Towards more Reliable Transfer Learning

Zirui Wang
Jaime Carbonell

Language Technologies Institute

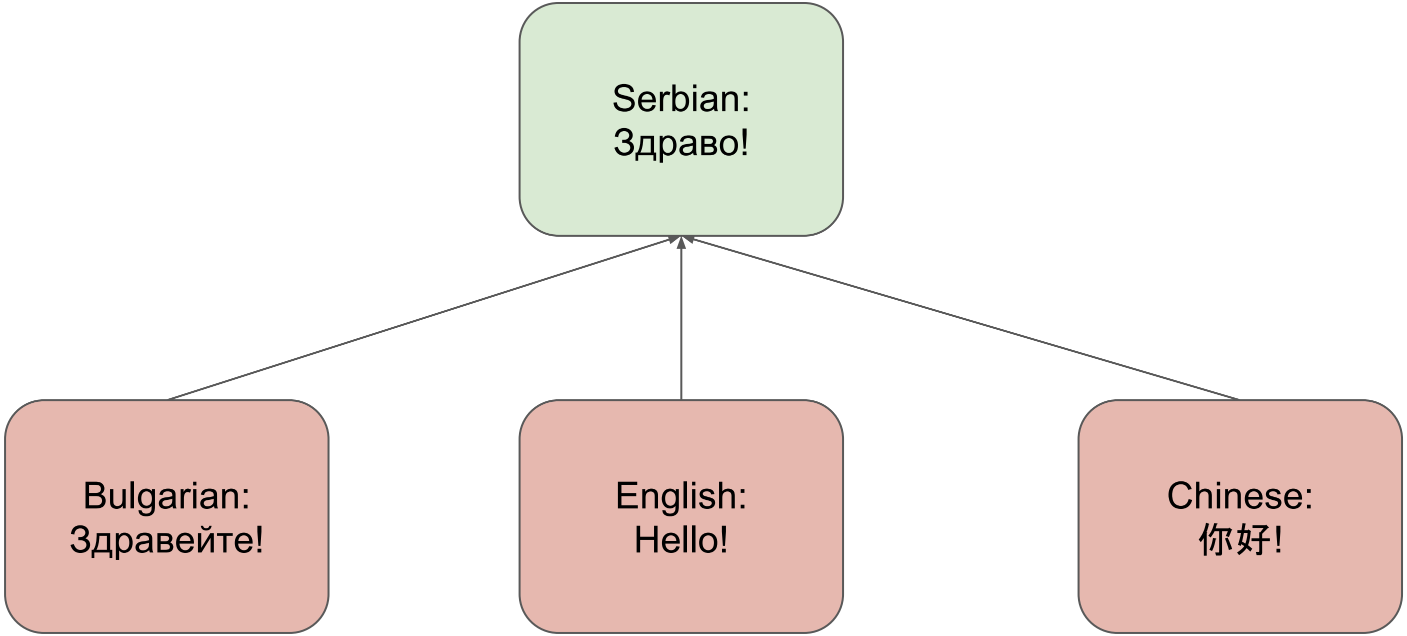
Carnegie Mellon University
School of Computer Science

Multi-Source Transfer Learning

1. Textual Task:
   a. Spam detection
   b. Sentiment analysis
   c. Cross-lingual document classification
2. Visual Task:
   a. Object recognition (e.g. Office31)
   b. Visual QA
3. Practical Task:
   a. Dicease diagnostics
   b. Urban computing

Not all sources are created equal

Low Resource Languages for Emergent Incidents (LORELEI)



Challenge: diverse proximity, diverse reliability

Two related tasks:

1. How to conduct transfer learning?
   - Peer-weighted multi-source transfer learning (PW-MSTL)
2. Active learning on sources
   - Adaptive multi-source active transfer (AMSAT)

1. Peer-weighted multi-source transfer learning (PW-MSTL)

Peer

Definition: peers of a source are other sources included in the task

How to utilize peer:

1. Use peers to help evaluate source reliability
2. Help a source to classify an instance when its confidence is too low

Algorithm 1 PW-MSTL

Input: $S = S^L \cup S^U$: source data; T: target data; μ: concentration factor; b_1: confidence tolerance; T: test data size;

for $k = 1, ..., K$ **do**
  Compute α^k by solving (6).
  Train a classifier $\hat{h}_k$ on the α^k weighted S_k^L.
end for

