Adapting RNN Sequence Prediction Model to Multi-label Set Prediction

Kechen Qin Cheng Li Virgil Pavlu Javed A. Aslam
Khoury College of Computer Sciences
Northeastern University

報告者：
07370002 王成嘉

Content

1. Introduction
2. Mapping Sequences to Sets
3. Adapting RNN Sequence Prediction Model to Multi-label Set Prediction
    1. How is our new formulation different?
    2. Training by Maximizing Set Probability
4. Results and Analysis
    1. Experimental Setup
    2. Experimental Results
    3. Analysis: Sequence Probability Dsitribution
5. Case Analysis
6. Conclusion

Abstract

Set V.S. Sequence

- Multi-label classification(MLC) problem for text
- Target: a set of labels
- RNN models define probabilities for sequences but not for sets
- obtain a set probability, including:
  - pre-specifying the label order
  - relating the sequence probability to the set probability in ad hoc ways

Abstract

- Our Formulation
  - derived from a principled notion of set probability
  - sum of probabilities of corresponding permutation sequences for the set
  - new training objective: maximizes set probability
  - new prediction objective: find the most probable set on a test document
  - give the RNN model freedom to discover the best label order

....

作者表示自己的模型就是比較好!

Introduction

- Multi-label text classification is an important ML task
- predict a set of labels to associate with a given document

x

Football

sport

Football action is back

BHAVNEET SINGH

The early morning scene that welcomes one on arrival at the football ground at Gymkhana is one of frenzied activity and intense action by around 90 players who, braving the chilly weather, are hoping to keep tryst with destiny.

This has been the case from December 19 when an exclusive camp began under the overall supervision of the rejuvenated and born-again Andhra Pradesh Football Association (APFA).

The Association has done well by hiring the services of proven former and incumbent coaches from Sports Authority of Andhra Pradesh (SAAP) to monitor the progress of the players, who are set to embark on a career jump.

The players are participating in the gruelling training-cum-selection camp with the hope of making it to the State squad that is to take part in the ensuing Santosh Trophy.

Earlier, Saturda's regimen began after the players and coaches observed a two-minute silence for L Venkatram Reddy, the former secretary of APOA, who passed away in the city on Friday.

The South Pool competition of the symbol of domestic football supremacy will be hosted in Chennai from January 24 to February 2.

The scene at Gymkhana has been best described by veteran stopper-back and current DSDO of Hyderabad A Aleem Khan, who opined that a festive-like atmosphere has pervaded the ground.

Action shots from Saturday's practice session at Gymkhana.

Indeed, such a scene was amiss for over a decade now, thanks to the absence of any top grade football activity and the fracas in the almost rudderless APFA.

It is almost after a gap of ten years that a full-throttle selection is being organised in the city. Coaches John Victor and M Hari are marshalling the resources and testing their abilities.

In contention are players from various institutions and clubs, including State Bank of Hyderabad (SBH), South Central Railway (SCR), AGORC, CCOB, Shastri Soccer Club, Bolaram Sporting Club and Hyderabad Sporting Club. The camp will be enlarged when players from the districts report by Monday.

It is a good augury that employers like SBH and AGORC have been recruiting football players under the sports quota, despite the upheavals. While SBH appointed eight footballers this fiscal, the latter has roped in four, which serve as the perfect incentive to the budding players to stay focussed on the job.

According to John Victor, "we put them through various drills to improve their ball control and timing of passing in the morning while the evening session is dedicated to practice matches so that they get into competitive gear and peak at the appropriate time. As the chief coach, my task is to spot the best of the lot because they will carry the State's responsibilities in the premier championship."

Asserting that the probables were brimming with confidence, which gives hope of a decent showing in Chennai, SAAP coach Hari opines,

"The ambience is brilliant and motivational for the players, which has been missing for a very long time."

In a couple of days, the number would be short-listed to around 30 who will then undergo strenuous advanced coaching and conditioning before the final squad is announced.

The task, though, is not as easy because South is a tough proposition as it features heavyweights like Kerala and Tamil Nadu in addition to

Karnataka, Andaman and Nicobar and Pondicherry. Aleem Khan is quite pragmatic when he reasons that South pool is the toughest in the country and it will be an uphill task for Hyderabad to lock horns with seasoned campaigners.

"It is too early to assess the potentialities and comment on how the team will perform. But one thing is for sure, talent is available in abundance, all it needs is encouragement. We are doing our best to provide them with that," he said, while praising APFA President Dr Rafath Ali for providing the impetus to take the sport forward.

With old warhorses at the helm, one gets reassured that Andhra Pradesh football is in safe hands. It also gives credence to the belief that once the league structure commences and players get to parade their skills with more regularity, the quest to regain erstwhile glory would be tenable and meaningful. For now the onus is on APFA.

2018 world cup

Russian

多標籤轉為單標籤問題

BR

X	Y_1	Y_2	Y_3	Y_4
$x^{(1)}$	0	1	1	0
$x^{(2)}$	1	0	0	0
$x^{(3)}$	0	1	0	0
$x^{(4)}$	1	0	0	1
$x^{(5)}$	0	0	0	1

