ASNLU at NTCIR-14 Finnum Task: Incorporating Knowledge into DNN for Financial Numeral Classification

ChaoChun Liang

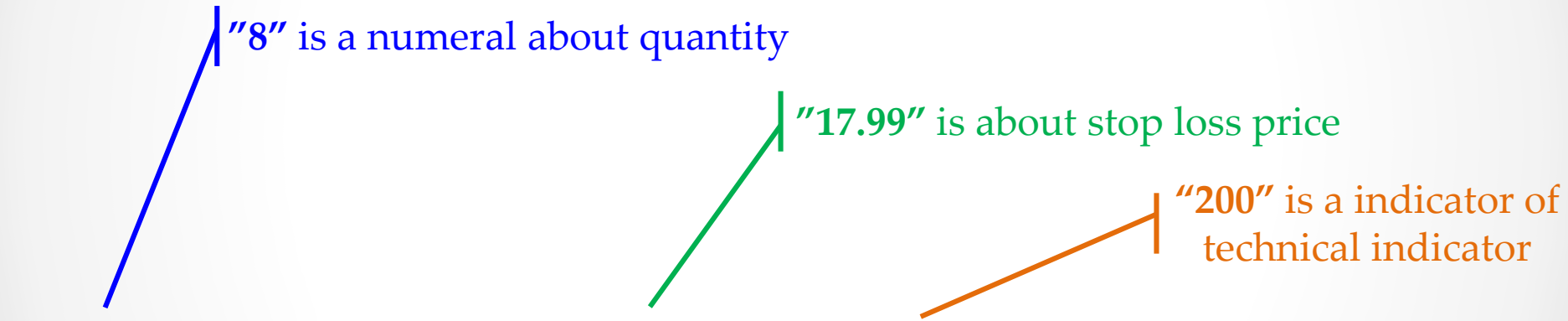
Institute of Information Science
Academia Sinica, Taipei
June 12, 2019

Outline

- Proposed Approaches
- Experimental Results
- Discussion
- Conclusion

Task Overview

- **Purpose: To understand the fine-grained numeral information in financial Tweet**



Proposed Approach 1/5

Tweet	**8**	breakouts:	$CHMT	(stop:	**$17.99**	).
Main Category	**Quantity**	O	O	O	**Monetary**	O
Sub Category	**Quantity**	O	O	O	**stop loss**	O

- **Model the Numeral Classification as a Sequence Labeling Process**
  - Input Word Sequence: *W1, W2, … Wn*
  - Output Label Sequence: *T1, T2, … Tn*

$$Ti \in \begin{cases} O \cup M & \text{, for Main-Category Classification, (\#=8)} \\ O \cup S & \text{, for Sub-Category Classification, (\#=18)} \end{cases}$$

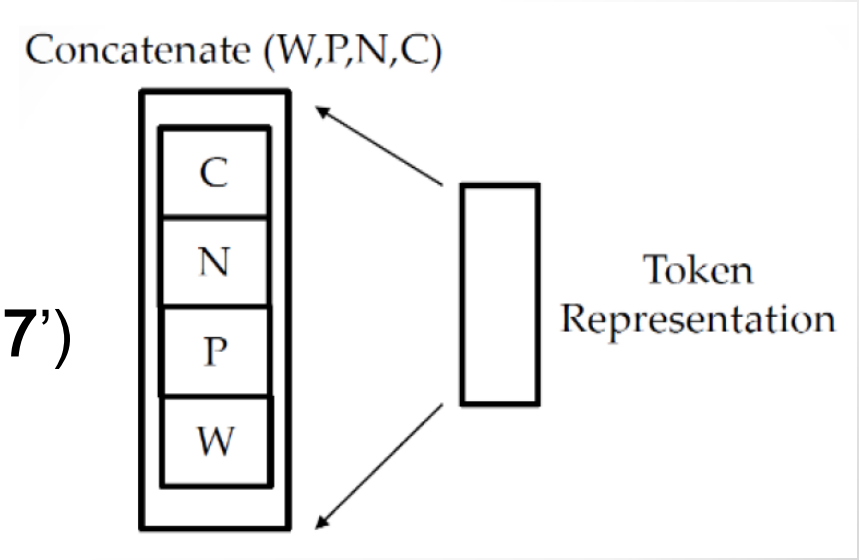
M: main category class set, S: sub-category class set
O: Not a target word to be classified

