

## Introduction

**A person or other entity is often associated with multiple URL endpoints on the web**

- ▶ Barack Obama  $\Leftrightarrow$  <https://barackobama.com/> and [https://en.wikipedia.org/wiki/Barack\\_Obama](https://en.wikipedia.org/wiki/Barack_Obama)
- ▶ Donald Trump  $\nleftrightarrow$  <https://twitter.com/realDonaldTrump> and <https://www.instagram.com/ivankatrump>

Motivates the task of **webpage coreference classification** for a given entity!

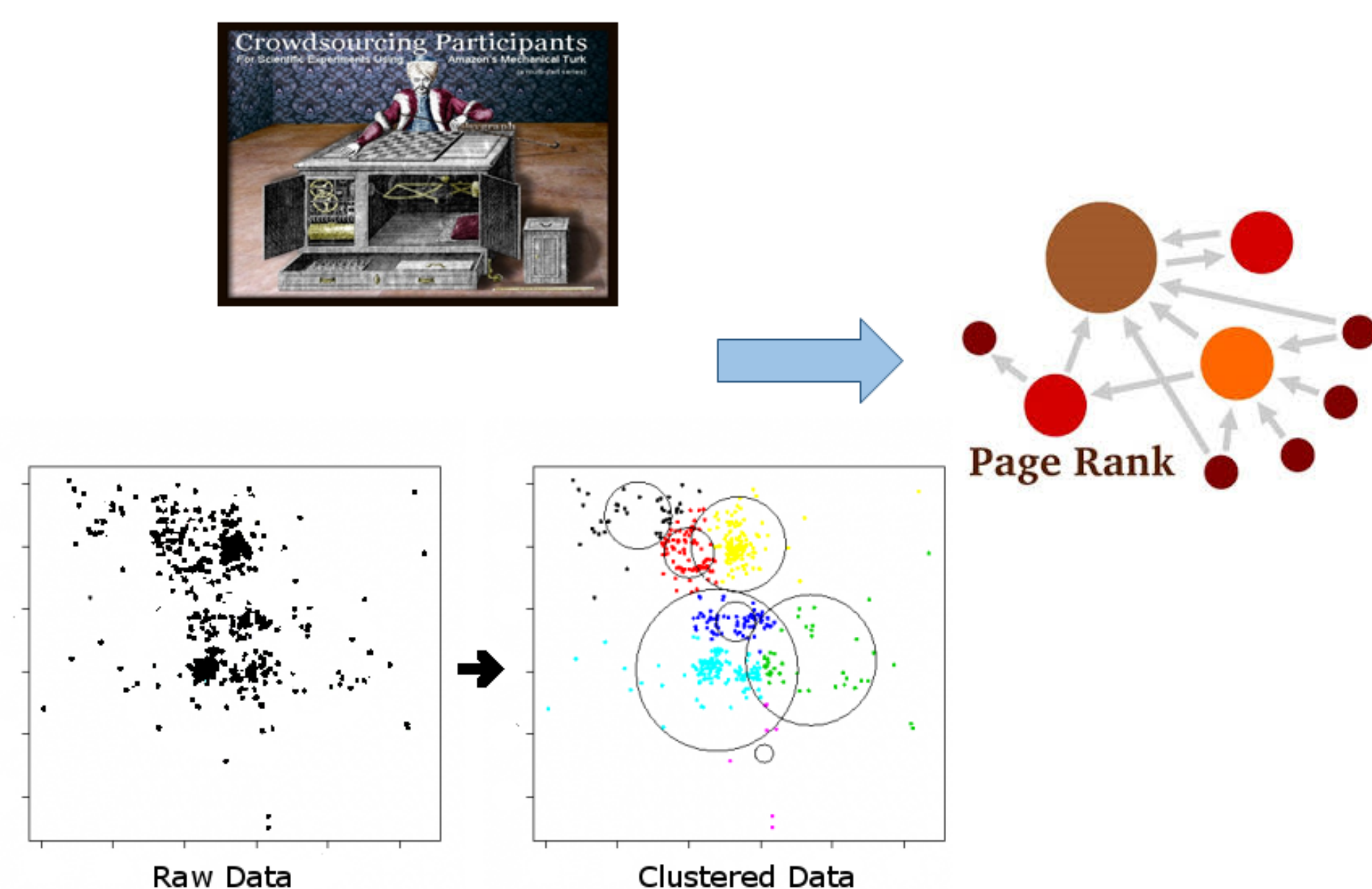
## Problem Setup

- ▶ We assume access to web KBs based on automatic crawling
