Human Assisted Machine Learning: Consensus Driven Data Curation

Ganesh Ramakrishnan, Department of Computer Science and Engineering, IITB

January 17, 2021

Acknowledgements for this work: Ashish Kulkarni, Oishik Chatterjee, Sunita Sarawagi

Problem Definition

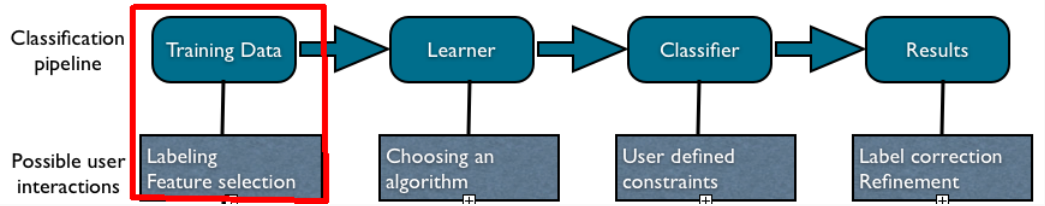
$\mathscr{X} = \{\mathbf{x}_i\}_{i \in \{1 \cdots n\}}$	set of data instances
$\mathscr{Y} = \{1 \cdots K\}$	label set

Multi-label classification problem: learn $f : \mathbf{X} \to \mathscr{Z}$. **Reality: We do not begin here..**

D_L	(partially) labeled dataset
D_U	a large unlabeled dataset
$\Lambda = \{\lambda_k\}_{k \in \{1 \cdots m\}}$	m independent labelers, both human and machine, possibly noisy and untrained λ_k
$\tau_{ij} \in \mathscr{Y}$	response of labeler $\lambda_j \in \Lambda$ when triggered (and $\tau_{ij} = 0$ otherwise

Realistic Goal: Expand D_L, minimize labeling cost and train any λ^j's that are trainable.

Human Interaction at the level of (Training) Data



1. **Consensus between Human/Rule-based Labelers and Machine Labelers**
   - **Data Programming using Continuous and Quality-Guided Labeling Functions [AAAI 2020]**
   - **An Interactive Multi-Label Consensus Labeling Model for Multiple Labeler Judgments [AAAI 2018]**
   - **Synthesis of Programs from Multimodal Datasets [AAAI 2018]**
   - **Interactive Martingale Boosting [IJCAI 2016]**
2. Addressing hierarchical labeling settings and Data Subset selection for Labeling
   - **Learning From Less Data: Diversified Subset Selection and Active Learning in Image Classification Tasks [WACV 2019]**
   - **OCR On-the-Go: Robust End-to-end Systems for Reading License Plates and Street Signs [ICDAR 2019]**
   - **Beyond clustering: Sub-DAG Discovery for Categorising Documents [CIKM 2016]**
   - **Summarizing Multi-Document Topic Hierarchies using Submodular Mixtures [ACL 2015]**

Personalized classifiers: evolving a classifier from a large reference knowledge graphs [SIGIR Workshop 2014]

Motivating Example: Image Labeling

Labeler A	Labeler B	Labeler C
Clock 0.6 Gauge 0.3	CafePress ABARTH Wall Clock CafePress Maltese Portrait Wall Clock	Indoors 68% Meat Eater 67% Room 66% Object 65% Vertebrate 65%
Banana 0.5 Vegetable 0.4 Fruit 0.3	Plantains (Musa balbisiana)	Flower 77% Human 67% Food 67% Group of People 62% Indoors 60% Activity 56% ClubSport 55%

Multiple labelers; Multiple labels; Little or no training data.

Objective 1: Labeler reliability

Labeler A	Labeler B	Labeler C
Clock 0.6 Gauge 0.3	CafePress ABARTH Wall Clock CafePress Maltese Portrait Wall Clock	Indoors 68% Meat Eater 67% Room 66% Object 65% Vertebrate 65%
Banana 0.5 Vegetable 0.4 Fruit 0.3	Plantains (Musa balbisiana)	Flower 77% Human 67% Food 67% Group of People 62% Indoors 60% Activity 56% ClubSport 55%

Objective 1: Labeler reliability

Labeler A	Labeler B	Labeler C
Clock 0.6 Gauge 0.3	CafePress ABARTH Wall Clock CafePress Maltese Portrait Wall Clock	Indoors 68% Meat Eater 67% Room 66% Object 65% Vertebrate 65%
Banana 0.5 Vegetable 0.4 Fruit 0.3	Plantains (Musa balbisiana)	Flower 77% Human 67% Food 67% Group of People 62% Indoors 60% Activity 56% ClubSport 55%

Labelers (Labeling functions or Models) might have complementary expertise. Can the labelers mutually benefit from their individual expertise?

Objective 2: Collective Prediction

	Labeler A	Labeler B	Labeler C
	Clock 0.6 Gauge 0.3	CafePress ABARTH Wall Clock CafePress Maltese Portrait Wall Clock	Indoors 68% Meat Eater 67% Room 66% Object 65% Vertebrate 65%
	Banana 0.5 Vegetable 0.4 Fruit 0.3	Plantains (Musa balbisiana)	Flower 77% Human 67% Food 67% Group of People 62% Indoors 60% Activity 56% ClubSport 55%

Objective 2: Collective Prediction

Labeler A	Labeler B	Labeler C
Clock 0.6 Gauge 0.3	CafePress ABARTH Wall Clock CafePress Maltese Portrait Wall Clock	Indoors 68% Meat Eater 67% Room 66% Object 65% Vertebrate 65%
Banana 0.5 Vegetable 0.4 Fruit 0.3	Plantains (Musa balbisiana)	Flower 77% Human 67% Food 67% Group of People 62% Indoors 60% Activity 56% ClubSport 55%

How do we aggregate the label predictions from individual labelers, to generate consolidated high confidence label predictions for each instance?

Labeling Functions on ‘Spouse’ Extraction Task [Bach *et. al.*, 2017]

SpouseDict = {’spouse’, ’married’, wife’, ’husband’, ’ex-wife’, ’ex-husband’}

FamilyDict = {’father’, ’mother’, ’sister’, ’brother’, ’son’, ’daughter’,’grandfather’, ’grandmother’, ’uncle’, ’aunt’,’cousin’ }$\bigotimes${+”,+’-in-law’}

OtherDict = {’boyfriend’, ’girlfriend’ ’boss’, ’employee’, ’secretary’, ’co-worker’}

SeedSet = {(’Barack Obama’, ’Michelle Obama ’), (’Jon Bon Jovi’,’Dorothea Hurley’),(’Ron Howard’,’Cheryl Howard’),.....}

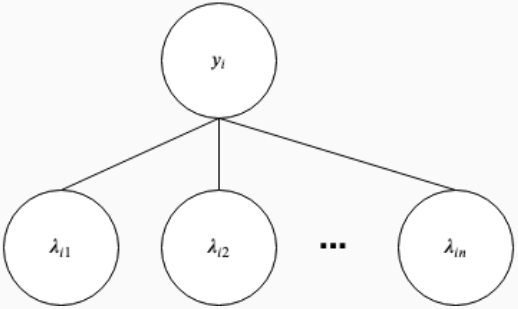
Id	Description
LF1	If some word in **SpouseDict** is present between E_1 and E_2 or within 2 words of either, return 1 else return 0
LF2	If some word in **FamilyDict** is present between E_1 and E_2, return -1 else return 0.
LF3	If some word in **OtherDict** is present between E_1 and E_2, return -1 else return 0.
LF4	If both E_1 and E_2 occur in **SeedSet**, return 1 else return 0.
LF5	If the number of word tokens lying between E_1 and E_2 are less than 4, return 1 else return 0.

Table 1: Discrete labeling functions (LFs) based on dictionary lookups or thresholded distance for the *spouse* relationship extraction task

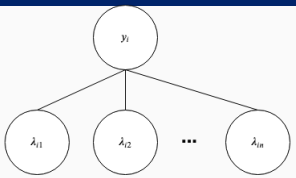
Problem Setting 1 (only Labelers are LFs)

Motivation: Lack of labeled data. Human designed labeling functions (LFs) assigning noisy labels to instances. Use these labels to generate labeled data. *(Data Programming)*
Problem Statement: Learn a generative model of true label distributions over LFs.
Data Programmming using Snorkel (Bach et al. 2017):

- Discrete Probabilistic Graphical Model
- Shared Parameters for agreement and disagreement of labeling functions



Problem Setting 1: Data Programmming using Snorkel *[Bach et al. 2017)*



Limitations:

1. Training instability due to unsupervised nature of the problem.
2. Highly sensitive to initialization, number of epochs, learning rate, *etc.*
3. Does not support continuous labeling functions.

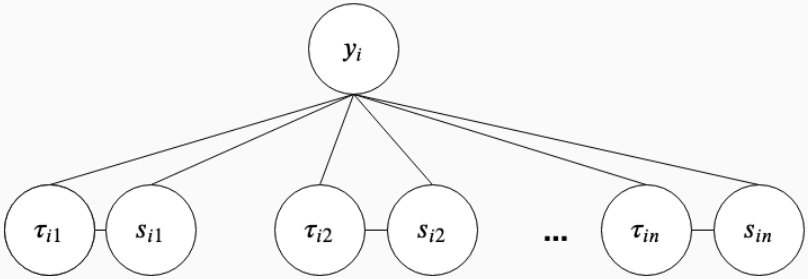
Joint probability distribution of y (true label for an instance(**X**) and $\Lambda_i = (\lambda_{i1}, \lambda_{i2}, , \lambda_{in})$ (labels assigned by the n labeling functions) is

$$P_\theta(y, \Lambda_i) = \frac{1}{Z_\theta} \exp(\sum_{j=1}^{n} \phi_j(y, \lambda_{ij}))$$

$$\phi_j(y, \lambda_{ij}) = \begin{cases} \theta_{jy} & \text{if } \lambda_{ij} = k_j, \\ -\theta_{jy} & \text{if } \lambda_{ij} \neq k_j, \lambda_{ij} \neq 0, \\ 0 & \text{otherwise.} \end{cases}$$

Problem Setting 1: Our Solution - CAGE *[AAAI 2020]*

CAGE stands for **C**ontinuous **A**nd quality **G**uided lab **E**ling functions.



Given n labeling functions $(\lambda_1, \lambda_2, , \lambda_n)$ which can be **continuous** or discrete.

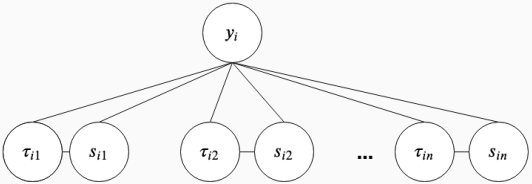
- Each LF λ_j, for a given instance x_i,
  - outputs a label $\tau_{ij} = k_j$ if triggered
  - else outputs $\tau_{ij} = 0$
- Further, if λ_j is a continuous LF, it also outputs a score $s_{ij} \in (0, 1)$.

For each x_i, we model the joint probability of the true label and the (labels, scores)

Problem Setting 1: Continuous LFs in CAGE

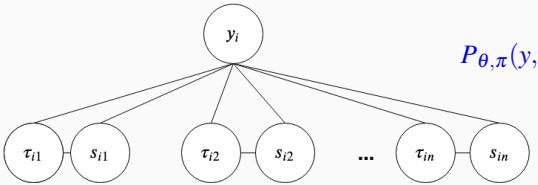
Id	Class	Description
LF1	+1	max [cosine(word-vector(u), word-vector(v))-0.8]$_+$: $u \in$ **SpouseDict** and $v \in$ {words between E_1, E_2}.
LF2	-1	max [cosine(word-vector(u), word-vector(v))-0.8]$_+$: $u \in$ **FamilyDict** and $v \in$ {words between E_1, E_2}.
LF3	-1	max [cosine(word-vector(u), word-vector(v))-0.8]$_+$: $u \in$ **OtherDict** and $v \in$ {words between E_1, E_2}.
LF4	-1	max [0.2 - Norm-Edit-Dist(E_1, E_2, u, v)]$_+$: $(u, v), (v, u) \in$ **SeedSet**.
LF5	+1	[1 - (number of word tokens between E_1 and E_2)/5.0]$_+$

CAGE: The Probabilistic Model



Joint probability distribution of y (true label for an instance(**X**) and $\Lambda_i = (\lambda_{i1}, \lambda_{i2}, , \lambda_{in})$ (labels assigned by the n labeling functions) and their scores $(s_{i1}, s_{i2}, , s_{in})$ is .....

CAGE: The Probabilistic Model



$$P_{\theta,\pi}(y,\tau_i,s_i) = \frac{1}{Z_\theta}\prod_{j=1}^{n}\psi_\theta(\tau_{ij},y)\left(\psi_\pi(\tau_{ij},s_{ij},y)\right)^{\text{cont}(\lambda_j)}$$

Joint probability distribution of y (true label for an instance(**X**) and $\Lambda_i = (\lambda_{i1},\lambda_{i2},,\lambda_{in})$ (labels assigned by the n labeling functions) and their scores $(s_{i1},s_{i2},,s_{in})$ is .....

$$\psi_\theta(\tau_{ij},y) = \begin{cases} \exp(\theta_{jy}) & \text{if } \tau_{ij} \neq 0, \\ 1 & \text{otherwise.} \end{cases}$$

$$\psi_\pi(\tau_{ij},s_{ij},y) = \begin{cases} \textit{Beta}(s_{ij};\alpha_a,\beta_a) & \text{if } k_j = y \;\&\; \tau_{ij} \neq 0, \\ \textit{Beta}(s_{ij};\alpha_d,\beta_d) & \text{if } k_j \neq y \;\&\; \tau_{ij} \neq 0, \\ 1 & \text{otherwise} \end{cases}$$

Note $\alpha_a = q_j^c\pi_{jy}$ and $\beta_a = (1-q_j^c)\pi_{jy}$ are parameters of the agreement distribution and $\alpha_d = (1-q_j^c)\pi_{jy}$ and $\beta_d = q_j^c\pi_{jy}$ are parameters of the disagreement distribution, where π_{jy} is constrained to be strictly positive. To impose $\pi_{jy} > 0$ while also maintaining differentiability, we reparametrize π_{jy} as $\exp(\rho_{jy})$.

$$Z_\theta = \sum_y \prod_j \sum_{\tau_j \in \{k_j,0\}} \psi_\theta(\tau_j,y)\int_{s_j=0}^{1}\psi_\pi(\tau_j,s_j,y) = \sum_{y\in\mathscr{Y}}\prod_j(1+\exp(\theta_{jy}))$$

Relationship with Snorkel

CAGE Model Potential

$$\psi_\theta(\tau_{ij}, y) = \begin{cases} \exp(\theta_{jy}) & \text{if } \tau_{ij} \neq 0, \\ 1 & \text{otherwise.} \end{cases}$$

Setting $\theta_{j,+1} = -\theta_{j,-1}$ in the CAGE model .....

Relationship with Snorkel

CAGE Model Potential

$$\psi_\theta(\tau_{ij}, y) = \begin{cases} \exp(\theta_{jy}) & \text{if } \tau_{ij} \neq 0, \\ 1 & \text{otherwise.} \end{cases}$$

Snorkel Model Potential

$$\phi_j(y, \lambda_{ij}) = \begin{cases} \theta_{jy} & \text{if } \lambda_{ij} = j, \\ -\theta_{jy} & \text{if } \lambda_{ij} \neq j, \lambda_{ij} \neq 0, \\ 0 & \text{otherwise.} \end{cases}$$

Setting $\theta_{j,+1} = -\theta_{j,-1}$ in the CAGE model .....

Few simplifications in CAGE lead to the snorkel model

- Coupling of θ_{jy} parameters in $\phi_j(y, \lambda_{ij})$
- Not including continuous LFs and the associated potentials $\left(\psi_\pi(\tau_{ij}, s_{ij}, y)\right)^{\text{cont}(\lambda_j)}$
- Ignoring quality guides q_j^t (next)...

CAGE: The Training Objective

$$\max_{\theta,\pi} LL(\theta,\pi|D) + R(\theta,\pi|\{q_j^t\})$$

CAGE: The Training Objective

$$\max_{\theta,\pi} LL(\theta,\pi|D) + R(\theta,\pi|\{q_j^t\})$$

WHY $R(\theta,\pi|\{q_j^t\})$? Unsupervised likelihood training inherently unstable

$$LL(\theta,\pi|D) = \sum_{i=1}^{m} \log \sum_{y\in\mathscr{Y}} P_{\theta,\pi}(\tau_i, s_i, y)$$

$$= \sum_{i=1}^{m} \log \sum_{y\in\mathscr{Y}} \prod_{j=1}^{n} \psi_j(\tau_{ij}, y) \left(\psi_j(s_{ij}, \tau_{ij}, y)\right)^{\text{cont}(\lambda_j)} - m \log Z_\theta$$

Recall, from the previous slide,

$$P_{\theta,\pi}(y, \tau_i, s_i) = \frac{1}{Z_\theta} \prod_{j=1}^{n} \psi_\theta(\tau_{ij}, y) \left(\psi_\pi(\tau_{ij}, s_{ij}, y)\right)^{\text{cont}(\lambda_j)}$$

CAGE: The Training Objective

$$\max_{\theta,\pi} LL(\theta,\pi|D) + R(\theta,\pi|\{q_j^t\})$$

WHY $R(\theta,\pi|\{q_j^t\})$? Unsupervised likelihood training inherently unstable

WHAT is $R(\theta,\pi|\{q_j^t\})$? Options:

- Match learned joint distribution $P_{\theta,\pi}$ of y and τ_j with user-provided quality guides q_j^t
  - For continuous LFs parameterize Beta distribution to combine quality guides and learning.
  - Empowers programmer to stabilize training via easy quality guides q_j^t on a LF (e.g. accuracy ≥ 0.5)

$$LL(\theta,\pi|D) = \sum_{i=1}^{m} \log \sum_{y\in\mathscr{Y}} P_{\theta,\pi}(\tau_i, s_i, y)$$

$$= \sum_{i=1}^{m} \log \sum_{y\in\mathscr{Y}} \prod_{j=1}^{n} \psi_j(\tau_{ij}, y) \left(\psi_j(s_{ij}, \tau_{ij}, y)\right)^{\text{cont}(\lambda_j)} - m \log Z_\theta$$

Recall, from the previous slide,

$$P_{\theta,\pi}(y, \tau_i, s_i) = \frac{1}{Z_\theta} \prod_{j=1}^{n} \psi_\theta(\tau_{ij}, y) \left(\psi_\pi(\tau_{ij}, s_{ij}, y)\right)^{\text{cont}(\lambda_j)}$$

CAGE: The Training Objective

$$\max_{\theta,\pi} LL(\theta,\pi|D) + R(\theta,\pi|\{q_j^t\})$$

$$LL(\theta,\pi|D) = \sum_{i=1}^{m} \log \sum_{y\in\mathscr{Y}} P_{\theta,\pi}(\tau_i, s_i, y)$$

$$= \sum_{i=1}^{m} \log \sum_{y\in\mathscr{Y}} \prod_{j=1}^{n} \psi_j(\tau_{ij}, y) \left(\psi_j(s_{ij}, \tau_{ij}, y)\right)^{\text{cont}(\lambda_j)} - m \log Z_\theta$$

Recall, from the previous slide,

$$P_{\theta,\pi}(y, \tau_i, s_i) = \frac{1}{Z_\theta} \prod_{j=1}^{n} \psi_\theta(\tau_{ij}, y) \left(\psi_\pi(\tau_{ij}, s_{ij}, y)\right)^{\text{cont}(\lambda_j)}$$

WHY $R(\theta,\pi|\{q_j^t\})$? Unsupervised likelihood training inherently unstable

WHAT is $R(\theta,\pi|\{q_j^t\})$? Options:

- Match learned joint distribution $P_{\theta,\pi}$ of y and τ_j with user-provided quality guides q_j^t
  - For continuous LFs parameterize Beta distribution to combine quality guides and learning.
  - Empowers programmer to stabilize training via easy quality guides q_j^t on a LF (e.g. accuracy ≥ 0.5)
- Other options considered:
  - Sign penalty on raw parameters to favor agreement
  - Constraints on LF accuracy calculated on data

CAGE: The Training Objective & Constraints $R(\theta, \pi | \{q_j^t\})$

$$\max_{\theta,\pi} LL(\theta, \pi|D) + R(\theta, \pi|\{q_j^t\})$$

Matching learned joint distribution $P_{\theta,\pi}$ of y and τ_j with user-provided quality guides q_j^t :

$$LL(\theta, \pi|D) = \sum_{i=1}^{m} \log \sum_{y \in \mathscr{Y}} P_{\theta,\pi}(\tau_i, s_i, y)$$

$$= \sum_{i=1}^{m} \log \sum_{y \in \mathscr{Y}} \prod_{j=1}^{n} \psi_j(\tau_{ij}, y) \left(\psi_j(s_{ij}, \tau_{ij}, y)\right)^{\text{cont}(\lambda_j)} - m \log Z_\theta$$

$$R(\theta|\{q_j^t\}) = \sum_j q_j^t \log P_\theta(y = k_j | \tau_j = k_j)$$

$$+ (1 - q_j^t) \log(1 - P_\theta(y = k_j | \tau_j = k_j))$$

Recall, from the previous slide,

$$P_{\theta,\pi}(y, \tau_i, s_i) = \frac{1}{Z_\theta} \prod_{j=1}^{n} \psi_\theta(\tau_{ij}, y) \left(\psi_\pi(\tau_{ij}, s_{ij}, y)\right)^{\text{cont}(\lambda_j)}$$

$$P_\theta(y = k_j | \tau_j = k_j) = \frac{P_\theta(y = k_j, \tau_j = k_j)}{P_\theta(\tau_j = k_j)}$$

$$= \frac{\mathsf{M}_j(k_j) \prod_{r \neq j} (1 + \mathsf{M}_r(k_j))}{\sum_{y \in \mathscr{Y}} \mathsf{M}_j(y) \prod_{r \neq j} (1 + \mathsf{M}_r(y))}$$

Data independent, Efficient, and Accurate!

Experimental Setup

Datasets:

1. Spouse: Relation extraction dataset - label candidate pairs of entities in a sentence as expressing a ‘spouse’ relation or not
2. Spam SMS: Binary spam/no-spam classification dataset with 5574 documents: 3700 unlabeled-train and 1872 labeled-test
3. CDR: Relation extraction dataset where the task is to detect whether or not a sentence expresses a ‘chemical cures disease’ relation
4. Dedup: 32 thousand pairs of noisy citation records with fields like Title, Author, Year etc. Task - detect if record pairs are duplicates
5. Ionosphere
6. Iris

Training Setup:

1. $q_{tj}(Discrete) = 0.9$
2. $q_{cj}(Continuous) = 0.85$
3. Learning rate = 0.001
4. Epochs = 100

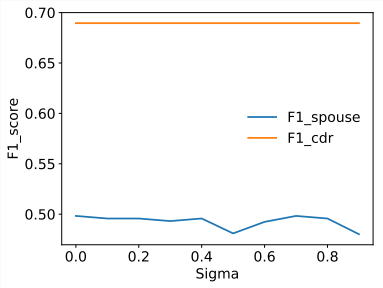
Overall Results

	Datasets					
	Spouse	CDR	SMS	Ion	Iris	Dedup
Majority	0.17	0.53	0.23	0.79	0.84	-
Snorkel	0.41	0.66	0.34	0.70	0.87	-
CAGE_{-C-G}	0.48	0.69	0.34	0.81	0.87	-
CAGE_{-C}	0.50	0.69	0.45	0.82	0.87	-
CAGE	**0.58**	**0.69**	**0.54**	**0.97**	**0.87**	**0.79**

Overall Results (F1) with predictions from various generative models contrasted with the Majority baseline.

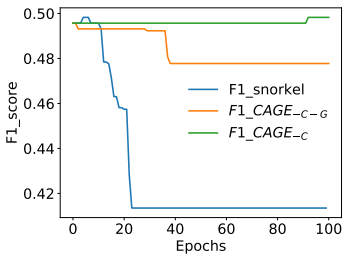
Quality Guides

	Spouse	CDR	Sms	Ion
CAGE$_{-C-G+-P}$	0.48	0.69	0.34	0.81
CAGE$_{-C-G}$	0.48	0.69	0.34	0.81
CAGE$_{-C,dataG}$	0.48	0.69	0.34	0.81
CAGE$_{-C}$	**0.50**	**0.69**	**0.45**	**0.82**

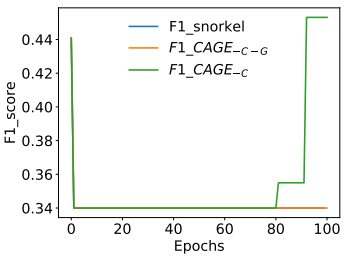


F1 with increasing distortion in the guess of the LF quality guide, q_j^t.

