Pathology Notes Project - IDSC

Spring 2022

Matthew Rossi

Pathology Notes Project

Broad Goal: build workflows to extract meaningful, useful data from unstructured cancer pathology reports pulled from the University of Miami's Jackson Memorial Hospital.

Inspiration

Hierarchical Convolutional Attention Networks for Text Classification

Shang Gao, Arvind Ramanathan, and **Georgia Tourassi**
{gaos, ramanathana, tourassig}@ornl.gov
Computational Science and Engineering Division
Oak Ridge National Laboratory
Oak Ridge, TN, USA

2018

Abstract

machine translation has self-attention mecha- in place of recurrent crease training speed model accuracy. We his approach with the onal filters and a hi- to create a document that is both highly ac-

patterns useful for NLP tasks, especially over long segments of text, they can be slow to train compared to other deep learning architectures – in order to calculate the gradients associated with any given word in a sequence, an RNN must backpropogate through all previous words in that sequence, resulting in backpropogation functions far more complex than those in feedforward or convolutional architectures.

CNNs, traditionally used for computer vision, have also been applied to NLP tasks with notable

Classifying cancer pathology reports with hierarchical self-attention networks

Shang Gao[a,*], John X. Qiu[a], Mohammed Alawad[a], Jacob D. Hinkle[a], Noah Schaefferkoetter[a], Hong-Jun Yoon[a], Blair Christian[a], Paul A. Fearn[b], Lynne Penberthy[b], Xiao-Cheng Wu[c], Linda Coyle[d], Georgia Tourassi[a,*], Arvind Ramanathan[a,*]

[a] Computational Sciences and Engineering Division, Health Data Sciences Institute, Oak Ridge National Laboratory, Oak Ridge, TN, USA
[b] Surveillance Informatics Branch, Division of Cancer Control and Population Sciences, National Cancer Institute, Bethesda, MD, USA
[c] Louisiana Tumor Registry, Louisiana State University Health Sciences Center School of Public Health, New Orleans, LA, USA
[d] Information Management Services Inc, Calverton, MD, USA

ARTICLE INFO

Keywords:
Cancer pathology reports
Clinical reports
Deep learning
Natural language processing
Text classification

2019

ABSTRACT

We introduce a deep learning architecture, hierarchical self-attention networks (HiSANs), designed for classifying pathology reports and show how its unique architecture leads to a new state-of-the-art in accuracy, faster training, and clear interpretability. We evaluate performance on a corpus of 374,899 pathology reports obtained from the National Cancer Institute's (NCI) Surveillance, Epidemiology, and End Results (SEER) program. Each pathology report is associated with five clinical classification tasks – site, laterality, behavior, histology, and grade. We compare the performance of the HiSAN against other machine learning and deep learning approaches commonly used on medical text data – Naive Bayes, logistic regression, convolutional neural networks, and hierarchical attention networks (the previous state-of-the-art). We show that HiSANs are superior to other machine learning and deep learning text classifiers in both accuracy and macro *F*-score across all five classification tasks. Compared to the previous state-of-the-art, hierarchical attention networks, HiSANs not only are an order of magnitude faster to train, but also achieve about 1% better relative accuracy and 5% better relative macro *F*-score.

Our Dataset

- 200k pathology report documents.
- Patient ID and ICD-10 diagnosis codes.
- No dates, no tumor ID.
- Each document contains all reports for a patient with the same ICD-10 code.

ICD-10 Codes

International Classification of Diseases- 10th edition

- Used for insurance/billing purposes
- Format: CXX.YYY
  - XX is “top-level classification”
  - YYY is “subclassification”
- C00-D49: Neoplasms (Tumors)
  - C is malignant, D is benign or unknown behavior
- Ex: C50.112

Our Dataset

- 1394 unique diagnoses
  - Average class has only ~140 examples
- 141 unique top-level diagnoses
  - Average class has ~1400 examples
- Shortest document is 971 characters
- Longest document is 8192 characters

Goals for this Semester

- Deep learning methods:
  - CNN, HCAN, HiSAN
    - Bug detection + correction
    - Optimization
- Organize shareable repository

Processing Pipeline

- Dataset management: slicing datasets using fetch_subset.py
- Text pre-processing
- Embedding using Word2Vec (Feature Extraction)
- Evaluate accuracy using macro score

Data Management - Resampling

balance_classes(X, y, max_class_size=None)

- Up-samples minority classes to max_class_size.
- By default, max_class_size takes the size of the largest class.