Proposed Approach 2/5

- Propose a *token representation with external knowledge* to represent the word meaning in Tweet sentences
- Implement three vanilla neural network models

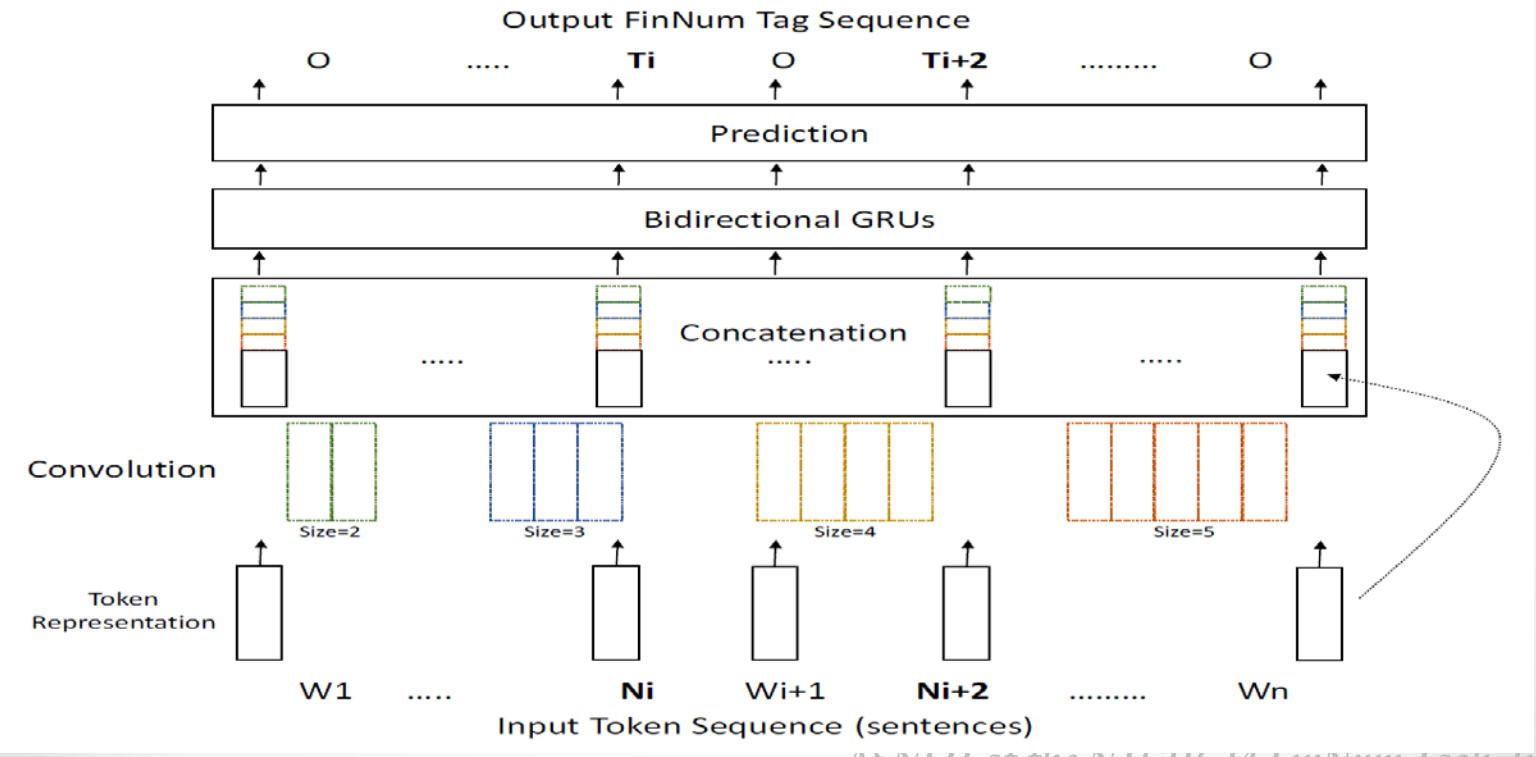
Proposed Approach 3/5

- **Token Representation**
  - **W**: Pre-trained Word Embedding
  - **P**: Part-of-Speech, **N**: Named entity Type
  - **C**: Category-Pattern Feature (#=6)
    - Company. (‘$NTNX’)
    - Money. (‘$**20**’ or ’**13**$’)
    - Product number. (‘PS**4**’)
    - Date. (’**11/09/17**’ or ’**11-09-17**’)
    - Time. (‘**6:45**’ or ‘**3:25** p.m.’)
    - Number. (’**68**’)



Proposed Approach 4/5

- CNN (detect local patterns, e.g. '85%')
- RNN (capture context information)
- RNN+CNN (capture local info. in RNN)



Proposed Approach 5/5

- **Rescoring in Prediction Time:**
  - Exclude the Out-of-Category ('O') label from the candidate set for each target numeral to avoid inconsistency.

Experiment Setting

