# Gradual Machine Learning for Entity Resolution

Boyi Hou, Qun Chen, Jiquan Shen, Xin Liu, Ping Zhong, Yanyan Wang, Zhaoqiang Chen, Zhanhuai Li
School of Computer Science, Northwestern Polytechnical University
Xi'an, Shaanxi
{ntoskrnl,shenjiquan,liuxin,pingzhong,wangyanyan,chenzhaoqiang}@mail.nwpu.edu.cn,{chenbenben,lizhh}@nwpu.edu.cn

## ABSTRACT

Usually considered as a classification problem, entity resolution can be very challenging on real data due to the prevalence of dirty values. The state-of-the-art solutions for ER were built on a variety of learning models (most notably deep neural networks), which require lots of accurately labeled training data. Unfortunately, high-quality labeled data usually require expensive manual work, and are therefore not readily available in many real scenarios. In this demo, we propose a novel learning paradigm for ER, called *gradual machine learning*, which aims to enable effective machine labeling without the requirement for manual labeling effort. It begins with some easy instances in a task, which can be automatically labeled by the machine with high accuracy, and then gradually labels more challenging instances based on iterative factor graph inference. In gradual machine learning, the hard instances in a task are gradually labeled in small stages based on the estimated evidential certainty provided by the labeled easier instances. Our extensive experiments on real data have shown that the proposed approach performs considerably better than its unsupervised alternatives, and its performance is also highly competitive compared to the state-of-the-art supervised techniques. Using ER as a test case, we demonstrate that gradual machine learning is a promising paradigm potentially applicable to other challenging classification tasks requiring extensive labeling effort.

Video: https://youtu.be/99bA9aamsgk

## CCS CONCEPTS

• **Computing methodologies** → **Learning paradigms**.

## KEYWORDS

gradual machine learning; entity resolution; unsupervised learning

**ACM Reference Format:**
Boyi Hou, Qun Chen, Jiquan Shen, Xin Liu, Ping Zhong, Yanyan Wang, Zhaoqiang Chen, Zhanhuai Li. 2019. Gradual Machine Learning for Entity Resolution. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3308558.3314121


*WWW '19, May 13–17, 2019, San Francisco, CA, USA*

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3314121

## 1 INTRODUCTION

The task of entity resolution (ER) aims at finding the records that refer to the same real-world entity [5]. Consider the example shown in Figure 1. ER needs to match the paper records between two tables $T_1$ and $T_2$. The pair of $< r_{1i}, r_{2j} >$, in which $r_{1i}$ and $r_{2j}$ denote a record in $T_1$ and $T_2$ respectively, is called a *matching* pair if and only if $r_{1i}$ and $r_{2j}$ refer to the same paper. In the example, $r_{11}$ and $r_{21}$ are *matching* while $r_{11}$ and $r_{22}$ are *unmatching*. The state-of-the-art solutions for ER were built on a variety of learning models (e.g., deep neural network [9]), which require lots of accurately labeled training data. Unfortunately, high-quality labeled data usually require expensive manual work, are therefore not easily available.

$T_1$

| ID | Title | Author | Venue | Year |
|---|---|---|---|---|
| $r_{11}$ | Belief Reasoning in MLS Deductive Databases | H Jamil | SIGMOD Conference | 1999 |
| $r_{12}$ | Efficient Index Structures for String Databases | T Kahveci, A Singh | VLDB | 2001 |
| | ..... | | | |

$T_2$

| ID | Title | Author | Venue | Year |
|---|---|---|---|---|
| $r_{21}$ | Belief Reasoning in MLS Deductive Databases | HM Jamil | SIGMOD Conference | 1999 |
| $r_{22}$ | Reasoning on Regular Path Queries | D Calvanese | SIGMOD RECORD | 2003 |
| | ..... | | | |

**Figure 1: An ER Example**

It can be observed that the dependence of the popular learning models (e.g., DNN) on high-quality labeled data is not limited to the task of ER. The dependence is actually crucial for their success in various domains (e.g., image and speech recognition [13]). However, in the real scenarios, where high-quality labeled data is scarce, their efficacy can be severely compromised. To address the limitation resulting from such dependence, we propose a novel learning paradigm for ER, called *gradual machine learning*, in which *gradual* means proceeding in small stages. Inspired by the gradual nature of human learning, which is adept at solving the problems with increasing hardness, gradual machine learning begins with some easy instances in a task, which can be automatically labeled by the machine with high accuracy, and then gradually reasons about the labels of the more challenging instances based on the observations provided by the labeled instances.

