Adversarial Connective-exploiting Networks for Implicit Discourse Relation Classification

Lianhui Qin, Zhisong Zhang, Hai Zhao, *Zhiting Hu*, Eric P. Xing

Shubham Jain

Discourse Relations

- Connect linguistic units (like sentences) semantically

- Types:
  - Explicit:

    I like the food, but I am full. (Relation: Comparison)

    Use Connectives

  - Implicit:

    Never mind. You already know the answer.

    Connectives can be inferred

Implicit discourse relation

Units : Never mind. You already know the answer.

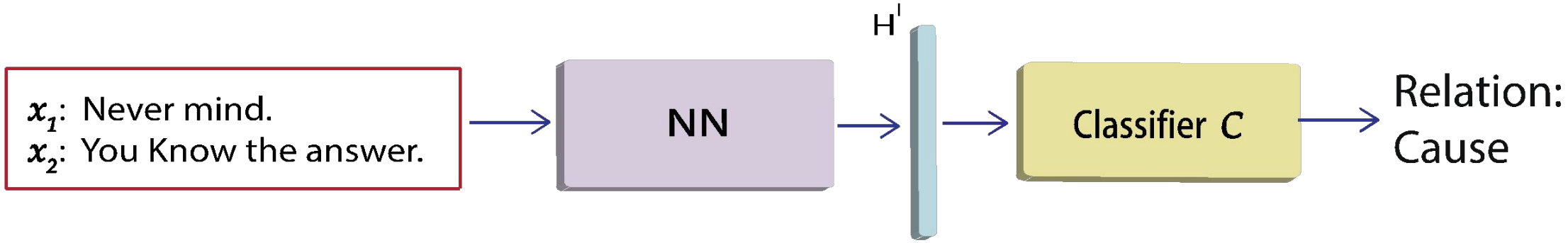
Connective: Never mind. Because you already know the answer.

Sentence 1 : Never mind.
Sentence 2 : You already know the answer.
[*Implicit connective*]: Because
[*Discourse relation*]: Cause

Discourse relation Classification

- Connectives are very important cues

- Explicit discourse relation : **> 85%**
- Implicit discourse relation : **< 50%** (with end to end neural nets !!!)



The Idea

- Human annotators adds the connectives to the dataset to find the relation

- Example from Penn Discourse Treebank (PDTB) benchmark

  Never mind. You already know the answer.

