Towards A More Holistic Approach in Entity Resolution with Active Learning Algorithms

Ge Wang

This dissertation is submitted in partial fulfilment of the requirements for the Master's degree in Information Science, UCL.

INSTG099 MSc Dissertation
Submission Date: 03/09/2018
Supervisor: Dr. Luke Dickens
Electronic Word Count: 11547
Reference Style Guide: GB/T 7714

Abstract

We researched on solving entity resolution problems using active learning approaches, and focused on general-based methods. We did a literature review and presented the general pipeline for solving ER tasks with AL approaches. We critically reviewed the existing literature to thematise existing approaches in order to identify three exemplars for evaluation. We critically reflected on the findings of the exemplars and synthesised a new method, CombinedSEL. The method is critically evaluated both theoretically and empirically. It was found to outperform previous approaches. Limitations and future works were discussed as well.

Declaration

I have read and understood the College and Departmental statements and guidelines concerning plagiarism. I declare that:

- This submission is entirely my own original work.
- Wherever published, unpublished, printed, electronic or other information sources have been used as a contribution or component of this work, these are explicitly, clearly and individually acknowledged by appropriate use of quotation marks, citations, references and statements in the text. It is 11547 words in length.

Contents

List of Figures

List of Tables

Acknowledgements

First and foremost I would like to thank my supervisor, Dr Luke Dickens, for his valuable and constructive suggestions during the planning and development of this research work. His willingness to give his time so generously has been very much appreciated.

I am also grateful to the authors who has kindly helped me throughout the implementation of my experiments. I wish to acknowledge the help provided by Professor Peter Christen, for his patience in answering my queries.

Finally, I would like to thank my friends and family who have supported me during the course of this dissertation. Without their assistance this work would not have been possible.

Chapter 1

Introduction

1.1 Introduction

In this work we studied the problem of entity resolution, specifically through the means of general-based active learning algorithms. Exploits of this kind often take advantage of classifiers now being able to actively pick the example pairs that is most informative for the training of the classifier. Such active learning algorithms are often categorised into general-based and domain-based. While domain-based ones are specifically designed for a particular type of task, they are often not applicable across domains. Thus a general-based algorithm that is able to perform well across a large number of domains is of interest. Also, the entity resolution tasks are usually of large size and it is particularly important for the algorithms to be scalable to large datasets.

Our research investigated the state-of-the-art active learning approaches, and extended techniques previously used in the context of building upon an algorithmic synthesis of the approaches investigated.

1.2 Motivation

The problem of entity resolution finds application in a wide range of domains, including bibliographical records, public health, web search results etc. Due to constraints on time and programmer effort it is necessary to find out a way to actively train the classifiers. As a result, there has been an increasing interest into active learning algorithms applicable to both very large datasets and across domains.

To our knowledge, there has not been a comprehensive survey and critical review of the active learning algorithms, specially in the sense to identify the common aspects and themes. We are thus motivated to fill this gap, and wish to focus our research onto the general-based approaches, that can be applicable to any type of classifiers and across multiple domains.

Previous approaches exploits human labelers through different techniques, and there are gaps between these approaches. We aim to close this gap by synthesising the current techniques used, and improve the algorithms performance by building upon this synthesis.

1.3 Contributions of this Work

To our knowledge, there has not been a recent comprehensive review of the active learning methods that is used for entity resolution, we thus aim to provide an up-to-date critical review for that. In the previous work there is a gap between two types of strategies on obtaining labels from human oracle. Also, no known approaches have attempted to synthesis the three main types of general-based active learning methods for entity resolution (cluster-based, conflict-based, uncertainty-based). In an attempt to close these gaps, we propose a novel approach that is built upon an algorithmic synthesis of the identified exemplars.