X	Y_1
$x^{(1)}$	0
$x^{(2)}$	1
$x^{(3)}$	0
$x^{(4)}$	1
$x^{(5)}$	0

X	Y_2
$x^{(1)}$	1
$x^{(2)}$	0
$x^{(3)}$	1
$x^{(4)}$	0
$x^{(5)}$	0

X	Y_3
$x^{(1)}$	1
$x^{(2)}$	0
$x^{(3)}$	0
$x^{(4)}$	0
$x^{(5)}$	0

X	Y_4
$x^{(1)}$	0
$x^{(2)}$	0
$x^{(3)}$	0
$x^{(4)}$	1
$x^{(5)}$	1

Classifier Chain

X	y1
x1	0
x2	1
x3	0

Classifier 1

X	y1	y2
x1	0	1
x2	1	0
x3	0	1

Classifier 2

X	y1	y2	y3
x1	0	1	1
x2	1	0	0
x3	0	1	0

Classifier 3

X	y1	y2	y3	y4
x1	0	1	1	0
x2	1	0	0	0
x3	0	1	0	0

Classifier 4

Introduction

Binary Relevance

- a set of label candidates $\mathcal{L} = \{1, 2, ..., L\}$
- aim to build a classifier
- maps a document x to a set of labels $\mathbf{y} \subset \mathcal{L}$
- binary vector $\mathbf{y} \in \{0, 1\}^L$
- predict each label independently
- y_ℓ presence or absence of a label
- 忽略了標籤的相關性

Introduction

- To capture **label dependencies** by building a joint probability estimation over all labels:

$$p(\mathbf{y} = (y_1, y_2, \dots, y_L)|x)$$

Probabilistic Classifier Chain (PCC)

- **binary decision is made for each label sequentially**
- learns labels one-by-one
- predefined fixed order
- use one classifier to estimate the probability of each label
- given all previous labels predictions: $p(y_l|y_1, \dots, y_{l-1}, x)$
- Drawback:
  - errors in early probability estimations tend to affect subsequent predictions
  - become massive when L is large

BR

X	Y_1	Y_2	Y_3	Y_4
$x^{(1)}$	0	1	1	0
$x^{(2)}$	1	0	0	0
$x^{(3)}$	0	1	0	0
$x^{(4)}$	1	0	0	1
$x^{(5)}$	0	0	0	1

X	Y_1
$x^{(1)}$	0
$x^{(2)}$	1
$x^{(3)}$	0
$x^{(4)}$	1
$x^{(5)}$	0

X	Y_2
$x^{(1)}$	1
$x^{(2)}$	0
$x^{(3)}$	1
$x^{(4)}$	0
$x^{(5)}$	0

X	Y_3
$x^{(1)}$	1
$x^{(2)}$	0
$x^{(3)}$	0
$x^{(4)}$	0
$x^{(5)}$	0

X	Y_4
$x^{(1)}$	0
$x^{(2)}$	0
$x^{(3)}$	0
$x^{(4)}$	1
$x^{(5)}$	1

Classifier Chain

X	y1
x1	0
x2	1
x3	0

Classifier 1

X	y1	y2
x1	0	1
x2	1	0
x3	0	1

Classifier 2

X	y1	y2	y3
x1	0	1	1
x2	1	0	0
x3	0	1	0

Classifier 3

X	y1	y2	y3	y4
x1	0	1	1	0
x2	1	0	0	0
x3	0	1	0	0

Classifier 4

Introduction

RNN

- applied to MLC by mapping **label set to a sequence**
- Only predicts the positive labels
- decision chain length = positive labels

PCC & RNN

- Both rely heavily on label order in training and prediction
- In multi-label data, labels are given as sets
- RNN defines sequence probability
- PCC defines set probability

Introduction

Arranging sets as sequence

- ordering alphabetically
- frequency
- label hierarchy
- label ranking algorithm

Previous experimental results show that which order to choose can have a significant impact on learning and prediction

- Vinyals et al., 2016

Introduction

Previous work

- RNN can explore different label orders and converge to some order automatically (Vinyals et al., 2016)
- RNN sequence model to multi-label set prediction without specifying the label order

This Paper:

- Propose new training and prediction objectives
- based on a principled notion of set probability
- gives RNN model freedom to discover the best label order

Mapping Sequences to Sets

Review RNN designed for sequences

- Input Sequences of outcomes: $\mathbf{s} = (s_1, s_2, ..., s_T)$
- Particular order: $s_t \in \{1, 2, ..., L\}$
- probability distribution over all possible output sequences given the input in the form :

$$p(\mathbf{s} = (s_1, s_2, ..., s_T)|x) = \prod_{t=1}^{T} p(s_t|x, s_1, s_2, ..., s_{t-1})$$

At prediction time, find Sequence with the highest probability: $\mathbf{s}^* = \arg\max_{\mathbf{s}} p(\mathbf{s}|x)$

- Use Beam Search
- with the attention mechanism, label probability distribution at time t:

$$p(s_t|x, s_1, s_2, ..., s_{t-1}) \sim softmax(\phi(c_t, h_t, s_{t-1}))$$

Mapping Sequences to Sets

Apply RNN to multi-label problems, one approach…

- map the given set of labels **y** to a sequence $\mathbf{s} = (s_1, s_2, \ldots, s_T)$
- globally fixed order (frequency, in PCC)
- Mapping is done
- (Nam) RNN is trained with the standard maximum likelihood objective :

$$maximize \sum_{n=1}^{N} \log p(\mathbf{s}^{(n)} | x^{(n)}) \qquad (1)$$

Mapping Sequences to Sets

- (Vinyals) dynamically choose the sequence order as most probable during training:

$$maximize \sum_{n=1}^{N} \max_{\mathbf{s}\in\pi(\mathbf{y}^{(n)})} \log p(\mathbf{s}|x^{(n)}) \qquad (2)$$

- $\pi(\mathbf{y}^{(n)})$ stands all permutations of set $\mathbf{y}^{(n)}$
- Drawback:
  - cannot be used in the early training stages
  - early order choice is reinforced by this objective and can be stuck upon permanently

Mapping Sequences to Sets

- (Vinyals) proposes two smoother alternative objective:
  - consider many random orders for each label set

$$maximize \sum_{n=1}^{N} \sum_{\mathbf{s} \in \pi(\mathbf{y}^{(n)})} \log p(\mathbf{s}|x^{(n)}) \qquad (3)$$

