From Group to Individual Labels using Deep Features

Dimitrios Kotzias[1] Misha Denil[2] Nando De Freitas[2,3] Padhraic Smyth[1]

[1]University of California, Irvine
[2]University of Oxford
[3]Canadian Institute for Advanced Research

[1]{dkotzias,smyth}@ics.uci.edu
[2]{mdenil,nando}@cs.ox.ac.uk

Hareesh Bahuleyan
Borealis AI

Multi-Instance Learning (General) Setting

- Training instances (e.g. sentences) : X = {x_i}, i = 1 . . .N
- NO Labels at the instance level
- Instead, we have labels for a group of instances (e.g. document) $\mathcal{D} = \{(\mathcal{G}_k, \ell_k)\}_{k=1,\ldots,K}$
- **Modelling**: (assume binary setting)
  - ➡ Predict 1 if the given group contains at least 1 positive label
  - ➡ Predict 0 if the given group contains NO positive labels
- **Examples**:
  - ➡ CV: Given an image whether a specific object of interest is present or not ?
  - ➡ NLP: Does a document speak about a topic of interest or not ? (if atleast one mention of the topic)

Focus of this paper

- Consider product reviews from Amazon - can easily obtain “group” level labels for sentiment
- Generating labels at the instance-level (e.g. for sentences within reviews) is much more time consuming
- **Contribution**:
  - An approach to the problem of *using group-level labels* to learn instance-level classification models

An Example

- Observations:
  - Positively labelled reviews may still have a set of negative sentence mentions (and vice versa)
  - Hence, relax the MIL whole-part assumption
  - **Reformulation:** The presence of a combination of instance types determines the label of the group

Paul Bettany did a great role as the tortured father whose favorite little girl dies tragically of disease.
For that, he deserves all the credit.
However, the movie was mostly about exactly that, keeping the adventures of Darwin as he gathered data for his theories as incomplete stories told to children and skipping completely the disputes regarding his ideas.
Two things bothered me terribly: the soundtrack, with its whiny sound, practically shoving sadness down the throat of the viewer, and the movie trailer, showing some beautiful sceneries, the theological musings of him and his wife and the enthusiasm of his best friends as they prepare for a battle against blind faith, thus misrepresenting the movie completely.
To put it bluntly, if one were to remove the scenes of the movie trailer from the movie, the result would be a non descript family drama about a little child dying and the hardships of her parents as a result.
Clearly, not what I expected from a movie about Darwin, albeit the movie was beautifully interpreted.

Modelling Approach

Task Specific Pre-Training

- Train a CNN to predict the document level label
- Simultaneously learns sentence and document level representations
- The representations hopefully capture task specific information
- The intermediate sentence representations (**x_i**) will be useful in the next stage