- **Pre-trained Embedding**
  - GLOVE 840.300D
- **CNN**
  - Kernel sizes of 2,3,4 and 5
  - 32 filters for each kernel
- **RNN**
  - Bi-GRUs with 128 hidden nodes
- **Dropout 0.5**

Overall Performance

	CNN		RNN		RNN+CNN	
	Micro	Macro	Micro	Macro	Micro	Macro
None	81.83	69.54	84.22	73.36	82.71	69.63
+POS&NE	88.21	79.14	88.45	78.63	**89.72**	**80.93**
+POS&NE +Pattern	87.73	78.47	88.76	83.55	**89.24**	81.50

Task-1 Test Set Performance

	CNN		RNN		RNN+CNN	
	Micro	Macro	Micro	Macro	Micro	Macro
None	69.88	58.66	75.22	71.72	73.94	65.54
+POS&NE	75.14	65.77	78.49	72.37	78.17	70.16
+POS&NE +Pattern	76.41	68.5	79.36	70.5	**79.12**	**72.51**

Task-2 Test Set Performance

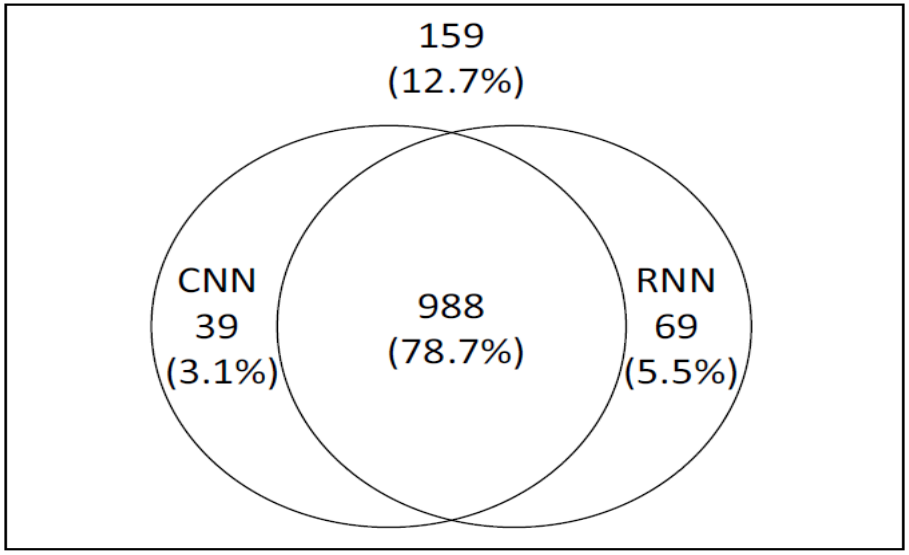
*"**None**" denotes the NN models without incorporating any knowledge.*
*"**POS&NE**" denotes the NN models with both POS and NE information.*
*"**Pattern**" denotes the NN models that incorporate category patterns specified by handcrafted rules.*