  - Sample sequences follow model predictive distribution not uniform distribution

$$maximize \sum_{n=1}^{N} \sum_{\mathbf{s} \in \pi(\mathbf{y}^{(n)})} p(\mathbf{s}|x^{(n)}) \log p(\mathbf{s}|x^{(n)}) \qquad (4)$$

Mapping Sequences to Sets

- Training : schedule the transition among objectives
- Prediction : find the most probable set $\mathbf{s}^* = \arg\max_{\mathbf{s}} p(\mathbf{s}|x)$

- This is done by finding the most probable sequence
- treating it as a $\hat{\mathbf{y}} = set(\mathbf{s}^*)$

- many sequences: argmax with low probability
- ignore sequences except top one lead to neglecting important information

Adapting RNN Sequence Prediction Model to Multi-label Set Prediction

Set-RNN

- We define set probability as sum of sequences probabilities
- all sequence permutations of the set, namely:

$$p(\mathbf{y}|x) \; = \; \sum_{\mathbf{s}\in\pi(\mathbf{y})} p(\mathbf{s}|x)$$

- Probability distribution over all possible sets:

$$\sum_{\mathbf{y}} p(\mathbf{y}|x) = \sum_{\mathbf{y}} \sum_{\mathbf{s}\in\pi(\mathbf{y})} p(\mathbf{s}|x) = \sum_{\mathbf{s}} p(\mathbf{s}|x) = 1$$

Adapting RNN Sequence Prediction Model to Multi-label Set Prediction

Set-RNN

- maximize the likelihood of given label sets, namely

$$\prod_{n=1}^{N} p(\mathbf{y}^{(n)}|x^{(n)}) = \prod_{n=1}^{N} \sum_{\mathbf{s}\in\pi(\mathbf{y}^{(n)})} p(\mathbf{s}|x^{(n)})$$

$$\Downarrow$$

$$maximize \sum_{n=1}^{N} \log \sum_{\mathbf{s}\in\pi(\mathbf{y}^{(n)})} p(\mathbf{s}|x^{(n)}) \qquad (5)$$

Methods	Training objectives	Prediction objectives
seq2seq-RNN	*maximize* $\sum_{n=1}^{N} \log p(\mathbf{s}^{(n)} \| x^{(n)})$	$\hat{\mathbf{y}} = set(\mathbf{s}^*)$, $\mathbf{s}^* = \arg\max_{\mathbf{s}} p(\mathbf{s} \| x)$
Vinyals-RNN-max	*maximize* $\sum_{n=1}^{N} \max_{\mathbf{s} \in \pi(\mathbf{y}^{(n)})} \log p(\mathbf{s} \| x^{(n)})$	$\hat{\mathbf{y}} = set(\mathbf{s}^*)$, $\mathbf{s}^* = \arg\max_{\mathbf{s}} p(\mathbf{s} \| x)$
Vinyals-RNN-uniform	*maximize* $\sum_{n=1}^{N} \sum_{\mathbf{s} \in \pi(\mathbf{y}^{(n)})} \log p(\mathbf{s} \| x^{(n)})$	$\hat{\mathbf{y}} = set(\mathbf{s}^*)$, $\mathbf{s}^* = \arg\max_{\mathbf{s}} p(\mathbf{s} \| x)$
Vinyals-RNN-sample	*maximize* $\sum_{n=1}^{N} \sum_{\mathbf{s} \in \pi(\mathbf{y}^{(n)})} p(\mathbf{s} \| x^{(n)}) \log p(\mathbf{s} \| x^{(n)})$	$\hat{\mathbf{y}} = set(\mathbf{s}^*)$, $\mathbf{s}^* = \arg\max_{\mathbf{s}} p(\mathbf{s} \| x)$
set-RNN (ours)	*maximize* $\sum_{n=1}^{N} \log \sum_{\mathbf{s} \in \pi(\mathbf{y}^{(n)})} p(\mathbf{s} \| x^{(n)})$	$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} p(\mathbf{y} \| x)$

Table 1: Comparison between previous and our *set-RNN* training and prediction objectives.

3.1How is our new formulation different?

$$maximize \sum_{n=1}^{N} \boxed{\sum_{\mathbf{s}\in\pi(\mathbf{y}^{(n)})} \log p(\mathbf{s}|x^{(n)})} \quad (3)$$

$$p(\mathbf{y}|x) = \prod_{\mathbf{s}\in\pi(\mathbf{y})} p(\mathbf{s}|x)$$

- **Multiplication**
- To maximize the set probability, RNN model has to assign equally high probabilities to all sequence permutations of the given label set

$$maximize \sum_{n=1}^{N} \log \boxed{\sum_{\mathbf{s}\in\pi(\mathbf{y}^{(n)})} p(\mathbf{s}|x^{(n)})} \quad (5)$$

$$p(\mathbf{y}|x) = \sum_{\mathbf{s}\in\pi(\mathbf{y})} p(\mathbf{s}|x)$$

- **Summation**
- different documents can have different label orders
- gives RNN model more freedom on label order

3.2 Training by Maximizing Set Probability

- Training an RNN with (5) requires summing up sequence (permutation) probabilities for a set y
- approximate this sum by only considering the top K highest probability
- variant of beam search for sets with width K
- with the search candidates in each step restricted to only labels in the set (see Algorithm 1 with ALL = 1)
- This approximate inference procedure is carried out repeatedly before each batch training step

Algorithm 1: Beam_Search

Input : Instance x
Subset of labels considered $G \subset \mathcal{L}$
Boolean flag ALL: 1 if sequences must contain all G labels; 0 if partial sequences are allowed