We note that there already exist many learning paradigms for a variety of classification tasks, including transfer learning [10], lifelong learning [4], curriculum learning [2] and self-training learning [8] to name a few. Transfer learning focused on using the labeled training data in a domain to help learning in another target domain. Lifelong learning studied how to leverage the knowledge mined from past tasks for the current task. Curriculum learning

investigated how to organize a curriculum (the presenting order of training examples) for better performance. Self-training learning aimed to improve the performance of a supervised learning algorithm by incorporating unlabeled data into the training data set. More recently, Snorkel [11] aimed to enable automatic and massive machine labeling by specifying a wide variety of labeling functions, whose results are supposed to be fed to DNN models.

However, the following two properties of gradual machine learning make it fundamentally different from the existing learning paradigms:

- Distribution misalignment between easy and hard instances in a task. The scenario of gradual machine learning does not satisfy the i.i.d (independent and identically distributed) assumption underlying most machine learning models: the labeled easy instances are not representative of the unlabeled hard instances. The distribution misalignment between the labeled and unlabeled instances renders most existing learning models unfit for gradual machine learning.
- Gradual learning by small stages in a task. Gradual machine learning proceeds in small stages. At each stage, it chooses to only label the instance with the highest degree of evidential certainty in a task based on the observations provided by the labeled instances. The process of iterative labeling can be performed in an unsupervised manner without requiring any human intervention.

The contributions of this demo can be summarized as follows: (1) a general paradigm of gradual machine learning **(Section 2)**; (2) a solution for ER based on the proposed paradigm **(Section 3)**; (3) an empirical study validating the efficacy of the proposed solution **(Section 4)**.

## 2 LEARNING PARADIGM

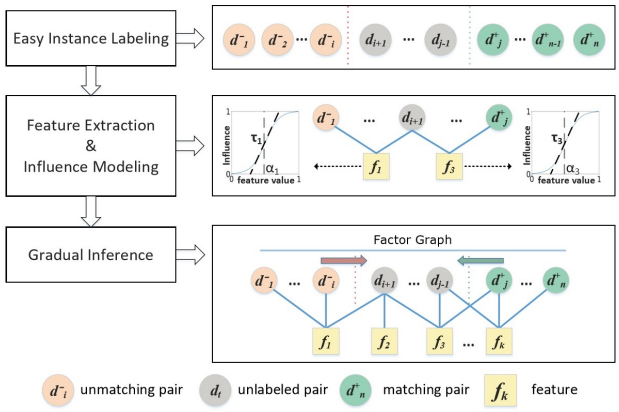


**Figure 2: Paradigm Overview.**

The paradigm of gradual machine learning, as shown in Figure 2, consists of the following three essential steps:

- **Easy Instance Labeling.** Given a classification task, it is usually very challenging to accurately label all the instances in the task without good-coverage training examples. However, the work can become much easier if we only need to automatically label some easy instances in the task. In the case of ER, while the pairs with the medium similarities are usually challenging for machine labeling, highly similar (resp. dissimilar) pairs have fairly high probabilities to be equivalent (resp. inequivalent). They can therefore be chosen as easy instances. In real scenarios, easy instance labeling can be performed based on the simple user-specified rules or the existing unsupervised learning techniques. Gradual machine learning begins with the observations provided by the labels of easy instances. Therefore, the high accuracy of automatic machine labeling on easy instances is critical for its ultimate performance on a given task.
- **Feature Extraction and Influence Modeling.** Features serve as the medium to convey the knowledge obtained from the labeled easy instances to the unlabeled harder ones. This step extracts the common features shared by the labeled and unlabeled instances. To facilitate effective knowledge conveyance, it is desirable that a wide variety of features are extracted to capture as much information as possible. For each extracted feature, this step also needs to model its influence over the labels of its relevant instances.
- **Gradual Inference.** This step gradually labels the instances with increasing hardness in a task. Since the scenario of gradual learning does not satisfy the i.i.d assumption, we propose to fulfill gradual learning from the perspective of evidential certainty. As shown in Figure 2, we construct a factor graph, which consisting of the labeled and unlabeled instances and their common features. Gradual learning is conducted over the factor graph by iterative factor graph inference. At each iteration, it chooses the unlabeled instance with the highest degree of evidential certainty for labeling. The iteration is repeatedly invoked until all the instances in a task are labeled. Note that in gradual inference, a newly labeled instance at the current iteration would serve as an evidence observation in the following iterations.