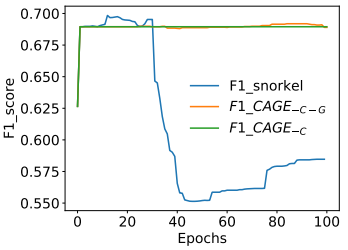
Quality Guides & Performance wrt epochs



(a) Spouse



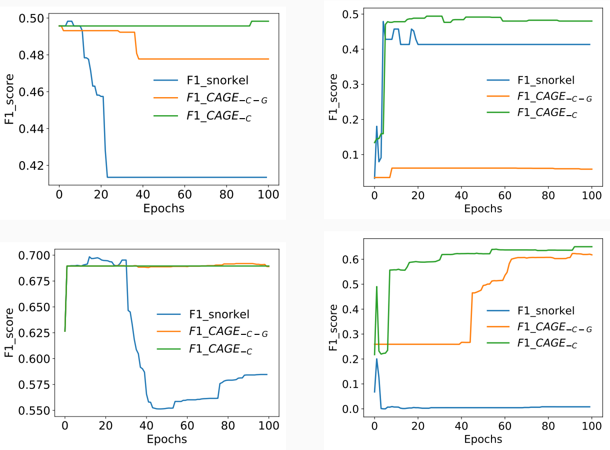
(b) SMS



(c) CDR

F1 with increasing number of training epochs compared across snorkel, CAGE_{-C-G} and CAGE_{-C}, for three datasets. For each dataset, in the absence of guides, we observe unpredictable variation in test F1 as training progresses.

Sensitivity to initialization



(a) Agreeing initialization (b) Random Initialization.

F1 with increasing number of training epochs compared across snorkel, $CAGE_{-C-G}$ and $CAGE_{-C}$, for two datasets: Spouse (top-row) and CDR(bottom-row). $CAGE_{-C}$ is able to recover from any initialization whereas methods without guides fare even worse with random initialization.

Problem Setting 1: Summarily

- **Continuous** LFs improve recall.
- Snorkel's unsupervised likelihood training is inherently unstable.
- CAGE allows **quality guides** to stabilize learning.
- Elegant method of incorporating guides into likelihood training.

Recall Objective 1: Labeler reliability

Labeler A	Labeler B	Labeler C
Clock 0.6 Gauge 0.3	CafePress ABARTH Wall Clock CafePress Maltese Portrait Wall Clock	Indoors 68% Meat Eater 67% Room 66% Object 65% Vertebrate 65%
Banana 0.5 Vegetable 0.4 Fruit 0.3	Plantains (Musa balbisiana)	Flower 77% Human 67% Food 67% Group of People 62% Indoors 60% Activity 56% ClubSport 55%

Labelers might have complementary expertise. Can the labelers mutually benefit from their individual expertise?

Recall Objective 2: Collective Prediction

	Labeler A	Labeler B	Labeler C
	Clock 0.6 Gauge 0.3	CafePress ABARTH Wall Clock CafePress Maltese Portrait Wall Clock	Indoors 68% Meat Eater 67% Room 66% Object 65% Vertebrate 65%
	Banana 0.5 Vegetable 0.4 Fruit 0.3	Plantains (Musa balbisiana)	Flower 77% Human 67% Food 67% Group of People 62% Indoors 60% Activity 56% ClubSport 55%

How do we aggregate the label predictions from individual labelers, to generate consolidated high confidence label predictions for each instance?

Recall Objective 2: Collective Prediction

	Labeler A	Labeler B	Labeler C
	Clock 0.6 Gauge 0.3	CafePress ABARTH Wall Clock CafePress Maltese Portrait Wall Clock	Indoors 68% Meat Eater 67% Room 66% Object 65% Vertebrate 65%
	Banana 0.5 Vegetable 0.4 Fruit 0.3	Plantains (Musa balbisiana)	Flower 77% Human 67% Food 67% Group of People 62% Indoors 60% Activity 56% ClubSport 55%

How do we aggregate the label predictions from individual labelers, to generate consolidated high confidence label predictions for each instance?

Can we also account for inter-label correlation and labeler reliability?

Objective 3: Inter-label correlation

	Labeler A	Labeler B	Labeler C
	Clock 0.6 Gauge 0.3	CafePress ABARTH Wall Clock CafePress Maltese Portrait Wall Clock	Indoors 68% Meat Eater 67% Room 66% Object 65% Vertebrate 65%
	Banana 0.5 Vegetable 0.4 Fruit 0.3	Plantains (Musa balbisiana)	Flower 77% Human 67% Food 67% Group of People 62% Indoors 60% Activity 56% ClubSport 55%

Objective 3: Inter-label correlation

Labeler A	Labeler B	Labeler C
Clock 0.6 Gauge 0.3	CafePress ABARTH Wall Clock CafePress Maltese Portrait Wall Clock	Indoors 68% Meat Eater 67% Room 66% Object 65% Vertebrate 65%
Banana 0.5 Vegetable 0.4 Fruit 0.3	Plantains (Musa balbisiana)	Flower 77% Human 67% Food 67% Group of People 62% Indoors 60% Activity 56% ClubSport 55%

Certain labels, especially in a large label space, might be mutually (semantically) correlated. This correlation may be exploited to minimize labeling cost

Objective 4: Active learning sampling

Labeler A	Labeler B	Labeler C
Clock 0.6 Gauge 0.3	CafePress ABARTH Wall Clock CafePress Maltese Portrait Wall Clock	Indoors 68% Meat Eater 67% Room 66% Object 65% Vertebrate 65%
Banana 0.5 Vegetable 0.4 Fruit 0.3	Plantains (Musa balbisiana)	Flower 77% Human 67% Food 67% Group of People 62% Indoors 60% Activity 56% ClubSport 55%

Objective 4: Active learning sampling

Labeler A	Labeler B	Labeler C
Clock 0.6 Gauge 0.3	CafePress ABARTH Wall Clock CafePress Maltese Portrait Wall Clock	Indoors 68% Meat Eater 67% Room 66% Object 65% Vertebrate 65%
Banana 0.5 Vegetable 0.4 Fruit 0.3	Plantains (Musa balbisiana)	Flower 77% Human 67% Food 67% Group of People 62% Indoors 60% Activity 56% ClubSport 55%

What should be the ideal active learning sampling strategy for training some 'Machine' Labelers that might ultimately result in a high quality labeled dataset while minimizing the overall labeling budget?

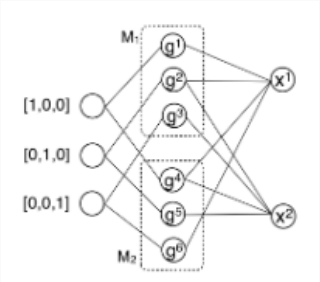
The problem of multi-labeler multi-label classification with little labeled data [AAAI 2018]

1. **Inter-label correlation:** Certain labels, especially in a large label space, might be mutually (semantically) correlated. This correlation may be exploited to minimize labeling cost.
2. **Labeler reliability:** Labelers might have complementary expertise. Can the labelers mutually benefit from their individual expertise?
3. **Collective Prediction:** How do we aggregate the label predictions from individual labelers, while accounting for inter-label correlation and labeler reliability to generate consolidated high confidence label predictions for each instance?
4. **Active learning sampling:** What should be the ideal active learning sampling strategy that might ultimately result in a high quality labeled dataset while minimizing the overall labeling budget?

Ersthwhile approaches present frameworks for addressing one of the above questions.
We address all the above.
Our paper on 'Program Induction' also at AAAI '18 is related to learning Labeling Functions

Consensus Model - Multiple labels

Symbol	Meaning
v	Number of groups ($= m \times l$), with index c
A	$n \times v$ matrix such that $a_{i,c}$ is the prediction of label (c mod l) on instance $\mathbf{x}_i$ by model $\lfloor c/l \rfloor$
τ	$v \times l$ matrix of probability distributions on label nodes
U	$n \times l$ matrix such that $\mathbf{u_{i,j}}$ is the probability that label j is relevant to $\mathbf{x}_i$
Q	$v \times l$ matrix such that $\mathbf{q}_{c,j}$ is the probability of seeing label j given the label corresponding to group node g_c
r_c	Labeler reliability



$$\min_{U,Q} \sum_{i=1}^{n} \sum_{c=1}^{v} r_c^{t'-1} \times a_{ic} \|\mathbf{u_i^{t'}} - \mathbf{q}_c^{t'}\|^2 + \alpha \sum_{c=1}^{v} \|\mathbf{q}_c^{t'} - \mathbf{b}_c\|^2 \quad (1)$$

s.t.

$$u_{ij}^{t'} \geq 0, \sum_{j=1}^{l} u_{ij}^{t'} = 1, i = 1, ..., n$$

$$q_{cj}^{t'} \geq 0, \sum_{j=1}^{l} q_{cj}^{t'} = 1, c = 1, ..., v$$

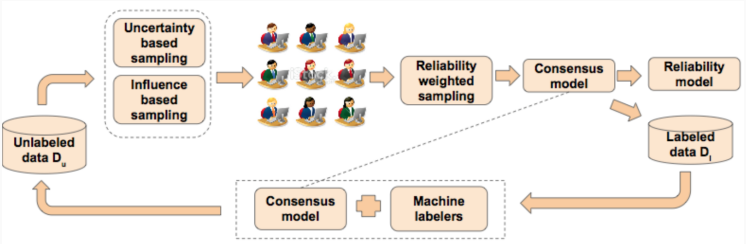
Model Comparison: Illustrative example

Image	Majority	Consensus model [AAAI '18]	Consensus with Robust F1 measures [AAAI'17]
	• buildings • plants • reflection • water	• buildings • **nighttime** • reflection • **sky** • water	• buildings • **castle** • nighttime • reflection • sky • water

Model comparison: Illustrative example

Thank You

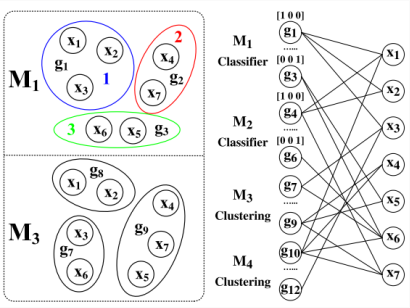
Our Approach



- System bootstraps by asking human labelers to label a sufficiently large initial set of unlabeled instances;
- Iteratively "grows" its training data by sampling the most influential unlabeled instance for labeling;
- In every iteration, the system tries to (i) improve the output from machine labelers by training them on high-confidence labeled data (ii) label the unlabeled data by conservatively querying human labelers (iii) infer reliability of labelers for each label.
- **On-going work: Making the Labeling functions Continuous using embeddings**

Consensus Model - Single label case

Symbol	Meaning
v	Number of groups $\mathbf{g}_c$'s ($= m \times l$), with index c
$A_{n\times v}$	$[a_{ic} = 1$ if x_i is assigned to group $\mathbf{g}_c]$
$B_{v\times l}$	$[b_{cj} = 1$ if label associated with $\mathbf{g}_c$ is $j]$
$U_{n\times l}$	$[u_{ij} = P(y = j\|\mathbf{x}_i)]$ - distribution over classes for a given example $\mathbf{x}_i$
$Q_{v\times l}$	$[q_{cj} = P(y = j\|\mathbf{g}_c)]$ - distribution over classes for a given group $\mathbf{g}_c]$
k_c	$\sum_{j=1}^{l} b_{cj}$

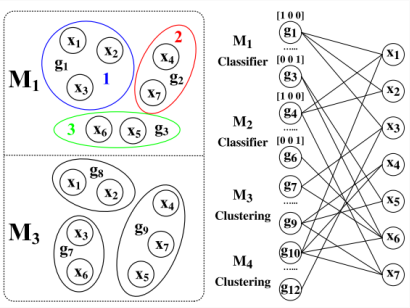


$$\min_{Q,U}\Big(\sum_{i=1}^{n}\sum_{c=1}^{v} a_{ic}\|\mathbf{u}_i - \mathbf{q}_c\|^2 + \alpha \sum_{c=1}^{v} k_c\|\mathbf{q}_c - \mathbf{b}_c\|^2\Big) \tag{2}$$

1. First term $\Rightarrow$ if an **instance** x_i **is linked to group** $\mathbf{g}_c$ **(**$a_{ic} = 1$**)**, their class probability distributions are close.
2. Second term $\Rightarrow$ **distribution on group nodes** after consensus is close to initial distribution.

Consensus Model - Single label case

Symbol	Meaning
v	Number of groups $\mathbf{g}_c$'s ($= m \times l$), with index c
$A_{n\times v}$	$[a_{ic} = 1$ if x_i is assigned to group $\mathbf{g}_c]$
$B_{v\times l}$	$[b_{cj} = 1$ if label associated with $\mathbf{g}_c$ is $j]$
$U_{n\times l}$	$[u_{ij} = P(y = j\|\mathbf{x}_i)]$ - distribution over classes for a given example $\mathbf{x}_i$
$Q_{v\times l}$	$[q_{cj} = P(y = j\|\mathbf{g}_c)]$ - distribution over classes for a given group $\mathbf{g}_c]$
k_c	$\sum_{j=1}^{l} b_{cj}$



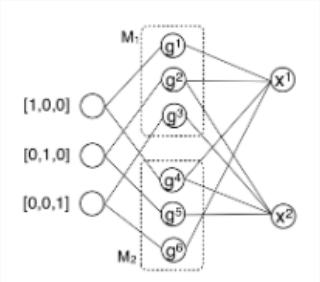
$$\min_{Q,U}\Big(\sum_{i=1}^{n}\sum_{c=1}^{v} a_{ic}\|\mathbf{u}_i - \mathbf{q}_c\|^2 + \alpha \sum_{c=1}^{v} k_c\|\mathbf{q}_c - \mathbf{b}_c\|^2\Big) \tag{3}$$

s.t.

$$\mathbf{u}_i \geq 0, |\mathbf{u}_i| = 1, i = 1, ..., n$$
$$\mathbf{q}_c \geq 0, |\mathbf{q}_c| = 1, c = 1, ..., v$$

Consensus Model - Multiple labels

Symbol	Meaning
v	Number of groups ($= m \times l$), with index c
A	$n \times v$ matrix such that $a_{i,c}$ is the prediction of label (c mod l) on instance $\mathbf{x}_i$ by model $\lfloor c/l \rfloor$
B	$v \times l$ matrix of probability distributions on label nodes
U	$n \times l$ matrix such that $\mathbf{u_{i,j}}$ is the probability that label j is relevant to $\mathbf{x}_i$
Q	$v \times l$ matrix such that $\mathbf{q}_{c,j}$ is the probability of seeing label j given the label corresponding to group node g_c
r_c	Labeler reliability



$$\min_{U,Q} \sum_{i=1}^{n} \sum_{c=1}^{v} r_c^{t'-1} \times a_{ic} \|\mathbf{u_i^{t'}} - \mathbf{q}_c^{t'}\|^2 + \alpha \sum_{c=1}^{v} \|\mathbf{q}_c^{t'} - \mathbf{b}_c\|^2 \quad (4)$$

s.t.

$$u_{ij}^{t'} \geq 0, \sum_{j=1}^{l} u_{ij}^{t'} = 1, i = 1, ..., n$$

$$q_{cj}^{t'} \geq 0, \sum_{j=1}^{l} q_{cj}^{t'} = 1, c = 1, ..., v$$

Labeler Reliability

- Labeler reliability $r_c^{t'}$ of k^{th} labeler on j^{th} label (that is, $c = (k,j)$) updated as

$$r_c^{t'} \leftarrow r_c^{t'-1} + \gamma(\kappa_c^{t'} - r_c^{t'-1}) \tag{5}$$

- κ_c: Agreement measure between the k^{th} labeler and the consensus model on label j across all labeled instances
- $\gamma < 1$ is a constant.
- Our choice for κ_c: Cohen Kappa [Cohen & Jacob, '68], a standard metric for inter-rater agreement

Active Learning: Uncertainty-based Sampling (Unc-R)

- *Sample that instance from the unlabeled pool which has minimum overall agreement:* $\mathbf{x}^* = \min_{\mathbf{x} \in D_U} \kappa_{\mathbf{x}}$.
- Overall inter-labeler agreement $\kappa_{\mathbf{x}}$ for instance $\mathbf{x}$:

$$\kappa_{\mathbf{x}} = \frac{1}{m} \sum_{k=1}^{m} \kappa(\mathbf{b}_L^k(\mathbf{x}), \mathbf{u}_L(\mathbf{x})) \tag{6}$$

where,
$\mathbf{u}_L$ is the consensus model obtained using m machine labelers, trained with the labeled dataset D_L;
κ measures agreement between label sets;
$\mathbf{b}_L^k(\mathbf{x}_i)$ is the output of k^{th} prediction function.

Active Learning: Influence-based Sampling (Inf-R)

- The expected agreement on the unlabeled dataset D_U, for a consensus model obtained from D_L:

$$\sigma_L = \frac{1}{|D_U|} \sum_{\mathbf{x} \in D_U} \sum_{\mathbf{z} \in \mathscr{Z}} \kappa(\mathbf{u}_L(\mathbf{x}), \mathbf{z}) P(\mathbf{z}|\mathbf{x}) \tag{7}$$

  After sampling an instance $\mathbf{x}^* \in D_U$, let $D_{L'} = D_L \cup \mathbf{x}^*$ be the new labeled dataset and the expected agreement on D_U based on $D_{L'}$:

$$\sigma_{L'} = \frac{1}{|D_U|} \sum_{\mathbf{x} \in D_U} \sum_{\mathbf{z} \in \mathscr{Z}} \kappa(\mathbf{u}_{L'}(\mathbf{x}), \mathbf{z}) P(\mathbf{z}|\mathbf{x}) \tag{8}$$

- Sample $\mathbf{x}^*$ that offers maximum improvement in the expected agreement.

$$\mathbf{x}^* = \arg\max_{\mathbf{x} \in D_U} \sum_{\mathbf{z} \in \mathscr{Z}} (\kappa(\mathbf{u}_L(\mathbf{x}), \mathbf{z}) - \kappa(\mathbf{u}_{L'}(\mathbf{x}), \mathbf{z})) P(\mathbf{z}|\mathbf{x}) \tag{9}$$

Active Learning: Influence-based Sampling (Inf-R)

- The expected agreement on the unlabeled dataset D_U, for a consensus model obtained from D_L:

$$\sigma_L = \frac{1}{|D_U|} \sum_{\mathbf{x} \in D_U} \sum_{\mathbf{z} \in \mathscr{Z}} \kappa(\mathbf{u}_L(\mathbf{x}), \mathbf{z}) P(\mathbf{z}|\mathbf{x}) \tag{7}$$

  After sampling an instance $\mathbf{x}^* \in D_U$, let $D_{L'} = D_L \cup \mathbf{x}^*$ be the new labeled dataset and the expected agreement on D_U based on $D_{L'}$:

$$\sigma_{L'} = \frac{1}{|D_U|} \sum_{\mathbf{x} \in D_U} \sum_{\mathbf{z} \in \mathscr{Z}} \kappa(\mathbf{u}_{L'}(\mathbf{x}), \mathbf{z}) P(\mathbf{z}|\mathbf{x}) \tag{8}$$

- Sample $\mathbf{x}^*$ that offers maximum improvement in the expected agreement.

$$\mathbf{x}^* = \arg\max_{\mathbf{x} \in D_U} \sum_{\mathbf{z} \in \mathscr{Z}} (\kappa(\mathbf{u}_L(\mathbf{x}), \mathbf{z}) - \kappa(\mathbf{u}_{L'}(\mathbf{x}), \mathbf{z})) P(\mathbf{z}|\mathbf{x}) \tag{9}$$

- Several mathematical tricks involved in estimating the above expression.

Improvement in Expected Agreement: (Optional)

Iterative Updates in closed form:

$$Q_A^* = (I - D_\lambda S_A)^{-1} D_{1-\lambda} B \tag{10}$$

$$U_{Q_A^\star} = D_n^{-1} A Q_A^* \tag{11}$$

where,
$D_v = diag\{\sum_{i=1}^{n} a_{ic}\}_{v\times v}$, $D_n = diag\{\sum_{c=1}^{v} a_{ic}\}_{n\times n}$, $K_v = diag\{\sum_{j=1}^{l} b_{cj}\}_{v\times v}$, $D_\lambda = (D_v + \alpha K_v)^{-1} D_v$, $D_{1-\lambda} = (D_v + \alpha K_v)^{-1}(\alpha K_v)$, $S_A = D_v^{-1} A' D_n^{-1} A$.
Consensus prediction for a new instance:

$$\mathbf{u}_L(\mathbf{x}) = \frac{A_L(x) Q_A^\star}{1^T A_L(x)} \text{ and } \mathbf{u}_{L'}(\mathbf{x}) = \frac{A_L(x) Q_{\tilde{A}_x}^\star}{1^T A_L(x)} \tag{12}$$

where, $A_L(x) = [f_L^1(\mathbf{x}) f_L^2(\mathbf{x}) \cdots \mathbf{b}_L^k(\mathbf{x})]$ is a $1 \times v$ matrix represents output of all prediction functions $\mathbf{b}_L$ and $\tilde{A}_x = \begin{bmatrix} A \\ A_L(x) \end{bmatrix}$. $Q_{\tilde{A}_x}^\star$ can be efficiently computed.

Estimate Conditional Probability $P(\mathbf{z}|\mathbf{x})$ (Optional)

Estimating $P(\mathbf{z}|\mathbf{x})$ for all possible $\mathbf{z} \in \mathscr{Z}$ is intractable due to the exponential search space ($\mathscr{Z} = \{0,1\}^l$).

Relaxations

- Assume the labels to be independent, $P(\mathbf{z}|\mathbf{x}) = \prod_j P(z^j|\mathbf{x})$;
- Relax the search space by considering a subset that represents the most possible label combinations for $\mathbf{x}$
  - $P(z^j|\mathbf{x})$ as $h(\langle \mathbf{w}^j, \mathbf{x}'^j \rangle + w_0^j)$, where, $\mathbf{x}'^j \in \mathbb{R}^m$ is a feature vector corresponding to $\mathbf{x}$, comprising m labelers' output for the j-th label, w_0^j and $\mathbf{w}^j$ are model parameters trained on the labeled data $\mathscr{L}$.
  - Next, we set $z^j = 0, \forall j$ s.t. $P(z^j|\mathbf{x}) < 0.5 - \delta$ and $z^j = 1, \forall j$ s.t. $P(z^j|\mathbf{x}) \geq 0.5 + \delta$. For all the remaining labels, we consider both classes $\{0,1\}$ and it is this subset of label combinations on which we compute the expected agreement.

