University

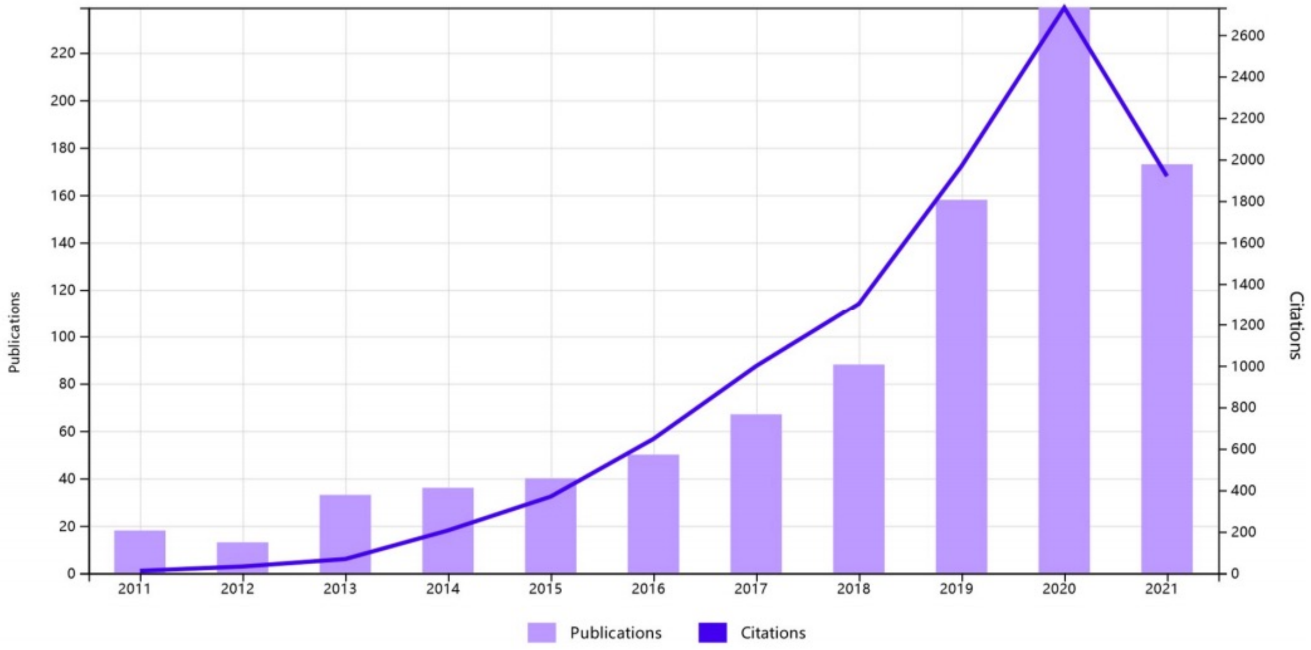
NAACL2022, July. 2022, Seattle Washington

Outline

- Background
- Related work
- Challenge and Motivation
- Methodology
- Experiments
- Conclusions and Future

Background

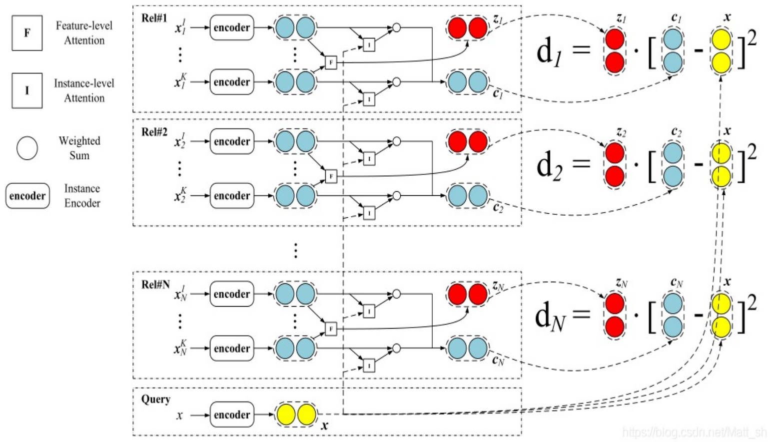
- Text classification is widely used to many realistic scenes, e.g. intention classification, faq-based question answering etc.
- However, the **class space changes** frequently in realistic scenes and the **annotated examples** are usually **scarce** and hard to label.
- Consequently, study of classification with few examples is much important and attracts a lot of attention.