## 3 SOLUTION FOR ER

### 3.1 Easy Instance Labeling

Given an ER task consisting of record pairs, the solution identifies the easy instances by the simple rules specified on record similarity. The set of easy instances labeled as *matching* is generated by setting a high lowerbound on record similarity. Similarly, the set of easy instances labeled as *unmatching* is generated by setting a low upperbound on record similarity. The effectiveness of the rule-based approach can be explained by the monotonicity assumption of precision [1]. With the metric of pair similarity, the underlying intuition of monotonicity assumption is that the more similar two records are, the more likely they refer to the same real-world entity. According to the monotonicity assumption, we can *statistically* state that a pair with a high (resp. low) similarity has a correspondingly high probability of being an equivalent (resp. inequivalent) pair. These record pairs can be deemed to be easy in that they can be automatically labeled by the machine with high accuracy. In comparison, the instance pairs having the medium similarities are more challenging because labeling them either way by the machine would introduce considerable errors.

## 3.2 Feature Extraction and Influence Modeling

Given an ER workload, we extract three types of features from its pairs, which include:

(1) Attribute value similarity. Different attributes may require different similarity metrics.
(2) Similarity based on the maximal number of common consecutive tokens in string attributes. Consecutive tokens can usually provide additional information besides that implied by the attribute value similarity features.
(3) The tokens occurring in both records or in one and only one record. Representing a token by $o_i$, we denote the feature of $o_i$ occurring in both records by *Same*($o_i$), and the feature of $o_i$ occurring in one and only one record by *Diff*($o_i$).Unlike the previous two types of features, which treat an attribute value as a whole, this type of feature considers the influence of each individual token on pair equivalence probability.

These three types of features can provide good coverage of the information contained in record pairs. We observe that all the three types of features can be supposed to satisfy the monotonicity assumption of precision. Therefore, for each feature $f$, we model its influence over pair labels by a monotonous sigmoid function with two parameters, $\alpha_f$ and $\tau_f$ as shown in Figure 2, which denote the $x$-value of the function's midpoint and the steepness of the curve respectively. Formally, given a feature $f$ and a pair $d$, the influence of $f$ w.r.t $d$ is represented by

$$P_f(d) = \frac{1}{1 + e^{-\tau_f(x_f(d) - \alpha_f)}}, \tag{1}$$

in which $x_f(d)$ represents $f$'s value w.r.t $d$. Since the third type of features has the constant value of 1, we first align them with record similarity and then model their influence by sigmoid functions. It is worthy to point out that monotonicity of precision is a universal assumption underlying the effectiveness of the existing machine metrics for classification tasks. Our proposed solution for feature influence modeling can, therefore, be potentially generalized for other classification tasks.

## 3.3 Gradual Inference

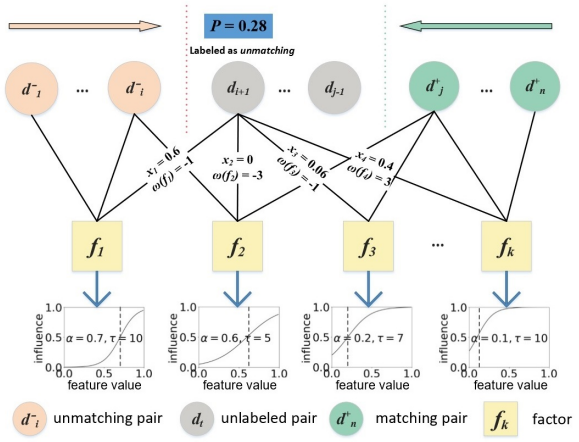


**Figure 3: An Example of Factor Graph.**

To enable gradual machine learning, we construct a factor graph, $G$, which consists of the labeled easy instances, the unlabeled hard instances and their common features. Gradual machine learning is attained by iterative factor graph inference on $G$. In $G$, the labeled easy instances are represented by the *evidence variables*, the unlabeled hard instances by the *inference variables*, and the features by the *factors*. The value of each variable represents its corresponding pair's equivalence probability. An evidence variable has the constant value of 0 or 1, which indicate the status of *unmatching* and *matching* respectively. It participates in gradual inference, but its value remains unchanged during the inference process. The values of the inference variables should instead be inferred based on $G$. An example of factor graph inference has been shown in Figure 3.