Evaluation Setup

Table 2: Datasets

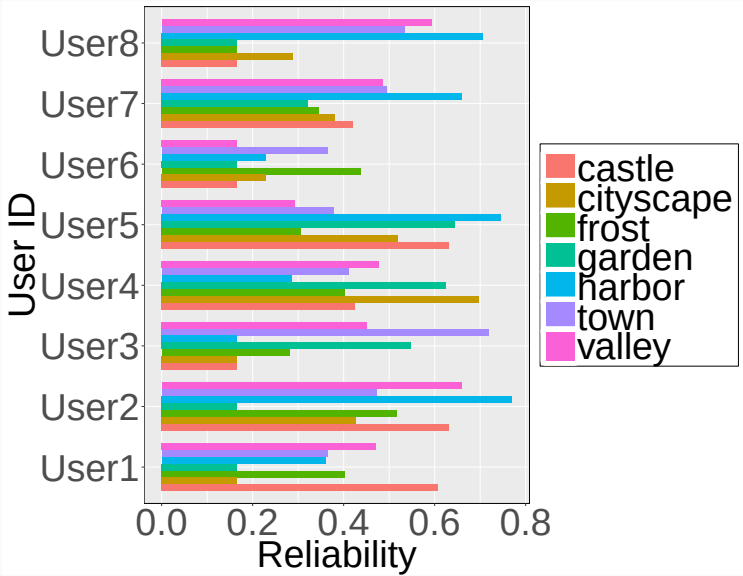
Dataset	#Instances	#Features	#Labels
Scene†	500	128	33
Flags†	194	19	7
Medical*	978	1,449	45
Enron*	1,672	1,001	53
Slashdot*	3,782	1,101	22
Corel5k*	5,000	499	374
Mediamill*	43,907	120	101

*Simulated labelers [Shao et. al. '15] †Human labelers

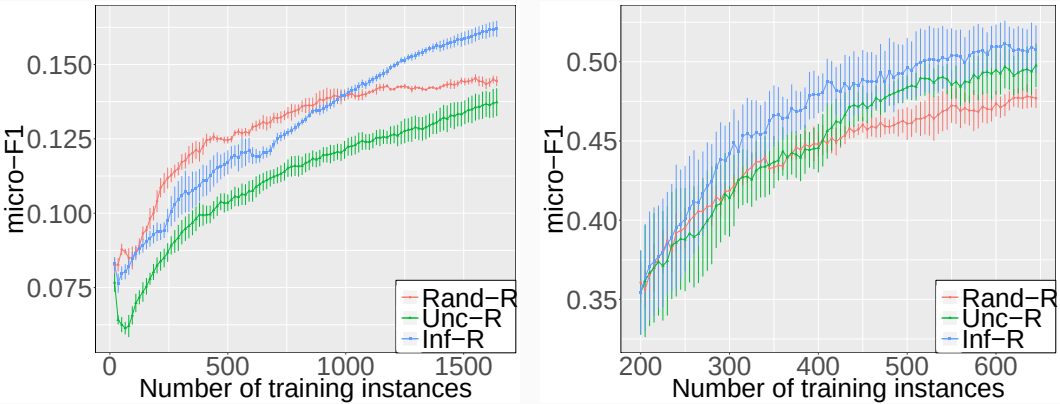
- randomly sample 30% instances as test data; labeled carefully by set of 3 experts
- remaining labeled data used for consensus based (active) learning on 8 (crowd) labelers

Micro-averaged F_1 as evaluation measure

Estimating Labeler Reliability



Effect of Active Learning



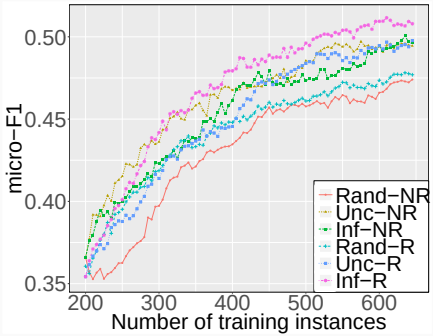
(a) Corel dataset

(b) Medical dataset

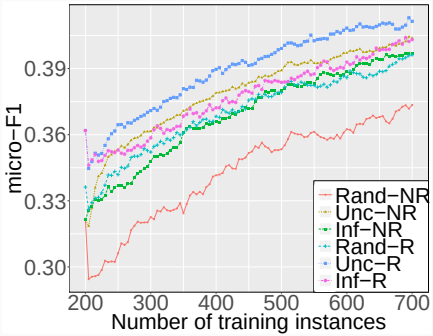
Figure 2: Average Micro-F_1 (with standard error) for PL and AL

Effect of User Reliability

Reliability weighted model incorporating labeler reliability through Inf-R (Infulence + Reliability) or Unc-R (Uncertainty + Reliability) indeed leads to a better consensus and more accurate labeling.



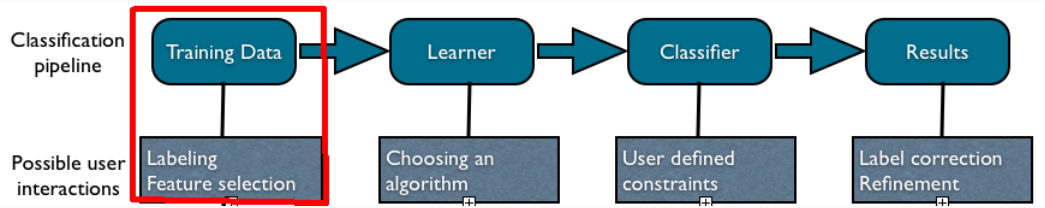
(a) Medical dataset



(b) Enron dataset

Figure 3: Effect of labeler reliability

Human Interaction at the level of (Training) Data



1. Consensus between Human/Rule-based Labelers and Machine Labelers (possibly trained on a small labeled data) in multilabel settings
2. **Addressing hierarchical labeling settings** and **Data Subset selection for Labeling**

An Interactive Multi-Label Consensus Labeling Model for Multiple Labeler Judgments [AAAI 2018]

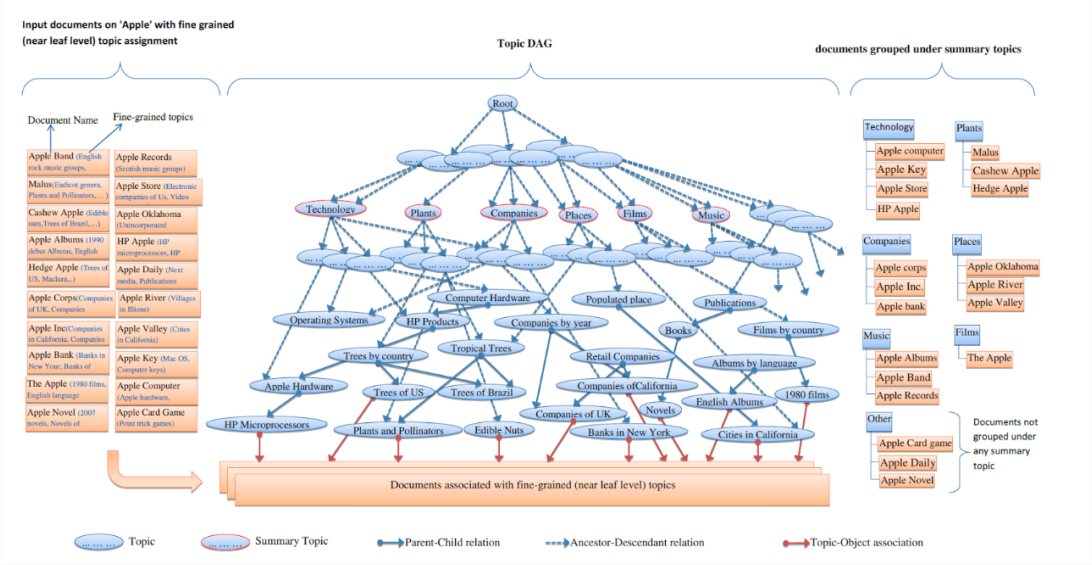
Learning From Less Data: Diversified Subset Selection and Active Learning in Image Classification Tasks [WACV 2019]

Beyond clustering: Sub-DAG Discovery for Categorising Documents [CIKM 2016]

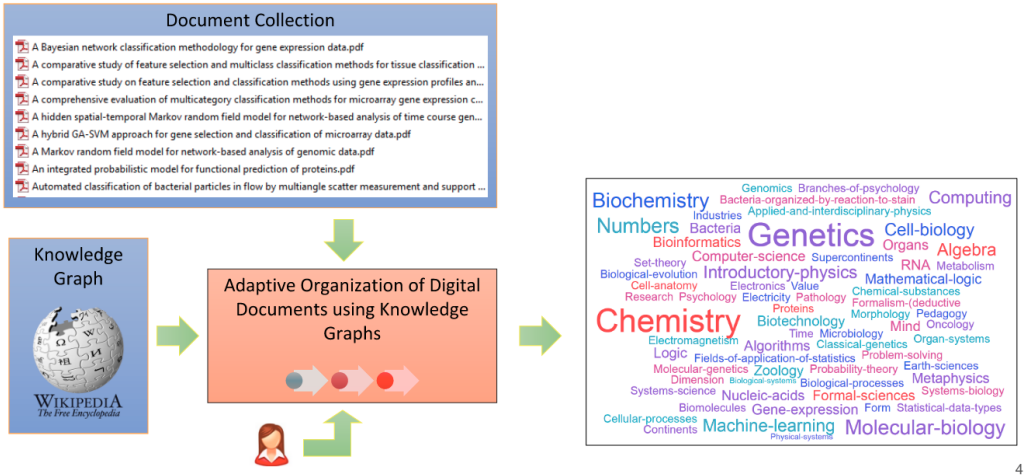
Summarizing Multi-Document Topic Hierarchies using Submodular Mixtures [ACL 2015]

Personalized classifiers: evolving a classifier from a large reference knowledge graphs [SIGIR Workshop 2014]

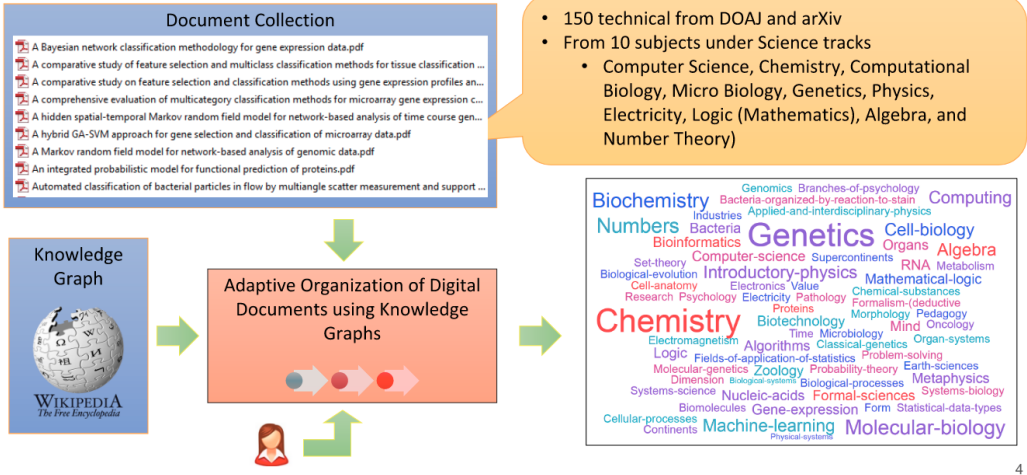
Extracting relevant subset from a Knowledge Graph: Motivating Example



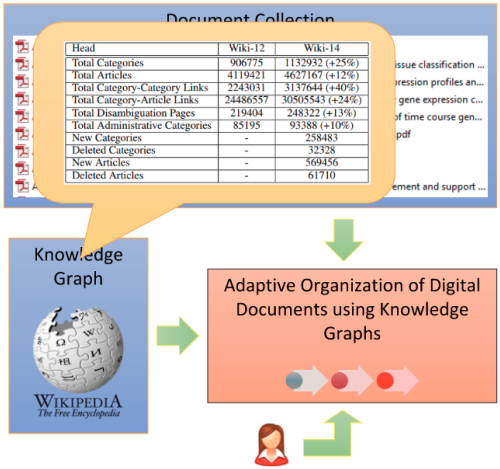
Going a step further: Dealing with larger document collections

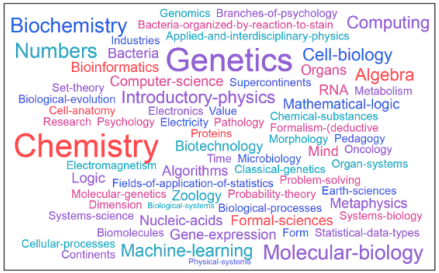


Going a step further: Dealing with larger document collections

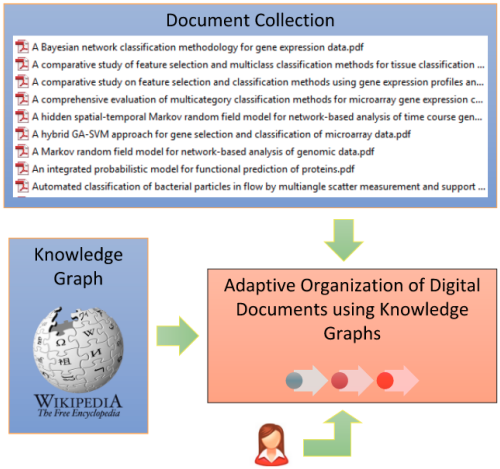


Going a step further: Dealing with larger document collections

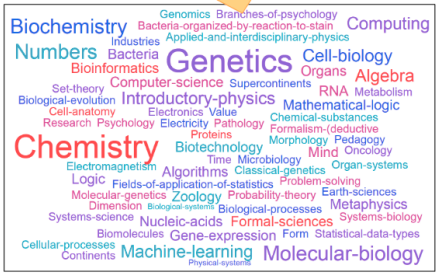




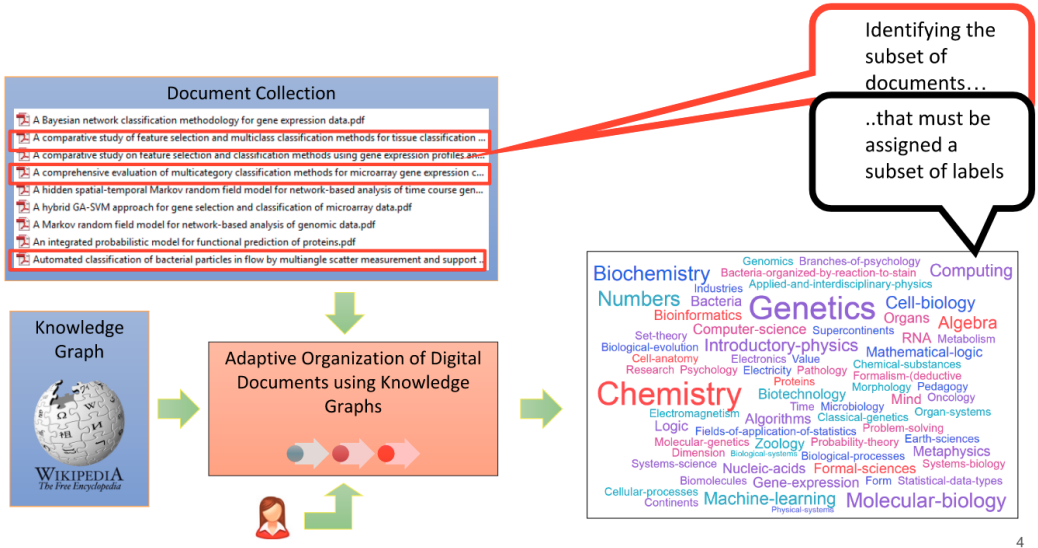
Going a step further: Dealing with larger document collections



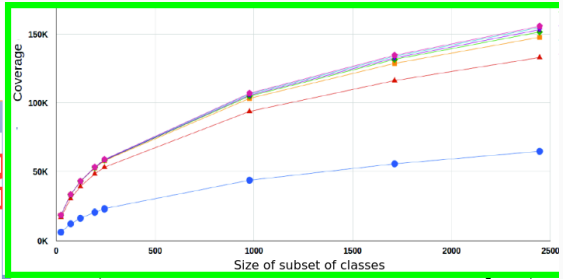
- Running through pipeline of our method
- Word Cloud of 100 categories
  - Font Size = # docs under it
- Manual Comparison
  - Out of top 20 categories 15 are in agreement with true categories

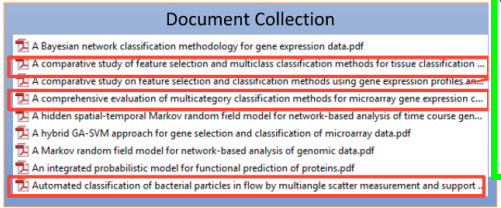


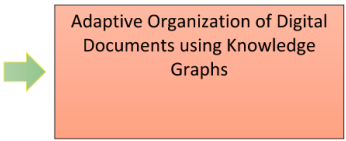
Going a step further: Dealing with larger document collections

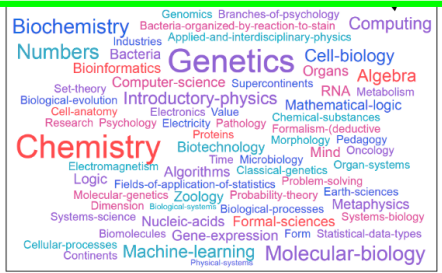


Key Observation: Diminishing Returns with increasing subset size

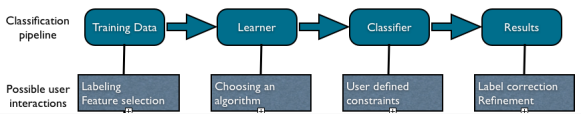








Human-Machine Interaction and Consensus in ML: Summary

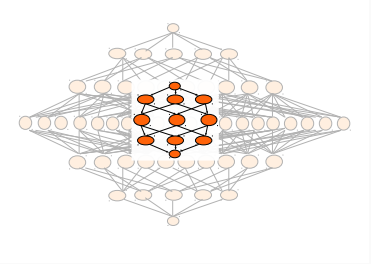


- **Training Data:**
  - **Consensus for (actively) curating training data [AAAI '18a, ongoing]**, Subset Selection on labels [ACL '15], [CIKM '16] and instances with applications in active learning [WACV '19a], [**NCETIS, IITB**]
- **Learner/Classifier:**
  - Learning Interpretable Logical Features [ML '09, ICML '11, AAAI '12, EMNLP '12]
  - **Incorporating Domain Patterns in Sequence-to-sequence Models for image, text and speech recognition [ICDAR '17, InterSpeech '18]**
  - Incorporating background (symbolic) knowledge in Model Learning [AAAI '16, PAKDD '16, **AAAI '18b**]
  - Incorporating Ontologies [CIKM '16, ISWC '13, ISWC '16, NAACL '18]
- **Results**: **Non-decomposable and Interactively Defined Performance Measures [NAACL '15, CIKM '16, IJCAI '16, AAAI '17, PAKDD '18]**: https://www.cse.iitb.ac.in/~av/fuss/s-2017/ganesh-video.webm

The generic problem: Subset selection over Directed Acyclic Graph (DAG)

Supervised subset selection over lattices: Hierarchical Kernel Learning (HKL) framework: [JMLR '15, AAAI '12, ILP '12, EMNLP '12, ICML '11]

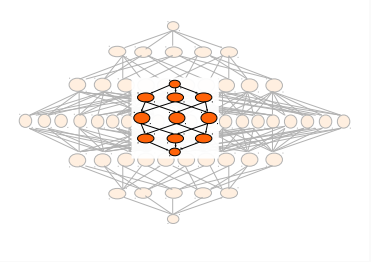
1. convex relaxation via l_1/l_ρ block-norm regularizer ($\rho \in (1,2]$)
2. efficient mirror descent-based active set algorithms on the dual (convex, Lipschitz conts., sub-differential objective over a simplex) with an efficiently computable sub-differential
3. generalization to multi-class, multi-label and multi-task
4. generalization to disjunctions/conjunctions of propositions, first order logic and sequentially structured output spaces



The generic problem: Subset selection over Directed Acyclic Graph (DAG)

Supervised subset selection over lattices: Hierarchical Kernel Learning (HKL) framework: [JMLR '15, AAAI '12, ILP '12, EMNLP '12, ICML '11]

1. convex relaxation via l_1/l_ρ block-norm regularizer ($\rho \in (1,2]$)
2. efficient mirror descent-based active set algorithms on the dual (convex, Lipschitz conts., sub-differential objective over a simplex) with an efficiently computable sub-differential
3. generalization to multi-class, multi-label and multi-task
4. generalization to disjunctions/conjunctions of propositions, first order logic and sequentially structured output spaces

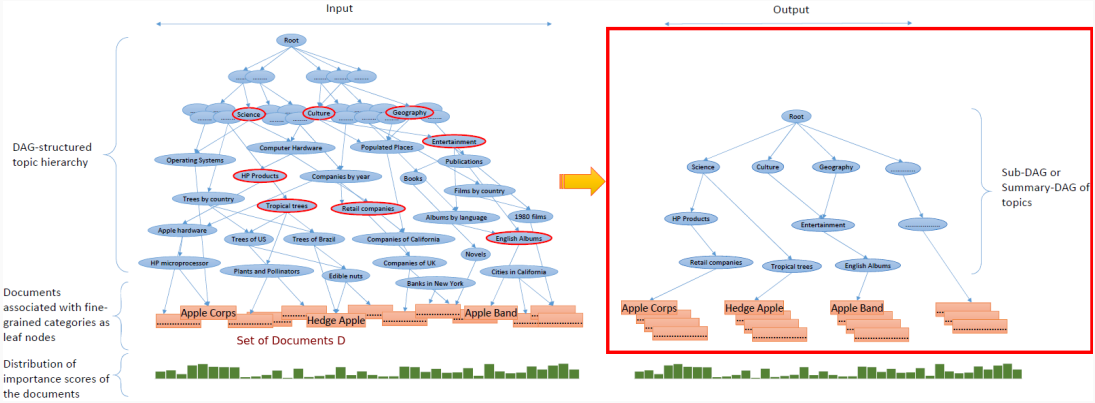


Summarizing by maximizing submodular/supermodular mixtures, learnt through max margin formulations. [ACL '15, CIKM '16, PAKDD '16, AAAI '18b, WACV '18a, WACV 18b]

Summarizing DAG-structured hierarchies

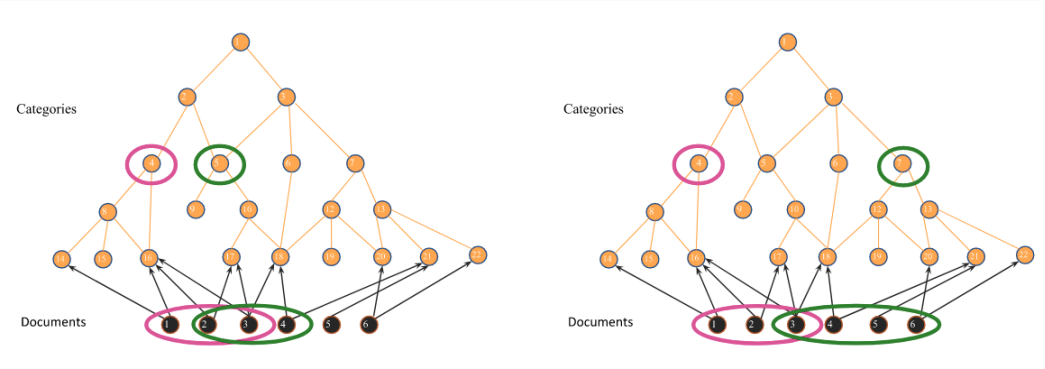
- $G(V, E)$: DAG structured hierarchy of V topics (or ground elements). E encodes parent-child (isa) relationship

Output: A representative subset S^* of size $K \in \mathbb{Z}_+$, that 'best' describes **D**

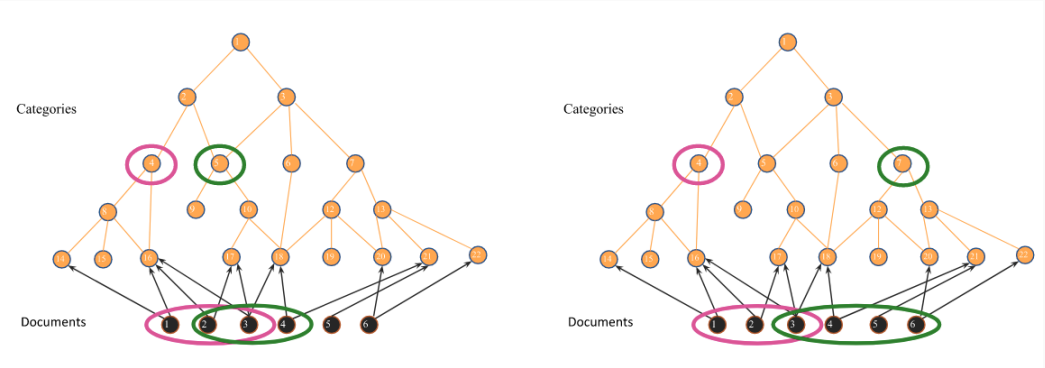


- **D**: Set of documents associated (hard/soft) with one or more of these topics.

Submodularity: Example of Coverage Functions (to maximize)



Submodularity: Example of Coverage Functions (to maximize)

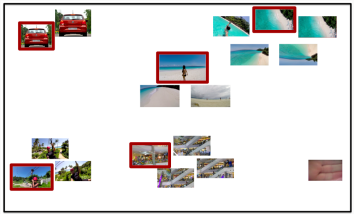


A set function $f^{\text{sub}}(.)$ is said to be submodular if for any element v and sets $A \subseteq B \subseteq V \setminus \{v\}$, $\mathsf{f}(A \cup \{v\}) - \mathsf{f}(A) \geq \mathsf{f}(B \cup \{v\}) - \mathsf{f}(B)$. Eg: Coverage, which we would like to maximize

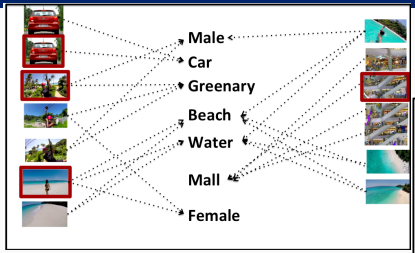
A function $f^{\text{sup}}(.)$ is said to be supermodular if the inequality above is reversed

Several submodular functions are also monotone and denoted $f^{\text{msub}}(.)$

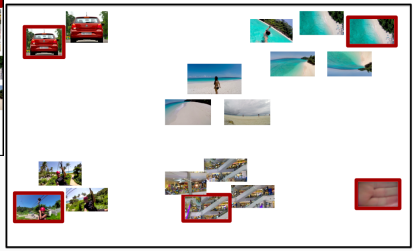
More submodular functions (to maximize): Data subsetting



Facility Location/Representation Function: Characterized by similarity s_{ij} between elements i and j



Weighted Feature Coverage Function: Characterized by set of concepts $\mathscr{U}$ with w_a being weight of concept a. Each element $i \in X$ contains a subset $U_i \in \mathscr{U}$ of concepts (for example, an image covers a table, chair and person).