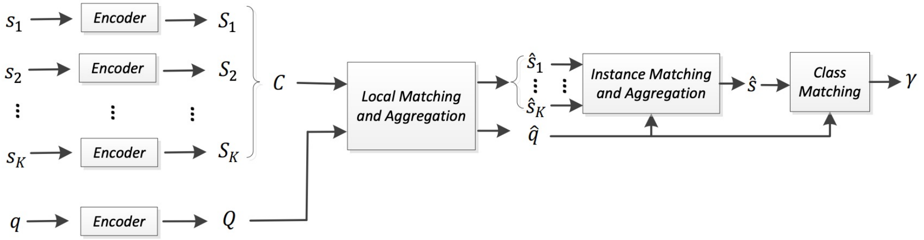
The number of FSL-related papers published

Related work

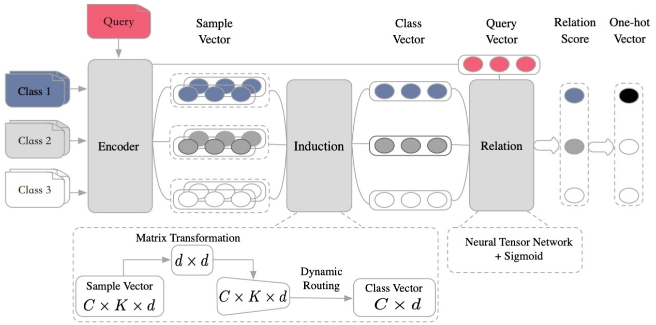
- Data augmentation
- Transfer learning
- Text matching methods
- Meta-learning methods
  - Optimization-based: learning optimization algorithm
  - Model-based: learning model parameters
  - **Metric-based: learning embedding space and metric**
- ....



Proto-hatt(Gao et al., 2019)



MLMAN(Ye and Ling, 2019)



Induction network(Geng et al., 2019)

Challenge and Motivation

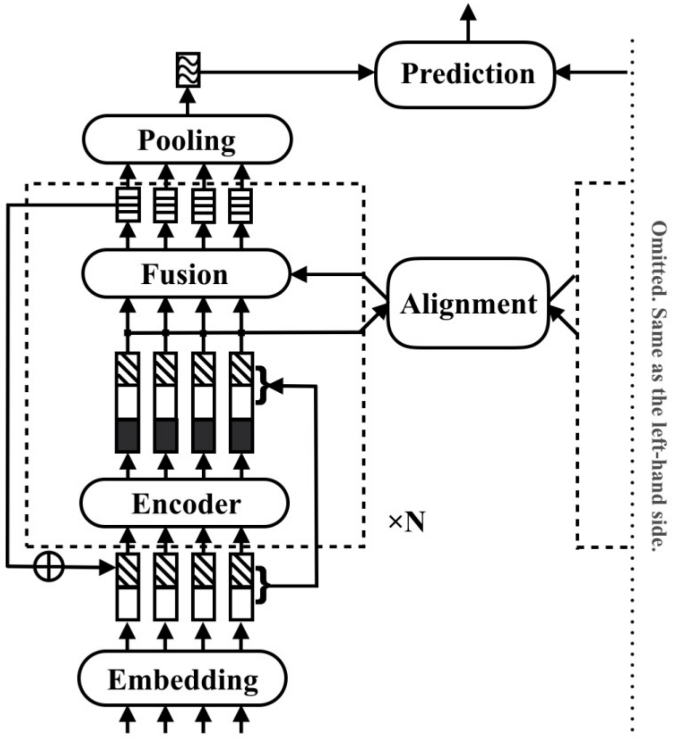
◆ Challenges

- Difficult to use one prototype to represent one class for the diverse expresses.
  e.g. "green streets are healthy streets", "the real heroes of Pakistan" and "what next for Kurdistan ?" are from the same class: "WORLD NEWS"

- Difficult to deploy online due to the worse performance (usually thousands of classes in realistic scenery).