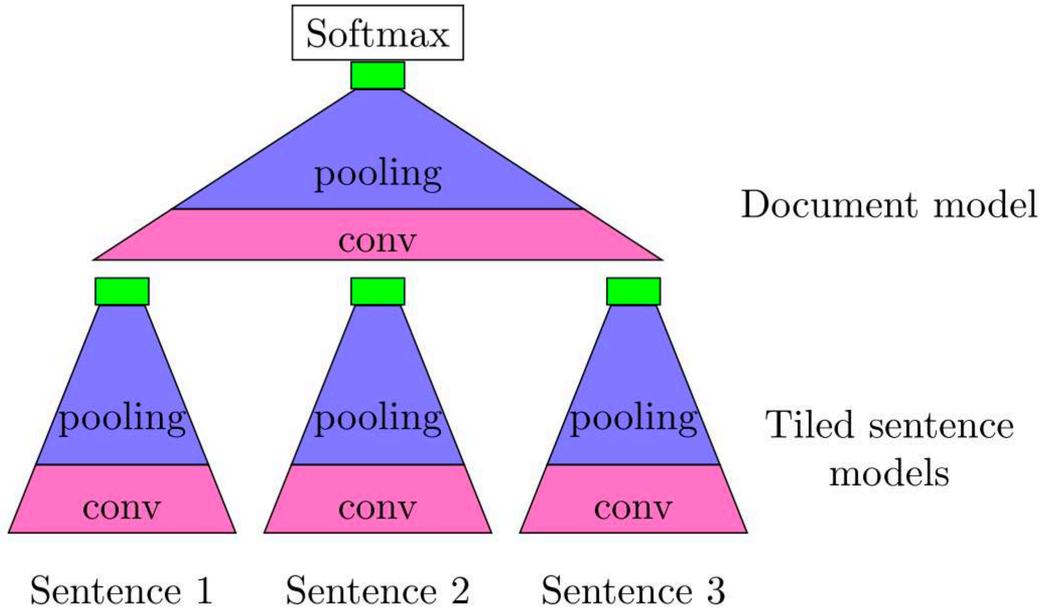
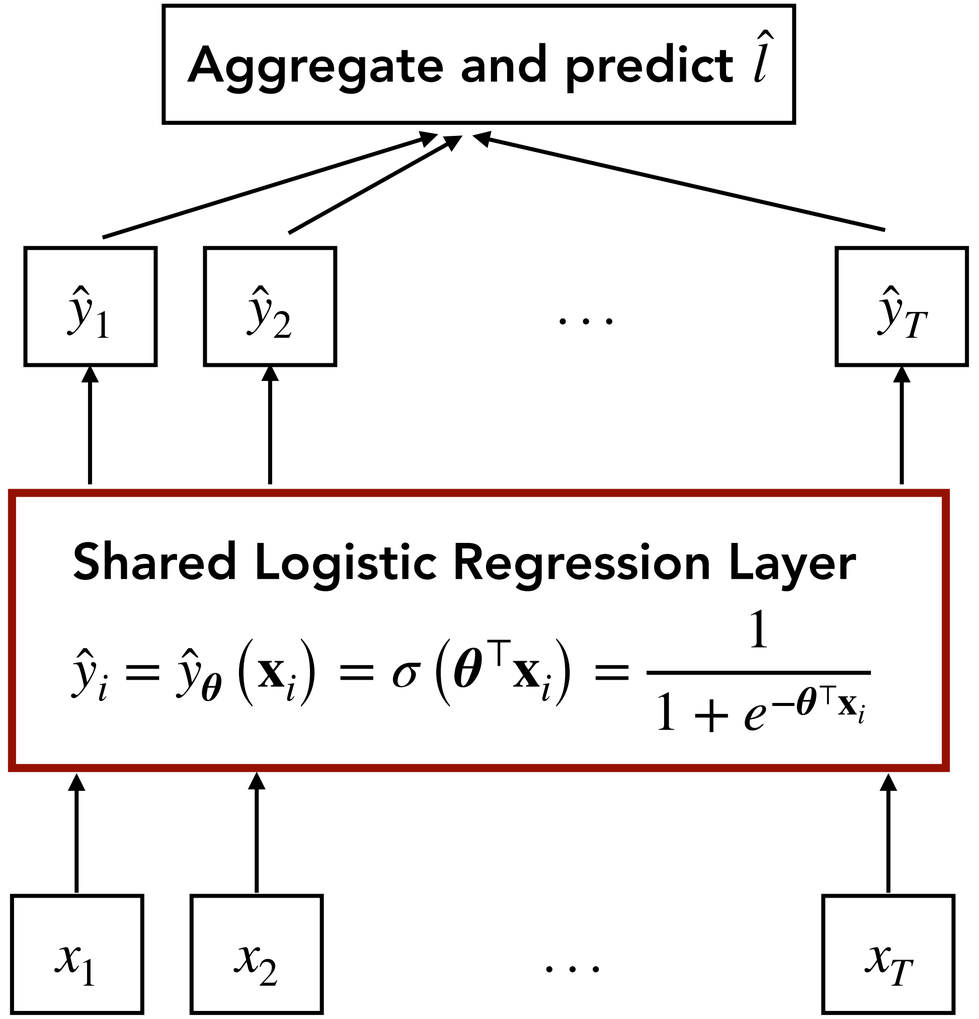


Figure 1: Model from Denil *et al.* [7]. The green squares indicate embedding vectors for sentences (atop the tiled sentence models) and for documents (atop the document model).

MIL Model

- **Forward Pass:**
  - Feed the sentence representations (x_i) to the model to predict instance level labels
  - Aggregate sentence predictions $y_\theta(x_i)$ to obtain the document level prediction



MIL Objective Function

$$J(\boldsymbol{\theta}) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\mathcal{K}(\mathbf{x}_i,\mathbf{x}_j)\Delta_1\left(\hat{y}_i,\hat{y}_j\right) + \frac{\lambda}{K}\sum_{k=1}^{K}\Delta_2\left(\hat{\ell}_k,\ell_k\right)$$

MIL Objective Function

$$J(\boldsymbol{\theta}) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)\Delta_1\left(\hat{y}_i, \hat{y}_j\right) + \frac{\lambda}{K}\sum_{k=1}^{K}\Delta_2\left(\hat{\ell}_k, \ell_k\right)$$

- K groups in a batch
- Compute loss by comparing prediction to ground-truth (log loss)

MIL Objective Function

$$J(\boldsymbol{\theta}) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) \Delta_1 \left(\hat{y}_i, \hat{y}_j \right) + \frac{\lambda}{K} \sum_{k=1}^{K} \Delta_2 \left(\hat{\ell}_k, \ell_k \right)$$

- N: total number of instances coming from all groups in a batch

- Kernel Function/ Distance Function to compute level of similarity or dis-similarity between instances in a batch

MIL Objective Function

$$J(\boldsymbol{\theta}) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) \Delta_1 \left(\hat{y}_i, \hat{y}_j \right) + \frac{\lambda}{K} \sum_{k=1}^{K} \Delta_2 \left(\hat{\ell}_k, \ell_k \right)$$

- Predictions made at the instance level.
- But no ground truth available
- Similar sentences should produce similar labels
- Squared loss

MIL Objective Function

$$J(\boldsymbol{\theta}) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) \Delta_1 \left(\hat{y}_i, \hat{y}_j \right) + \frac{\lambda}{K} \sum_{k=1}^{K} \Delta_2 \left(\hat{\ell}_k, \ell_k \right)$$