Diversity: Characterized by distance d_{ij} between elements i and j.

Name	$f(X)$	Nature	Pre-compute stats	Std complexity	Reduced complexity.
Facility Location	$\sum_{i\in V} \max_{k\in X} s_{ik}$	f^{sub}	$[\max_{k\in X} s_{ik}, i \in V]$	$O(n^2)$	$O(n)$
Saturated Coverage	$\sum_{i\in V} \min\{\sum_{j\in X} s_{ij}, \alpha_i\}$	f^{sub}	$[\sum_{j\in X} s_{ij}, i \in V]$	$O(n^2)$	$O(n)$
Graph Cut	$\lambda \sum_{i\in V}\sum_{j\in X} s_{ij} - \sum_{i,j\in X} s_{ij}$	f^{sub}	$[\sum_{j\in X} s_{ij}, i \in V]$	$O(n^2)$	$O(n)$
DPP	$\log\det(S_X))$	f^{sub}	$\text{SVD}(S_X)$	$O(\lvert X\rvert^3)$	$O(\lvert X\rvert^2)$
Feature Based	$\sum_{i\in\mathscr{U}} \psi(w_i(X))$	f^{sub}	$[w_i(X), i \in \mathscr{U}]$	$O(n\lvert\mathscr{U}\rvert)$	$O(\lvert\mathscr{U}\rvert)$
Set Cover	$w(\cup_{i\in X} U_i)$	f^{sub}	$\cup_{i\in X} U_i$	$O(n\lvert\mathscr{U}\rvert)$	$O(\lvert\mathscr{U}\rvert)$
Prob. Set Cover	$\sum_{i\in\mathscr{U}} w_i[1-\prod_{k\in X}(1-p_{ik})]$	f^{sub}	$[\prod_{k\in X}(1-p_{ik}), i \in \mathscr{U}]$	$O(n\lvert\mathscr{U}\rvert)$	$O(\lvert\mathscr{U}\rvert)$
Dispersion Min	$\min_{k,l\in X, k\neq l} d_{kl}$	f^{d}	$\min_{k,l\in X, k\neq l} d_{kl}$	$O(\lvert X\rvert^2)$	$O(\lvert X\rvert)$
Dispersion Sum	$\sum_{k,l\in X} d_{kl}$	f^{sup}	$[\sum_{k\in X} d_{kl}, l \in X]$	$O(\lvert X\rvert^2)$	$O(\lvert X\rvert)$
Dispersion Min-Sum	$\sum_{k\in X} \min_{l\in X} d_{kl}$	f^{sub}	$[\min_{k\in X} d_{kl}, l \in X]$	$O(\lvert X\rvert^2)$	$O(\lvert X\rvert)$

- s_{ij} is the similarity between elements i and j
- $\mathscr{U}$ is a set of concepts with w_a being weight of concept a. Each element $i \in X$ contains a subset $U_i \in \mathscr{U}$ of concepts. ψ is a concave function.
- d_{ij} is the distance between elements i and j.

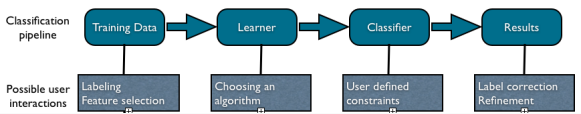
Name	$f(X)$	Nature	Pre-compute stats	Std complexity	Reduced complexity.
Facility Location	$\sum_{i\in V}\max_{k\in X} s_{ik}$	f^{sub}	$[\max_{k\in X} s_{ik}, i\in V]$	$O(n^2)$	$O(n)$
Saturated Coverage	$\sum_{i\in V}\min\{\sum_{j\in X} s_{ij}, \alpha_i\}$	f^{sub}	$[\sum_{j\in X} s_{ij}, i\in V]$	$O(n^2)$	$O(n)$
Graph Cut	$\lambda\sum_{i\in V}\sum_{j\in X} s_{ij} - \sum_{i,j\in X} s_{ij}$	f^{sub}	$[\sum_{j\in X} s_{ij}, i\in V]$	$O(n^2)$	$O(n)$
DPP	$\log\det(S_X))$	f^{sub}	SVD(S_X)	$O(\lvert X\rvert^3)$	$O(\lvert X\rvert^2)$
Feature Based	$\sum_{i\in\mathscr{U}}\psi(w_i(X))$	f^{sub}	$[w_i(X), i\in\mathscr{U}]$	$O(n\lvert\mathscr{U}\rvert)$	$O(\lvert\mathscr{U}\rvert)$
Set Cover	$w(\cup_{i\in X}U_i)$	f^{sub}	$\cup_{i\in X}U_i$	$O(n\lvert\mathscr{U}\rvert)$	$O(\lvert\mathscr{U}\rvert)$
Prob. Set Cover	$\sum_{i\in\mathscr{U}} w_i[1-\prod_{k\in X}(1-p_{ik})]$	f^{sub}	$[\prod_{k\in X}(1-p_{ik}), i\in\mathscr{U}]$	$O(n\lvert\mathscr{U}\rvert)$	$O(\lvert\mathscr{U}\rvert)$
Dispersion Min	$\min_{k,l\in X,k\neq l} d_{kl}$	f^{d}	$\min_{k,l\in X,k\neq l} d_{kl}$	$O(\lvert X\rvert^2)$	$O(\lvert X\rvert)$
Dispersion Sum	$\sum_{k,l\in X} d_{kl}$	f^{sup}	$[\sum_{k\in X} d_{kl}, l\in X]$	$O(\lvert X\rvert^2)$	$O(\lvert X\rvert)$
Dispersion Min-Sum	$\sum_{k\in X}\min_{l\in X} d_{kl}$	f^{sub}	$[\min_{k\in X} d_{kl}, l\in X]$	$O(\lvert X\rvert^2)$	$O(\lvert X\rvert)$

- s_{ij} is the similarity between elements i and j
- $\mathscr{U}$ is a set of concepts with w_a being weight of concept a. Each element $i\in X$ contains a subset $U_i\in\mathscr{U}$ of concepts. ψ is a concave function.
- d_{ij} is the distance between elements i and j.

We have also developed several new functions (based on $rating(.)$ functions, positive, repetitive and negative constraints expresed through X_P, X_R and X_N respectively), such as

$$f(X) = \sum_{x_i\in X_P}|X\cap x_i|\times\left(1+\frac{|X\cap x_i|}{|x_i|}\right)\times e^{\alpha\times rating(x_i)} + \sum_{x_i\in X_R}\min(|X\cap x_i|,\beta)\times\left(1+\frac{\min(|X\cap x_i|,\beta)}{\min(|x_i|,\beta)}\right)\times e^{\alpha\times rating(x_i)} - \sum_{x_i\in X_N}|X\cap x_i|*k$$

Human-Machine Interaction and Consensus in ML: Summary



- **Training Data:**
  - **Consensus for (actively) curating training data [AAAI '18a, ongoing]**, Subset Selection on labels [ACL '15], [CIKM '16] and instances with applications in active learning [WACV '19a], [**NCETIS, IITB**]
- **Learner/Classifier:**
  - Learning Interpretable Logical Features [ML '09, ICML '11, AAAI '12, EMNLP '12]
  - **Incorporating Domain Patterns in Sequence-to-sequence Models for image, text and speech recognition [ICDAR '17, InterSpeech '18]**
  - Incorporating background (symbolic) knowledge in Model Learning [AAAI '16, PAKDD '16, **AAAI '18b**]
  - Incorporating Ontologies [CIKM '16, ISWC '13, ISWC '16, NAACL '18]
- **Results**: **Non-decomposable and Interactively Defined Performance Measures [NAACL '15, CIKM '16, IJCAI '16, AAAI '17, PAKDD '18]**: See FUSS talk

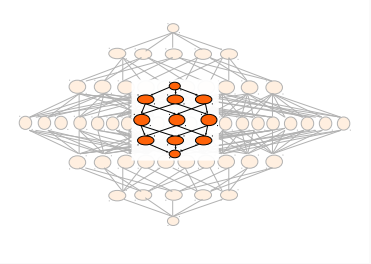
The Bigger Pictures: Plans going ahead

- Modeling interaction, consensus and incorporation of domain knowledge in complex problems
  - Examples: Machine Translation (MT), Automatic Question Generation, Caption generation for Audio-Visual input, (query-driven) Summarization
  - Initial success published for MT in 2016 & 2018, need to adopt to deep learning models
  - Very excited/assured by our success on OCR and human interaction on its output
- Scaling up (multi-instance) multi-label classification approaches to millions of labels (exhibiting heavy tailed distributions)
- Optimizing deep learning on performance measures that matter: BLEU score (MT), ranking losses, interactions on confusion matrix

The generic problem: Subset selection over Directed Acyclic Graph (DAG)

Supervised subset selection over lattices: Hierarchical Kernel Learning (HKL) framework: [JMLR '15, AAAI '12, ILP '12, EMNLP '12, ICML '11]

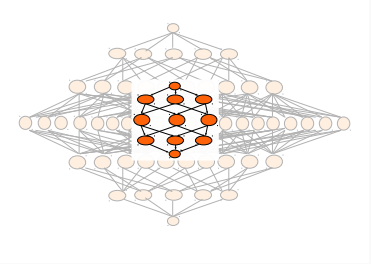
1. convex relaxation via l_1/l_ρ block-norm regularizer ($\rho \in (1,2]$)
2. efficient mirror descent-based active set algorithms on the dual (convex, Lipschitz conts., sub-differential objective over a simplex) with an efficiently computable sub-differential
3. generalization to multi-class, multi-label and multi-task
4. generalization to disjunctions/conjunctions of propositions, first order logic and sequentially structured output spaces



The generic problem: Subset selection over Directed Acyclic Graph (DAG)

Supervised subset selection over lattices: Hierarchical Kernel Learning (HKL) framework: [JMLR '15, AAAI '12, ILP '12, EMNLP '12, ICML '11]

1. convex relaxation via l_1/l_ρ block-norm regularizer ($\rho \in (1,2]$)
2. efficient mirror descent-based active set algorithms on the dual (convex, Lipschitz conts., sub-differential objective over a simplex) with an efficiently computable sub-differential
3. generalization to multi-class, multi-label and multi-task
4. generalization to disjunctions/conjunctions of propositions, first order logic and sequentially structured output spaces

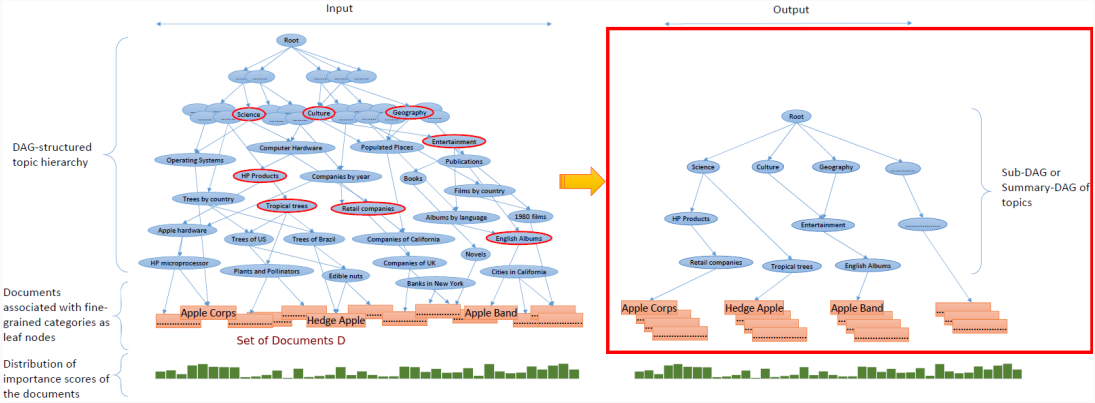


Summarizing by maximizing submodular/supermodular mixtures, learnt through max margin formulations. [ACL '15, CIKM '16, PAKDD '16, AAAI '18b, WACV '18a, WACV 18b]

Summarizing DAG-structured hierarchies

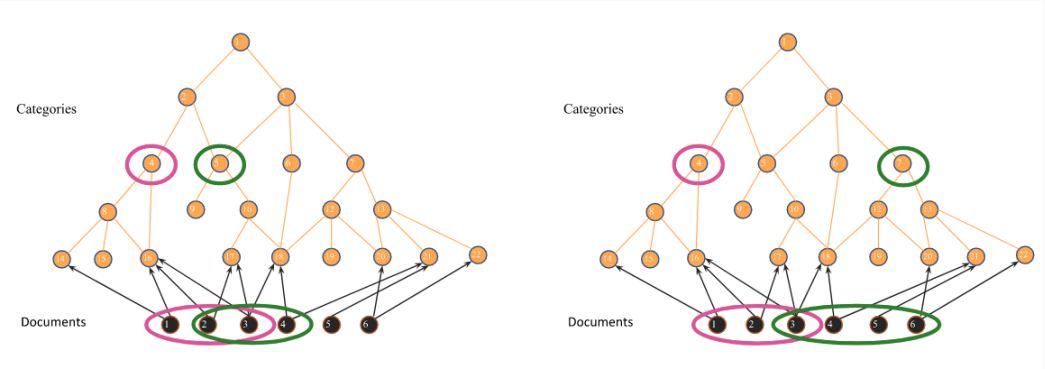
- $G(V, E)$: DAG structured hierarchy of V topics (or ground elements). E encodes parent-child (isa) relationship

Output: A representative subset S^* of size $K \in \mathbb{Z}_+$, that 'best' describes **D**

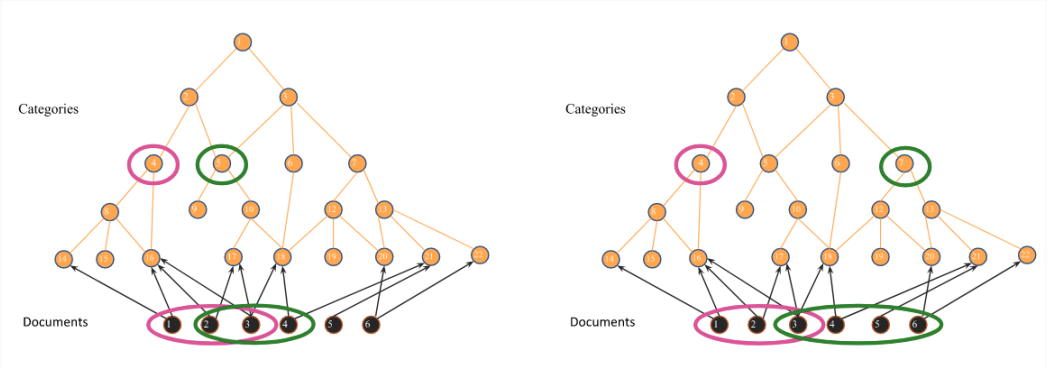


- **D**: Set of documents associated (hard/soft) with one or more of these topics.

Submodularity: Example of Coverage Functions (to maximize)



Submodularity: Example of Coverage Functions (to maximize)

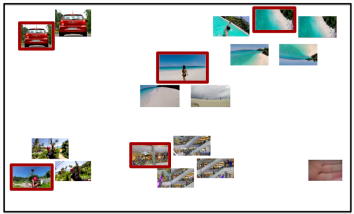


A set function $f^{\text{sub}}(.)$ is said to be submodular if for any element v and sets $A \subseteq B \subseteq V \setminus \{v\}$, $\mathsf{f}(A \cup \{v\}) - \mathsf{f}(A) \geq \mathsf{f}(B \cup \{v\}) - \mathsf{f}(B)$. Eg: Coverage, which we would like to maximize

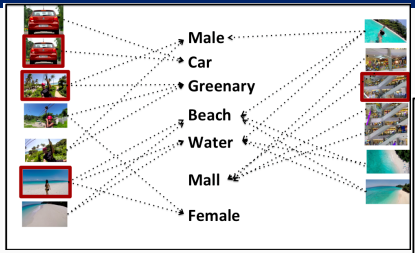
A function $f^{\text{sup}}(.)$ is said to be supermodular if the inequality above is reversed

Several submodular functions are also monotone and denoted $f^{\text{msub}}(.)$

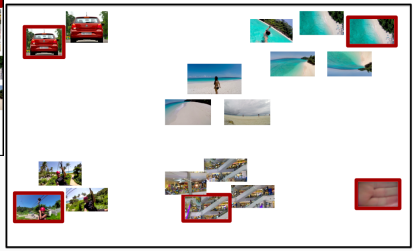
More submodular functions (to maximize): Data subsetting



Facility Location/Representation Function: Characterized by similarity s_{ij} between elements i and j



Weighted Feature Coverage Function: Characterized by set of concepts $\mathscr{U}$ with w_a being weight of concept a. Each element $i \in X$ contains a subset $U_i \in \mathscr{U}$ of concepts (for example, an image covers a table, chair and person).



Diversity: Characterized by distance d_{ij} between elements i and j.

Name	$f(X)$	Nature	Pre-compute stats	Std complexity	Reduced complexity.
Facility Location	$\sum_{i\in V} \max_{k\in X} s_{ik}$	f^{sub}	$[\max_{k\in X} s_{ik}, i \in V]$	$O(n^2)$	$O(n)$
Saturated Coverage	$\sum_{i\in V} \min\{\sum_{j\in X} s_{ij}, \alpha_i\}$	f^{sub}	$[\sum_{j\in X} s_{ij}, i \in V]$	$O(n^2)$	$O(n)$
Graph Cut	$\lambda \sum_{i\in V}\sum_{j\in X} s_{ij} - \sum_{i,j\in X} s_{ij}$	f^{sub}	$[\sum_{j\in X} s_{ij}, i \in V]$	$O(n^2)$	$O(n)$
DPP	$\log\det(S_X))$	f^{sub}	$\text{SVD}(S_X)$	$O(\lvert X\rvert^3)$	$O(\lvert X\rvert^2)$
Feature Based	$\sum_{i\in\mathscr{U}} \psi(w_i(X))$	f^{sub}	$[w_i(X), i \in \mathscr{U}]$	$O(n\lvert\mathscr{U}\rvert)$	$O(\lvert\mathscr{U}\rvert)$
Set Cover	$w(\cup_{i\in X} U_i)$	f^{sub}	$\cup_{i\in X} U_i$	$O(n\lvert\mathscr{U}\rvert)$	$O(\lvert\mathscr{U}\rvert)$
Prob. Set Cover	$\sum_{i\in\mathscr{U}} w_i[1-\prod_{k\in X}(1-p_{ik})]$	f^{sub}	$[\prod_{k\in X}(1-p_{ik}), i \in \mathscr{U}]$	$O(n\lvert\mathscr{U}\rvert)$	$O(\lvert\mathscr{U}\rvert)$
Dispersion Min	$\min_{k,l\in X, k\neq l} d_{kl}$	f^{d}	$\min_{k,l\in X, k\neq l} d_{kl}$	$O(\lvert X\rvert^2)$	$O(\lvert X\rvert)$
Dispersion Sum	$\sum_{k,l\in X} d_{kl}$	f^{sup}	$[\sum_{k\in X} d_{kl}, l \in X]$	$O(\lvert X\rvert^2)$	$O(\lvert X\rvert)$
Dispersion Min-Sum	$\sum_{k\in X} \min_{l\in X} d_{kl}$	f^{sub}	$[\min_{k\in X} d_{kl}, l \in X]$	$O(\lvert X\rvert^2)$	$O(\lvert X\rvert)$

- s_{ij} is the similarity between elements i and j
- $\mathscr{U}$ is a set of concepts with w_a being weight of concept a. Each element $i \in X$ contains a subset $U_i \in \mathscr{U}$ of concepts. ψ is a concave function.
- d_{ij} is the distance between elements i and j.

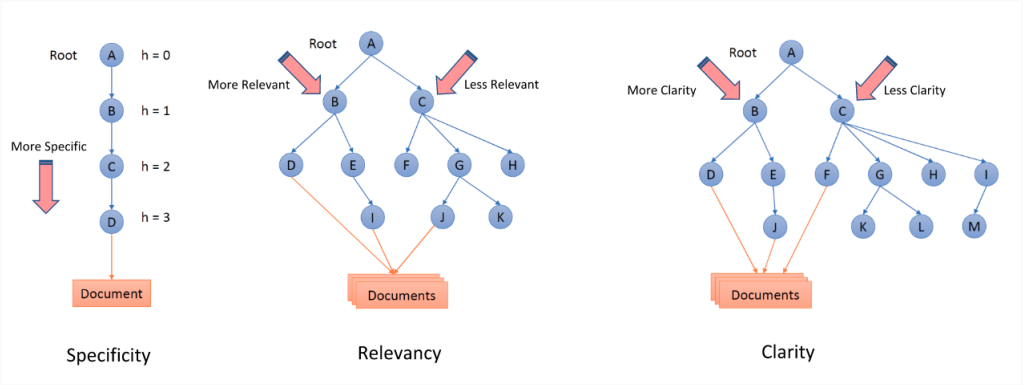
Name	$f(X)$	Nature	Pre-compute stats	Std complexity	Reduced complexity.
Facility Location	$\sum_{i\in V}\max_{k\in X} s_{ik}$	f^{sub}	$[\max_{k\in X} s_{ik}, i\in V]$	$O(n^2)$	$O(n)$
Saturated Coverage	$\sum_{i\in V}\min\{\sum_{j\in X} s_{ij}, \alpha_i\}$	f^{sub}	$[\sum_{j\in X} s_{ij}, i\in V]$	$O(n^2)$	$O(n)$
Graph Cut	$\lambda\sum_{i\in V}\sum_{j\in X} s_{ij} - \sum_{i,j\in X} s_{ij}$	f^{sub}	$[\sum_{j\in X} s_{ij}, i\in V]$	$O(n^2)$	$O(n)$
DPP	$\log\det(S_X))$	f^{sub}	SVD(S_X)	$O(\lvert X\rvert^3)$	$O(\lvert X\rvert^2)$
Feature Based	$\sum_{i\in\mathscr{U}}\psi(w_i(X))$	f^{sub}	$[w_i(X), i\in\mathscr{U}]$	$O(n\lvert\mathscr{U}\rvert)$	$O(\lvert\mathscr{U}\rvert)$
Set Cover	$w(\cup_{i\in X}U_i)$	f^{sub}	$\cup_{i\in X}U_i$	$O(n\lvert\mathscr{U}\rvert)$	$O(\lvert\mathscr{U}\rvert)$
Prob. Set Cover	$\sum_{i\in\mathscr{U}} w_i[1-\prod_{k\in X}(1-p_{ik})]$	f^{sub}	$[\prod_{k\in X}(1-p_{ik}), i\in\mathscr{U}]$	$O(n\lvert\mathscr{U}\rvert)$	$O(\lvert\mathscr{U}\rvert)$
Dispersion Min	$\min_{k,l\in X,k\neq l} d_{kl}$	f^{d}	$\min_{k,l\in X,k\neq l} d_{kl}$	$O(\lvert X\rvert^2)$	$O(\lvert X\rvert)$
Dispersion Sum	$\sum_{k,l\in X} d_{kl}$	f^{sup}	$[\sum_{k\in X} d_{kl}, l\in X]$	$O(\lvert X\rvert^2)$	$O(\lvert X\rvert)$
Dispersion Min-Sum	$\sum_{k\in X}\min_{l\in X} d_{kl}$	f^{sub}	$[\min_{k\in X} d_{kl}, l\in X]$	$O(\lvert X\rvert^2)$	$O(\lvert X\rvert)$

- s_{ij} is the similarity between elements i and j
- $\mathscr{U}$ is a set of concepts with w_a being weight of concept a. Each element $i\in X$ contains a subset $U_i\in\mathscr{U}$ of concepts. ψ is a concave function.
- d_{ij} is the distance between elements i and j.