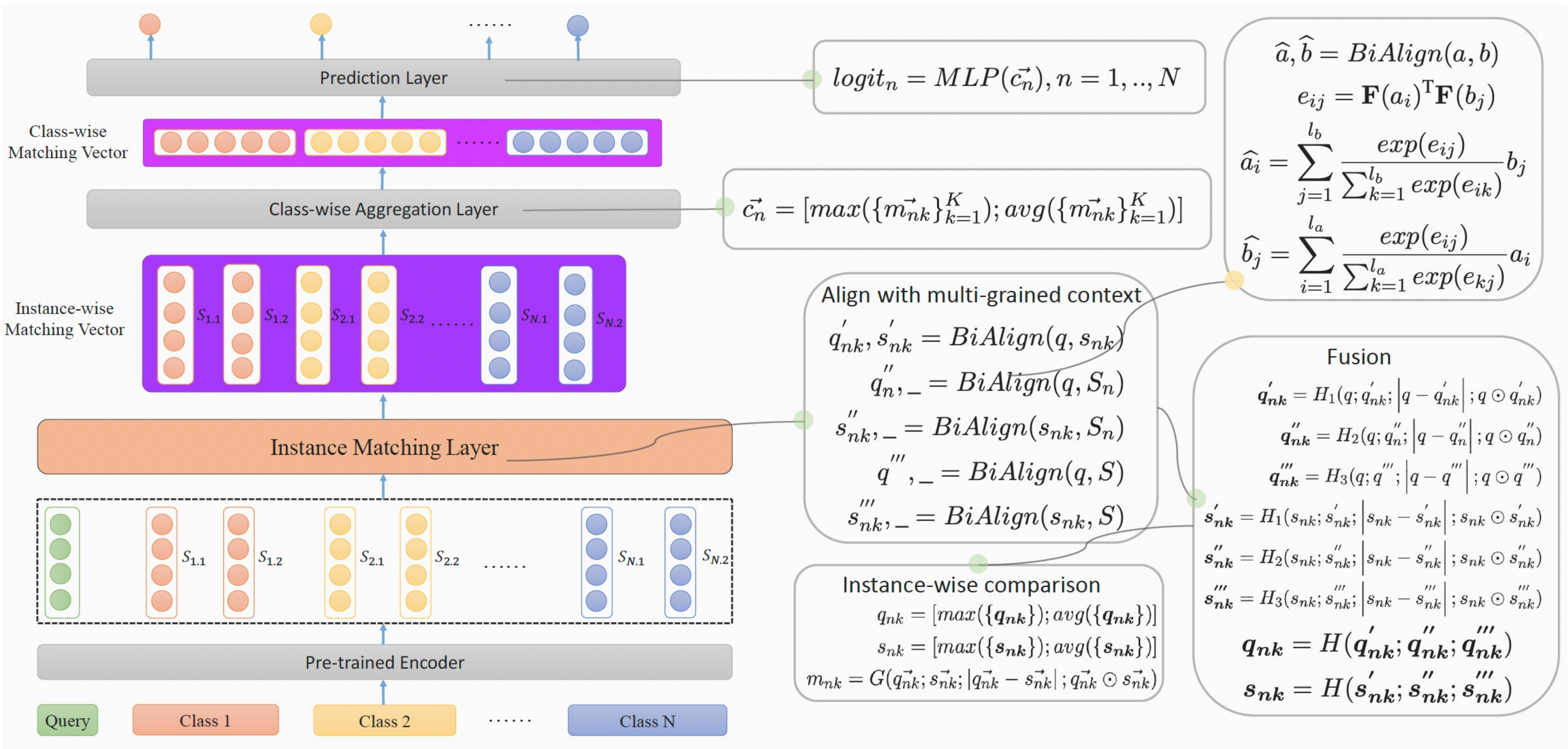
◆ Motivations

- In text matching area, many studies show that the **fine-grained comparison** and **matching** are effective. e.g. RE2(Yang et al.2019) etc.
- In searching area, **retrieval-then-ranking** is usually used to solve the problem of large amount of candidates.