Output: A list of top sequences and the associated probabilities

Let $\mathrm{s}_1,\mathrm{s}_2,\ldots,\mathrm{s}_K$ be the top K sequences found so far. Initially, all K sequences are empty. $\oplus$ means concatenation.
while *true* **do**
  // Step 1: Generate Candidate Sequences from each existing sequence $s_k \in K$ and all possible new labels $l \in G$:
  Expand all non-stopped sequences:
  $C = \{\mathrm{s}_k \oplus l | l \in G, STOP \notin s_k\}$
  Include stopped sequences:
  $C = C \cup \{\mathrm{s}_k | STOP \in s_k\}$
  Terminate non-stopped sequences:
  if $ALL == 0$ **then**
    $C = C \cup \{\mathrm{s}_k \oplus STOP | STOP \notin s_k\}$
  end
  // Step 2: Select highest probabilities sequences from candidate set C
  K = topK-argmax$_k\{\mathrm{prob}[s_k] | s_k \in C\}$
  if *all top K sequences end with $STOP$ or contain all labels in G* **then**
    Terminate the algorithm
  end
end
return *sequence list* $\mathrm{s}_1,\mathrm{s}_2,\ldots,\mathrm{s}_K$ *and the associated probabilities*

3.2 Training by Maximizing Set Probability

- find highest probability sequences for all training instances occurring in that batch
- The overall training procedure:

Algorithm 2: Training method for set-RNN

Input : Multi-label dataset $(x^{(n)}, \mathbf{y}^{(n)}), n = 1, 2, ..., N$

Output: Trained RNN model parameters

```
foreach batch do
    foreach (x^n, y^n) in the batch do
        Get top K sequences :
        {s_1^n, ..., s_K^n, p(s_1^n|x^n), ..., p(s_K^n|x^n)}=
         = Beam_Search(x^n, y^n, ALL = 1)
    end
    Update model parameters by maximizing
        sum_{(x^n,y^n) in batch} log sum_{s in {s_1^n,...,s_K^n}} p(s|x^n)
end
```

$$\sum_{(x^n, \mathbf{y}^n) \in \text{batch}} \log \sum_{\mathbf{s} \in \{\mathbf{s}_1^n, ..., \mathbf{s}_K^n\}} p(\mathbf{s}|x^n)$$

3.3 Predicting the Most Probable Set

Algorithm 3: Prediction Method for set-RNN

Input : Instance x

Output: Predicted label set $\hat{\mathbf{y}}$

Obtain K highest probability sequences :

$\{\mathrm{s}_1, ..., \mathrm{s}_K\}$ = Beam_Search(x,$\mathcal{L}$, $ALL = 0$)

Map each sequence s_k to the corresponding set $\mathbf{y}_k$ and remove duplicate sets (if any)

foreach $\mathbf{y}_k$ **do**

    Get K most probable sequences associated with $\mathbf{y}_k$ and their probabilities :

    $\{\mathrm{s'}_1, ..., \mathrm{s'}_K, p(\mathrm{s}'_1|x), ..., p(\mathrm{s}'_K|x)\}$=

    = Beam_Search(x,$\mathbf{y}_k$, $ALL = 1$)

    Set probability is approx by summing up :

$$p(\mathbf{y}_k|x) \approx \sum_{\mathrm{s} \in \{\mathrm{s}'_1, ..., \mathrm{s}'_K\}} p(\mathrm{s}|x)$$

end

$\hat{\mathbf{y}} = argmax_{\mathbf{y}_k}(p(\mathbf{y}_k|x))$

set $\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} p(\mathbf{y}|x)$

- two-level beam search
- run standard RNN beam search (Algorithm 1 with ALL = 0)
- generate a list of highest probability sequences
- consider the label set associated with each label sequence
- evaluate each set probability (Algorithm 1 with ALL = 1)
- find top few highest probability sequences associated with the set
- sum up their probabilities
- choose the highest probability as the prediction
- the most probable set may not correspond to the most probable sequence

Results and Analysis

4.1 Experimental Setup

- filter out stopwords and punctuations
- Truncat document
  - TheGuardian & AAPD: max 500
  - Slashdot & RCV1-v2: max 120
- less words than the maximum number :Zero padding
- Numbers and out-of-vocabulary words are replaced with special tokens
- Words, user tags and labels are all encoded as 300-dimensional vectors using WORD2VEC (Mikolov et al., 2013)
- We implement RNNs with attention using TENSORFLOW-1.4.0 (Abadi et al., 2016)
- The dynamic function for RNNs is chosen to be GRU
  - 2 layers
  - at most 50 units in decoder
  - GRU unit is 300
- dropout rate = 0.3
- train the model with Adam optimizer (Kingma and Ba, 2014)
- with learning rate 0:0005
- Beam size :12

Data	#Train	#Test	Cardinality	#Labels	Doc length
Slashdot	19,258	4,814	4.15	291	64
RCV1-v2	40,000	10,000	3.17	101	121
TheGuardian	37,638	9,409	7.41	1,527	505
AAPD	53,840	1,000	2.41	54	163

Table 2: Statistics of the datasets.

Results and Analysis

4.1 Experimental Setup

- Evaluation metrics

$$\text{label-F1} = \frac{1}{L}\sum_{\ell=1}^{L} \frac{2\sum_{n=1}^{N} y_\ell^{(n)} \hat{y}_\ell^{(n)}}{\sum_{n=1}^{N} y_\ell^{(n)} + \sum_{n=1}^{N} \hat{y}_\ell^{(n)}}$$

$$\text{instance-F1} = \frac{1}{N}\sum_{n=1}^{N} \frac{2\sum_{\ell=1}^{L} y_\ell^{(n)} \hat{y}_\ell^{(n)}}{\sum_{\ell=1}^{L} y_\ell^{(n)} + \sum_{\ell=1}^{L} \hat{y}_\ell^{(n)}}$$