We have also developed several new functions (based on $rating(.)$ functions, positive, repetitive and negative constraints expresed through X_P, X_R and X_N respectively), such as

$$f(X) = \sum_{x_i\in X_P} |X\cap x_i| \times \left(1+\frac{|X\cap x_i|}{|x_i|}\right)\times e^{\alpha\times rating(x_i)} + \sum_{x_i\in X_R}\min(|X\cap x_i|,\beta)\times\left(1+\frac{\min(|X\cap x_i|,\beta)}{\min(|x_i|,\beta)}\right)\times e^{\alpha\times rating(x_i)} - \sum_{x_i\in X_N}|X\cap x_i|*k$$

Modular functions: Eg: Cost/Benefit Functions



$c(X)$ is a modular function if $c(X) = \sum_{x \in X} c(x)$, where $c(x)$ is some cost/benefit of element x.

Maximizing submodular functions under constraints

- Problem 1 is knapsack constrained submodular maximization: $\max_{X \subseteq V, c(X) \leq b} f(X)$

  Goal: Find a summary with a fixed cost.A special case is cardinality constraint $(c(x) = 1)$
- Problem 2 is called the Submodular Cover Problem $\min_{f(X) \geq \theta} c(X)$

  with θ being a coverage constraint.

  Goal: Find a minimum cost subset X with sufficient coverage. A special case is the set cover problem. Problem 2 can be seen as a Dual version of Problem 1
- A simple greedy algorithm obtains a $1 - \frac{1}{e}$ approximation guarantee for monotone submodular function maximization
- The form we mostly consider: $S^* \in \operatorname{argmax}_{X \subseteq V : |X| \leq K} \sum_i w_i f_i(X)$

Maximizing mixtures of (monotone) sub/supermodular functions

We can show that a greedy algorithm for $\max\{f(X) \mid X \subseteq V, |X| \leq k\}$ for the following combination of non-negative functions

$$f(X) = \alpha f^{\text{msub}}(X) + \beta f^{\text{sub}}(X) + \gamma f^{\text{sup}}(X) + \delta f^{\text{d}}(X)$$

gives the following approximation guarantees:

1. $1 - 1/e$ if $\alpha \geq 0$ and $\beta = \gamma = \delta = 0$.
2. $1/2$ if $\delta > 0$ and $\alpha = \beta = \gamma = 0$.
3. $1/e$ if $\alpha \geq 0, \beta > 0$ and $\gamma = \delta = 0$.
4. $1/4$ if $\alpha > 0, \delta > 0$ and $\beta = \gamma = 0$
5. $1/2e$ if $\alpha, \beta, \delta \geq 0$ and $\gamma = 0$.
6. $\frac{(1-e^{(1-\kappa^l)\kappa_k})}{\kappa_k}$ where $k(X) = f^{\text{msub}}(X)$ and $l(X) = f^{\text{sup}}(X)$, if $\alpha, \gamma \geq 0$ and $\beta = \delta = 0$.
7. $\frac{(1-e^{(1-\kappa^l)\kappa_k})}{2\kappa_k}$ where $k(X) = f^{\text{msub}}(X)$ and $l(X) = f^{\text{sup}}(X)$, if $\alpha, \gamma, \delta \geq 0$ and $\beta = 0$.
8. inapproximable unless P = NP, if $\beta, \gamma > 0$ and $\alpha, \delta \geq 0$.

Large Margin Learning

- We optimize the weights w of the scoring function $F_w(\cdot)$ in a large-margin structured prediction framework

$$\min_{w \geq 0, \|w\|_1 = 1} \quad \sum_{S \in \mathscr{S}} \left[\max_{S' \in \mathscr{Y}} \left[F_w(S') + \mathscr{L}(S') \right] - F_w(S) \right] + \frac{\lambda}{2} \|w\|_2^2, \tag{13}$$

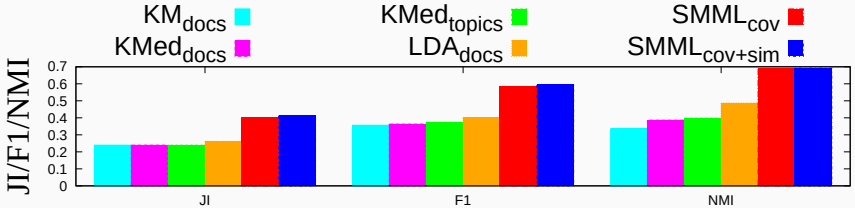
  where $\mathscr{S} = \{S_1, S_2, \cdots, S_N\}$ is the ground-truth summary, $\mathscr{L}(\cdot)$ is the loss function, and $\mathscr{Y}$ is a structured output space (for example $\mathscr{Y}$ is the set of summaries that satisfy a certain budget B, i.e., $\mathscr{Y} = \{S' \subseteq V : |S'| \leq B\}$). We assume

- Loss is normalized, $0 \leq \mathscr{L}(S') \leq 1, \forall S' \subseteq V$
- $w \geq 0$, ensures that the final mixture is submodular.
- w are learnt using stochastic gradient descent

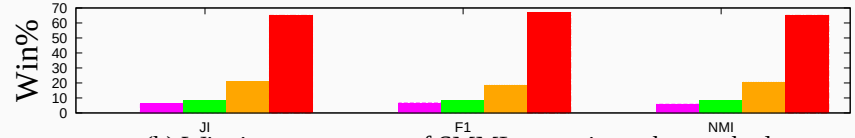
Evaluation for Hierarchy Summarization

- Evaluated on the task of automatic generation of Wikipedia disambiguation page
- Comparing the clusters induced by the summary topics with the clusters on Wikipedia disambiguation page
- Compared with K-Means, K-Medoids, LDA
- **KM_{docs}:** K-Means algorithm run on documents
- **$KMed_{docs}$:** K-Medoids algorithm with documents
- **$KMed_{topics}$:** K-Medoids run on topics as TF-IDF vectors of words.
- **LDA_{docs}:** LDA algorithm with the number of topics set to the number of true clusters on the Wikipedia disambiguation page
- **$SMML_{cov}$:** Submodular mixtures without similarity based functions
- **$SMML_{cov+sim}$:** All classes of submodular functions

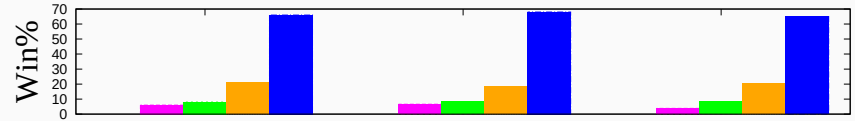
Results



(a) Comparing metrics with baselines

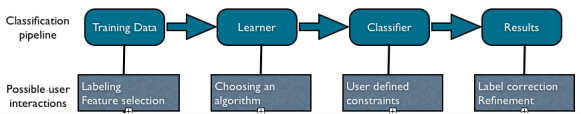


(b) Winning percentages of $SMML_{cov}$ against other methods



(c) Winning percentages of $SMML_{cov+sim}$ against other methods

Human-Machine Interaction and Consensus in ML: Summary



- **Training Data:**
  - **Consensus for (actively) curating training data [AAAI '18a, ongoing]**, Subset Selection on labels [ACL '15], [CIKM '16] and instances with applications in active learning [WACV '19a], [**NCETIS, IITB**]
- **Learner/Classifier:**
  - Learning Interpretable Logical Features [ML '09, ICML '11, AAAI '12, EMNLP '12]
  - **Incorporating Domain Patterns in Sequence-to-sequence Models for image, text and speech recognition [ICDAR '17, InterSpeech '18]**
  - Incorporating background (symbolic) knowledge in Model Learning [AAAI '16, PAKDD '16, **AAAI '18b**]
  - Incorporating Ontologies [CIKM '16, ISWC '13, ISWC '16, NAACL '18]
- **Results**: **Non-decomposable and Interactively Defined Performance Measures [NAACL '15, CIKM '16, IJCAI '16, AAAI '17, PAKDD '18]**: https://www.cse.iitb.ac.in/~av/fuss/s-2017/ganesh-video.webm

The Bigger Pictures: Plans going ahead

- Modeling interaction, consensus and incorporation of domain knowledge in complex problems
  - Examples: Machine Translation (MT), Automatic Question Generation, Caption generation for Audio-Visual input, (query-driven) Summarization
  - Initial success published for MT in 2016 & 2018, need to adopt to deep learning models
  - Very excited/assured by our success on OCR and human interaction on its output
- Scaling up (multi-instance) multi-label classification approaches to millions of labels (exhibiting heavy tailed distributions)
- Optimizing deep learning on performance measures that matter: BLEU score (MT), ranking losses, interactions on confusion matrix

Thank You

Loss Function and Inference

Loss:

- Loss function is defined as

$$\mathscr{L}_{\text{jaccard}}(S,T) = 1 - \frac{1}{k}\sum_{s\in S}\max_{t\in T}\frac{\left|\Gamma(s)\cap\Gamma(t)\right|}{\left|\Gamma(s)\cup\Gamma(t)\right|} \tag{14}$$

  where S is the inferred topics and T is the true topics, $k = |S| = |T|$ is the number of topics.
- When the clustering produced by the inferred and the true topics are similar, Jaccard loss is 0. When they are completely dissimilar, the loss is maximum, i.e., 1.
- Jaccard loss is a modular function

Inference:

- Greedy algorithm
- Solution is within $1 - 1/e$ factor from the optimal solution

How Good is Our Consensus Model?

Dataset	Judgments		Majority		MLCM-r		MIML^{perf}	
	κ	F_1	κ	F_1	κ	F_1	κ	F_1
Flags	.6 - .8	.83-.9	.874	.934	.874	.934	.873	.933
Scene	.32 - .6	.35 - .65	.637	.675	.645	.682	**.641**	**.679**
Scene*	.34 - .79	.37 - .81	.824	.846	.851	.869	**.894**	**.907**

*Corrected ground truth

Table 3: How good is the consensus output

Effect of Batch Size on Active Learning

Starting from 50 labeled instances, we ran active learning iterations until the labeled data size reached 350. The sampling batch size was set to 2, 5, 10, 25, 50 and 100.

(a) AL-U

(b) AL-I

A smaller batch size generally results in better performance. Smaller batch sizes offer more opportunities to evaluate the unlabeled data on continuously improving models, thereby sampling the most informative instances.

Run-Time Analysis

For a fixed batch size:

1. Began with 50 labeled instances of the medical dataset
2. Iterated until we had 350 labeled instances.

We report the total wall-clock time for each batch size.

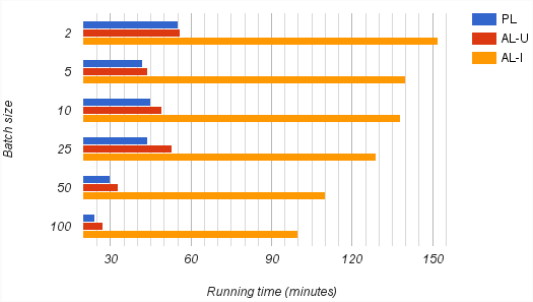


Figure 6: Effect of Batch size on running time

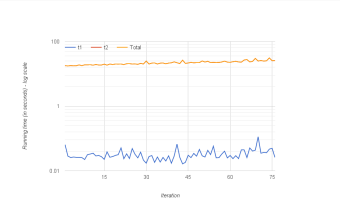
- As the batch size increases, running time decreases for each strategy.
- As expected, AL-I takes significantly more time as it involves evaluation of the impact of labeling of each unlabeled instance on the expected agreement.

Reducing Drudgery in Generated Labeled Data

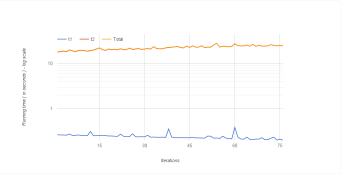
- We already saw that AAAI ’18 work can help learn with less data more effectively. But it is only in retrospect after labeling was performed
- Imagine we have our MTP reports for this round and we want to organize/label them
- Identifying subset.. It is also possible to reduce the drudgery involved in labeling data by identifying a subset of a large unlabeled datset that would be the most appropriate to be labeled. A special class of subset selection functions, viz., mixtures of submodular functions, naturally models notions of diversity, coverage and representation. This class can be used to eliminate redundancy, thus lending itself well for summarizing training data sets through subset selection. It also helps improve the efficiency of active learning in further reducing human labeling efforts. The data subsetting can also be augmented by identifying a subset of labels that are more relevant to the dataset at hand. Further, most often, these labels are nodes in a DAG (Directed Acyclic Graph). We present models for identifying label subsets over DAGs using submodular mixtures and present methods to learn these mixtures. I will also summarize our other efforts toward human assisted machine learning - (a) domain knowledge based feature induction and (b) optimizing performance measures that are

Run-Time Analysis (contd.)

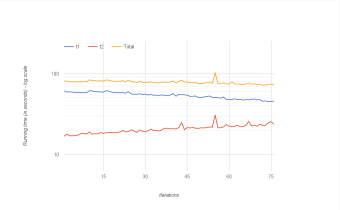
1. Started with 1000 labeled instances, we ran 75 iterations of learning, in batches of 5.



(a) Passive Learning



(b) AL-U



(c) AL-I

Figure 7: Effect of size of unlabeled instances (iteration) on running time. $t1$: time for searching the unlabeled instances; $t2$: time for updating the consensus model.

- $t1$ decreases as the number of unlabeled instances progressively reduces with each learning iteration.
- For AL-I, $t1$ dominates the total running time; progressively decreases.
- In case of PL and AL-U, $t1$ is much lower than $t2$; gradual rise in their total running

Summarizing DAG-structured hierarchies of patterns

Grammar for patterns

V *(Non Terminals)*	**Drug**, **Disease**, **T1**, **NP**, **S**
Σ *(Terminals)*	*ACOMPLIA ,Actraphane,severe, hepatic, impairment, diabetes*, Patients, with, for, <1,6,11>, <2,4,7>,<1,6,7>, <2,4,5>, <1,8,11>, …
R *(Production Rule)*	**Drug** -> *ACOMPLIA* | *Actraphane* **Disease** -> *severe hepatic impairment* | *diabetes* **T1** -> Patients with Disease | Drug for Disease **NP** -> <1,0,1> | <1,6,7> | <1,6,11> | <1,8,11> **NP** -> <2,0,1> | <2,4,5> | <2,4,7> **S** -> **Drug** |**Disease** |**T1** |**NP** |
S *(Dummy Start symbol)*	**S**

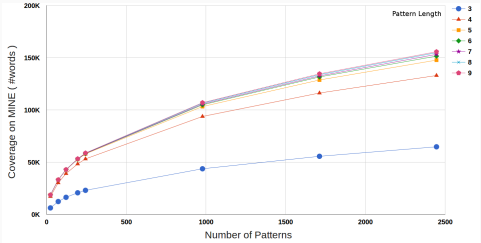
PATTERN: in patients with ■CAT1■ (568)	**contains ■CAT2■ mg of ■CAT3■ (91)**
in patients with HIT type II	capsule contains 25 mg of lenalidomide
in patients with CNS metastases	tablet contains 300 mg of maraviroc
in patients with ESRD	syringe contains 100 mg of anakinra
in patients with normal and impaired renal function	tablet contains 2.3 mg of sucrose
in patients with previous history of pancreatitis	capsule contains 200 mg of pregabalin
in patients with cirrhosis of the liver	vial contains 10 mg of the active substance
	tablet contains 30 mg of aripiprazole

Summarizing DAG-structured hierarchies of patterns

Grammar for patterns

V *(Non Terminals)*	**Drug**, **Disease**, **T1**, **NP**, **S**
Σ *(Terminals)*	*ACOMPLIA ,Actraphane,severe, hepatic, impairment, diabetes*, Patients, with, for, <1,6,11>, <2,4,7>,<1,6,7>, <2,4,5>, <1,8,11>, …
R *(Production Rule)*	**Drug** -> *ACOMPLIA* | *Actraphane* **Disease** -> *severe hepatic impairment* | *diabetes* **T1** -> Patients with **Disease** | **Drug** for **Disease** **NP** -> <1,0,1> | <1,6,7> | <1,6,11> | <1,8,11> **NP** -> <2,0,1> | <2,4,5> | <2,4,7> **S** -> **Drug** |**Disease** |**T1** |**NP** |
S *(Dummy Start symbol)*	**S**

Diminishing Returns with increasing subset size



PATTERN: in patients with ■CAT1■ (568)	**contains ■CAT2■ mg of ■CAT3■ (91)**
in patients with HIT type II	capsule contains 25 mg of lenalidomide
in patients with CNS metastases	tablet contains 300 mg of maraviroc
in patients with ESRD	syringe contains 100 mg of anakinra
in patients with normal and impaired renal function	tablet contains 2.3 mg of sucrose
in patients with previous history of pancreatitis	capsule contains 200 mg of pregabalin
in patients with cirrhosis of the liver	vial contains 10 mg of the active substance
	tablet contains 30 mg of aripiprazole

Summarizing DAG-structured hierarchies of nodes

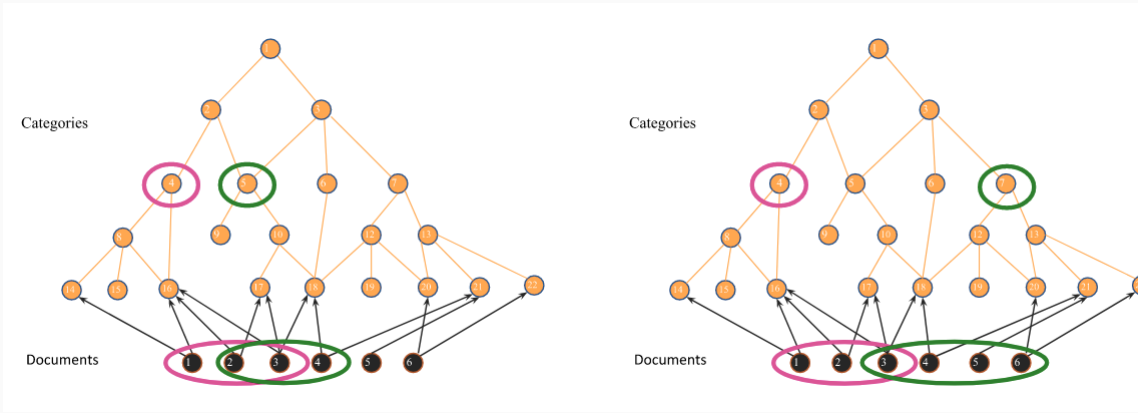
Summarizing DAG-structured hierarchies of nodes over a given set of instances by maximizing submodular mixtures, learnt through max margin formulations (RJB '07, CVR '08, BIRJ '15, KSKOR '16):

1. Generating Wikipedia disambiguation pages: Wikipedia categories as nodes and Wikipedia entities as instances
2. Building compact lexicons of patterns for cross-domain machine translation: Patterns as nodes and Sentences as instances
3. Generating candidate multi-labels a machine learning dataset: Candidate labels as nodes and Documents as instances

Desirable properties

- $\Gamma(s)$: Set of documents (transitively) covered by a topic s. Natural extension to set S is $\Gamma(S) = \cup_{s \in S} \Gamma(s)$
- $\Gamma^{\alpha(s)} \subseteq \Gamma(t)$ has path length between a document and s upper bounded by α
- **Goal:** Identify summary set of topic $S \subseteq V$ with following properties.
- **Coverage:** S should cover most of the documents. A document d is said to be covered by a topic t if $d \in \Gamma(t)$
- "Quality" Criterion: **Specificity/Clarity/Relevance/Coherence:** Help us choose a set of topics that are neither too abstract nor overly specific.

Desirable Property: Coverage Functions



Coverage components capture “coverage” of a set of documents.

- **Weighted Set Cover Function:** Given $S \subseteq V, f(S) = \sum_{d \in \Gamma(S)} w_d = w(\Gamma(S))$, assigns weights to the documents based on their relative importance (e.g., in Wikipedia

Similarity-based Functions

Similarity components capture the similarity between two topics.

- Defined through a similarity matrix: $\mathbf{S} = \{s_{ij}\}_{i,j \in V}$. Given $i, j \in V$, $s_{ij} = |\Gamma(i) \cap \Gamma(j)|$, (number of documents commonly covered)
- **Facility Location:** $f(S) = \sum_{i \in V} \max_{j \in S} s_{ij}$, is a natural model for k-medoids and exemplar based clustering.
- **Penalty based diversity:** A similarity matrix may be used to express a form of coverage of a set S but penalized with a redundancy term: $f(S) = \sum_{i \in V, j \in S} s_{ij} - \lambda \sum_{i \in S} \sum_{j \in S,} s_{i,j}$; here $\lambda \in [0, 1]$.

QC Functions as Barrier Modular Mixtures

- A modular function for every QC function:

  $f^{\alpha}_{\text{specificity}}(s) = \begin{cases} 1 & \text{if the height of topic } s \text{ is at least } \alpha \\ 0 & \text{otherwise} \end{cases}$ for every possible value of α. This creates a submodular mixture with as many components as the number of possible values of α.

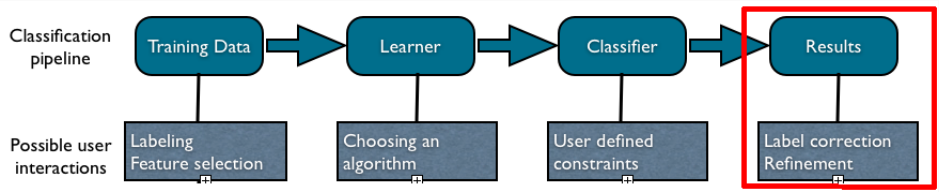
  $f^{\beta}_{\text{clarity}}(s) = \begin{cases} 1 & \text{if the clarity of topic } s \text{ is at least } \beta \\ 0 & \text{otherwise} \end{cases}$ for every possible (discretized to make it countably finite) value of β. And,

  $f^{\gamma}_{\text{relevance}}(s) = f_{\text{cov}}\left(s|\Gamma^{\gamma(s)}\right)$, where $f_{\text{cov}}(.)$ is the coverage submodular function and $s|X$ indicates coverage of a topic s over a set of documents X.

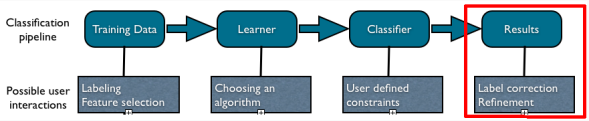
Fidelity Functions

- A function representing the fidelity of a set S to another reference set R is one that gets a large value when the set S represents the set R.
- R can be produced from other algorithms such as k-means, LDA and its variants or from a manually tagged corpus.
- **Topic Coherence:** This function scores a set of topics S high when $\Gamma(S)$ resembles the clusters of documents produced by an external source (k-means, LDA or manual). Given an external source that clusters the documents, producing T clusters $L_1, L_2, ..., L_T$ (for T topics), topic coherence is defined as: $f(S) = \sum_{t \in T} \max_{k \in S} w_{k,t}$ where $w_{k,t} = \text{harmonic\_mean}(w^p_{k,t}, w^r_{k,t})$ and $w^p_{k,t} = \frac{|\Gamma(k) \cap L_t|}{|\Gamma(k)|}$ and $w^r_{k,t} = \frac{|\Gamma(k) \cap L_t|}{|L_t|}$. Note that, $w^p_{k,t} \geq 0$ and $w^r_{k,t} \geq 0$ are the precision and recall of the resemblance

Results: Evaluation function vs. Loss function

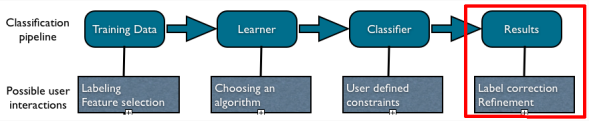


Results: Evaluation function vs. Loss function



- Frequently Used Losses: *Decomposable measures* derived by averaging the performance on individual data points $\mathbf{L}(\mathbf{f}(\mathbf{X}), \mathbf{Y}) = \frac{1}{N} \sum_{i=1}^{N} L(\mathbf{f}(\mathbf{x}_i), \mathbf{y}_i)$.
  - Eg: Hamming distance [Bi and Kwok 2013] Precision [Hsu 2009], Recall [Steck 2010], Cross Entropy, Square Loss, Hinge Loss etc.
  - Decomposable and easy to optimize
  - Tend to neglect performance on rare labels [Narasimhan 2014]
- Evaluation Measures (However): Often

Results: Evaluation function vs. Loss function



- Frequently Used Losses: *Decomposable measures* derived by averaging the performance on individual data points $\mathbf{L}(\mathbf{f}(\mathbf{X}), \mathbf{Y}) = \frac{1}{N}\sum_{i=1}^{N} L(\mathbf{f}(\mathbf{x}_i), \mathbf{y}_i)$.
  - Eg: Hamming distance [Bi and Kwok 2013] Precision [Hsu 2009], Recall [Steck 2010], Cross Entropy, Square Loss, Hinge Loss etc.
  - Decomposable and easy to optimize
  - Tend to neglect performance on rare labels [Narasimhan 2014]
- Evaluation Measures (However): Often *Non-decomposable measures* that cannot be decomposed into a sum of terms measured over individual examples
  - Eg: **F-measure**, **Diversity Measures** , **BLEU Score**, Normalized Discounted Cumulative Gain, Average Precision, Mean Average Precision, Precision@k, etc.