RE2(Yang et al., 2019)

Methodology



- MGIMN consists of four modules: Encoder Layer, **Instance Matching Layer**, **Class Aggregation Layer** and Prediction Layer.
- Our key innovation is to **perform instance-wise comparisons** followed by **class aggregation** instead of constructing **compact representation** (called prototype) which expresses the entire meaning of class.

Methodology

☛ Instance-wise Matching ☛ Class-wise Aggregation

達摩院
ALIBABA DAMO ACADEMY

Instance-wise Matching

Align with multi-grained context

$$q'_{nk}, s'_{nk} = BiAlign(q, s_{nk})$$
$$q''_{n}, \_ = BiAlign(q, S_n)$$
$$s''_{nk}, \_ = BiAlign(s_{nk}, S_n)$$
$$q''', \_ = BiAlign(q, S)$$
$$s'''_{nk}, \_ = BiAlign(s_{nk}, S)$$

For alignment, we also consider their alignments with **global context** information apart from the **local alignment** between **query** and **each support instance.**

Fusion

$$\boldsymbol{q'_{nk}} = H_1(q; q'_{nk}; \left|q - q'_{nk}\right|; q \odot q'_{nk})$$
$$\boldsymbol{q''_{nk}} = H_2(q; q''_{n}; \left|q - q''_{n}\right|; q \odot q''_{n})$$
$$\boldsymbol{q'''_{nk}} = H_3(q; q'''; \left|q - q'''\right|; q \odot q''')$$
$$\boldsymbol{s'_{nk}} = H_1(s_{nk}; s'_{nk}; \left|s_{nk} - s'_{nk}\right|; s_{nk} \odot s'_{nk})$$
$$\boldsymbol{s''_{nk}} = H_2(s_{nk}; s''_{nk}; \left|s_{nk} - s''_{nk}\right|; s_{nk} \odot s''_{nk})$$
$$\boldsymbol{s'''_{nk}} = H_3(s_{nk}; s'''_{nk}; \left|s_{nk} - s'''_{nk}\right|; s_{nk} \odot s'''_{nk})$$
$$\boldsymbol{q_{nk}} = H(\boldsymbol{q'_{nk}}; \boldsymbol{q''_{nk}}; \boldsymbol{q'''_{nk}})$$
$$\boldsymbol{s_{nk}} = H(\boldsymbol{s'_{nk}}; \boldsymbol{s''_{nk}}; \boldsymbol{s'''_{nk}})$$

For fusion, we fuse **original representation** and **three kinds of aligned representation** to **enhance semantics.**

Instance-wise comparison

$$\vec{q_{nk}} = [max(\{\boldsymbol{q_{nk}}\}); avg(\{\boldsymbol{q_{nk}}\})]$$
$$\vec{s_{nk}} = [max(\{\boldsymbol{s_{nk}}\}); avg(\{\boldsymbol{s_{nk}}\})]$$
$$m_{nk} = G(\vec{q_{nk}}; \vec{s_{nk}}; \left|\vec{q_{nk}} - \vec{s_{nk}}\right|; \vec{q_{nk}} \odot \vec{s_{nk}})$$

For comparison, instead of calculating prototype, we perform **fine-grained comparison** between **query** and **support instance**, and achieve instance matching vector.

Class-wise aggregation

$$\vec{c_n} = [max(\{\vec{m_{nk}}\}_{k=1}^{K}); avg(\{\vec{m_{nk}}\}_{k=1}^{K})]$$

This layer **aggregates instance-wise matching vectors** into **class-wise matching vectors** for final prediction.