Training Models

- Create training set, validation set, and test set.
- Shuffle data prior to each epoch

Accuracy

Micro Accuracy

$$= \frac{\#\ correctly\ classified}{total\ \#\ of\ datapoints}$$

F1 Macro Accuracy

Harmonic mean of precision and recall, averaged across classes

Classification Tasks

- Abstracts data (baseline)
  - 8k datapoints, 8 classes
- Top-level Site Prediction
  - 92k datapoints, 67 classes
- Breast Cancer Subsite Prediction
  - 6k datapoints, 8 classes

Hyperparameters

- Dropout (Regularization)
- Resampling
- Number of epochs
- Number of attention heads (HiSAN only)

Baseline – Abstracts Dataset

- Text of 8000 pubmed abstracts
- 1000 datapoints per class
- Target label is subject
- 8 classes:
  - chemistry, diagnosis, genetics, metabolism, pathology, physiology, psychology, surgery

Abstracts Dataset Hyperparameters

	HCAN	HiSAN
Epochs	10	25
Dropout	0.2	0.2
Attention Heads	N/A	16
Resampling?	No	No

Abstracts Dataset Results

	HCAN (dropout=0.2)	HiSAN (dropout=0.2, attention_heads =16)
Epochs	10	25
CPU Time	2492s (41.5 min)	3290s (54.8min)
Avg. CPU Time per Epoch	249.2s (4.15 min)	131.6s (2.2 min)

Abstracts Dataset Results

	HCAN (dropout=0.2)	HiSAN (dropout=0.2, attention_heads =16)
Epochs	10	25
Best Train Micro Acc	0.9263	0.9042
Best Train Macro Acc	0.9265	0.9042
Best Val Micro Acc	0.7106	0.7581
Best Val Macro Acc	~	0.7595

Top-Level Site Prediction

- Target label: CXX
- 92k Datapoints, 67 classes
  - (About half of the dataset)
- Same classes as the HiSAN paper
  - (for comparison)

Top-Level Site Class Imbalance

```
C00:  124    C01:  657    C02: 1115    C03:   83    C04:  255    C05:  255
C06:  883    C07:  433    C08:  172    C09:  813    C10:  834    C11:  221
C12:   45    C13:  228    C14:  212    C15: 1141    C16: 1016    C17:  162
C18: 3661    C19:  548    C20:  791    C21:  469    C22: 2110    C23:  137
C24:  362    C25: 3006    C26:   55    C30:  383    C31:  371    C32: 1605
C33:   40    C34: 5720    C37:   67    C38:   91    C40:  228    C41: 1231
C44: 7735    C47:   69    C48:  330    C49: 3470    C50: 16791   C51:  151
C52:   86    C53: 1087    C54: 1446    C55:  456    C56: 1366    C57:  133
C60:   94    C61: 6246    C62:  716    C63:   23    C64: 2956    C65:  237
C66:  256    C67: 3450    C68:  609    C69:  494    C70:   44    C71: 1666
C72:   53    C73: 2888    C74:  145    C75:   51    C76: 2044    C77: 3459
C80: 3941    >>> _
```

Top-Level Site Prediction

	HCAN	HiSAN (no resampling, Attention_heads =8)	HiSAN (no resampling, Attention_heads =16)
Epochs	10	16	21
CPU Time	~55000s (15 hr)	~40000s (11 hr)	~26000s (7hr)
Avg. CPU Time per Epoch	~5500s (1.5 hr)	~2500s (42 mins)	~1250s (21 mins)

Top-Level Site Prediction

HCAN	No Resampling
Dropout=0.1	Train: 0.7488; 0.5410 Val macro: 0.3184
Dropout=0.2	Train: 0.7197; 0.4555 Val macro: 0.3352