Note that the influence of a feature over a pair is specified by the sigmoid function as shown in Eq. 1. Therefore, in the factor graph, we represent the factor weigh of $f$ w.r.t $d$ by

$$\omega_f(d) = \theta_f(d) \cdot \log(\frac{P_f(d)}{1 - P_f(d)}) = \theta_f(d) \cdot \tau_f(x_f(d) - \alpha_f), \tag{2}$$

in which $\log(\cdot)$ codes the estimated influence of $f$ on $d$ by sigmoid regression, and $\theta_f(d)$ represents the confidence on influence estimation. We consider $\theta_f(d)$ as the confidence on the regression result provided by its corresponding sigmoid function, and estimate it based on the theory of regression error bound [3].

A factor graph infers the equivalence probability of a pair $d$, $P(d)$, by:

$$P(d) = \frac{\prod\limits_{f \in F_d} e^{\omega_f(d)}}{1 + \prod\limits_{f \in F_d} e^{\omega_f(d)}}, \tag{3}$$

in which $F_d$ denotes the feature set of the pair $d$. The process of gradual inference essentially learns the parameter values ($\alpha$ and $\tau$) of all the features such that the inferred results maximally match the evidence observations on the labeled instances. Formally, the objective function can be represented by

$$(\hat{\alpha}, \hat{\tau}) = \arg\min_{\alpha,\tau} -\log \sum_{V_I} P_{\alpha,\tau}(\Lambda, V_I), \tag{4}$$

in which $\Lambda$ denotes the observed labels of evidence variables, $V_I$ denotes the inference variables in $G$, and $P_{\alpha,\tau}(\Lambda, V_I)$ denotes the joint probability of the variables in $G$. Since the variables in $G$ are conditionally independent, $P_{\alpha,\tau}(\Lambda, V_I)$ can therefore be represented by:

$$P_{\alpha,\tau}(\Lambda, V_I) = \prod_{d \in \Lambda \cup V_I} P_{\alpha,\tau}(d). \tag{5}$$

Accordingly, the objective function can be simplified into

$$(\hat{\alpha}, \hat{\tau}) = \arg\min_{\alpha,\tau} -\log \sum_{d \in \Lambda} P_{\alpha,\tau}(d). \tag{6}$$

Given a factor graph, $G$, at each stage, gradual inference first reasons about the parameter values of the features and the equivalence probabilities of the unlabeled pairs by maximum likelihood, and then labels the unlabeled pair with the highest degree of evidential certainty. We define evidential certainty as the inverse of entropy [12], which is formally defined by

$$H(d) = -(P(d) \cdot \log_2 P(d) + (1 - P(d)) \cdot \log_2(1 - P(d))), \tag{7}$$

in which $H(d)$ denotes the entropy of $d$. An inference variable once labeled would become an evidence variable and serve as an evidence observation in the following iterations. The iteration is repeatedly invoked until all the inference variables are labeled. In our programs, we deploy SciPy [7] to implement the process of factor graph inference.

However, repeated inference by maximum likelihood estimation over a large-sized factor graph of the whole variables is usually very time-consuming [14]. It is also unnecessary because at each iteration, only the inference variables receiving considerable evidential support from evidence variables need to be considered for labeling. Therefore, we have proposed a scalable solution for gradual inference. It first selects the top-$m$ unlabeled variables with the most evidential support in $G$ as the candidates for labeling. To reduce the invocation frequency of factor graph inference, it then approximates probability estimation on the $m$ candidates by a more efficient algorithm. Finally, it infers via maximum likelihood the probabilities of only the top-$k$ most promising variables among the $m$ candidates. For each variable in the final set of $k$ candidates, its probability is not inferred over the entire graph of $G$, but over a potentially much smaller subgraph. More technical details of scalable gradual inference can be found in our technical report [6], but omitted here due to space limit.