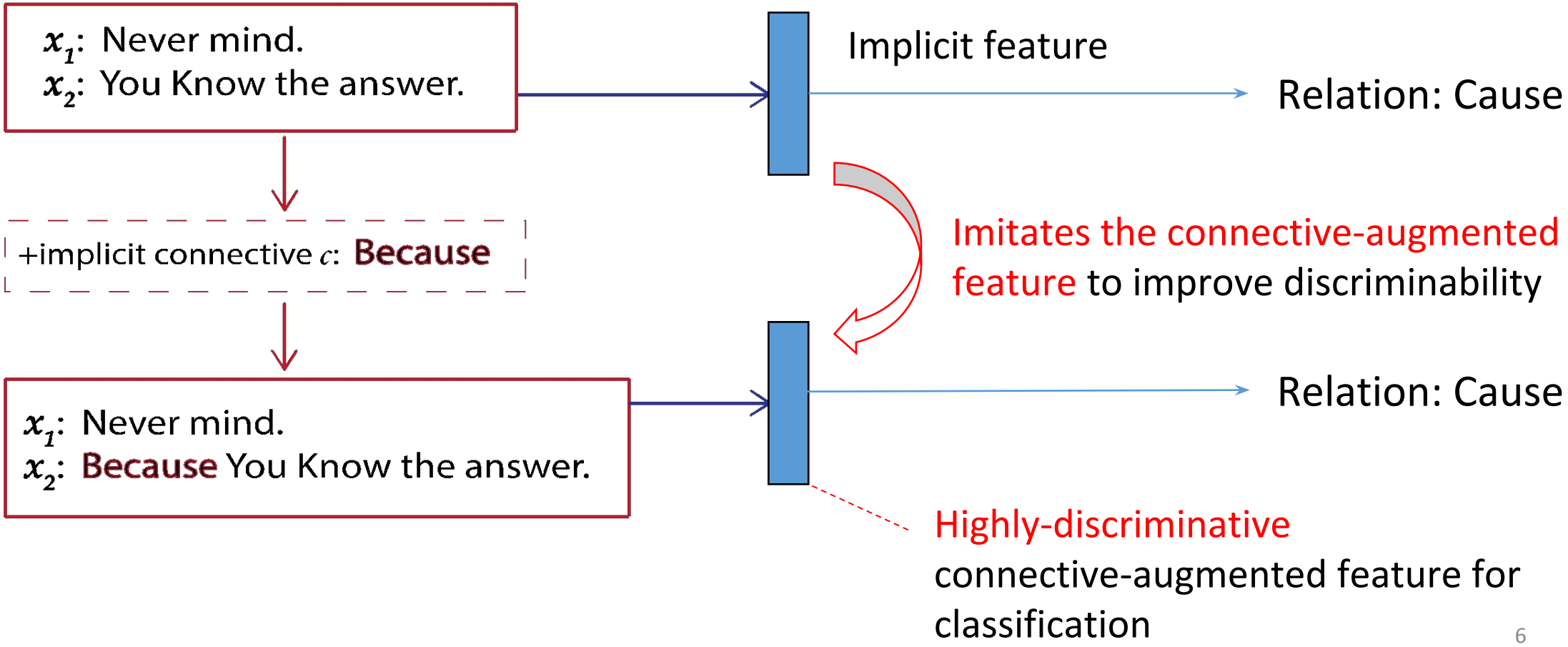
  - Add the implicit connective

    Never mind. because You already know the answer.

  - Determine the relation

Idea

- Use the annotated implicit connectives in the *training data*



Feature imitation

- Due to the connective cue, there is a huge gap in the features
- Failed with using things like L2 distance reduction
- It was necessary to use adaptive scheme to ensure discriminability : **Adversarial networks**

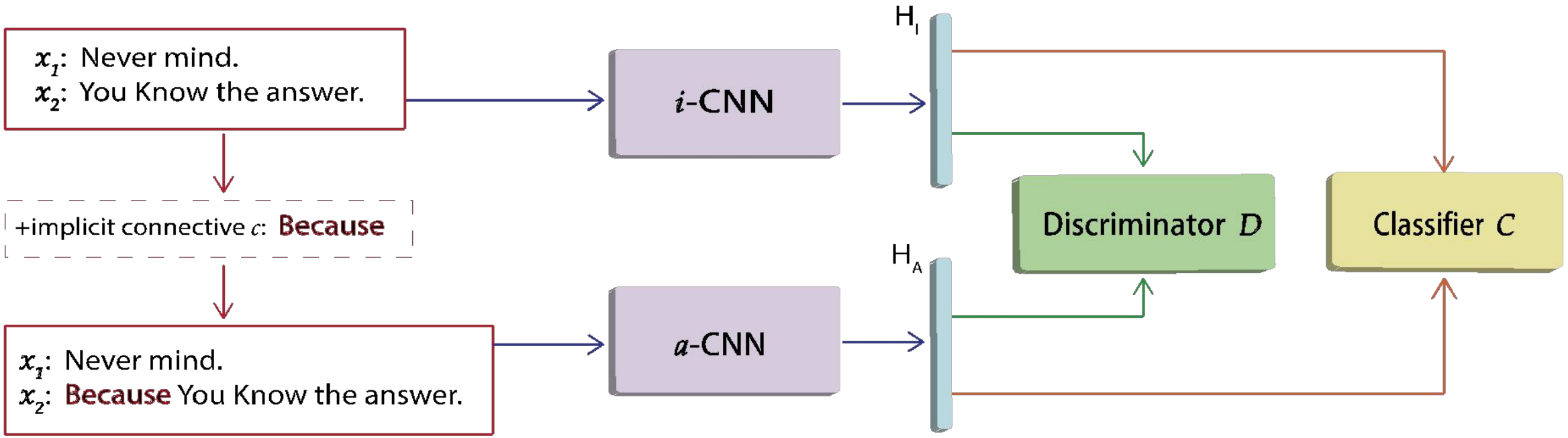
Adversarial Networks

- Proposed by Goodfellow et al., 2014
- Idea :

  Say we want to generate images from a vector.