  - ▶ social networks (e.g. [facebook.com/](https://facebook.com/)), news aggregation endpoints (e.g. [nytimes.com/topic/person/](https://nytimes.com/topic/person/)) and organisation directories (e.g. [www.gtlaw.com/People/](https://www.gtlaw.com/People/))
- ▶ Given a training dataset  $D$  with pairs of web URLs
  - ▶ Initially all the pairs are unlabeled ( $D_u \leftarrow D, D_p, D_n = \phi$ )
  - ▶ Learn a model  $f(\phi(U_i, U_j)) \rightarrow y$ , for URL pair  $U_i$  and  $U_j$
  - ▶ Target  $y \in \{0, 1\}$

## Distant Supervision

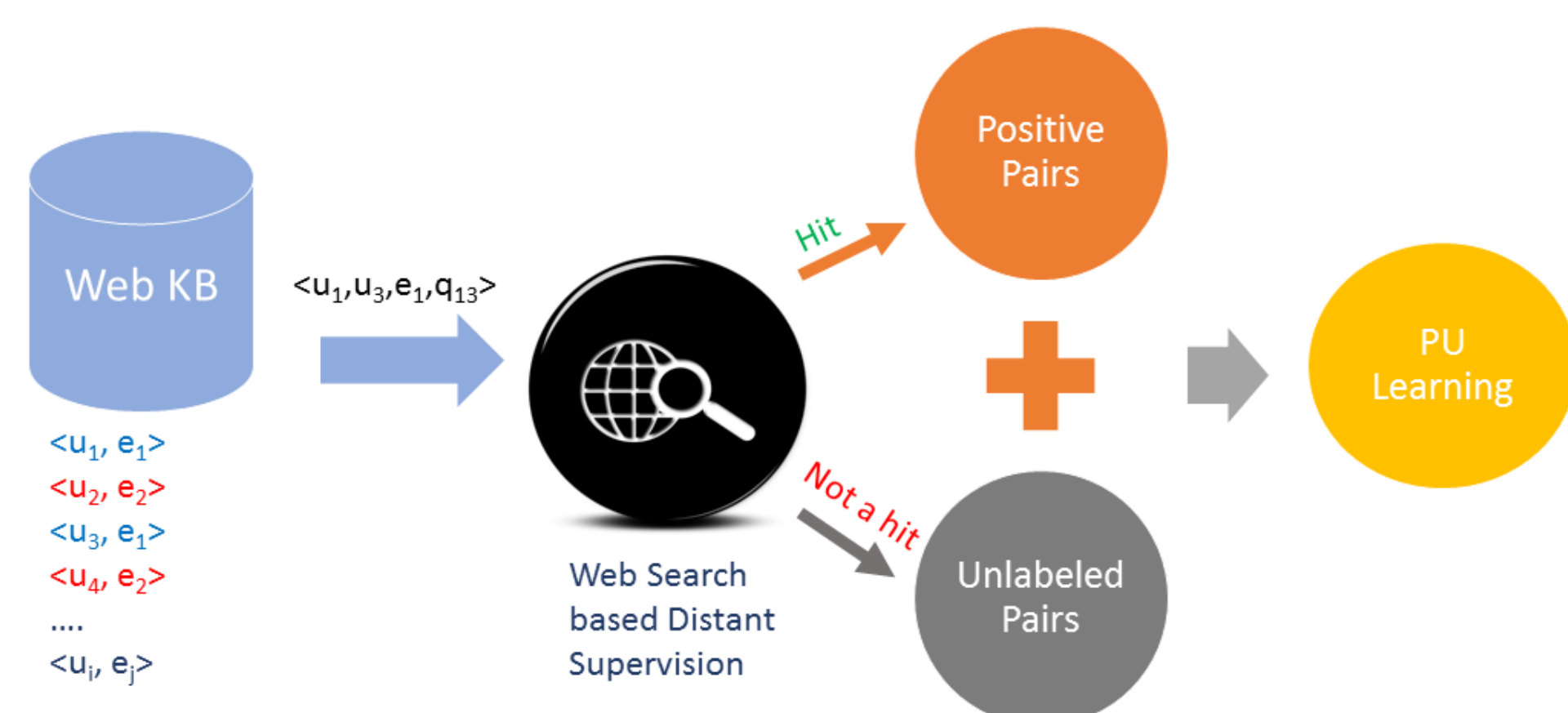
To strike a balance between unsupervised and supervised methods that require annotated data