Decomposable Objective Function - Examples

(Mean) Square Error Loss :

$$(y^{(i)} - \mathbf{w}^T x^{(i)})^2 \tag{15}$$

Hinge loss (in Linear SVMs):

$$\max \left(1 - y^{(i)}(\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}^{(i)}) + b), 0\right) \tag{16}$$

Decomposable Objective Function - Examples

Cross Entropy : The cross entropy for the distributions p and q over a given set is defined as follows (discrete p and q)

$$H(p,q) = -\sum_{x} p(x) \log q(x). \tag{17}$$

Logistic Loss : The logistic regression model predicts an output $y \in \{0,1\}$, given an input vector $\mathbf{x}$. The probability is modeled using the logistic function $g(z) = 1/(1+e^{-z})$ and the loss function is

$$\frac{1}{N}\sum_{n=1}^{N} H(p_n, q_n) = -\frac{1}{N}\sum_{n=1}^{N} \left[y_n \log \hat{y}_n + (1-y_n) \log(1-\hat{y}_n) \right] \tag{18}$$

Non-Decomposable (Loss) Functions

- Functions that cannot be decomposed into a simple sum of individual terms measured over each example in the dataset
- Examples :
  F-measure, **Diversity Measures** , **BLEU Score**, Normalized Discounted Cumulative Gain, Average Precision, Mean Average Precision, Precision@k, etc.
- These are often used to evaluate (the effectiveness) of models and optimizing for them directly is therefore often desirable.
- Optimizing such functions, however, is most often, challenging.

Non-Decomposable Objective Function examples

Confusion Matrix :

		Predicted	
		Negative	Positive
Actual	Negative	**a**	**b**
	Positive	**c**	**d**

The entries in the confusion matrix have the following:

- a is the number of correct predictions that an instance is negative
- b is the number of incorrect predictions that an instance is positive,
- c is the number of incorrect of predictions that an instance negative, and
- d is the number of correct predictions that an instance is positive.

Non-Decomposable Objective Function examples - F Measure

Using Confusion matrix to calculate the following :

- Accuracy (AC) = $\frac{a+d}{a+b+c+d}$
- True Positive(TP) = $\frac{d}{c+d}$
- False Positive(FP) = $\frac{b}{a+b}$
- True Negative(TN) = $\frac{a}{a+b}$
- False Negative(FN) = $\frac{c}{c+d}$
- F measure = $\frac{(\beta^2+1)*P*TP}{\beta^2*P+TP}$

Non-Decomposable Objective Function examples - F Measure

F-measure : F-measure or balanced F-score (F1 score) is the harmonic mean of precision and recall.

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \tag{19}$$

The general formula for positive real β is:

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}} \tag{20}$$

Non-Decomposable Objective Function

- There are in general 2 ways to optimize a non-decomposable objective function :
  - **Direct optimization** : Formulate the learning problem to directly optimize this measure
  - **Indirect optimization** : Approximate this measure with a simpler one and try to optimize it aiming to indirectly optimize the original non-decomposable performance measure.
- Due to complexity of most objective functions they are hard to optimize directly
- Eg : Cannot use gradient descent for optimization

Example 1: Bargaining Through Confusion Matrix

		Predicted	
		0	**1**
Actual	**0**	916	43
	1	51	508

- A machine learning model trained to recognize **Spam Emails**.
- The user desires to reduce the False Positives as much as possible on Held-out dataset and specifies some minimum acceptability level
- Can the ML model meet the user requirements or 'bargain' with the user if unable to?

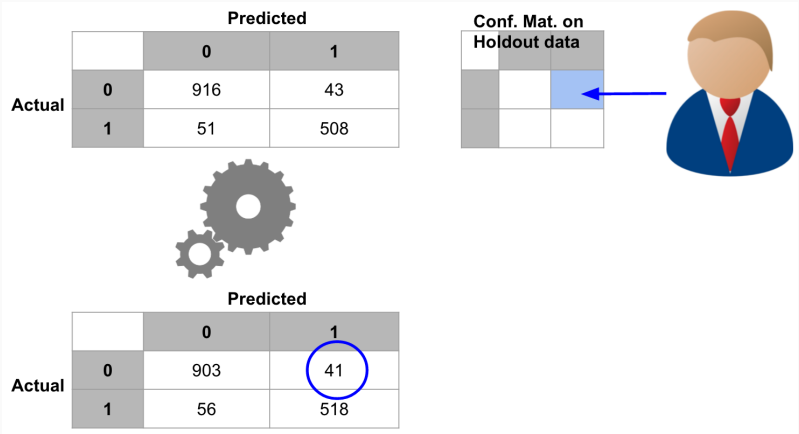
Example 1: Bargaining Through Confusion Matrix

		Predicted	
		0	**1**
Actual	**0**	916	43
	1	51	508

- A machine learning model trained to recognize **Spam Emails**.
- The user desires to reduce the False Positives as much as possible on Held-out dataset and specifies some minimum acceptability level
- Can the ML model meet the user requirements or 'bargain' with the user if unable to?

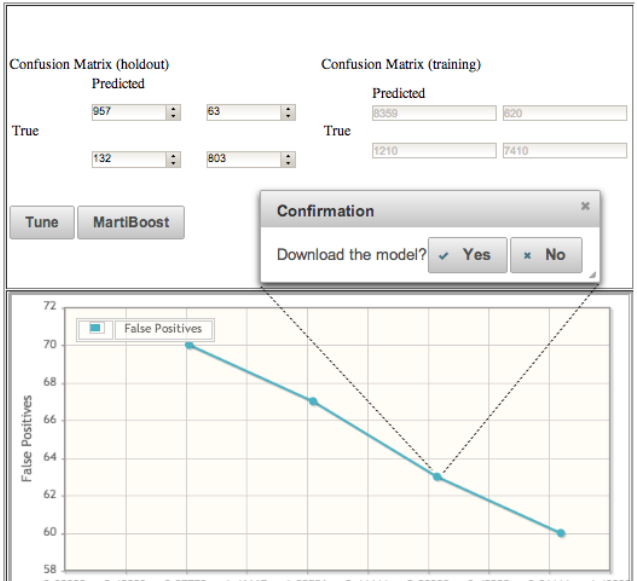
Extended to multiple classes while provisioning for arbitrary levels of accuracy [IJCAI 2016]

Example 1: User Bargains through Confusion Matrix



Goal: Match the User Specified True Positive (TP) and False Negative (FN) rates

Example 1: Model (Seller) Obliges and Explains



Classification with Asymmetric cost

- Misclassification cost of instances is not always the same for all classes
- Low FP desirable for spam detection
- Low FN desirable for disease detection

Challenges

1. What is the acceptable (low) number of FPs/FNs for a given data?
2. How can we “tune” a model, **with the provision of multiple iterations**, to meet user-specified preferences?
3. **How can the models be quickly ”tuned” on large datasets?**

Research Challenge

- $\mathscr{D}^k$ denotes distribution on dataset $\mathscr{D}$ restricted to examples in class C_k: $\{x \in \mathscr{X} : c(x) = l\}$
- Hypothesis $h : \mathscr{X} \to \{1 \ldots K\}$ is said to have K-sided advantage γ^k (for class k) with respect to $\mathscr{D}^k$ if it satisfies $Pr_{x \in \mathscr{D}^k}[h(x) = k] \geq \frac{1}{K} + \gamma^k$
- Thus, $\frac{1}{K} + \gamma^k$ are desired (minimum) accuracies $Acc^k_{des} = TP_k/(TP_k + FN_k)$ on $\mathscr{D}^k$
- **Q1**: How do we enable a user to decide an acceptable classifier performance γ^k?
- **Q2**: How do we tune a classifier to best meet these user preferences Acc^k_{des}?

Math Framework in the Bargaining...

$\mathscr{X}_{Train}$	Training data
$\mathscr{X}_{Tune}$	Tuning data
$\mathbf{w}$	Model parameters
$\mathbf{d}$	Model hyperparameters
$\mathbf{y}^{(i)}$	k-dimensional label vector
$y_k^{(i)}$	classification score for class k
$\mathbf{p}^{(i)}(\mathbf{d};\mathbf{w}^*)$	current state vector
$\mathbf{s}^{(i)}$	target state vector

Re-estimate **parameters** on Train Data to minimize divergence between target and current states (involves boosting)

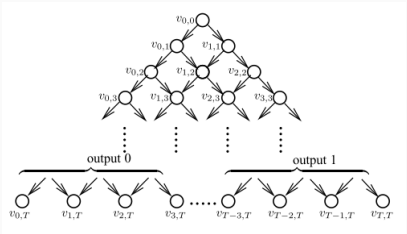
$$\mathbf{g}(\mathbf{d};\mathbf{w}^*) = \sum_{i=1}^{|\mathscr{X}_{Tune}|} KL(\mathbf{s}^{(i)}||\mathbf{p}^{(i)}(\mathbf{d};\mathbf{w}^*))$$

$$\mathbf{d}^* = \underset{\mathbf{d}\in\mathbb{R}^l}{\operatorname{argmin}}\ \mathbf{g}(\mathbf{d};\mathbf{w}^*) \tag{21}$$

$$\mathbf{w}^* = \underset{\mathbf{w}\in\mathbb{R}^n}{\operatorname{argmin}} \frac{1}{2}\mathbf{w}^T C\mathbf{w} - \sum_{i=1}^{|\mathscr{X}_{Train}|} \log \mathbf{p}^{(i)}(\mathbf{d}^*;\mathbf{w})$$

Martingale Boost: Expanding Scope for Bargaining

Typical boosting algorithm constructs accurate decision functions using weak base predictors. Drawback: **Noise intolerance**



- **Martingale Boosting** [Long 2005, Kalai 2005]: Branching program based boosting algorithm shown to have noise tolerance.
- Branching is over weak learners. (illustated on the left for $K = 2$)

Contributions: Increasing order of Importance

1. Extension of branching based program Martingale Boosting to multiple classes
   - Extensions of claims on upper bounds on errors that decrease exponentially as the square of the advantage γ_k
2. Weaker Error Bounds for increasing number of classes
3. Extensive experimental comparisons against (Adaptive) Martingale Boosting
4. **Exploiting the Multiple levels of Martingale Boosting to provision for multiple iterations of interactive model tuning based on held-out data set**

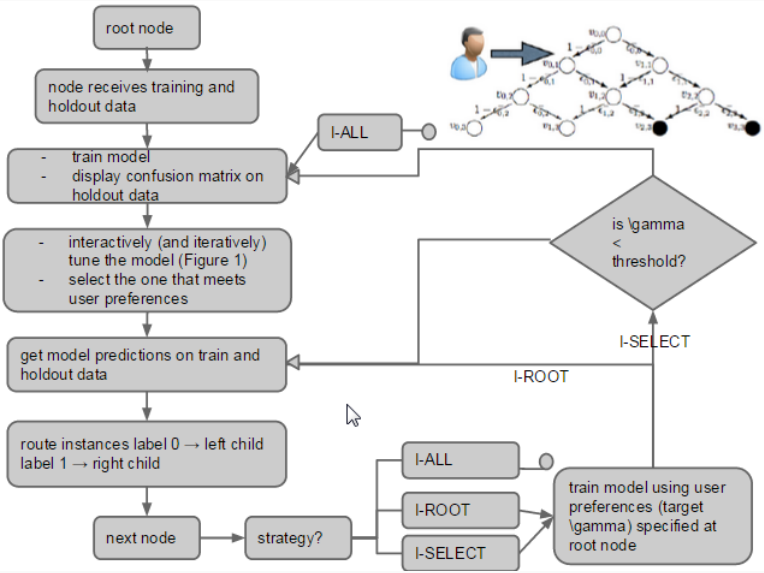
Granularity of Interaction on Tuning data

- I-ALL: (Our skyline) For small MB programs, users tune every node
  - Manual tuning effort is quadratic in the levels of MB
- I-ROOT: (Default MB-tune) Take human preferences at root node and assume the same target advantage (γ_k) at other nodes
- I-SELECT: selective manual tuning at only those nodes k where the advantage γ_k is below a threshold (0.3)

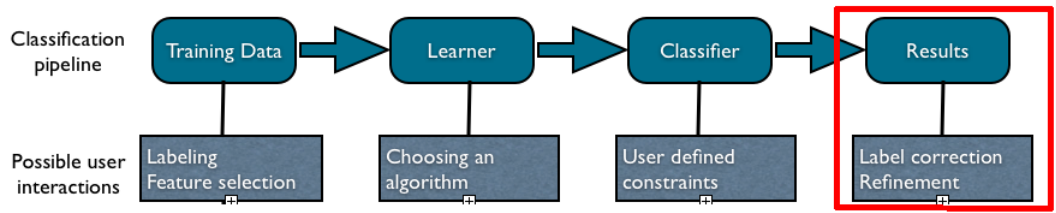
Early node freezing

- As the martingale program grows top-down, the distribution of examples reaching at some of the nodes (especially those at the extreme ends of a level) tends to be heavily biased towards a class label
- freeze a node $v_{i,j}$ if $\min_{b\in\{+,-\}} p_{i,j}^b < \frac{\varepsilon}{L(L+1)}$ for some error rate ε [Long 2005]

Interactive Martingale Boosting



Example 2: Optimizing User-specified Performance Measures for Multi-Instance Multi-Labeled Learning



Example 2: Optimizing Measures for Multi-Instance Multi-Labeled Learning

Knowledge base

r	e_1	e_2
BornIn	Barack Obama	U. S.
PresidentOf	Barack Obama	U. S.

Sentences	Latent Label
- *Barack Obama was born in Honolulu, Hawaii, United States.*	*BornIn*
- *Obama left United States this Saturday for a UN summit in Geneva.*	*none*
- *President Obama defended his administrations' collection of phone records in the U.S.*	*PresidentOf*

Example 2: Optimizing Measures for Multi-Instance Multi-Labeled Learning

Knowledge base

r	e_1	e_2
BornIn	Barack Obama	U. S.
PresidentOf	Barack Obama	U. S.

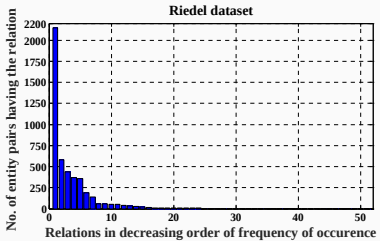
Sentences	Latent Label
- *Barack Obama was born in Honolulu, Hawaii, United States.*	*BornIn*
- *Obama left United States this Saturday for a UN summit in Geneva.*	*none*
- *President Obama defended his administrations' collection of phone records in the U.S.*	*PresidentOf*

Need (to optimize for) performance measures that account for

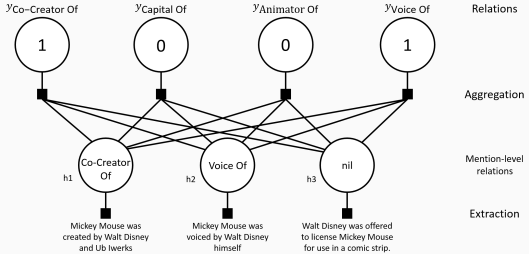
- Multi-instance labeling
- Multiple labels per instance bag
- Skew in label distribution
- Future work: Temporal Scoping

Optimizing Multivariate Performance measures for Multi Instance, Multi Label Problems with Applications to Deep Learning in Vision, Natural Language Processing [NAACL 20015, AAAI 2017]

Multiple Labels and Multiple Instances



(a)



(b)

Figure 8: (Left) The Riedel dataset exhibits a heavy tail in its label distribution, most relations are extremely rare. (Right) Mention level hidden labels can be used to compute the active relations for the entity-pair `(Walt Disney, Mickey Mouse)`.

Multiple Labels and Multiple Instances

Table 4: Dataset Statistics: Collaboration with MSR on Extreme Classificaton

	#bags	#labels	$\frac{\text{labels}}{\text{point}}$	$\frac{\text{points}}{\text{label}}$	$\frac{\text{instances}}{\text{bag}}$
Riedel	4350	52	1.08	90.4	6.6
Scene	2000	5	1.24	494.4	9
Reuters	2000	7	1.15	329.71	3.56
Wikipedia	billions	millions	...	...	....

MIML

- Compute parameters $\mathbf{w}$ to minimize a User Defined Performance (or Loss) function $\Delta(\mathbf{Y}', \mathbf{Y})$ (such as $_\beta$ or $_\beta$ over the dataset) over weakly supervised multi-instance labels $\mathbf{H}$, while reducing the complexity

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||_2^2 + C \max_{\mathbf{Y}'} \left\{ \max_{\mathbf{H}} \mathbf{w} \cdot \Psi(\mathbf{X}, \mathbf{H}, \mathbf{Y}') - \max_{\mathbf{H}} \mathbf{w} \cdot \Psi(\mathbf{X}, \mathbf{H}, \mathbf{Y}) + \Delta(\mathbf{Y}', \mathbf{Y}) \right\} \quad (22)$$

- Approach 1: Concave Convex Procedure, in Conjunction with Dual Decomposition [NAACL 2015]
- Approach 2: A simple algorithm MIML^{perf} that uses a novel plug-in technique significantly outperforming every other technique while also giving order of magnitude run-time reductions [AAAI 2017, Ongoing work for Extreme Classification]

Scalebale Plugin Based Performance Classifier

Algorithm 1 MIML^{perf}: Training Routine

Input: Data $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$, expression rate κ, perf. measure Δ

For all (i, j) such that $y_{i,j} = 1$, set random $\kappa \cdot n_i$ entries of the vector $\mathbf{z}^{(i,j)}$ to 1 *// Initialize hidden labels randomly*

while not converged **do**

Step 1: Fix hidden variables, update plug-in classifiers

  for every label $j \in [L]$ **do**

    $\mathcal{D}^j \leftarrow \{\{(\mathbf{x}_i^{(k)}, z_k^{(i,j)})\}_{k=1}^{n_i}\}_{i=1}^{n}$ *// Prepare datasets*

    $g^j \leftarrow$ CPE-train$(\mathcal{D}^j)$ *// Train CPE models*

  end for

  $\boldsymbol{\eta} \leftarrow$ Tune-thresholds$((\mathbf{X}, \mathbf{Y}), \mathbf{p}; \Delta)$ *// Optimize* Δ

Step 2: Fix plug-in classifiers, update hidden variables

  for $(i, j) \in [N] \times [L]$ such that $y_{i,j} = 1$ **do**

    $\mathbf{z}^{(i,j)} \leftarrow \mathbf{0}^{n_i}$ *// Reset hidden labels*

    $c^{(i,j)} \leftarrow \sum_{k=1}^{n_i} \mathbb{I}\{g^j(\mathbf{x}_i^{(k)}) \geq \eta_j\}$

    $S^{(i,j)} \leftarrow$ Sample $c^{(i,j)}$ entries of $\mathbf{z}^{(i,j)}$ according to g^j

    $z_k^{(i,j)} \leftarrow 1$ for all $k \in S^{(i,j)}$ *// Reestimate hidden labels*

  end for

end while

Theoretical Analysis: Generalization Guarantee

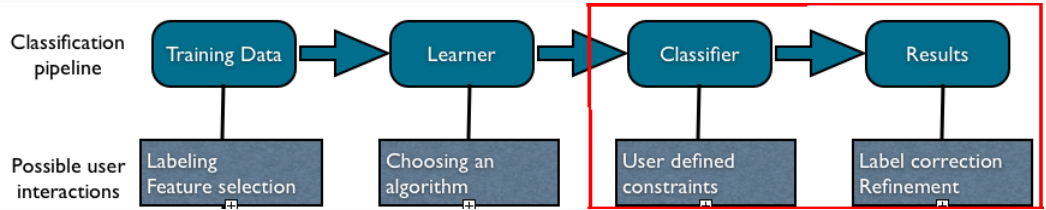
Let the training set $(\mathbf{X}, \mathbf{Y})$ be chosen randomly from some fixed but unknown distribution and let $(\mathbf{X}^t, \mathbf{Y}^t)$ denote a random test set drawn from the same distribution. Let π denote the minimum frequency of any label.

Let the instances be represented as d dimensional features $\mathbf{x}_i^{(k)} \in \Re^d$. Then for any N such that $\sqrt{\frac{1}{N}\left(\log\frac{12}{\delta} + d\log\frac{2eN}{d}\right)} < \frac{\beta\pi}{2(1+\beta)}$, we have, with probability at least $1-\delta$, for some constant C

$$\left|\mathbf{E}\left(\beta\left(\hat{\mathbf{f}}; \mathbf{X}^t, \mathbf{Y}^t\right)\right) - \beta\left(\hat{\mathbf{f}}; \mathbf{X}, \mathbf{Y}\right)\right| \leq C\sqrt{\frac{1}{N}\left(d + \log\frac{1}{\delta}\right)}$$

The result is stated for linear models for sake of simplicity and can be extended to hypothesis spaces with finite capacity as well. A similar result holds true for F-micro measure.

Example 4: Incorporating Domain Knowledge and Non-Decompos Losses into Sequence to Sequence Models



- Illustration on OCR Correction
- Illustration on Automated Question Generation
- Illustration on Audio Set Classification

Automatic question and answer generation

Sachin Ramesh Tendulkar is a former Indian cricketer and captain, widely regarded as one of the greatest batsmen of all time. He took up cricket at the age of eleven, made his Test debut on 15 November 1989 against Pakistan in Karachi at the age of sixteen, and went on to represent Mumbai domestically and India internationally for close to twenty-four years.....

Automatic question and answer generation

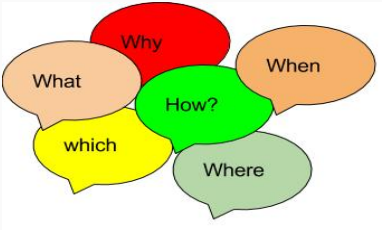
Sachin Ramesh Tendulkar is a former Indian cricketer and captain, widely regarded as one of the greatest batsmen of all time. He took up cricket at the age of eleven, made his Test debut on 15 November 1989 against Pakistan in Karachi at the age of sixteen, and went on to represent Mumbai domestically and India internationally for close to twenty-four years.....

How would someone tell that you have read this text?

Automatic question and answer generation

Sachin Ramesh Tendulkar is a former Indian cricketer and captain, widely regarded as one of the greatest batsmen of all time. He took up cricket at the age of eleven, made his Test debut on 15 November 1989 against Pakistan in Karachi at the age of sixteen, and went on to represent Mumbai domestically and India internationally for close to twenty-four years.....

How would someone tell that you have read this text?



Automatic question and answer generation

A system to automatically generate questions and answers from text.

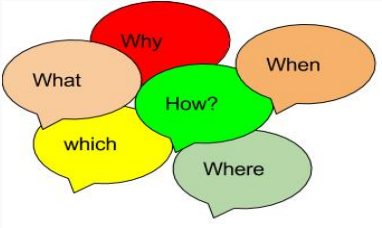
Some text

Sachin Tendulkar received the Arjuna Award in 1994 for his outstanding sporting achievement, the Rajiv Gandhi Khel Ratna award in 1997...

Questions

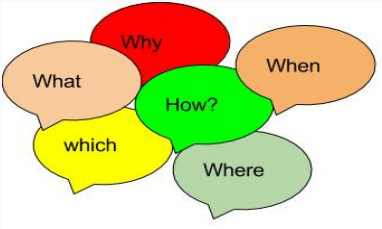
1. When did Sachin Tendulkar received the Arjuna Award?
   Ans: 1994
2. which award did sachin tendular received in 1994 for his outstanding sporting achievement?
   Ans: Arjuna Award
3. when did Sachin tendulkar received the Rajiv Gandhi Khel Ratna Award?
   Ans: 1997

Why is this problem Challenging?



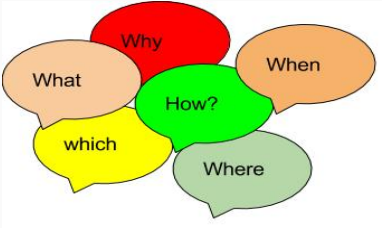
- **Question Must be Relevant to the Text**

Why is this problem Challenging?