Compute δ and $\mathbf{R}$ as explained in Section 4.2.
Compute ω as (5).

for $t = 1, ..., T$ **do**
  Observe testing example $x^{(t)}$.
  for $k = 1, ..., K$ **do**
    if $|\hat{h}_k(x^{(t)}| < b_1$ **then**
      Compute $\hat{p}_k^{(t)} = \sum_{m \in [K], m \neq k} \mathbf{R}_{km} |\hat{h}_m(x^{(t)})|$.
    else
      Compute $\hat{p}_k^{(t)} = |\hat{h}_k(x^{(t)})|$.
    end if
  end for
  Predict $\hat{y}^{(t)} = sign(\sum_{k \in [K]} \omega_k \hat{p}_k^{(t)})$.
end for

Kernel Mean Match (KMM) for the kth source:

$$\min_{\alpha^k} \left\| \frac{1}{n_k^L + n_k^U} \sum_{i=1}^{n_k^L + n_k^U} \alpha_i^k \Phi(x_i^{S_k}) - \frac{1}{n_T} \sum_{i=1}^{n_T} \Phi(x_i^T) \right\|_H^2$$

Algorithm 1 PW-MSTL

Input: $S = S^L \cup S^U$: source data; T: target data; μ: concentration factor; b_1: confidence tolerance; T: test data size;
for $k = 1, ..., K$ **do**
    Compute α^k by solving (6).
    Train a classifier $\hat{h}_k$ on the α^k weighted S_k^L.
end for
Compute δ and $\mathbf{R}$ as explained in Section 4.2.
Compute ω as (5).
for $t = 1, ..., T$ **do**
    Observe testing example $x^{(t)}$.
    for $k = 1, ..., K$ **do**
        if $|\hat{h}_k(x^{(t)})| < b_1$ **then**
            Compute $\hat{p}_k^{(t)} = \sum_{m \in [K], m \neq k} \mathbf{R}_{km} |\hat{h}_m(x^{(t)})|$.
        else
            Compute $\hat{p}_k^{(t)} = |\hat{h}_k(x^{(t)})|$.
        end if
    end for
    Predict $\hat{y}^{(t)} = sign(\sum_{k \in [K]} \omega_k \hat{p}_k^{(t)})$.
end for

Compute inter-source relationship and source-target distances (we used MMD but any measurement should be fine):

$$\mathbf{R}_{i,j} = \begin{cases} \frac{\exp(\beta_1 \hat{\epsilon}_{S_i}(\hat{h}_j))}{\sum_{j' \in [K], j' \neq i} \exp(\beta_1 \hat{\epsilon}_{S_i}(\hat{h}'_j))}, & i \neq j \\ 0, & \text{otherwise} \end{cases}$$

$$\delta_i = \frac{exp(-\beta_2 MMD^{\rho}(S_i, T))}{\sum_k exp(-\beta_2 MMD^{\rho}(S_k, T))}$$

$$\mathrm{MMD}[\mathcal{F}, p, q] := \sup_{f \in \mathcal{F}} \left(\mathbf{E}_p[f(x)] - \mathbf{E}_q[f(y)] \right)$$

$$\mathrm{MMD}[\mathcal{F}, X, Y] := \sup_{f \in \mathcal{F}} \left(\frac{1}{m} \sum_{i=1}^m f(x_i) - \frac{1}{n} \sum_{i=1}^n f(y_i) \right)$$

Algorithm 1 PW-MSTL

```
Input: S = S^L ∪ S^U: source data; T: target data; μ: concentration factor; b_1: confidence tolerance; T: test data size;
for k = 1, ..., K do
    Compute α^k by solving (6).
    Train a classifier ĥ_k on the α^k weighted S_k^L.
end for
Compute δ and R as explained in Section 4.2.
Compute [ω] as (5).
for t = 1, ..., T do
    Observe testing example x^(t).
    for k = 1, ..., K do
        if |ĥ_k(x^(t))| < b_1 then
            Compute p̂_k^(t) = Σ_{m∈[K], m≠k} R_km |ĥ_m(x^(t))|.
        else
            Compute p̂_k^(t) = |ĥ_k(x^(t))|.
        end if
    end for
    Predict ŷ^(t) = sign(Σ_{k∈[K]} ω_k p̂_k^(t)).
end for
```

Source Importance Weight:

$$\omega = \delta \cdot [\underbrace{\boxed{\mu}}_{\text{concentration factor}}\mathbf{I_K} + (1-\mu)\mathbf{R}]$$

Algorithm 1 PW-MSTL

Input: $S = S^L \cup S^U$: source data; T: target data; μ: concentration factor; b_1: confidence tolerance; T: test data size;

for $k = 1, ..., K$ **do**

  Compute α^k by solving (6).