  - **Generator :** generate similar to a “correct values” to fool the discriminator
  - **Discriminator :** discriminate between the thing generated by the generator and the actual “correct values”

The model



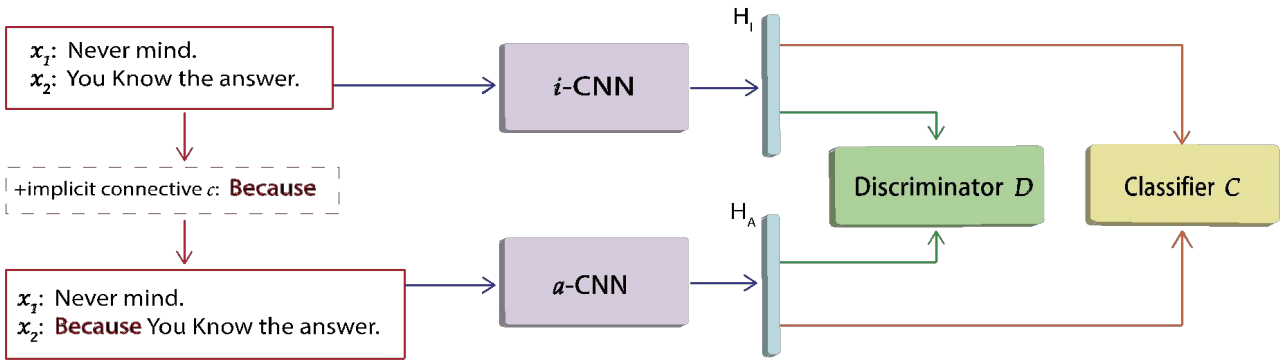
- *i-CNN* wants to mimic *a-CNN* and both wants to maximize the classification accuracy from C
- Discriminator wants to discriminates between H_I and H_A

Network training

Repeat :

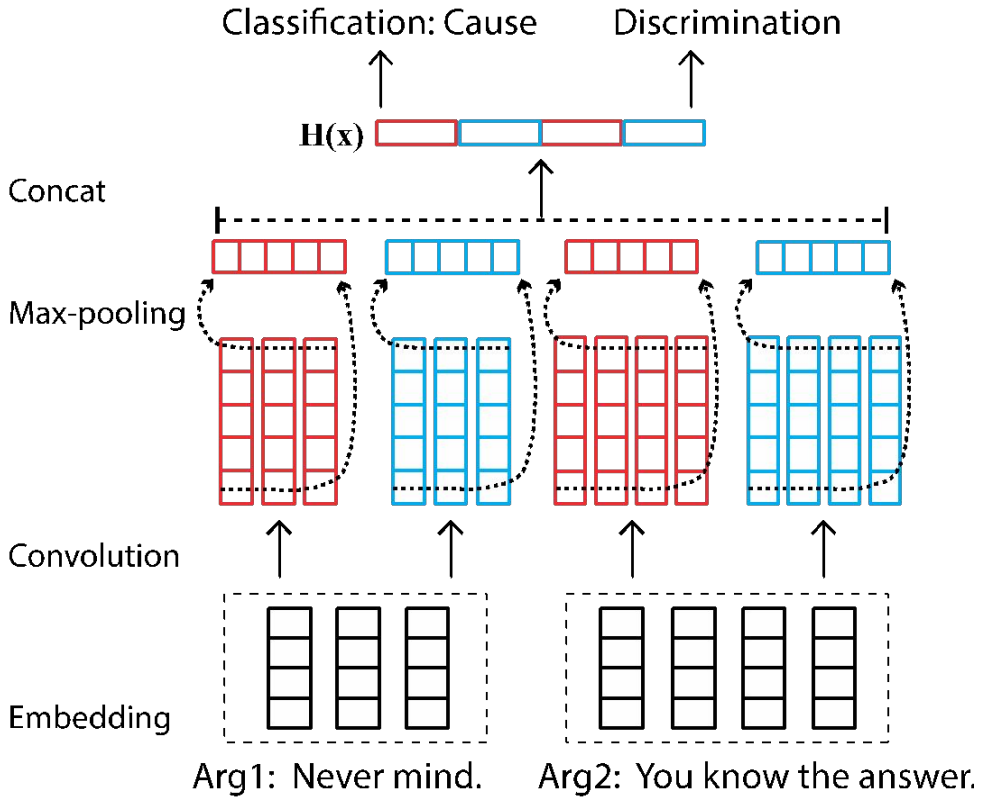
- Train *i-CNN* and *C* to maximize classification accuracy and fool D
- Train *a-CNN* to maximize classification accuracy
- Train **D** to distinguish between the two features