- **Question Must be Relevant to the Text**
- **Answer Must be Unambiguous**

Why is this problem Challenging?



- **Question Must be Relevant to the Text**
- **Answer Must be Unambiguous**
- **Question must be challenging and well formed**

Our initial Contribution [PAKDD 2018]

Sentence: It was adopted into an Oscar-winning film in 1962 by director Robert Mulligan, with a screenplay by Horton Foote.
Feature Tagged Sentence: It|PRP|O|nsubjpass was|VBD|O|auxpass adapted|VBN|O|ROOT into|IN|O|case
.... Horton|N|N|O|Compound Foote|N|N|O|Compound

With Features:	Who was the director of the film ?
With Features and Answer as "Horton Foote":	Who wrote the movie ?
With Features and Answer as "Robert Mulligan":	Who was the director of the Oscar-winning movie ?

Our initial Contribution [PAKDD 2018]

Sentence: It was adopted into an Oscar-winning film in 1962 by director Robert Mulligan, with a screenplay by Horton Foote.
Feature Tagged Sentence: It|PRP|O|nsubjpass was|VBD|O|auxpass adapted|VBN|O|ROOT into|IN|O|case
.... Horton|N|N|O|Compound Foote|N|N|O|Compound

With Features:	Who was the director of the film ?
With Features and Answer as "Horton Foote":	Who wrote the movie ?
With Features and Answer as "Robert Mulligan":	Who was the director of the Oscar-winning movie ?

- Pointer network based method for automatic answer selection.

Our initial Contribution [PAKDD 2018]

Sentence: It was adopted into an Oscar-winning film in 1962 by director Robert Mulligan, with a screenplay by Horton Foote.
Feature Tagged Sentence: It|PRP|O|nsubjpass was|VBD|O|auxpass adapted|VBN|O|ROOT into|IN|O|case
.... Horton|N|N|O|Compound Foote|N|N|O|Compound

With Features:	Who was the director of the film ?
With Features and Answer as "Horton Foote":	Who wrote the movie ?
With Features and Answer as "Robert Mulligan":	Who was the director of the Oscar-winning movie ?

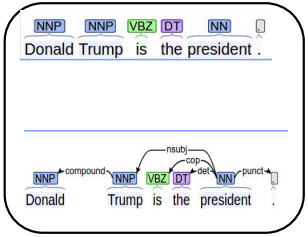
- Pointer network based method for automatic answer selection.
- Sequence to sequence model with attention and augmented with rich set of linguistic features and answer encoding

Sentence:

Donald Trump is the President.

Question:

Who is Donald Trump ?



POS Tag and Dependency Label

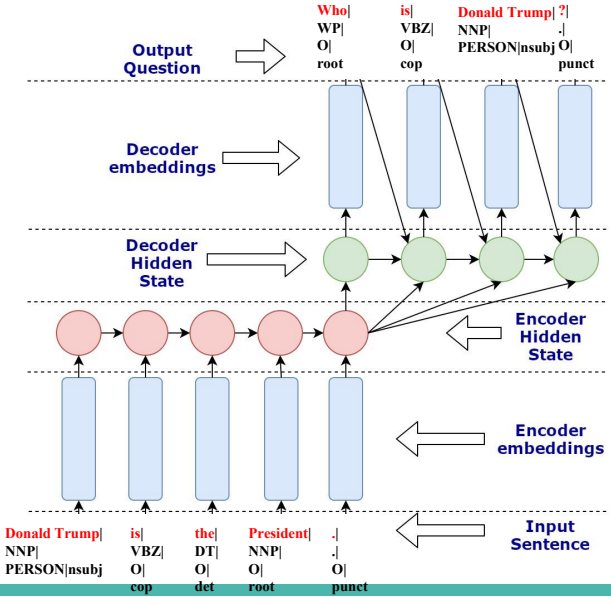
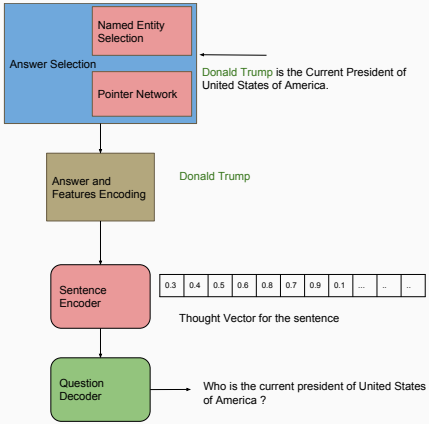
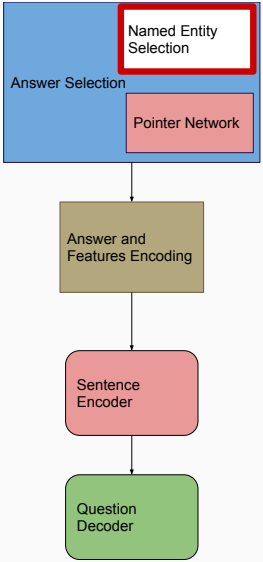


Figure 10: Question generation

The High level Architecture



Named Entity Selection

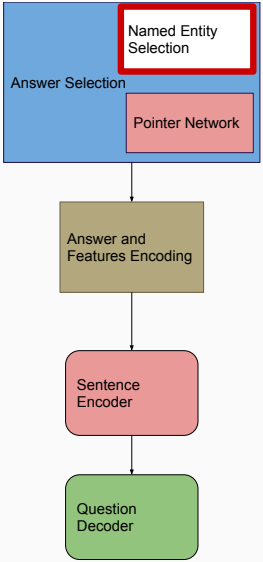


- Sentence $S = (w_1, w_2, ..., w_n)$ is encoded using a 2-layer LSTM network into hidden states $H = (h_1^s, h_2^s, ..., h_n^s)$.

---

h_n^s is final state
h_{mean}^s is the mean of all activations
h_{mean}^{ne} is mean of activations in NE span $(h_i^s, ..., h_j^s)$
[a]Most relevant answer to ask question about

Named Entity Selection

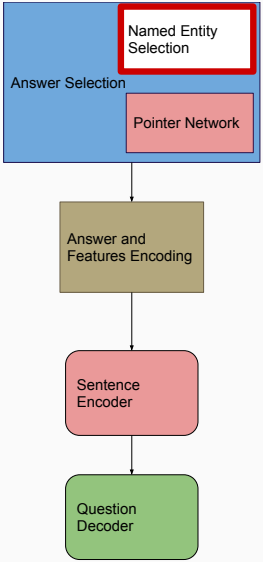


- Sentence $S = (w_1, w_2, ..., w_n)$ is encoded using a 2-layer LSTM network into hidden states $H = (h_1^s, h_2^s, ..., h_n^s)$.
- For each NE, $NE = (n_i, ..., n_j)$, create representation (**R**) $=< h_{mean}^{ne} >$,

---

h_n^s is final state
h_{mean}^s is the mean of all activations
h_{mean}^{ne} is mean of activations in NE span $(h_i^s, ..., h_j^s)$
[a]Most relevant answer to ask question about

Named Entity Selection

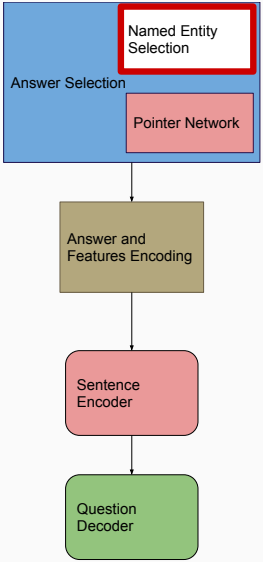


- Sentence $S = (w_1, w_2, ..., w_n)$ is encoded using a 2-layer LSTM network into hidden states $H = (h_1^s, h_2^s, ..., h_n^s)$.
- For each NE, $NE = (n_i, ..., n_j)$, create representation (**R**) $=< h_{mean}^{ne} >$,
- **R** is fed to MLP along with $< h_n^s; h_{mean}^s; >$ to get probability of named entity being a pivotal answer[a].

---

h_n^s is final state
h_{mean}^s is the mean of all activations
h_{mean}^{ne} is mean of activations in NE span $(h_i^s, ..., h_j^s)$
[a]Most relevant answer to ask question about

Named Entity Selection

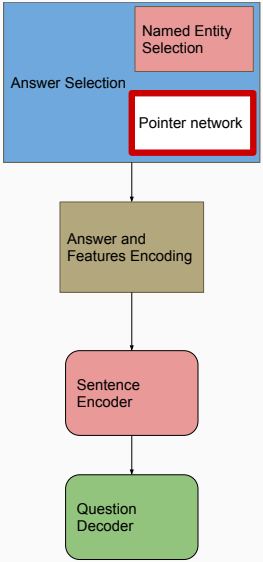


- Sentence $S = (w_1, w_2, ..., w_n)$ is encoded using a 2-layer LSTM network into hidden states $H = (h_1^s, h_2^s, ..., h_n^s)$.
- For each NE, $NE = (n_i, ..., n_j)$, create representation (**R**) $=< h_{mean}^{ne} >$,
- **R** is fed to MLP along with $< h_n^s; h_{mean}^s; >$ to get probability of named entity being a pivotal answer[a].
- $P(NE_i|S) = softmax(\mathbf{R}_i.W + B)$

---

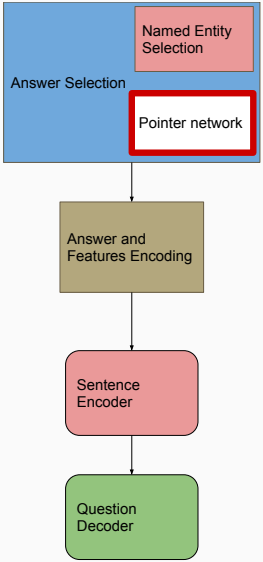
h_n^s is final state
h_{mean}^s is the mean of all activations
h_{mean}^{ne} is mean of activations in NE span $(h_i^s, ..., h_j^s)$
[a]Most relevant answer to ask question about

Answer selection using Pointer networks



- Given encoder hidden states $H = (h_1, h_2, \ldots, h_n)$, the probability of generating $O = (o_1, o_2, \ldots, o_m)$ is : $P(O|S) = \prod P(o_i|o_1, o_2, o_3, \ldots, o_{i-1}; H)$

Answer selection using Pointer networks



- Given encoder hidden states $H = (h_1, h_2, \ldots, h_n)$, the probability of generating $O = (o_1, o_2, \ldots, o_m)$ is : $P(O|S) = \prod P(o_i|o_1, o_2, o_3, \ldots, o_{i-1}; H)$
- Probability distribution is modeled as:

$$u^i = v^T tanh(W^e \hat{H} + W^d D_i) \quad (23)$$

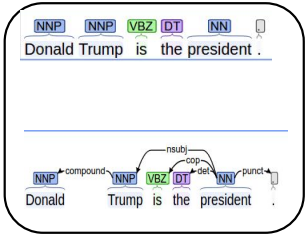
$$P(O|S) = softmax(u^i) \quad (24)$$

Sentence:

Donald Trump is the President.

Question:

Who is Donald Trump ?



POS Tag and Dependency Label

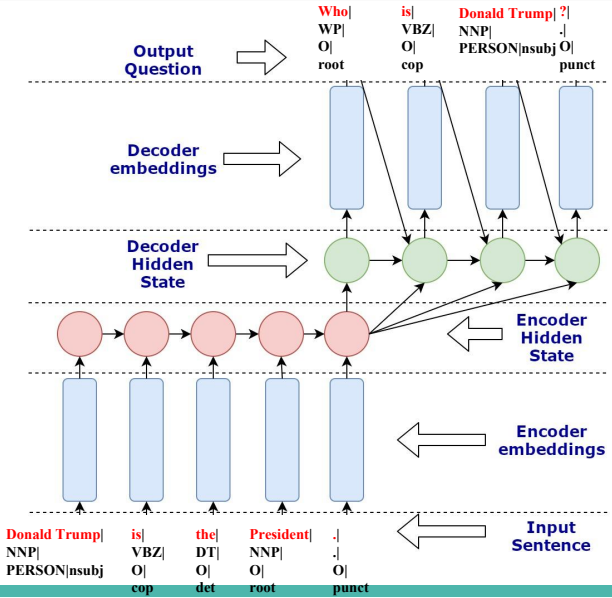
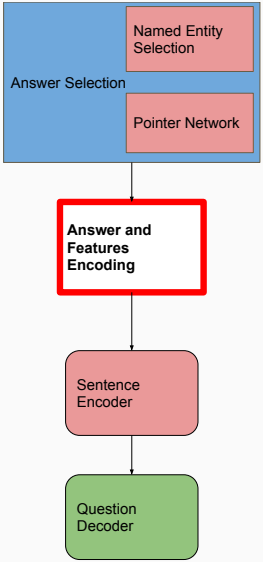


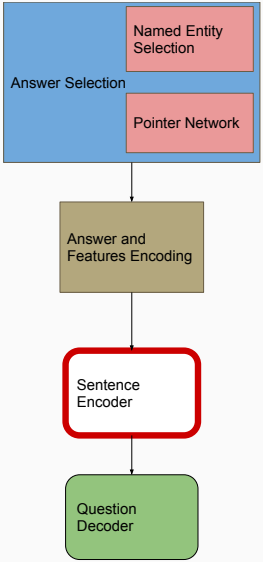
Figure 11: Question generation

Features and Answer Encoding



- POS tag, Named Entity tag, Dependency label as linguistic features.
- Rich set of linguistic features help model learn better generalize transformation rules.
- Dependency label is the edge label connecting each word with the parent in the dependency tree.

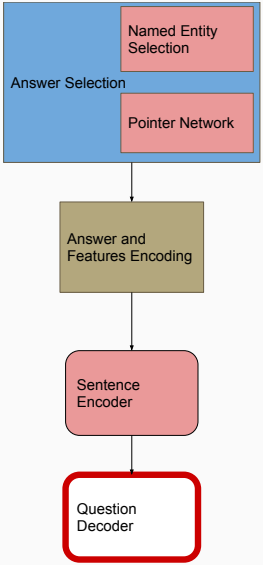
Sentence Encoder



- BiLSTM to capture both left context and the right context.
- $$\overrightarrow{\hat{h}_t} = f(\overrightarrow{W} w_t + \overrightarrow{V}\overrightarrow{\hat{h}_{t-1}} + \overrightarrow{b}), \overleftarrow{\hat{h}_t} = f(\overleftarrow{W} w_t + \overleftarrow{V}\overleftarrow{\hat{h}_{t+1}} + \overleftarrow{b}) \quad (25)$$
- $$\hat{h}_t = g(Uh_t + c) = g(U[\overrightarrow{\hat{h}_t}, \overleftarrow{\hat{h}_t}] + c) \quad (26)$$

$\hat{h}_t$ is the thought vector $W, V,$ *and* $U \in R^{n \times m}$ are trainable parameters, $w_t \in R^{p \times q \times r}$ is feature encoded word embedding at time step t.

Question Decoder



- 2-layer LSTM network.
- Decoder:

$$P(Q|S;\theta) = softmax(W_s(tanh(W_r[h_t, c_t] + b)) \quad (27)$$

- Beam search with beam_size 3 to decode question.
- Suitably modified decoder integrated with an attention mechanism to handle rare word problem.

where W_s and W_r are weight vectors and *tanh* is the activation function.

Attention Mechanism

Attention distribution:

$$e_i^t = v^t tanh(W_{eh}h_i + W_{sh}s_t + b_{att}) \tag{28}$$

$$a^t = softmax(e^t) \tag{29}$$

$$c_t^* = \sum_i a_i^t h_i \tag{30}$$

Probability distribution over vocabulary is:

$$P_{vocab} = sofmax(W_v[s_t, c_t^*] + b_v) \tag{31}$$

Overall loss is calculated as:

$$LOSS = \frac{1}{T}\sum_{t=0}^{T} -\log P_{vocab}(word_t) \tag{32}$$

Human evaluation results

System	p1(%)	p2(%)	p3(%)
QG [Xinya 2017]	51.6	48	52.3
QG+F	59.6	57	64.6
QG+F+NE	57	52.6	67
QG+GAE	44	35.3	50.6
QG+F+AES	51	47.3	55.3
QG+F+AEB	61	60.6	**71.3**
QG+F+GAE	**63**	**61**	67

Table 5: Human evaluation results on $\mathscr{S}^{te}$. Parameters are, **p1**: percentage of syntactically correct questions, **p2**: percentage of semantically correct questions, **p3**: percentage of relevant questions.

f

F	Features
NE	Named entity selection
AES	Sequence pointer network
AEB	Boundary pointer network
GAE	Ground truth answer encoding

Automatic evaluation results

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L
QG [Xinya 2017]	39.97	22.39	14.39	9.64	14.34	37.04
QG+F	41.89	24.37	15.92	10.74	15.854	37.762
QG+F+NE	41.54	23.77	15.32	10.24	15.906	36.465
QG+GAE	43.35	24.06	14.85	9.40	15.65	37.84
QG+F+AES	**43.54**	**25.69**	**17.07**	**11.83**	**16.71**	**38.22**
QG+F+AEB	**42.98**	**25.65**	**17.19**	**12.07**	**16.72**	**38.50**
QG+F+GAE	**46.32**	**28.81**	**19.67**	**13.85**	**18.51**	**41.75**

Some sample questions generated

Sentence 1: The museum was founded by the nurse and explorer Kate Marsden and Reverend J.C. Thompson FGS.
Answer Predicted: Nurse and explorer Kate Marsden
Question Generated: who founded the museum ?

Sentence 2: American idol premiered in June 2002 and became the surprise summer hit show of 2002.
Answer Predicted: June 2002
Question Generated: When did American idol begin ?

Sentence 3: Shuman then constructed a full-scale steam engine powered by low-pressure water, enabling him to patent the entire solar engine system by 1912.
Answer Predicted: 1912
Question Generated: When was the solar engine system invented ?

Sentence 4: Journalist Vinod Verma arrested for alleged extortion bid on Chhattisgarh minister .
Answer Predicted: alleged extortion bid on chhattisgarh minister
Question Generated: what did journalist vinod arrested for ?

Sentence 5: Donald Trump is the current president of the United States .
Answer Predicted: Donald Trump
Question Generated: who is the current president of the United States ?

Sentence 6: Manhattan was on track to have an estimated 90,000 hotel rooms at the end of 2014, a 10% increase from 2013.
Answer Predicted: 90000
Question Generated: How many hotel rooms did Manhattan have ?

Need for more representative loss function

- Typically models optimize cross entropy loss while training.

Need for more representative loss function

- Typically models optimize cross entropy loss while training.
- BLEU metric is used for evaluation.

Need for more representative loss function

- Typically models optimize cross entropy loss while training.
- BLEU metric is used for evaluation.
- Need for a mechanism to deal with relatively rare word.

Need for more representative loss function

- Typically models optimize cross entropy loss while training.
- BLEU metric is used for evaluation.
- Need for a mechanism to deal with relatively rare word.
- Need to handle word repetitions problem while decoding questions.

Incorporating copy and coverage mechanism into the decoder

Copy

Text: Filipe Nyusi is the current president of Mozambique.

Question: Who is the current president of USA?

Copy is modeled by modifying the decoding probability of word w as a convex combination (mixture) of its being copied and of being emitted from vocabulary.

Coverage

Text: the royal canadian navy, headed by the commander of the royal canadian navy includes 33 warships and submarines deployed in two fleets.

Question: The royal Canadian navy navy is headed by whom?

Coverage is modeled by modifying the attention based on an additional vector of words already covered.

Adding copy mechanism

- We calculate $p_{cg} \in [0, 1]$ as:

$$p_{cg} = sigmoid(W_{eh}^T c_t^* + w_{sh}^T s_t + W_x x_t + b_{cg}) \tag{33}$$

- Final decoding prob. of a word is given by mixture model

$$p^*(word) = p_{cg} \sum_{w_i = word} a_i^t + (1 - p_{cg}) P_{vocab}(word) \tag{34}$$

Adding coverage mechanism

- We maintain word coverage vector(wcv) of words already decoded.

$$wcv^t = \sum_{t'=0}^{t-1} a^{t'} \tag{35}$$

- Typical attention:

$$e_i^t = v^t tanh(W_{eh} h_i + W_{sh} s_t + b_{att}) \tag{36}$$

- New attention:

$$e_i^t = v^t tanh(W_{wcv} wcv_i^t + W_{eh} h_i + W_{sh} s_t + b_{att}) \tag{37}$$

- Final loss becomes:

$$Loss_{final} = \frac{1}{T} \sum_{t=0}^{T} \log P^*(w_t) - \lambda_c \Sigma_i min(a_i^t, wcv_i^t) \tag{38}$$

Generator Evaluator Network for Question Generation

- Deep reinforcement learning based technique to directly BLEU, ROUGE etc.
- Generator is Seq2Seq model with copy and coverage mechanism.
- Evaluator evaluates the predicted sequence against the gold sequence and assigns a reward based on task specific metric.
- The generator is the *agent* and the generation of next word is an *action*.
- The (probability of) decoding of a word $P_\theta(word)$ is a (stochastic) *policy*.

Evaluator

- Evaluates the generated question against the gold question and returns a value(reward).
- Four metrics as evaluators 1) BLEU [Papineni 2002] 2) GLEU[1] 3) ROUGE-L 4) Neural Sentence matcher.
- Neural sentence matcher is based on decomposable attention model[Parikh 2016]
- Using BLEU scorer as evaluator we got 2 BLEU point improvement over our previous result.

Architecture of reinforcement learning framework

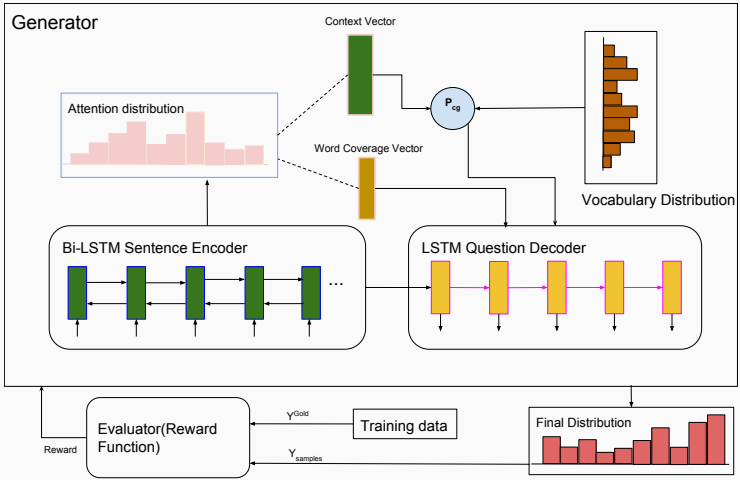


Figure 12: Architecture of reinforcement learning based framework for question generation

Automatic evaluation Results

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L
L2A	43.21	24.77	15.93	10.60	16.39	38.98
AutoQG	44.68	26.96	18.18	12.68	17.86	40.59
RL_{BLEU}	46.84	29.38	20.33	14.47	19.08	41.07
RL_{DAS}	44.64	28.25	19.63	14.07	18.12	42.07
RL_{GLEU}	45.20	29.22	20.79	15.26	18.98	43.47
RL_{ROUGE}	**47.01**	**30.67**	**21.95**	**16.17**	**19.85**	**43.90**

Table 6: Experimental results on the test set on automatic evaluation metrics. Best results for each metric (column) are **in bold**.

Human evaluation result

Model	Syntactically correct		Semantically correct		Relevant	
	Score	Kappa	Score	Kappa	Score	Kappa
L2A	39.2	0.49	39	0.49	29	0.40
AutoQG	51.5	0.49	48	0.78	48	0.50
RL_{BLEU}	47.5	0.52	49	0.45	41.5	0.44
RL_{DAS}	68	0.40	63	0.33	41	0.40
RL_{GLEU}	60.5	0.50	62	0.52	44	0.41
RL_{ROUGE}	**69.5**	0.56	**68**	0.58	**53**	0.43

Table 7: Human evaluation results (column “Score”) as well as inter-rater agreement (column “Kappa”) for each model on the test set. The scores are between 0-100, 0 being the worst and 100 being the best. Best results for each metric (column) are **in bold**.

THANK YOU

How Effective is IMB?