  Train a classifier $\hat{h}_k$ on the α^k weighted S^L_k.

end for

Compute δ and $\mathbf{R}$ as explained in Section 4.2.

Compute ω as (5).

for $t = 1, ..., T$ **do**

  Observe testing example $x^{(t)}$.

  for $k = 1, ..., K$ **do**

    if $|\hat{h}_k(x^{(t)}| < b_1$ **then**

      Compute $\hat{p}_k^{(t)} = \sum_{m \in [K], m \neq k} \mathbf{R}_{km} |\hat{h}_m(x^{(t)})|$.

    else

      Compute $\hat{p}_k^{(t)} = |\hat{h}_k(x^{(t)})|$.

    end if

  end for

  Predict $\hat{y}^{(t)} = sign(\sum_{k \in [K]} \omega_k \hat{p}_k^{(t)})$.

end for

- Classify testing instances by weighted vote.
- Allow peers to assist classify an instance if the confidence is too low.

Results

Table 1. Classification accuracy (%) on the target domain, given that source domains contain diverse {1%,5%,15%,30%} labeled data.

Method	Synthetic		Spam			Sentiment				
	case1	case2	user7	user8	user3	electronics	toys	music	apparel	dvd
KMM	82.7	88.8	92.0	91.8	89.7	77.6	77.4	71.0	78.3	72.4
KMM-A	87.3	91.4	92.0	92.0	91.8	74.6	76.3	70.3	75.8	72.4
A-SVM	70.8	89.4	84.5	87.8	86.8	70.8	73.7	67.7	73.6	62.6
DAM	75.8	91.0	83.8	85.4	86.8	71.3	73.7	68.0	75.1	62.5
PW-MSTL$_b$	85.5	90.8	91.5	92.6	90.3	78.0	78.7	70.7	79,5	73.2
PW-MSTL	**88.4**	**92.6**	**93.8**	**95.6**	**92.8**	**79.3**	**81.9**	**74.6**	**82.7**	**76.7**

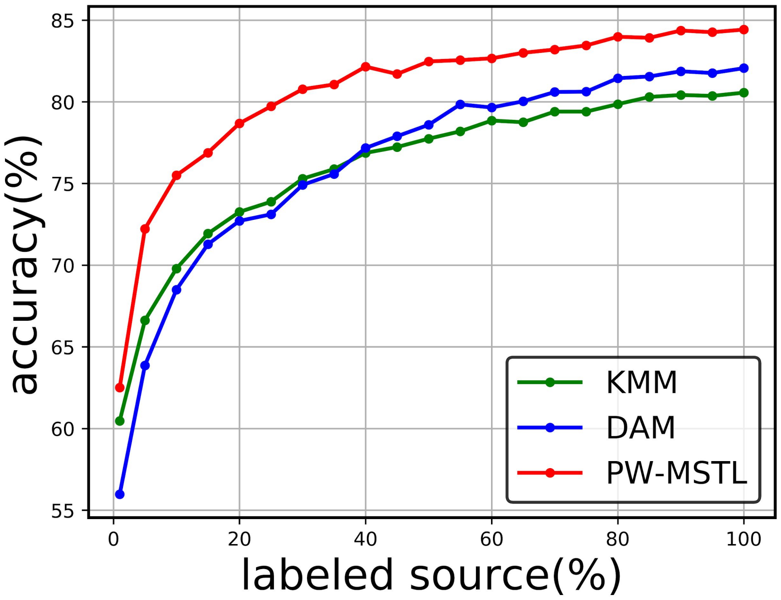
Results (continued)

Table 2. Classification accuracy (%) on the target domain, given that source domains contain the same fraction (%L) of labeled data.