- for each instance n
  - $y_\ell^{(n)}$ = 1 if label ℓ is a given label in ground truth
  - $\hat{y}_\ell^{(n)}$ = 1 if label ℓ is a predicted label

- instance-F1 : basically determined by the popular labels' performance
- label-F1 : sensitive to the performance on rare label

Results and Analysis

- **Binary Relevance (BR)**
- **BR-support (Binary Relevance with support inference)** (Wang et al., 2018)
  - trains binary classifiers independently
  - imposes label constraints at prediction time (僅考慮訓練期間觀察到的標籤集)

$$\hat{\mathbf{y}} = \arg\max_{\text{observed } \mathbf{y}} \prod_{\ell=1}^{L} p(y_\ell|x)$$

- **PCC**: chain of binary classification problems (predictions are made with Beam Search)
- **Seq2Seq** : maps each set to a sequence by decreasing label frequency
- **Vinyals**
  - **Vinyals-RNN-uniform, Vinyals-RNNsample,** and **Vinyals-RNN-max**
  - trained with different objectives that correspond to different transformations between sets and sequences
  - Vinyals-RNNsample and Vinyals-RNN-max are initialized by Vinyals-RNN-uniform
- **Vinyals-RNN-max-direct** :
  - Vinyals-RNN-max directly without Vinyals-RNN-uniform as initialization
- **Sequence Generation Model (SGM):**
  - similar to seq2seq-RNN
  - new decoder structure
  - computes a weighted global embedding based on all labels
  - Not just the top one at each timestep

Methods	Slashdot		RCV1-v2		TheGuardian		AAPD			
	label-F1	instance-F1	label-F1	instance-F1	label-F1	instance-F1	label-F1	instance-F1	hamming-loss	micro-F1
BR	.271	.484	.486	.802	.292	.572	.529	.654	.0230	.685
BR-support	.247	.516	.486	.805	.296	.594	.545	.689	**.0228**	.696
PCC	.279	.480	.595	.818	-	-	.541	.688	.0255	.682
seq2seq-RNN	.270	.528	.561	.824	.331	.603	.510	.708	.0254	.701
Vinyals-RNN-uniform	.279	.527	.578	.826	.313	.567	.532	.721	.0241	.711
Vinyals-RNN-sample	.300	.531	.590	.828	.339	.597	.527	.706	.0259	.697
Vinyals-RNN-max	.293	.530	.588	.829	.343	.599	.535	.709	.0256	.700
Vinyals-RNN-max-direct	.226	.518	.539	.808	.313	.583	.490	.702	.0257	.694
SGM	-	-	-	-	-	-	-	-	.0245	.710
set-RNN	**.310**	**.538**	**.607**	**.838**	**.361**	**.607**	**.548**	**.731**	.0241	**.720**

Table 3: Comparison of different approaches. "-" means result not available. For *hamming loss*, the lower the value is, the better the model performs. For all other measures, the higher the better.

- Set-RNN : performs the best in all metrics on all datasets
- Vinyals-RNN-max and Vinyals-sample : perform not bad but degrades significantly

- instance-F1 : basically determined by the popular labels' performance
- label-F1 : sensitive to the performance on rare label

Results and Analysis

4.2 Experimental Results

- set-RNN predicts rare labels better than seq2seq-RNN

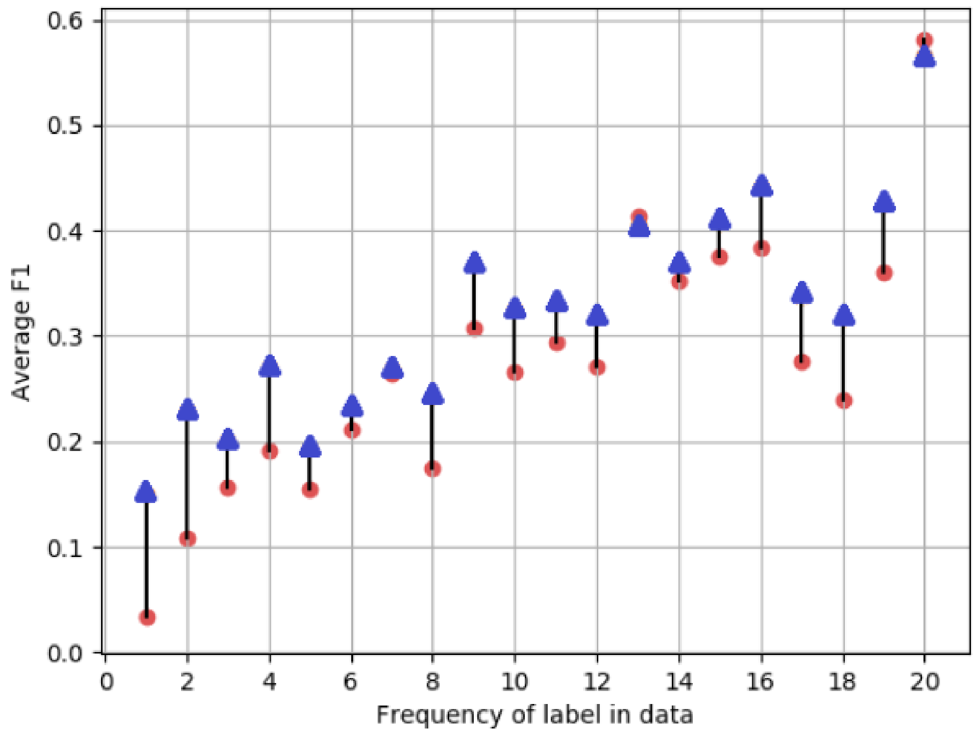

Figure 1: Average F1 over rare labels with the same frequency on TheGuardian dataset. Blue(Δ)=set-RNN, Red(·)=seq2seq-RNN.