Experiments

☛ Setting

Datasets	Standard FSL Setting		Generalized FSL Setting						
	#sentences	$C_{tr}/C_{val}/C_{test}$	C	K	#sc	#uc	$\#D_{tr}$	$\#D_{val}$	$\#D_{test}$
OOS	22500	50/50/50	150	5	50	100	7000	1250	1250
Liu	25478	18/18/18	54	5	18	36	8312	450	450
Amzn	3057	106/106/106	318	5	106	212	1043	530	530
Huffpost	41000	14/13/14	41	5	14	27	13860	340	340
FaqIr	1233	17/16/17	50	5	17	33	309	381	381

- Standard FSL Setting: we also **mainly follow the commonly used manner** (Snell et al., 2017 etc.)
- Generalized FSL Setting (Nguyen et al., 2020 etc.): A more challenging-yet-realistic evaluation method. In this setting, we **reform** task a **C-way K-shot classification** in which only subset of classes are seen in training phase.

Experiments

☛ Results

達摩院 ALIBABA DAMO ACADEMY

Main experiments

Methods	OOS			Liu			FaqIr		
	5-way	10-way	GFSL	5-way	10-way	GFSL	5-way	10-way	GFSL
Proto	92.20	87.91	61.94	82.46	73.23	47.66	89.83	81.56	60.78
Matching	89.78	84.41	58.34	78.25	67.45	41.95	86.74	78.77	53.85
Induction	80.44	70.92	34.00	65.58	51.56	24.73	71.62	56.99	20.10
Proto-HATT	92.84	89.11	65.52	82.38	75.29	51.27	85.01	76.17	62.62
MLMAN	95.99	93.41	74.39	87.39	79.82	57.24	94.77	89.49	74.42
MGIMN(ours)	**96.36**	**94.00**	**76.23**	**87.84**	**80.60**	**57.66**	**95.14**	**90.69**	**75.81**

Methods	Amzn		
	5-way	10-way	GFSL
Proto	78.40	69.02	41.03
Matching	75.73	64.17	38.34
Induction	64.02	50.12	20.09
Proto-HATT	78.05	69.00	41.81
MLMAN	85.64	79.39	46.71
MGIMN(ours)	**85.96**	**80.07**	**49.46**

Methods	Huffpost		
	5-way	10-way	GFSL
Proto	51.57	36.74	16.47
Matching	49.77	34.28	14.18
Induction	44.69	29.35	10.40
Proto-HATT	51.23	36.65	16.06
MLMAN	52.76	38.22	16.78
MGIMN(ours)	**54.98**	**40.12**	**19.61**

Ablation Study

Methods	Liu			Huffpost		
	5-way	10-way	GFSL	5-way	10-way	GFSL
MGIMN(ours)	**87.84**	**80.60**	**57.66**	**54.98**	**40.12**	**19.61**
w/o episode	86.22	78.99	56.67	54.14	39.53	18.69
w/o class	84.56	76.89	54.62	54.09	39.10	17.53
w/o instance	87.74	79.93	57.39	53.65	38.86	18.67
w/o instance&class&episode	80.53	70.94	42.54	51.81	37.10	16.48

Retrieval-then-classify

Methods	Liu(c=50)		OOS(c=150)		Amzn(c=318)	
	score	speed	score	speed	score	speed
MGIMN-overall	57.66	315	76.23	757	49.46	1630
RTC-BM25	54.97	55	74.80	56	44.76	58
RTC-oribert	52.93	60	70.55	65	31.09	70
RTC-mgimnbert	56.21	60	75.58	65	46.80	70

Conclusions and Future

- We propose a **novel** few-shot text classification framework (MGIMN), which performs instance-wise matching followed by class-wise aggregation.
- Experimental results show that our method **outperforms** than previous prototype-based methods and classic matching network in both FSL and GFSL evaluation settings.
- In the future, we will make more investigations on the **diverse interactive** ways.

Thanks