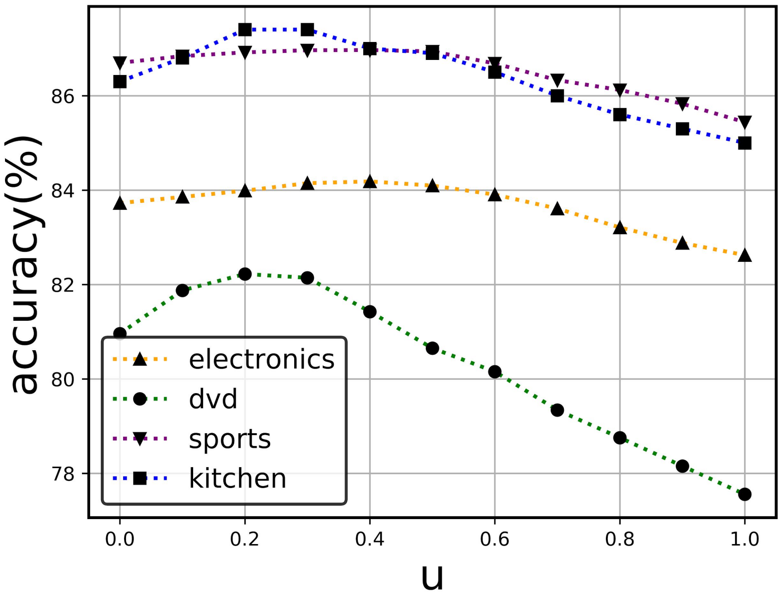
$\%_L$	Method	Synthetic	Spam			Sentiment				
			user7	user8	user3	electronics	toys	music	apparel	dvd
10%	KMM	87.0	89.1	91.2	90.3	75.0	74.6	68.3	75.6	70.2
	KMM-A	91.1	91.3	90.7	91.0	74.8	76.5	70.2	76.8	71.3
	A-SVM	89.4	88.4	91.9	89.2	77.1	78.1	69.9	78.2	68.9
	DAM	89.7	89.6	90.4	91.3	77.5	79.0	69.9	79.8	69.0
	PW-MSTL$_b$	90.2	89.7	92.4	92.1	77.7	78.7	69.7	78.9	73.5
	PW-MSTL	91.2	**92.5**	**94.9**	**93.1**	**79.8**	**81.5**	**73.3**	**81.3**	**76.4**
50%	KMM	95.6	92.6	94.0	91.8	81.6	81.7	75.0	82.2	76.9
	KMM-A	97.2	91.4	93.8	**94.7**	80.4	82.4	74.5	82.7	77.1
	A-SVM	96.4	91.5	95.2	93.4	81.7	83.4	74.7	84.3	76.0
	DAM	96.6	92.7	93.1	93.2	83.5	84.5	73.4	84.4	77.3
	PW-MSTL$_b$	96.6	92.9	95.2	93.5	83.6	84.7	74.4	85.0	80.4
	PW-MSTL	97.2	**94.5**	**95.7**	93.7	**84.8**	**86.4**	**76.9**	**87.2**	**82.0**

Results (continued)

Figure 1. (a) Incrementual accuracy on **dvd**



(b) Sensitivity analysis of concentration factor $\boldsymbol{\mu}$



2. Adaptive multi-source active transfer (AMSAT)

Two Questions

- Which source domain to pick?
- Which instance within selected domain to choose?

Algorithm 2 AMSAT

Input: $S = S^L \cup S^U$: source data; T: target data; μ: concentration factor; B: budget;

```
for k = 1, ..., K do
    Compute α^k by solving (6).
    Train a classifier ĥ_k on the α^k weighted S_k^L.
end for
for t = 1, ..., B do
    Compute β_i^(t) = n_i^L / Σ_i n_i^L.
    Draw a Bernoulli random variable P^(t) with probability D_KL(β^(t) || uniform).
    if P^(t) = 1 then
        Set Q^(t) = 1 / β^(t).
    else
        Compute ω^(t) as (5) and set Q^(t) = ω^(t).
    end if
    Draw k^(t) from [K] with distribution Q^(t).
    Select x^(t) according to (8) and query the label for it.
    Update S_{k(t)}^L ← S_{k(t)}^L ∪ {x^(t)}.
    Update S_{k(t)}^U ← S_{k(t)}^U \ {x^(t)}.
    Update classifier ĥ_{k(t)}.
end for
```

- Draw a rv depending on how unbalanced sources were.
- If sources are too unbalanced, more likely to explore less labeled sources.
- If sources are balanced, more likely to exploit more useful source.