4.2 Experimental Results

- analyze benefit
- test two prediction strategies
  - 1) finding the sequence with the highest probability and outputting the corresponding set

    (default prediction : all models except set-RNN)
  - 2) outputting the set with highest probability (default prediction : set-RNN)
- Vinyals-RNN-uniform, set-RNN benefit most from predicting the top set
- Perform promotion: model has to spread probability mass across different sequence permutations of the same set

Methods	Slashdot		RCV1-v2		TheGuardian		AAPD	
	label-F1	instance-F1	label-F1	instance-F1	label-F1	instance-F1	label-F1	instance-F1
seq2seq-RNN	.270→.269	.528→.528	.561→.561	.824→.824	.331→.336	.603→.603	.510→.511	.708→.709
Vinyals-RNN-uniform	.279→.288	**.527→.537**	.578→.587	.826→.833	**.313→.336**	**.567→.585**	**.532→.542**	.721→.724
Vinyals-RNN-sample	.300→.303	.531→.537	.590→.597	.828→.833	**.339→.351**	.597→.602	.527→.530	.706→.708
Vinyals-RNN-max	.293→.301	.530→.535	.588→.585	.829→.830	.343→.352	.599→.604	.535→.537	.709→.712
Vinyals-RNN-max-direct	.226→.228	.518→.519	.539→.538	.808→.808	.313→.316	.583→.584	.490→.490	.702→.701
set-RNN	**.297→.310**	**.528→.538**	**.593→.607**	.831→.838	**.349→.361**	**.595→.607**	.548→.548	.728→.731

Table 4: Predicting the most probable sequence vs. predicting the most probable set. Numbers before the arrow: predicting the most probable sequence. Numbers after the arrow: predicting the most probable set. We highlight scores which get significantly improved in bold (improvement is larger than 0.01).

Results and Analysis

4.3 Analysis: Sequence Probability Distribution

- How sharply (or uniformly) distributed the probabilities over different sequence permutations of the predicted set
- Normalize sequence probabilities related to the predicted set
- compute the **entropy**
- **Smaller entropy values = sharper distributions**
- predictions with different set sizes should be comparable
- Entropy/(log number of sequences)
- Result:

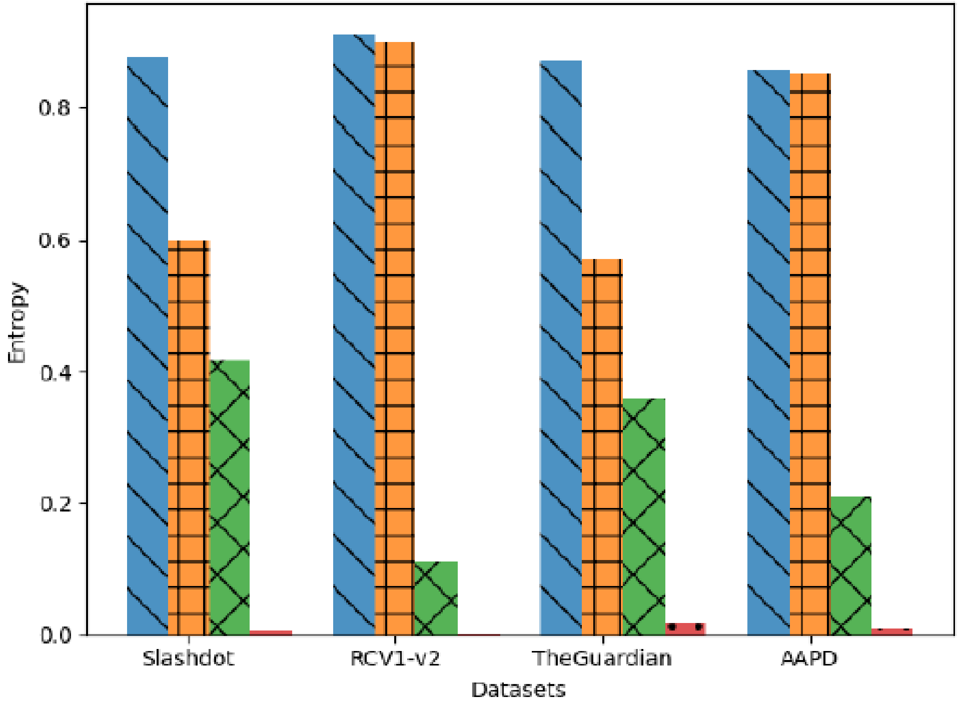

Figure 2: Entropy of sequence probability distribution for each model. Blue(\)=Vinyals-RNN-uniform, Orange(+)=set-RNN, Green(×)=Vinyals-RNN-max, Red(·)=seq2seq-RNN.

Results and Analysis

- Red(seq2seq-RNN):
  - trained with fixed label order & standard RNN objective
  - generates very sharp sequence distributions
  - only assigns probability to one sequence in the given order
  - entropy close to 0
  - predicting the set = predicting the top sequence
- Blue(Vinyals-RNN-uniform)
  - spreads probabilities across sequences
  - leads highest entropy
  - From Table 4, Vinyals-RNN-uniform perform improves
  - Training with object(3) is impossible to discover and concentrate on a particular natural label order
  - Overall Vinyals-RNN-uniform is not competitive even with the set-prediction enhancement.

