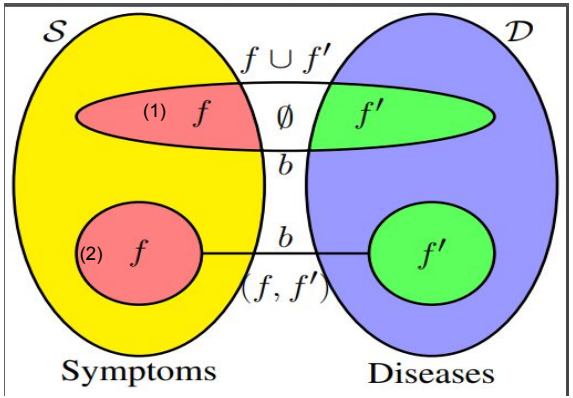
Problem Statement:

Formulate a framework that deal with the set-matching(SM) and hence solve the bipartite hyperedge prediction(BHP) problem.



f is subset of S(symptoms set), |f|>=1.
f' is subset of D(diseases set), |f|>=1.
(1) in figure => BH in hypergraph form.
(2) in figure => BH in set matching form.

Dataset Details:

Table 1: The list of bipartite hypergraph datasets used in this work, along with their vital statistics: # left nodes $|\mathcal{V}|$, # right nodes $|\mathcal{V}'|$, # left hyperedges $|\mathcal{F}|$, # right hyperedges $|\mathcal{F}'|$, and # bipartite hyperedges $|\mathcal{B}|$.

Dataset	Left nodes ($\mathcal{V}$)	Right nodes ($\mathcal{V}'$)	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{V}'\rvert$	$\lvert\mathcal{B}\rvert$	$\lvert\mathcal{F}\rvert$	$\lvert\mathcal{F}'\rvert$
tmdb-cc	Cast (actors)	Crew (other members)	4,556	3,802	2,825	2,824	2,744
tmdb-ck	Cast (actors)	Plot keywords	3,156	1,256	2,669	2,656	2,621
mag-acm-ak	Authors	Keywords	1,059	2,338	1,388	847	1,379

These dataset are prepared from **TMDB 5000 movie dataset(Kaggle)** and Microsoft **Academic Graph ACM subset**

Formulate **Cross-attention framework** to solve this problem.

This work was submitted and unfortunately got rejected in Nips2020 by Name "The CAT SET on the MAT: Cross Attention for Set Matching in Bipartite Hypergraphs".(still in progress).

Problem Statement:

Proposed Graph attention based methods to solve text classification problem.

Data, its preprocessing and graph construction is same as done in TEXT_GCN by Yao.
Graph Attention framework is inspired from GAT by P Veličković.

Datasets used:

Dataset	R8	R52	ohsumed	MR
training	4937	5879	3022	6398
test	2189	2568	4043	3554
nodes	15362	17992	21557	29426
sentence length	65.72	69.8	135.82	20.3

Graph Construction step:

$$A_{ij} = \begin{cases} \text{PMI}(i,j) & i,j \text{ are words, PMI}(i,j) > 0 \\ \text{TF-IDF}_{ij} & i \text{ is document, } j \text{ is word} \\ 1 & i = j \\ 0 & \text{otherwise} \end{cases}$$

Only documents have labels and a combined graph containing both words and documents is constructed

Problem Statement:

Tri-char grams with CNN to solve sentiment analysis

Use the pytorch implementation of Yoon Kim's paper.
Each word(length>3) of dataset got split into tri-char grams.
Word2vec model is trained over this training corpus of tri-char grams.
Word2vec embedding obtained from above training is used as embedding matrix.
Finally original single layer CNN along with Cross-entropy as loss function.

Datasets used:

Use the dataset Stanford Sentiment Treebank(SST-1).
An extension of MR but with train/dev/test splits provided
Fine-grained labels: very positive, positive, neutral, negative, very negative.
Dataset Size: 11855, test size: 2210,

Solution1:

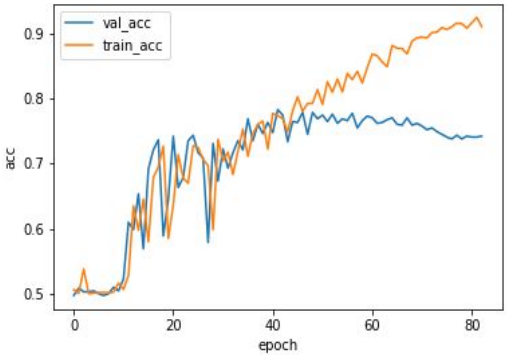
The framework is motivated from Hyper-Sagnn by Zhang.
It used self attention mechanism to predict the hyperedges.
Cross-entropy framework is based on this self-attention mechanism.
When Basic Cross-attention framework combined with self attention mechanism, good improvement is seen.
It uses Cross entropy as loss function.
Embedding used for nodes in hyperedges is node2vec for hypergraph.
Main reason for its rejections in Nips 2020 as per review is less number of datasets.

Solution2:

Main problem is directly applying vanilla graph attention network over the constructed graph does not consider its weight which make performance poor.

Initial feature matrix is identity matrix ie this performance is based on structural informations.

To consider its weight, we apply first GCN layer and then graph attention layer over it. This improved the performance a lot and make it par to state of the art on some datasets.



Accuracy plot MR dataset, on testdata:76.45%, at epoch 45

Courses:
Natural Language Processing by HSE university (Coursera Audit)
Currently going through numerical optimization on NPTEL.

Link to projects mentions:
Problem2:https://github.com/swyam/TEXTGAT.git
Problem3:https://github.com/swyam/CNN-text-classification-using-trichargrams.git

Personal Website: https://swyam.github.io