Algorithm 2 AMSAT

Input: $S = S^L \cup S^U$: source data; T: target data; μ: concentration factor; B: budget;

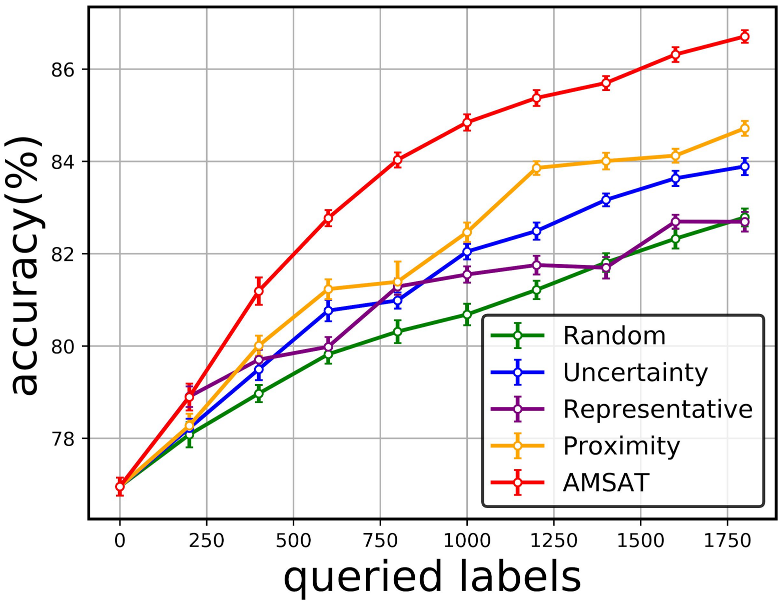
for $k = 1, \ldots, K$ **do**
    Compute α^k by solving (6).
    Train a classifier $\hat{h}_k$ on the α^k weighted S_k^L.
end for
for $t = 1, \ldots, B$ **do**
    Compute $\beta_i^{(t)} = \frac{n_i^L}{\sum_i n_i^L}$.
    Draw a Bernoulli random variable $P^{(t)}$ with probability $D_{KL}(\beta^{(t)} || uniform)$.
    if $P^{(t)} = 1$ **then**
        Set $Q^{(t)} = \frac{1}{\beta^{(t)}}$.
    else
        Compute $\omega^{(t)}$ as (5) and set $Q^{(t)} = \omega^{(t)}$.
    end if
    Draw $k^{(t)}$ from [K] with distribution $Q^{(t)}$.
    Select $x^{(t)}$ according to (8) and query the label for it.
    Update $S_{k(t)}^L \leftarrow S_{k(t)}^L \cup \{x^{(t)}\}$.
    Update $S_{k(t)}^U \leftarrow S_{k(t)}^U \setminus \{x^{(t)}\}$.
    Update classifier $\hat{h}_{k(t)}$.
end for

Kernel matching weighted uncertainty sampling:

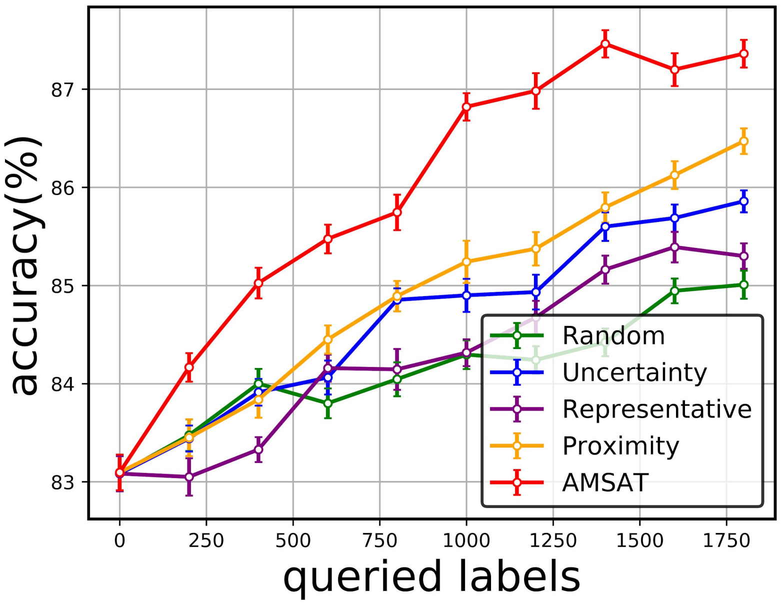
$$x = \underset{x_i \in S^U_{k(t)}}{\arg\max} E[(\hat{y}_i - y_i)^2 | x_i] \alpha_i^{k(t)}$$