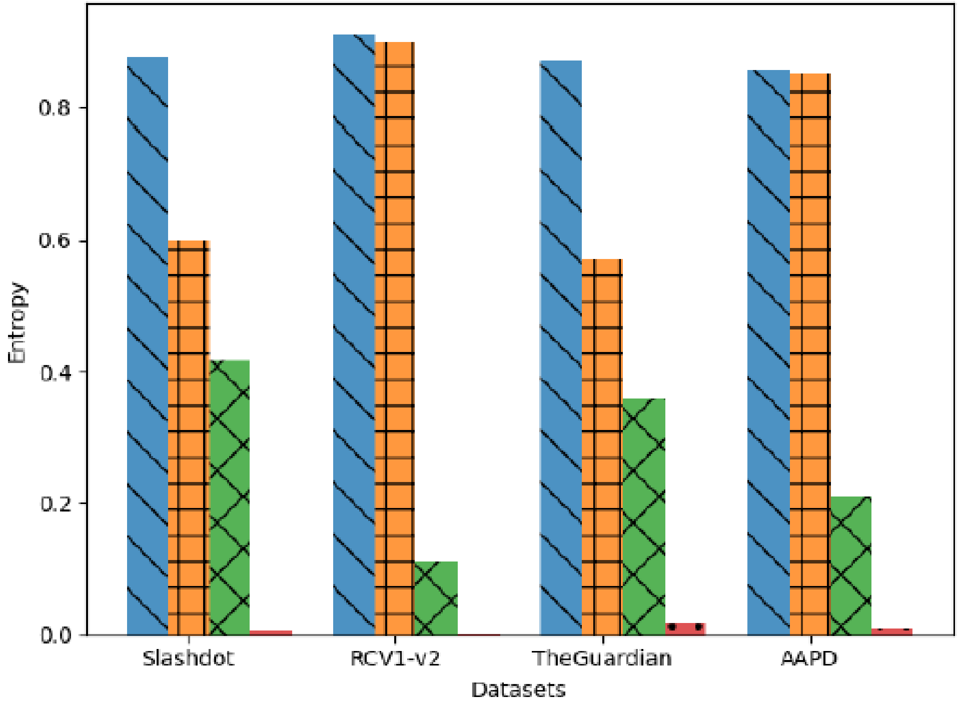

Figure 2: Entropy of sequence probability distribution for each model. Blue(\)=Vinyals-RNN-uniform, Orange(+)=set-RNN, Green(×)=Vinyals-RNN-max, Red(·)=seq2seq-RNN.

Results and Analysis

Vinyals-RNN-max & set-RNN

- Both allowed to assign probability mass to a subset of sequences
- Green(Vinyals-RNNmax)
  - sharper sequence distributions than set-RNN
  - because it intend to allocate most probability to the most probable sequence
  - due to max operator in training objective:

$$\mathit{maximize} \sum_{n=1}^{N} \max_{\mathbf{s} \in \pi(\mathbf{y}^{(n)})} \log p(\mathbf{s}|x^{(n)}) \qquad (2)$$

- Orange(set-RNN)
  - From Table 4:
    - set-RNN 表現最好
    - Vinyals-RNN-max表現普通

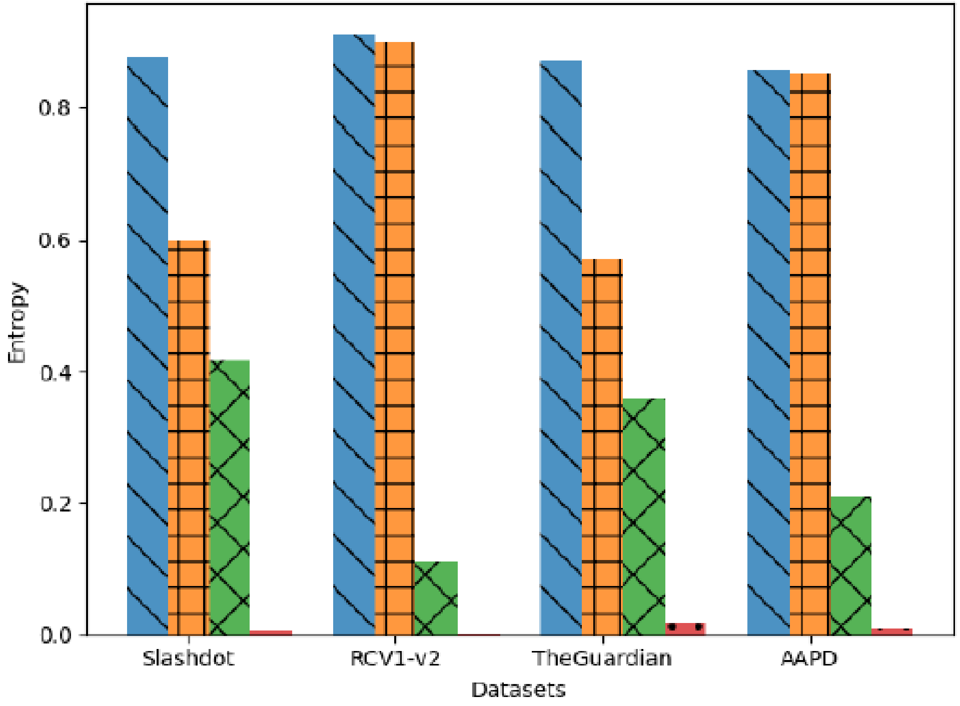

Figure 2: Entropy of sequence probability distribution for each model. Blue(\)=Vinyals-RNN-uniform, Orange(+)=set-RNN, Green(×)=Vinyals-RNN-max, Red(·)=seq2seq-RNN.