- ▶ We obtain positive examples using web-search-based distant supervision
- ▶ Search query George Clinton AND P-Funk fetches [https://en.wikipedia.org/wiki/George\\_Clinton\\_\(musician\)](https://en.wikipedia.org/wiki/George_Clinton_(musician))
  - ▶ But not <http://www.biography.com/people/george-clinton-537674>
- ▶ We build a positive and unlabelled (PU) learning model



## Proposed Approach

- ▶ We generate queries for URL pairs that share same entity name
- ▶ Employ a label propagation technique to expand the set of positive examples



- ▶ Any binary classifier can be trained on the expanded labeled set

## Query Generation

- ▶ We construct web search queries for distant supervision as follows:
  - ▶  $Q_i$ : Using the target entity name and context information from  $U_i$
  - ▶  $Q_j$ : Similar to the above, we generate context information from  $U_j$ .
- ▶ E.g., for URL pairs: [www.imperial.ac.uk/people/f.allen](http://www.imperial.ac.uk/people/f.allen) and <https://www.linkedin.com/in/franklin-allen-0557906> a query constructed is "*Franklin Allen Brevan Howard Centre*"

## Initialize Labels

- ▶ For each query in  $Q_i$ , we check to see if  $U_j$  is present in the top- $K$  search results  $S_{ij}$
- ▶ Conversely if  $U_i$  is present in the top- $K$  results,  $S_{ji}$  for each query in  $Q_j$ 
  - ▶  $[\exists q \in Q_i, \exists S_{ji} \mid U_j \in S_{ji} \vee \exists q \in Q_j, \exists S_{ij} \mid U_i \in S_{ij}] \implies \hat{y}_{ij} = 1$
  - ▶  $D_p \leftarrow D[\hat{y}_{ij} = 1], D_u \leftarrow D \setminus D[\hat{y}_{ij} = 1]$ .

## Labelling Unlabelled Pairs

- ▶ Step 1: Randomly select  $N$  instances from  $D_p$ , and hold them out in  $S_p$ .
- ▶ Step 2: Train a binary classifier  $\theta$ , taking  $D'_p = D_p \setminus S_p$  as positive instances and  $D_u$  as negative instances.
- ▶ Step 3:  $\mu_p = \frac{1}{|S_p|} \sum_{i:S_p} p(x_i = 1|\theta)$ , (using Platt scaling)
  - ▶  $D_p^* = x_u \in D_u : p(x_u = 1) > \mu_p$ .
  - ▶  $D_p \leftarrow D_p \cup D_p^*, D_n \leftarrow D_u \setminus D_p^*$