Experimental Results 1/3

	CNN		RNN		RNN+CNN	
	Micro	Macro	Micro	Macro	Micro	Macro
None	81.83	69.54	84.22	73.36	82.71	69.63

Task-1 testing set performance

- **Division of classification results between CNN and RNN models**



Experimental Results 2/3

	CNN		RNN		RNN+CNN	
	Micro	Macro	Micro	Macro	Micro	Macro
None	81.83	69.54	84.22	73.36	82.71	69.63
+POS&NE	88.21	79.14	88.45	78.63	**89.72**	**80.93**

Task-1 testing set performance

- **OOVs provide no useful Information**
  - OOVs: 30+% on Development and Test sets
- **Linguistic Information (POS&NE) attached to OOVs improved the performance significantly (4% ~ 10%).**

Experimental Results 3/3

	CNN		RNN		RNN+CNN	
	Micro	Macro	Micro	Macro	Micro	Macro
None	81.83	69.54	84.22	73.36	82.71	69.63
+POS&NE	88.21	79.14	88.45	78.63	**89.72**	**80.93**
+POS&NE +Pattern	87.73	78.47	88.76	83.55	**89.24**	81.50

Task-1 testing set performance

- Category-pattern features offer small improvements or even degrade performance.
- Not cover enough patterns for manually-encoded rules.

Discussion 1/2

- Issue-1: High OOV rate
- Issue-2: Diverse patterns in Tweet (Not enough coverage with handcrafted patterns)
- Solution: Numeral-Splitting
  - Most OOVs are concatenations of a numeral and other characters.
  - Split each token with numbers into individual sub-tokens.

OOV Rate	Dev	Test
Before	36%	39%
After	22%	23%

Discussion 2/2

- **Performance improves significantly.** E.g., 9% (micro), 18%(macro) in RNN+CNN("None").
- **Outperforms the handcrafted patterns.**

	CNN		RNN		RNN+CNN	
	Micro	Macro	Micro	Macro	Micro	Macro
None	81.83	69.54	84.22	73.36	82.71	69.63
+POS&NE +Pattern	87.73	78.47	88.76	83.55	**89.24**	81.50

Task-1 Test Set Performance (before Numeral Splitting)

	CNN		RNN		RNN+CNN	
	Micro	Macro	Micro	Macro	Micro	Macro
None	89.56	83.17	92.27	86.60	92.11	88.18
+POS&NE	90.68	83.60	91.95	**88.36**	**92.99**	88.25

Task-1 Test Set Performance (after Numeral Splitting)

Conclusion

- The proposed token representation (with linguistic knowledge) improves performance significantly.
- A suitable pre-processing (splitting numerals) to reduce OOV rates is essential.
- Jointly adopting both approaches could offer additional benefits.

Q & A

Thanks

Appendix – P10 1/2

- **Errors made by RNN were due to the model missing local patterns**
  - E.g., "num/num" (Temporal) in "10/24"
    "num%" (Percentage) in "7.8%"
- **Errors made by CNN were due to the model missing context information**
  - E.g., "*You **sold** ESPR at 11 and CLVS at 29 but thanks for this tip.*"

Appendix – P10 2/2

- **Errors made by RNN and CNN both were due to the number can not be categorized explicitly (i.e. need more information).**
  - E.g., "*$NGAS Buy on dips on $UGAZ $UNG. Dip to 3.075, NG is on wave 3 move to 3.27 on 8HR chart.*"

Appendix – P12

- Category F-Score of RNN with POS&NE and Category-Patterns

	+POS&NE	+POS&NE +Pattern
Monetary	0.9107	0.9085
Quantity	0.7727	0.7857
Percentage	0.9882	0.9882
Temporal	0.8978	0.8903
Product Number	0.3182	**0.6818**
Option	0.7727	0.7727
Indicator	0.7778	0.7037