Note : a-CNN is trained with C fixed as it is strong enough



Network details: CNNs

- *i-CNN*
  - Word - Embedding layers, Convolutions and max-pooling
- *a-CNN*
  - Word - Embedding layers, Convolutions
  - Average k-max pooling
    - Average of the top k values
    - Forces to “attend” the contextual features from the sentences

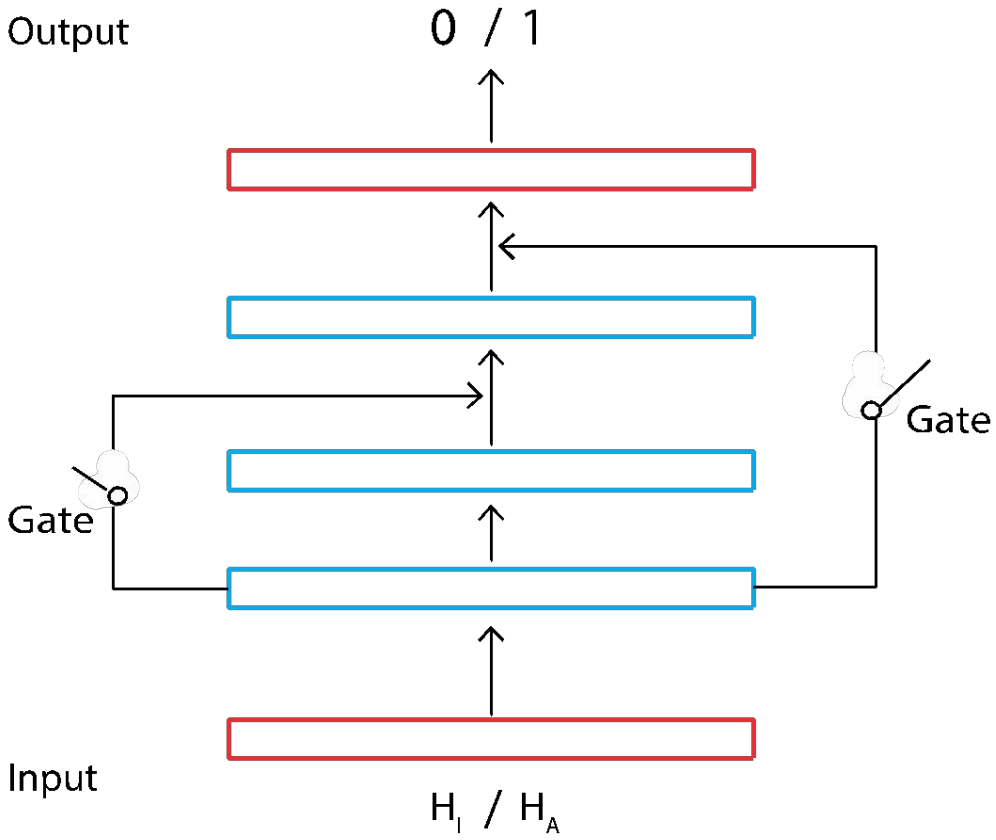


i-CNN

Network details: Discriminator

- Discriminator, D:
  - Multi fully connected layers (FCs)
  - Additional stacked gate to help in gradient propagation [Qin et al., 2016]

- Classifier, C:
  - Fully connected layer followed by softmax



Discriminator

Experiments

- PDTB benchmark dataset
  - Sentence pairs, relation labels, implicit connectives
- Multi-class classification task
  - 11 relation classes
  - Two slightly different settings as in previous work
- One-vs-all classification tasks
  - 4 Relation classes: Comparison, Contingency, Expansion, Temporal