## 4 EMPIRICAL EVALUATION & DEMO PLAN

In this section, we empirically evaluate the performance of our proposed approach (denoted by GML) on real data. We have empirically compared the proposed solution with four alternatives, including unsupervised rule-based (UR), unsupervised clustering (UC), support vector machine (SVM) and deep neural network (DNN) [9]. Our evaluation was conducted on 3 real datasets, DBLP-Scholar[1](denoted by DS), Abt-Buy[2] (denoted by AB) and Songs[3] (denoted by SG).

The detailed evaluation results are presented in Table 1, in which $r$ and $p$ stand for recall and precision respectively, and the results on F-1 have been highlighted. For the supervised approaches of SVM and DNN, we report their performance provided with different sizes of training data, which are measured by the fraction of training data among the whole workload. For DNN, the training data consists of the data used for model training and the data used for validation. Therefore, we report the fractions of both parts in the table. It can be observed that GML performs considerably better than the unsupervised alternatives, UR and UC. In most cases, their performance differences on F-1 are larger than 5%. The performance of GML in terms of F-1 is also highly competitive compared to both supervised approaches of SVM and DNN. GML beats SVM (with the maximum training data size at 30%) on all the three test datasets; GML also beats DNN (with the maximum training data size at 30%) on both AB and SG.

**Demo Plan.** We have implemented a demo system for gradual machine learning, whose screenshots are shown in Fig. 4. It consists of four components: Easy Instance Labeling (EIL), Feature Extraction(FE), Gradual Inference (GI) and finally Result Report (RR). EIL

[1]available at https://dbs.uni-leipzig.de/file/DBLP-Scholar.zip
[2]available at https://dbs.uni-leipzig.de/file/Abt-Buy.zip
[3]available at http://pages.cs.wisc.edu/~anhai/data/falcon_data/songs

**Table 1: Comparative Evaluation of GML**

| | GML | | | UR | | | UC | | |
|---|---|---|---|---|---|---|---|---|---|
| | r | p | F-1 | r | p | F-1 | r | p | F-1 |
| DS | 0.885 | 0.944 | **0.914** | 0.923 | 0.840 | **0.880** | 0.793 | 0.939 | **0.860** |
| AB | 0.461 | 0.790 | **0.582** | 0.645 | 0.428 | **0.514** | 0.806 | 0.268 | **0.402** |
| SG | 0.979 | 0.962 | **0.970** | 0.993 | 0.825 | **0.901** | 0.995 | 0.808 | **0.892** |

| | SVM | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 10% | | | 20% | | | 30% | | |
| | r | p | F-1 | r | p | F-1 | r | p | F-1 |
| DS | 0.890 | 0.918 | **0.903** | 0.892 | 0.918 | **0.904** | 0.896 | 0.921 | **0.908** |
| AB | 0.667 | 0.387 | **0.490** | 0.674 | 0.404 | **0.505** | 0.535 | 0.525 | **0.530** |
| SG | 0.995 | 0.855 | **0.920** | 0.992 | 0.925 | **0.957** | 0.991 | 0.945 | **0.968** |

| | DNN | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 10%(5%:5%) | | | 20%(15%:5%) | | | 30%(25%:5%) | | |
| | r | p | F-1 | r | p | F-1 | r | p | F-1 |
| DS | 0.949 | 0.869 | **0.907** | 0.945 | 0.956 | **0.950** | 0.982 | 0.929 | **0.955** |
| AB | 0.043 | 0.254 | **0.074** | 0.441 | 0.601 | **0.509** | 0.444 | 0.707 | **0.546** |
| SG | 0.777 | 0.830 | **0.802** | 0.952 | 0.900 | **0.925** | 0.938 | 0.970 | **0.954** |