Results

Figure 2. (a) Accuracy on **kitchen** (cold start)

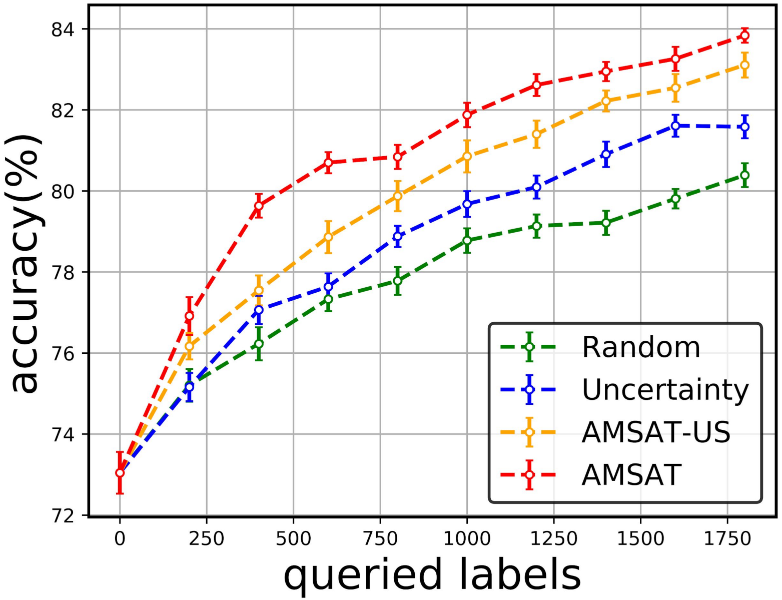


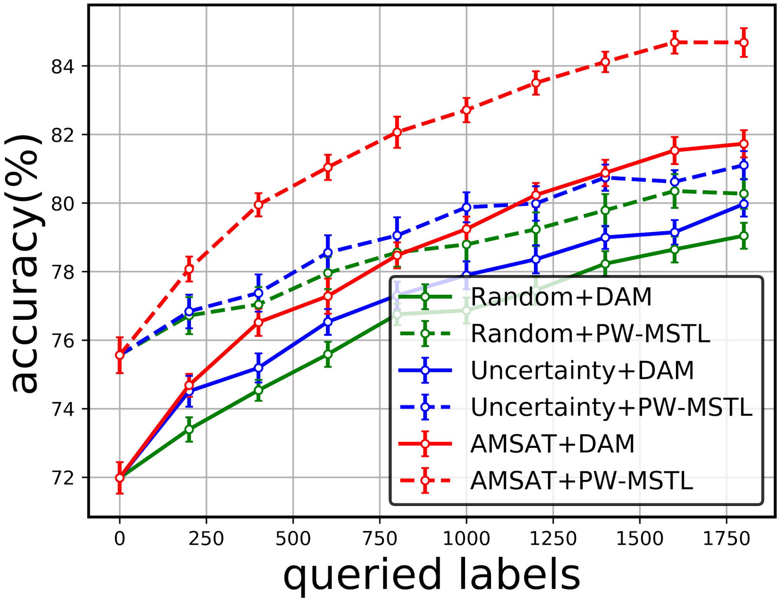
(b) Accuracy on **kitchen** (warm start)



Results (continue)

Figure 3. (a) Ablation study (b) Combined result





Conclusions

- **PW-MSTL** outperforms other MSTL approaches when sources are not equally reliable.
- **AMSAT** outperforms other active learning baselines and both source/instance picking strategies are effective.
- **Domain** is not restricted to text, both methods are general for other data types or base models.
- **Future**: study the relation between active learning in the source and negative transfer.

Q&A

- Why did you propose TWO methods in ONE paper? Are you trying to fill the space?
- Where the hell did you get these methods? Inspired by Confucius?
- Why do we want to perform active learning on sources in the first place? Why don’t we just do it in the target?
- Ok…I don’t believe in you. Can you give an example?
- Where is DNN/CNN/RNN/XNN? How could it be missing from your work?
- I think your work is naive/useless/foolish. Why do we even care?
- More question?

Towards more Reliable Transfer Learning

Zirui Wang
Jaime Carbonell

Language Technologies Institute

Carnegie Mellon University School of Computer Science