Top-Level Site Prediction

HiSAN with 8 attention heads	No Resampling	Resample to 1k	Upsample to Max (13299)
Dropout=0.1	0.6256; 0.3789		
Dropout=0.15	0.6257; 0.3781		
Dropout=0.2	0.6256; 0.3833		
Dropout=0.3	0.6233; 0.3676		
Dropout=0.4	0.6194; 0.3592	0.5390; 0.3468	0.5513; 0.3517

Top-Level Site Prediction

HiSAN with 16 attention heads	Dropout=0.2
Epochs	21
Accuracy	Train: 0.6475; 0.4555 Val: 0.6302; 0.3837

Breast Cancer Subsite Prediction

- Target label: C50.X
- 6k Datapoints, 8 classes (subsites):

1. nipple & areola
2. central portion
3. upper-inner quadrant
4. lower-inner quadrant
5. upper-outer quadrant
6. lower-outer quadrant
7. axillary tail
8. overlapping sites

Breast Cancer Subsite Prediction

```
words = [
    ["nipple", "areola"],
    ["center", "central"],
    ["upper", "inner"],
    ["lower", "inner"],
    ["upper", "outer"],
    ["lower", "outer"],
    ["axillary", "tail"],
    [],
    ["overlapping"],
]
n_classifiable = 0
for i in [0,1,2,3,4,5,6,8]:
    icd = 'C50.' + str(i)
    one_class = data[(data['c.icd10_after_spilt']==icd)]
    assert len(one_class) != 0
    lst = []
    for text in one_class['c.path_notes']:
        text = process_text(text)
        lst.append(True in [(word in text) for word in words[i]])
    n_classifiable += sum(lst)
    print("%s: %f" % (icd, sum(lst)/len(one_class)))
print("Percent classifiable:", n_classifiable/len(data))
```

Breast Cancer Subsite Prediction

```
words = [
    ["nipple", "areola"],
    ["center", "central"],
    ["upper", "inner"],
    ["lower", "inner"],
    ["upper", "outer"],
    ["lower", "outer"],
    ["axillary", "tail"],
    [],
    ["overlapping"],
]
n_classifiable = 0
for i in [0,1,2,3,4,5,6,8]:
    icd = 'C50.' + str(i)
    one_class = data[(data['c.icd10_after_spilt']==icd)]
    assert len(one_class) != 0
    lst = []
    for text in one_class['c.path_notes']:
        text = process_text(text)
        lst.append(True in [(word in text) for word in words[i]])
    n_classifiable += sum(lst)
    print("%s: %f" % (icd, sum(lst)/len(one_class)))
print("Percent classifiable:", n_classifiable/len(data))
```

```
C50.0: 0.233136
C50.1: 0.148583
C50.2: 0.170213
C50.3: 0.155303
C50.4: 0.193548
C50.5: 0.141230
C50.6: 0.466667
C50.8: 0.025275
Percent classifiable: 0.16290349556782072
```

Breast Cancer Subsite Task

	HCAN	HiSAN1	HiSAN2	HiSAN3
Epochs	10	45	20	29
Dropout	0.2	0.2	0.2	0.4
Attention Heads	N/A	16	16	16
Resampling?	Max	No	Max	Max

Breast Cancer Subsite Prediction

HiSAN	No resampling	Upsampling to Max (1448 datapoints)
Dropout=0.2	Train: 0.4851; 0.3596 Val: 0.2926; 0.1580	Train: 0.6416; 0.6353 Val: 0.2570; 0.1927
Dropout=0.4	~	Train: 0.5868; 0.5787 Val: 0.2399; 0.1939

Breast Cancer Subsite Prediction

HCAN	Upsampling to Max (1448 datapoints)
Dropout=0.2	Train: 0.8335; 0.8313 Val: 0.2415; 0.1882

Next steps...

- Apply pre-trained models (ClinicalBERT)
- Extract other data from reports: size, stage/grade, etc.
- Look for additional methods to remove noise from data to improve performance