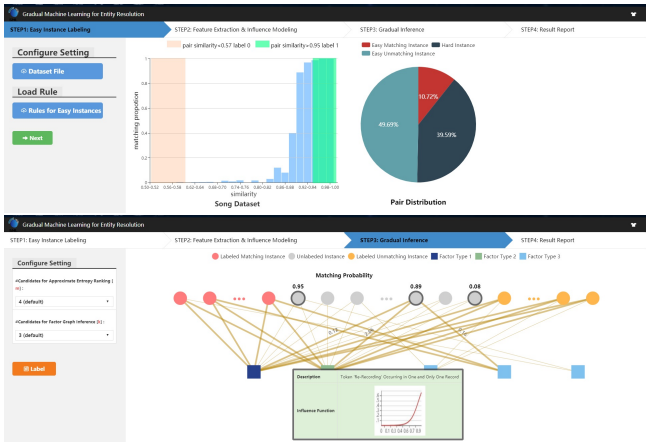


**Figure 4: Demo System Screenshots**

labels easy instances. FE extracts features, models the influence of features and constructs factor graph. GI demonstrates the process of scalable gradual inference, which includes top-$m$ candidate selection based on evidential support measurement, efficient entropy estimation and approximate factor graph inference. In GI, we have also designed an animation to visualize the iterative labeling process of GML. Finally, RR reports the experimental results. The attendees will be invited to operate the demo system step by step on a laptop using various datasets. They will be able to look into the details of the visualized process of GML.

## 5 ACKNOWLEDGEMENT

This work is supported by the Ministry of Science and Technology of China, National Key Research and Development Program (2016YFB1000703), National Natural Science Foundation of China (61732014, 61672432, 61472321 and 61502390).

# REFERENCES


[1] Arvind Arasu, Michaela Götz, and Raghav Kaushik. 2010. On Active Learning of Record Matching Packages. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data (SIGMOD '10)*. ACM, New York, NY, USA, 783–794. https://doi.org/10.1145/1807167.1807252

[2] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum Learning. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09)*. ACM, New York, NY, USA, 41–48. https://doi.org/10.1145/1553374.1553380

[3] Songxi Chen. 1994. Empirical Likelihood Confidence Intervals for Linear Regression Coefficients. *Journal of Multivariate Analysis* 49, 1 (1994), 24–40. https://doi.org/10.1006/jmva.1994.1011

[4] Zhiyuan Chen, Bing Liu, Ronald Brachman, Peter Stone, and Francesca Rossi. 2018. *Lifelong Machine Learning: Second Edition*. Morgan & Claypool. https://ieeexplore.ieee.org/document/8438617

[5] Peter Christen. 2012. *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Springer Publishing Company, Incorporated.

[6] Boyi Hou, Qun Chen, Xin Liu, Ping Zhong, Yanyan Wang, Zhaoqiang Chen, and Zhanhuai Li. 2019. Gradual Machine Learning for Entity Resolution (Technical Report). https://arxiv.org/abs/1810.12125 [Online].

[7] Eric Jones, Travis Oliphant, Pearu Peterson, et al. 2001–. SciPy: Open source scientific tools for Python. http://www.scipy.org/ [Online].

[8] Rada Mihalcea. 2004. Co-training and Self-training for Word Sense Disambiguation. In *Proceedings of the Eighth Conference on Computational Natural Language Learning, CoNLL 2004, Held in cooperation with HLT-NAACL 2004, Boston, Massachusetts, USA, May 6-7, 2004*. 33–40. http://aclweb.org/anthology/W/W04/W04-2405.pdf

[9] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep Learning for Entity Matching: A Design Space Exploration. In *Proceedings of the 2018 International Conference on Management of Data, SIGMOD 2018, Houston, TX, USA, June 10-15, 2018*. 19–34. https://doi.org/10.1145/3183713.3196926

[10] Sinno Jialin Pan and Qiang Yang. 2010. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering* 22, 10 (2010), 1345–1359. https://doi.org/10.1109/TKDE.2009.191

[11] Alexander J. Ratner, Stephen H. Bach, Henry R. Ehrenberg, and Chris Ré. 2017. Snorkel: Fast Training Set Generation for Information Extraction. In *Proceedings of the 2017 ACM International Conference on Management of Data (SIGMOD '17)*. ACM, New York, NY, USA, 1683–1686. https://doi.org/10.1145/3035918.3056442

[12] Claude E. Shannon. 1948. A mathematical theory of communication. *The Bell System Technical Journal* 27, 3 (1948), 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

[13] Dong Yu and Li Deng. 2014. *Automatic Speech Recognition: A Deep Learning Approach*. Springer Publishing Company, Incorporated.

[14] Xiaofeng Zhou, Yang Chen, and Daisy Zhe Wang. 2016. ArchimedesOne: Query Processing over Probabilistic Knowledge Bases. *Proc. VLDB Endow.* 9, 13 (2016), 1461–1464. https://doi.org/10.14778/3007263.3007284