- N: total number of instances coming from all groups in a batch

- Kernel Function/ Distance Function to compute level of similarity or dis-similarity between instances

- Predictions made at the instance level.
- But no ground truth available

- K groups in a batch
- Compute loss by comparing prediction to ground-truth (log loss)

Loss Intuitions

$$J(\boldsymbol{\theta}) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\mathcal{K}(\mathbf{x}_i,\mathbf{x}_j)\Delta_1\left(\hat{y}_i,\hat{y}_j\right) + \frac{\lambda}{K}\sum_{k=1}^{K}\Delta_2\left(\hat{\ell}_k,\ell_k\right)$$

- **First Term:**
  - Forces similar items across different groups to have similar labels and allows for *inter-group knowledge transfer*
  - Acts as a regularizer to leverage instance level similarity information

- **Second Term:**
  - Necessary for the multi-instance learning
  - Without the 2nd term, every instance can be assigned the same $\hat{y}_i$ and the loss will be zero!

Specific Form of the Loss Function

- RBF Kernel for the similarity function
- Logistic Regression (only learnable parameters)
- Aggregation of instance predictions via Averaging

$$J(\boldsymbol{\theta}) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} e^{(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2)} \Big(\sigma(\boldsymbol{\theta}^\top \mathbf{x}_i) - \sigma(\boldsymbol{\theta}^\top \mathbf{x}_j)\Big)^2 + \frac{\lambda}{K}\sum_{k=1}^{K}\left(\frac{1}{|\mathcal{G}_k|}\left(\sum_{i\in\mathcal{G}_k}\sigma(\boldsymbol{\theta}^\top \mathbf{x}_i)\right) - \ell_k\right)^2 \tag{3}$$

Results

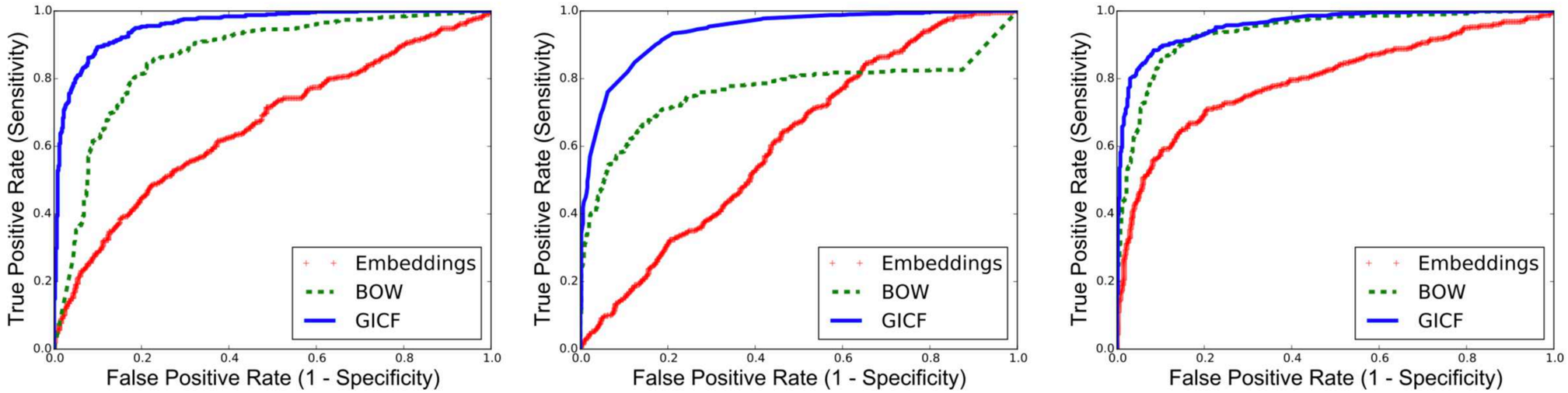
Group Level Prediction

	Accuracy			AUC		
	Amazon	IMDb	Yelp	Amazon	IMDb	Yelp
Logistic w/ BOW on Documents	85.8%	86.20%	**91.25%**	88.08%	88.32	**94.41**
Logistic w/ BOW on Sentences	88.3%	81.81%	78.16%	87.19%	82.67	67.87
Logistic w/ Embeddings on Documents	67.82%	58.23%	81.00%	61.24%	60.77	82.59
GICF w/ Embeddings on Sentences	**92.8%**	**88.56%**	88.73 %	**91.73%**	**88.36%**	92.36%

Table 3: Accuracy and Area-Under-the-Curve (AUC) scores for predicting labels at the group (document) level for the baselines and our proposed method (GICF). Training is always done at the group level. Testing on sentences corresponds to scoring each sentence separately and aggregating the results. BOW or embeddings corresponds to the features used.

Instance Level Prediction

- Created a evaluation set of 1000 reviews - containing sentence level labels (human annotated)



(a) Amazon sentence ranking (b) Imdb sentence ranking (c) Yelp Sentence ranking

Figure 2: ROC plots for instance level classification, for each of the baselines and our method for the three datasets

Thank You