## Datasets

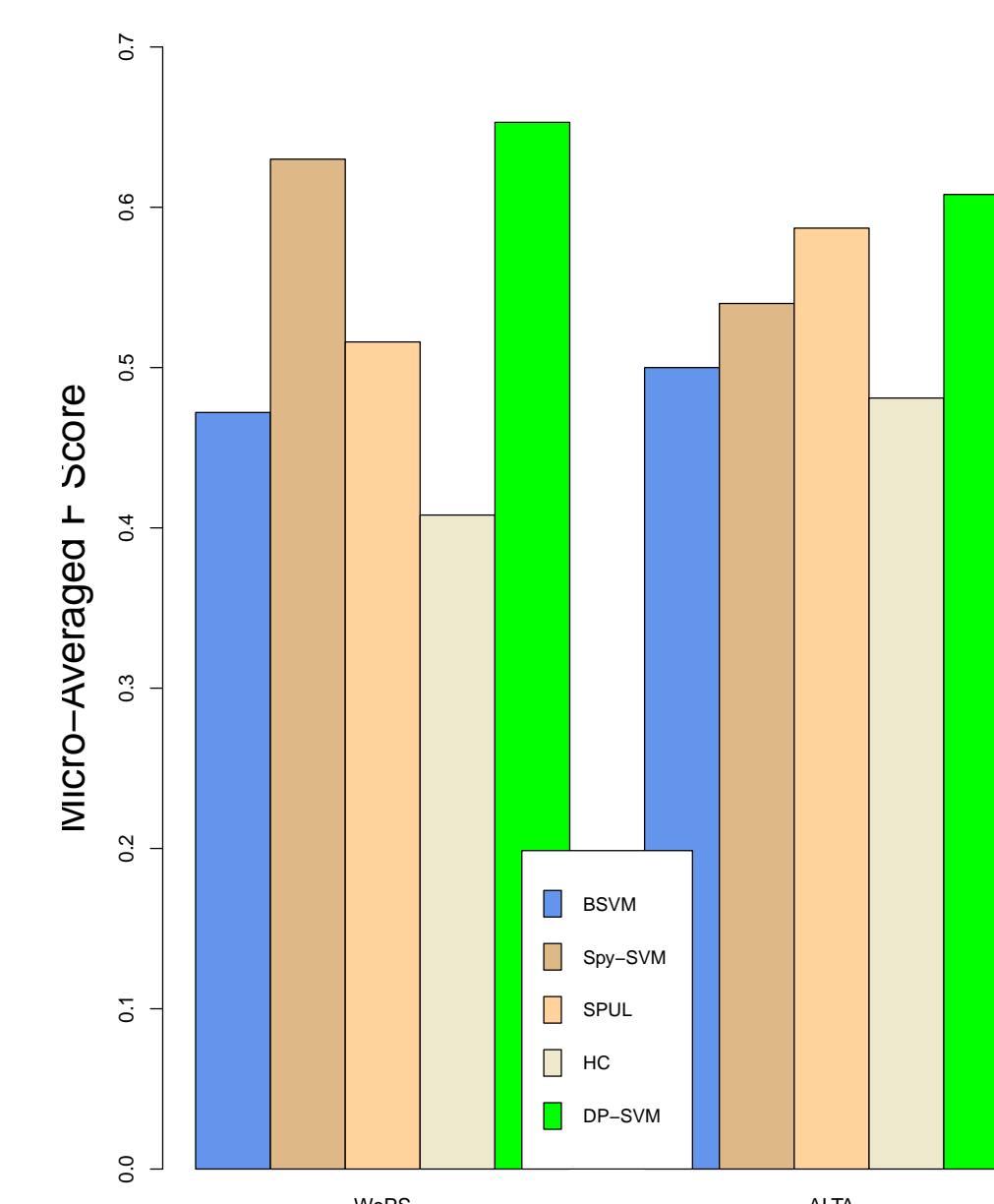
- ▶ SemEval-2007 WePS development set
  - ▶ Balanced 1000 end-point URL-pairs dataset from webpages for 49 people.
- ▶ ALTA-2016 shared task dataset
  - ▶ Balanced 400 end-point URL-pairs dataset that can refer to any entity

## Feature Representation

- ▶ Structural features such as document length difference, URL path length difference
- ▶ Semantic features such as unigram cosine similarity, cosine similarity over an average word-level word2vec representation, machine translation scores (BLEU, METEOR, TER)

## Experimental Results

- ▶ Baselines
  - ▶ Hierarchical Clustering (HC) - Unsupervised Approach
  - ▶ Biased SVM (BSVM) with costs for positive and negative classes
  - ▶ Spy-SVM (B. Liu et. al., ICML 2002)
  - ▶ SPUL (C. Elkan et. al., KDD 2008)
- ▶ Proposed Approach
  - ▶ DP-SVM (Linear Kernel SVM built using propagated distant labels)



## Conclusions

- ▶ Approach to determining whether two endpoint URLs refer to the same entity.
- ▶ Two key contributions:
  - ▶ use of distant supervision
  - ▶ application of PU Learning to the task