Results and Analysis

Slashdot and TheGuardian

- benefit more from predicting the most probable set than RCV1 and AAPD
- Because larger label cardinalities(基數)
- more permutations for one set potentially

Case Analysis 01

set-RNN works with two examples

- **RCV1-V2 (set-RNN)**
- most probable set (also correct set here) predicted by set-RNN :
  - {forex, markets, equity, money markets, metals trading, commodity} has the maximum total probability of 0.161
- Top sequences (decreasing probability order) :

PROB	SEQUENCE
0.0236	equity, markets, money markets, forex
0.0196	**forex, markets, equity, money markets, metals trading, commodity**
0.0194	**equity, markets, forex, money markets, metals trading, commodity**
0.0159	**markets, equity, forex, money markets, metals trading, commodity**
0.0157	**forex, money markets, equity, metals trading, markets, commodity**
0.0153	**forex, money markets, markets, equity, metals trading, commodity**
0.0148	markets, equity, money markets, forex
0.0143	**money markets, equity, metals trading, commodity, forex, markets**
0.0123	**markets, money markets, equity, metals trading, commodity, forex**
0.0110	**markets, equity, forex, money markets, commodity, metals trading**
0.0107	**forex, markets, equity, money markets, commodity, metals trading**
0.0094	**forex, money markets, equity, markets, metals trading, commodity**

Case Analysis 02

- **TheGuardian** (seq2seq-RNN V.S. set-RNN)
  - seq2seq-RNN :Tate Modern (incorrect but more popular label)
  - Set-RNN: Tate Britain (correct but less popular label)
- Training data: Exhibition is more frequent than Tate Britain and Tate Modern
- By decreasing frequency:
  - **Exhibition & Tate Modern: 19**
  - Exhibition & Tate Britain: 3
- By set level (co-occurs):
  - Exhibition & Tate Modern: 12
  - **Exhibition & Tate Britain: 22**
- 本例而言，強加入序列順序會使概率估計產生偏誤，從而導致預測錯誤。

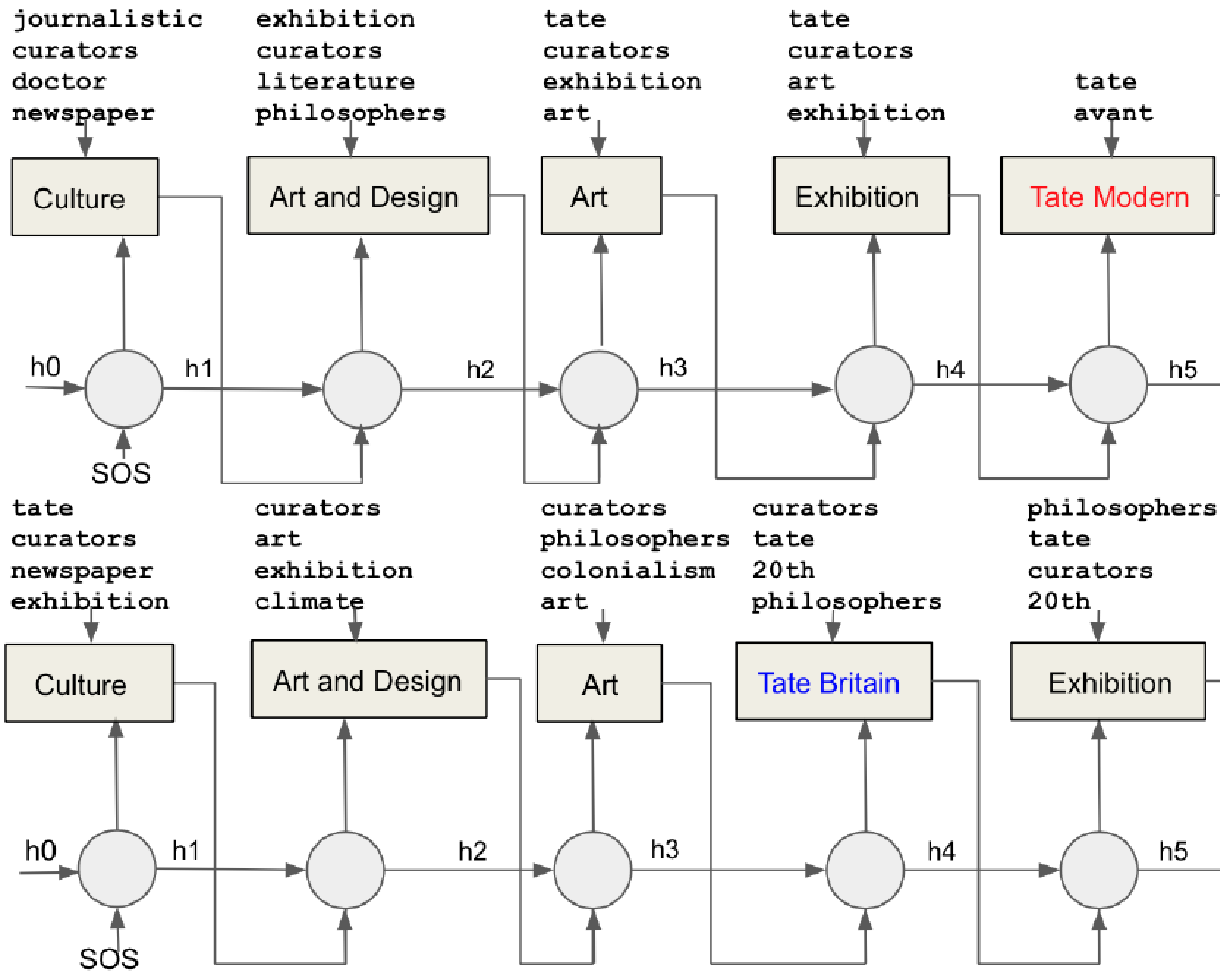

Figure 3: Top: best sequence by seq2seq-RNN; bottom: best sequence by set-RNN. Above models, at each time, we list the top unigrams selected by attention.

Conclusion

- RNN only directly defines probabilities for sequences not for sets
- Previous approaches:
  - Transform a set to a sequence in some pre-specified order
  - relate the sequence probability to the set probability in some ad hoc way
- our formulation
  - Derived from principled notion of set probability.
  - We define the set probability as the sum of all corresponding sequence permutation probabilities
- New training objective: maximizes the set probability
- New prediction objective: finds the most probable set
- give RNN model more freedom to automatically discover and utilize the best label orders