Multi-class classification task

- Accuracy (%) on two settings

	Model	PDTB-Lin	PDTB-Ji
1	Word-vector	34.07	36.86
2	CNN	43.12	44.51
3	Ensemble	42.17	44.27
4	Multi-task	43.73	44.75
5	ℓ_2-reg	44.12	45.33
6	Lin et al. (2009)	40.20	-
7	Lin et al. (2009)+Brown clusters	-	40.66
8	Ji and Eisenstein (2015)	-	44.59
9	Qin et al. (2016a)	43.81	45.04
10	Ours	**44.65**	**46.23**

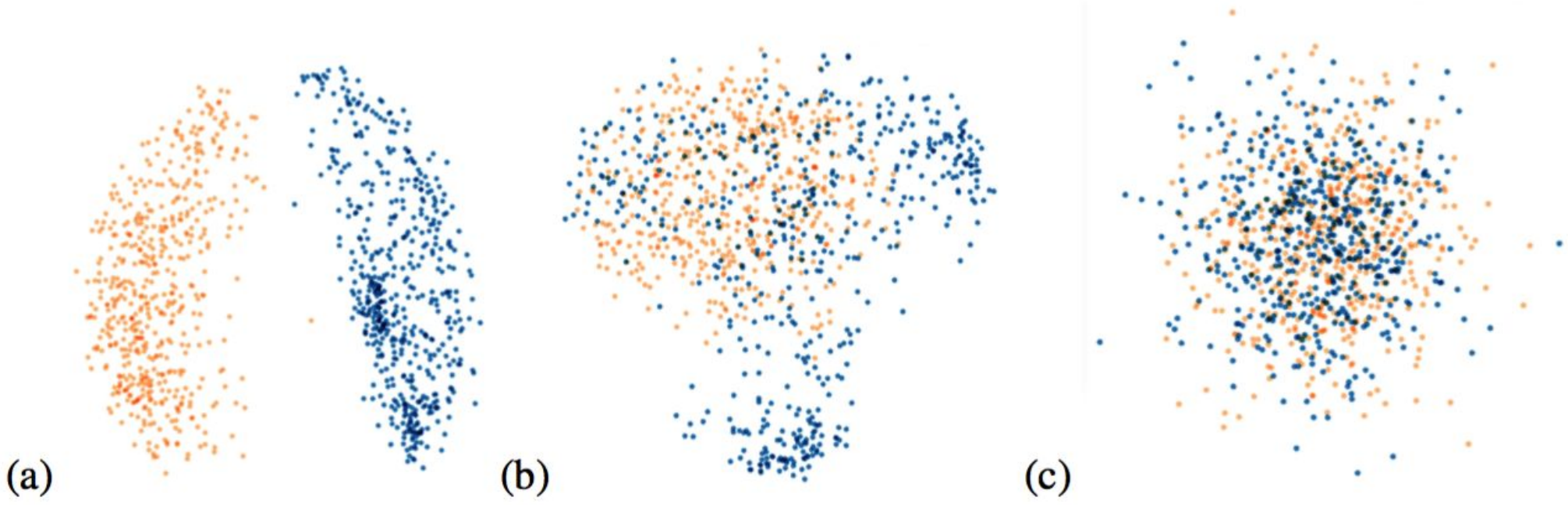
One-vs-all classification tasks

- Comparisons of F1 scores (%) for binary classifications

Model	COMP.	CONT.	EXP.	TEMP.
Pitler et al. (2009)	21.96	47.13	-	16.76
Qin et al. (2016c)	**41.55**	**57.32**	71.50	35.43
Zhang et al. (2016)	35.88	50.56	71.48	29.54
Zhou et al. (2010)	31.79	47.16	70.11	20.30
Liu and Li (2016)	36.70	54.48	70.43	**38.84**
Chen et al. (2016a)	40.17	54.76	-	31.32
Ours	40.87	54.56	**72.38**	36.20

Feature visualization

- *i*-CNN (blue) and *a*-CNN (orange) feature vectors
  - (a): without adversarial mechanism
  - (b)-(c): features as training proceeds in the proposed framework



Conclusions

- Connectives are very important cues
- Use the additional data during training to propose a new feature learning
- Proposed adversarial networks for feature learning with adaptive distance

Discussions

- Generalization
  - Can be used in task in which we can use additional data during training time to learn better

Thanks