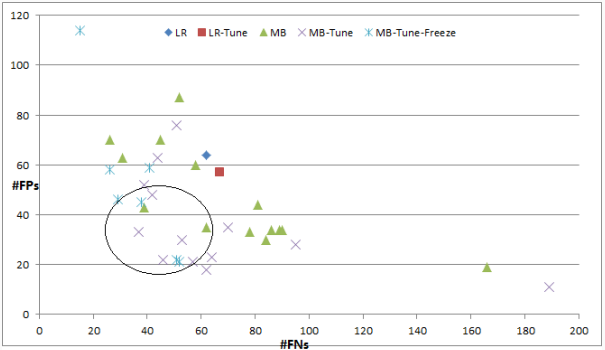


Figure 13: FP vs FN on the UCI Spam dataset using various classifiers

The models were tuned to minimize FPs, and thus, the models in the lower half of the plot are more desirable and the ones in the lower left quadrant achieve a better overall accuracy.

Effect of Base Learner and Boosting

- choice of base learner did not significantly affect the overall model performance
- MB, due to its non-linear, branching program-based structure performed better than AdaBoost

Dataset	RBF	LR	RBF-Tune[2]	LR-Tune	Ada-RBF	Ada-LR	MB-RBF	MB-LR
Ionosphere	91.03	90.71	92.55	93.43	92.96	94.97	95.58	**96.42**
Sonar	85.67	86.75	86.9	91.02	87.72	89.87	90.38	**91.08**

Table 8: Effect of changing the base learner and boosting algorithms

Comparison with Other Methods - 1

Dataset	Test Acc %	LR	LR-Tune	Ada	Ada-Tune	MB	MB-Tune(I-ROOT)	MB-Tune-Freeze
Spambase	Acc+	91.7	91.9	86.5	93.6	95.1	91.2 (±3.6)	89.8 (±.4)
	Acc-*	93.3	93.7	94.2	94.4	97.3	**98** (±.2)	97.8 (±.1)
	Acc	92.9	93	91.1	93.9	96.4	95.3 (±1.5)	94.6 (±.2)
Ionosphere	Acc+	81.8	85.7	86.9	88.7	91	92.8 (±7.2)	92.8 (±7.2)
	Acc-*	94.8	96.8	95.3	94.4	97.7	**99.3** (±2.8)	99.3 (±2.8)
	Acc	90.7	93.4	94.9	93.7	96.4	**97.1** (±4.5)	97.1 (±4.5)
Sonar	Acc+	88.4	93.2	87.7	91.4	89.4	92.8 (±3.2)	92.8 (±3.2)
	Acc-*	82.5	88.2	91.8	90.6	92.2	**95.8** (±.6)	95.8 (±.6)
	Acc	86.7	91	89.8	91	91.1	**94.4** (±1.1)	94.4 (±1.1)
Liver	Acc+*	81.1	81.2	84.0	83.4	85.8	**87.7** (±2.9)	82.7 (±3.6)
	Acc-	59.2	59.1	57.3	64.2	71.8	74.2 (±7.0)	73.7 (±6.7)
	Acc	76.8	77	73.5	82.5	81.6	**82.6** (±4.0)	79 (±4.1)

Table 9: Comparison of interactive martingale boosting with other methods on UCI datasets.

Comparison with Other Methods - 2

I-ROOT

- Outperforms other approaches on all datasets;
- Its Accuracy on the favored class generally higher than that achieved by MB
  - Might come at the cost of reduced accuracy on other classes;
- Early freezing $\Rightarrow$ Results only slightly worse, but with the advantage of significant reduction in the number of nodes in the DAG.
  - On Spambase dataset, 70 of the total 120 nodes got frozen.

Effect of Number of Levels on Model Accuracy

- Varied number of MB levels $L = 2$ to 15;
- model accuracy on the tuned class steadily rises with the number of levels and flattens at high values of L

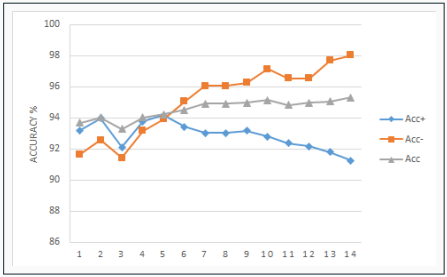


Figure 14: Effect of varying L on model accuracy evaluated on UCI Spambase

We set $L = 15$ in our experiments.

Running Time Analysis

	MB(ms)	MB-Tune(ms)	MB-Tune-Freeze(ms)
Spambase	1009	29394	27880
Ionosphere	288	1050	1030
Sonar	230	623	422
Liver	496	949	936

Table 10: Comparison of training time.

- Running time of IMB could be improved by (1) grouping of features to limit the number of hyperparameters; (2) multi-threaded implementation;
- MB-Tune-Freeze takes lesser time to train due to early freezing of nodes, with marginal impact on accuracy.

Comparison with Skyline: Granularity of Interaction

	Granularity (#Nodes tuned)	Acc+	Acc-	Acc
Spambase	I-ROOT	93.80	93.27	93.48
	I-ALL (6)	93.12	**94.37**	**93.86**
	I-SELECT (3)	92.86	93.97	93.22
Ionosphere	I-ROOT	88	96.67	93.57
	I-ALL (6)	90.71	**97.72**	**95.41**
	I-SELECT (2)	90.71	96.84	94.66

Table 11: Effect of granularity of user interaction evaluated on UCI Spambase but with MB restricted to 4 levels.

- I-ALL benefits from the interactive refinement at all nodes and does better in meeting user preferences;
- I-SELECT reduces the number of nodes requiring manual tuning with marginal impact

Representative Results

Dataset	Kernel	F-Macro			F-Micro			Avg. Precision		
		MIMLSVM	M3MIML	MIML^{perf}	MIMLSVM	M3MIML	MIML^{perf}	MIMLSVM	M3MIML	MIML^{perf}
Scene	lin	0.4292	0.386	**0.5427**	0.4475	0.3868	**0.528755**	0.6635	0.5754	**0.746562**
	rbf	0.6054	0.5872	**0.6201**	0.6016	0.5796	**0.615842**	0.7704	0.74857	**0.791984**
Reuters	lin	0.8274	0.7601	**0.8492**	0.8395	0.8067	**0.862955**	0.9489	0.9463	**0.9625**
	rbf	0.88387	0.69979	**0.8901**	0.8689	0.7866	**0.89422**	0.9467	0.9403	**0.970646**

	F-macro	Precision	Recall	F-micro	Training time
MM-F$_\beta$	0.1366	0.6599	0.6521	0.6559	4h 20m
MIML^{perf}	**0.228354**	0.792812	0.578109	**0.668648**	1m 49s

	F-macro	F-micro
Optimizing $_\beta$	**22.8354**	61.0615
Optimizing $_\beta$	18.984	**66.8648**

Motivation

Motivation: Example of domain adaptation in machine translation

In-domain parallel corpora is often scarce

- A statistical machine translation (SMT) model is trained on large amounts of cross-domain corpora;
- it fails to reliably translate an in-domain text **Koehn:2007:SMT**

In-domain bilingual lexicons are readily available

- a medical domain bilingual lexicon comprising technical and popular medical terminology;
- redundant phrases; e.g. "...*be given marketing authorisation*", appears 218 times in the EMEA medical corpus.

Domain Adaptation: deals with augmenting a cross-domain translation model to reliably translate an in-domain text **Ren:2009:MWE**; **Wu:2008:SPM**

Going beyond phrases

Certain phrases, which might themselves be infrequent, tend to have “consensus” when generalized to higher-level *patterns*.

Table 12: Examples of recurring patterns, sample snippets covered by them and their frequency (in brackets) for the EMEA corpus

PATTERN: in patients with ■CAT1■ (568)	contains ■CAT2■ mg of ■CAT3■ (91)
in patients with HIT type II	capsule contains 25 mg of lenalidomide
in patients with CNS metastases	tablet contains 300 mg of maraviroc
in patients with ESRD	syringe contains 100 mg of anakinra
in patients with normal and impaired renal function	tablet contains 2.3 mg of sucrose
in patients with previous history of pancreatitis	capsule contains 200 mg of pregabalin
in patients with cirrhosis of the liver	vial contains 10 mg of the active substance
	tablet contains 30 mg of aripiprazole

Pattern: n-grams of tokens, domain-specific categories or higher-level phrase classes (noun phrase, verb phrase etc.)

Challenges in pattern extraction for in-domain translation

In the absence of a parallel in-domain corpus, translation of extracted patterns requires manual effort, which poses two key challenges:

- **quality of a pattern:** syntactically well-formed patterns like “the CAT5 of treatment” might be easier for humans to translate than others like “CAT4 condition has”;
- **quality of the set of patterns:**
  - quality patterns could have instances that overlap in their spans in the corpus, not all of which need to be translated;
  - human translation effort might have a budget constraint and therefore, a compact set of patterns is desirable.

Diminishing returns

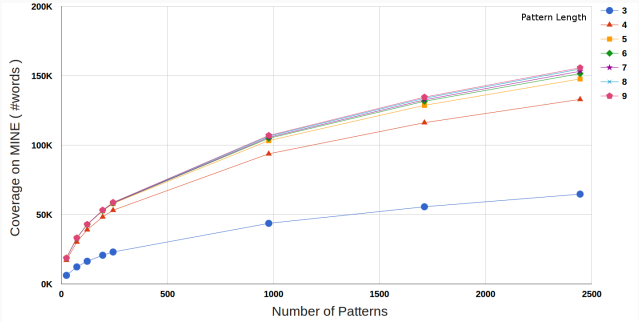


Figure 15: gain in coverage progressively diminishes with growth in the size of the subset

Diminishing returns (submodularity)

A set function $f(.)$ is said to be submodular if for any element v and sets $A \subseteq B \subseteq V \setminus \{v\}$, where V represents the ground set of elements, $f(A \cup \{v\}) - f(A) \geq f(B \cup \{v\}) - f(B)$.

It naturally models notions of coverage and diversity **Kempe:2003:KDD**; **Lin:2010:NAACL**

Our contributions

A framework to mine a set of quality patterns based on three key ideas:

1. Language of patterns: A pattern could either be lexical or a combination of words and higher-level categories.
2. Quality criteria for a pattern: based on simple (modular) aggregation of the instance costs.
3. Quality criteria for a set of patterns: either modular or based on element-wise non-decomposable submodular costs.

We incorporate these patterns along with their translations, as entries in a bilingual lexicon and study its effect on the translation accuracy for the domain adaptation of a baseline SMT model.

Problem of mining quality patterns

Corpus C:

SenId 1:	ACOMPLIA should not be used in patients with severe hepatic impairment
tok index:	0 1 2 3 4 5 6 7 8 9 10 11
SenId 2:	Actraphane is used in patients with diabetes
tok index:	0 1 2 3 4 5 6 7

Types

in form of	Name of type	Entries of List
Lexicon List (*UnDisambiguated in corpus C*)	**Drug**	*ACOMPLIA , Actraphane*
	Disease	*severe hepatic impairment, diabetes*
	T1 (Complex type)	*Patients with **Disease**, **Drug** for **Disease***
Span List (*Disambiguated in Corpus C*)	**NP**	<1,0,1>,<2,0,1><1,6,11>,<2,4,7>, <1,6,7>,<2,4,5>,<1,8,11>

Two notions of quality

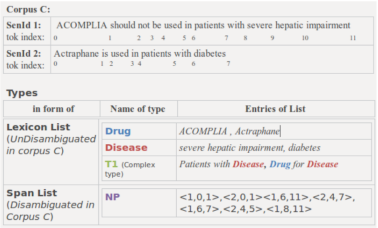
- quality of individual patterns;
- quality of a set of patterns

Problem definition

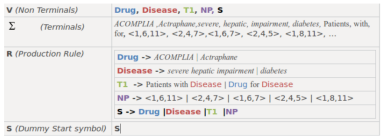
We are given a domain corpus $\mathscr{C}$ and optionally a set of "types"[3] $\mathscr{T}$. The problem of lexicon curation is to extract from $\mathscr{C}$, a set H of quality patterns, as per a quality function $Q_{\mathscr{C}}(h)$ for the quality of a pattern $h \in H$ in the corpus and a quality function $Q_{\mathscr{C}}(H)$ for the quality of the set H.

Our Proposed Solution Framework

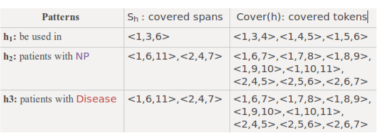
Components of the solution framework



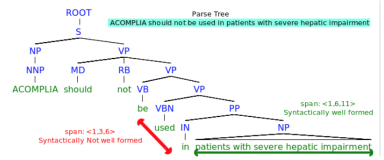
(a) Corpus and Types



(b) CFG Grammar G: (V, Σ, R, S)



(c) Patterns



(d) Syntactically well-formed span

Figure 16: Solution framework

Context Free Grammar G

The set V of non-terminals corresponds to the set of types $\mathscr{T}$, where, each type $T_i \in \mathscr{T}$ could be

- lexicon list (undisambiguated)
- spans (disambiguated) obtained as an output of an annotator; a span is a 3-tuple of sentence id, start and end token index within the sentence.

V *(Non Terminals)*	**Drug**, **Disease**, **T1**, **NP**, **S**
Σ *(Terminals)*	*ACOMPLIA ,Actraphane,severe, hepatic, impairment, diabetes*, Patients, with, for, <1,6,11>, <2,4,7>,<1,6,7>, <2,4,5>, <1,8,11>, …
R *(Production Rule)*	**Drug** -> *ACOMPLIA | Actraphane* **Disease** -> *severe hepatic impairment | diabetes* **T1** -> Patients with Disease | Drug for Disease **NP** -> <1,0,1> | <1,6,7> | <1,6,11> | <1,8,11> **NP** -> <2,0,1> | <2,4,5> | <2,4,7> **S** -> **Drug** |**Disease** |**T1** |**NP** |
S *(Dummy Start symbol)*	**S**

The set Σ of terminals then comprises:

- the set of lexical tokens in the entries of each type $T_i \in \mathscr{T}$;
- spans $< s, u, v >$ encoded as productions of the form $T_i \rightarrow < s_{i1}, u_{i1}, v_{i1} > | \ldots | < s_{ik}, u_{ik}, v_{ik} >$.

Pattern extractor

a program that uses the context free grammar G, to extract from $\mathscr{C}$, a set $\mathscr{H}$ of patterns;

Patterns	S_h : covered spans	Cover(h): covered tokens
h_1: be used in	<1,3,6>	<1,3,4>,<1,4,5>,<1,5,6>
h_2: patients with NP	<1,6,11>,<2,4,7>	<1,6,7>,<1,7,8>,<1,8,9>, <1,9,10>,<1,10,11>, <2,4,5>,<2,5,6>,<2,6,7>
h3: patients with Disease	<1,6,11>,<2,4,7>	<1,6,7>,<1,7,8>,<1,8,9>, <1,9,10>,<1,10,11>, <2,4,5>,<2,5,6>,<2,6,7>

- **Pattern:** a potential higher level domain type along with a set of spans S_h in the corpus from which it is extracted;
- For a span $\mu_i =< s_i, u_i, v_i > \in S_h$, we define $tokens(\mu_i) = \{< s_i, u_i, u_{i+1} >, \ldots, < s_i, v_{i-1}, v_i >\}$ as the set of all tokens covered by the span μ_i;
- **token coverage** of the pattern h is the set $cover(h) = \cup_{\mu \in S_h} tokens(\mu)$

Pattern quality criteria

Modular quality $Q_{\mathscr{C}}(h)$ of a pattern

- defined as a function $Q_{\mathscr{C}}(h) : \mathscr{H} \to [0, 1]$;
- the set $\mathscr{H}_Q = \{h \in \mathscr{H} | Q_{\mathscr{C}}(h) > r\}$ ($0 < r < 1$ is a quality threshold), is the set of all patterns in $\mathscr{H}$ that meet the quality criterion.

Quality $Q_{\mathscr{C}}(H)$ of a patterns set H

- defined as a function $Q_{\mathscr{C}}(H)$ from $2^{\mathscr{H}_Q} \to \mathbb{Z}$;
- either modular (e.g. $|H|$, $\sum_{h \in H} |cover(h)|$) or submodular (e.g. $|\cup_{h \in H} cover(h)|$).

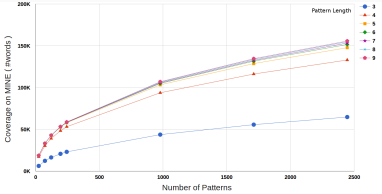


Figure 17: Coverage function

Examples of modular quality criteria for a pattern

1. Pattern consensus: number of spans covered by the pattern;
2. Informativeness: for a set $\mathscr{C}$ of corpora, $\frac{|\mathscr{C}|}{|\{\mathscr{C} \in \mathscr{C} : |S_h^{\mathscr{C}}| > 0\}|}$, where, $S_h^{\mathscr{C}}$ is the set of spans covered by h in corpus $\mathscr{C}$;
3. Syntactic well-formedness: spans covered by a pattern, such that, each forms a sub-tree in the parse of their sentence.
4. Lexical rule-based consensus: conform to a set of lexical rules; E.g., a pattern should not start or end with prepositions;
5. Semantic rule-based consensus: enforces a semantic constraint among the tokens in a span covered by the pattern. E.g., in the “mergers and acquisition” domain, we might constrain the semantic role of agents in the patterns to be either a *buyer* or a *seller*.

Patterns subset selection

$$H^* = \arg \max_{H \subseteq \mathscr{H}_Q} Q^2_{\mathscr{C}}(H) \text{ s.t. } Q^1_{\mathscr{C}}(H) < c \tag{39}$$

Or

$$H^* = \arg \min_{H \subseteq \mathscr{H}_Q} Q^1_{\mathscr{C}}(H) \text{ s.t. } Q^2_{\mathscr{C}}(H) > d \tag{40}$$

where, c and d are thresholds on the cost and the quality of H respectively.

This optimization has an efficient solution if the cost function $Q^1_{\mathscr{C}}(H)$ is modular and the quality function $Q^2_{\mathscr{C}}(H)$ is submodular (Iyer et. al. 2013).
Further, if $Q^2_{\mathscr{C}}(H)$ is a submodular and monotone function, then this problem can be solved greedily with theoretical guarantee of $1 - \frac{1}{\varepsilon}$ (Nemhauser et. al. 1978).

Summary and Conclusion

- Formulations and methods for identifying statistically optimal subsets from DAGs in Supervised feature selection and Partially supervised settings
  - Improved generalization via compact subset identification
  - **Bridged gap** between statistical and relational/rule learning communities
  - Investigated interesting feature classes
- Explored feature spaces of sizes $\sim 2^{50}$

Bibliography

References I

demo

Bibliography

- Francis Bach. High-dimensional non-linear variable selection through hierarchical kernel learning. CoRR, abs/0909.0844, 2009.
- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. J. Mach. Learn. Res., 3:993–1022, March 2003.
- Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology,
- Vitor Santos Costa, Ashwin Srinivasan, Rui Camacho, Hendrik Blockeel, Bart Demoen, Gerda Janssens, Wim Van Laer, James Cussens, and Alan Frisch.Query transformations for improving the efficiency of ilp systems. Journal of Machine Learning Research, 4:491, 2002.
- Krzysztof Dembczynski, Wojciech Kotlowski, and Roman Slowinski.Maximum likelihood rule ensembles. In ICML, pages 224–231, 2008.

- Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin.LIBLINEAR: A library for large linear classification. Journal of Machine Learning Research, 9:1871–1874, 2008.
- Peter A. Flach and Nicolas Lachiche.First-order bayesian classification with 1bc. 2000.
- Pratik Jawanpuria, Jagarlapudi Saketha Nath, and Ganesh Ramakrishnan.Efficient rule ensemble learning using hierarchical kernels. In ICML, pages 161–168, 2011.
- I. T. Jolliffe. Principal Component Analysis.Springer-Verlag, 1986.
- Balaji Krishnapuram, Lawrence Carin, Mario A. T. Figueiredo, and Alexander J. Hartemink.Sparse multinomial logistic regression: Fast algorithms and generalization bounds. IEEE Trans. Pattern Anal. Mach. Intell., 27(6):957–968, June 2005.

- K. Lang.20 newsgroups data set.
- Niels Landwehr, Bernd Gutmann, Ingo Thon, Luc De Raedt, and Matthai Philipose.Relational transformation-based tagging for activity recognition. Progress on Multi-Relational Data Mining, 89(1):111–129, 2009.
- Simon Lacoste-Julien, Fei Sha, and Michael I. Jordan.Disclda: Discriminative learning for dimensionality reduction and classification. In NIPS, pages 897–904, 2008.
- Haifeng Li, Tao Jiang, and Keshu Zhang. Efficient and robust feature extraction by maximum margin criterion. In In Advances in Neural Information Processing Systems 16, pages 157–165. MIT Press, 2003.
- John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira.Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

- Nada Lavrac, Filip Zelezny, and Peter A/ Flach. Rsd: Relational subgroup discovery through first-order feature construction. pages 149–165, July 2002.
- Eric McCreath and Arun Sharma.Lime: A system for learning relations. pages 336–374, 1998.
- Sriraam Natarajan, Hung H. Bui, Prasad Tadepalli, Kristian Kersting, and Weng keen Wong.Logical hierarchical hidden markov models for modeling user activities. In Proc. of ILP-08, 2008.
- Naveen Nair, Ganesh Ramakrishnan, and Shonali Krishnaswamy.Enhancing activity recognition in smart homes using feature induction. In Proceedings of the 13th international conference on Data warehousing and knowledge discovery, DaWaK'11, pages 406–418, Berlin, Heidelberg, 2011. Springer-Verlag.
- Naveen Nair, Amrita Saha, Ganesh Ramakrishnan, and Shonali Krishnaswamy.Rule Ensemble Learning In Structured Output Spaces. In Proceedings of American Assocition for Artificial Intelligence, AAAI'12, Toronto, Canada, 2012.
- L.R. Rabiner.A tutorial on hidden markov models and selected applications in speech

- Marie Szafranski, Yves Grandvalet, and Alain Rakotomamonjy.Composite kernel learning. Machine Learning, 79(1-2):73–103, 2010.
- Ashwin Srinivasan, Stephen Muggleton, Michael J. E. Sternberg, and Ross D. King.Theories for mutagenicity: A study in first-order and feature-based induction. Artif. Intell., 85(1-2):277–299, 1996.
- Ashwin Srinivasan and Ganesh Ramakrishnan.Parameter screening and optimisation for ilp using designed experiments. Journal of Machine Learning Research, 12:627–662, 2011.
- Ashwin Srinivasan. The aleph manual. 1999.
- Lucia Specia, Ashwin Srinivasan, Sachindra Joshi, Ganesh Ramakrishnan, and Maria das Graças Volpe Nunes.An investigation into feature construction to assist word sense disambiguation. Machine Learning, 76(1):109–136, 2009.

- Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun.Support vector machine learning for interdependent and structured output spaces. In Proceedings of the twenty-first international conference on Machine learning, ICML '04, pages 104–, New York, NY, USA, 2004. ACM.
- Emmanuel Tapia, Stephen Intille, and Kent Larson. Activity Recognition in the Home Using Simple and Ubiquitous Sensors Pervasive Computing. In Alois Ferscha and Friedemann Mattern, editors, Pervasive Computing, volume 3001 of Lecture Notes in Computer Science, chapter 10, pages 158–175. Springer Berlin / Heidelberg.
- Tim van Kasteren, Athanasios Noulas, Gwenn Englebienne, and Ben Kröse.Accurate activity recognition in a home setting. In Proceedings of the 10th international conference on Ubiquitous computing, UbiComp '08, pages 1–9, New York, NY, USA, 2008. ACM.
- Ian H. Witten, Eibe Frank, and Mark A. Hall.Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, Amsterdam, 3rd edition, 2011.

- Wanhong Xu.Supervising latent topic model for maximum-margin text classification and regression. In Mohammed Zaki, Jeffrey Yu, B. Ravindran, and Vikram Pudi, editors, Advances in Knowledge Discovery and Data Mining, volume 6118 of Lecture Notes in Computer Science, pages 403–414. Springer Berlin / Heidelberg.
- Jieping Ye and Shuiwang Ji.Discriminant analysis for dimensionality reduction: An overview of recent developments.
- Jun Zhu, Amr Ahmed, and Eric Xing.MedLDA: Maximum Margin Supervised Topic Models. Journal of Machine Learning Research, 1:1–48, 2010.
- Amrita Saha, Ashwin Srinivasan, and Ganesh Ramakrishnan. What kind of relational features are useful for Statistical learning?. Inducive Logic Programming, Dubrovnik, Croatia, 2012.
- Naveen Nair, Ajay Nagesh, and Ganesh Ramakrishnan. Markov Logic Chains. Inducive Logic Programming, Dubrovnik, Croatia, 2012.
- Ajay Nagesh, Ganesh Ramakrishnan, Laura Citicariu, Rajasekar Krishnamurthy,