The contributions of this dissertation are as follows:

1. We formalised the general pipeline for solving the problem of entity resolution. (Chapter 2)

2. We presented the most up-to-date critical review of the active learning algorithms used for entity resolution and identified the common themes and aspects within the field. We formalised the core requirements for an algorithm that can be used in real world scenarios. (Chapter 3)

3. We provide a critical analysis on the most representative active learning algorithms both theoretically and empirically. We made modifications and improvements upon the basis of the original algorithms, and made them more

suitable for general-based tasks. We synthesised the difference between the two type of techniques that are used to acquire labels from human oracle. We presented the related findings and offered suggestions on the suitable tasks for each of the identified exemplar. (Chapter 4 & 5)

4. We proposed a novel approach that is built upon the identified exemplars and evaluated it both theoretically and empirically. (Chapter 6)

5. We outlined a number of future research areas we believe are important to the process of entity resolution through active learning approaches. These areas may provide useful starting points for further research on the topic. (Chapter 7)

Chapter 2

Aim of Research

The purpose of this work is to provide a comprehensive review of the field of entity resolution, specifically through the means of general-based active learning approaches. We aim to answer the following research questions:

- What is the general pipeline of entity resolution?
- What are the state-of-the-art active learning algorithms that is being used for solving the problem of entity resolution?
  - What are the common aspects and themes of these methods? What are the core requirements for a practically applicable algorithm?
  - Does there exist general-based approaches that is applicable to any type of classifiers and datasets? And how well do they perform?
- Does there exist a general-based method that is able to improve upon these state-of-the-art methods?

In order to answer the questions above, we will do a literature review of the subject and critically review the active learning algorithms are currently being used. After that, we will select the most representative exemplars for critical analysis and reflect on the related findings.

Chapter 3

Literature Review

In this chapter, we wish to go through the core concepts we will be discussing throughout the dissertation. We aim to formalise the problem of entity resolution and present our readers with a general pipeline for entity resolution.

3.1 What is Entity Resolution?

Entity Resolution (ER) refers to the traditional problem of determining if two entities in the same or different datasets refer to the same real-world object. Also known as entity matching, deduplication, record-linkage and coreference resolution, it is a complex and ubiquitous problem that has been investigated since the very beginning of computer science [1].

3.2 General Pipieline for ER

Figure 2.1 outlines the general pipeline for ER. We considered the first two steps (cleaning&standardisation, blocking/indexing) as the crucial feature engineering process, the output candidate recored pairs will then be used to generate similarity vectors with certain similarity functions. The vectors will then be input into the matching algorithms (focus of this research) to generate matches and non-matches. The whole process will then be evaluated.

3.2.1 Step 1: Cleaning & Standardisation

Real-world data can often be dirty and contain noisy, incomplete and ill-formatted records. Ensuring the good quality of data is important for a successful ER task.

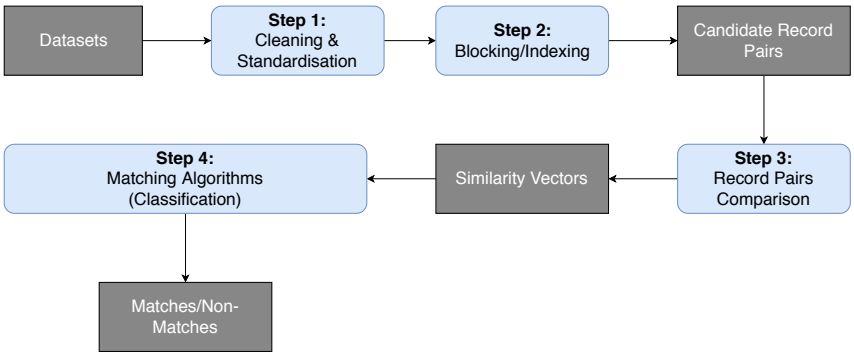

Figure 3.1: General pipeline of the ER process. The blocking/indexing step generates candidate record pairs, while the output of the comparison step are vectors containing numerical similarity values. The matching step (focus of this research) then classifies them into matches and non-matches.

This step includes adjusting lower/upper cases, removal of whitespaces, detecting and correcting known typographical errors, replacing abbreviations with proper name forms etc. [2] This is the process of converting the raw data input into well defined and consistent formats [3].

3.2.2 Step 2: Blocking/Indexing

When two datasets of size A and B are to be matched, comparisons need to be done between A $\times$ B pairs; For a single dataset of size A, the number of maximum comparisons needed is A $\times$ (A-1) / 2. Since the datasets are often of large volume, these comparisons are extremely expensive in computational cost thus approximate techniques are needed to scale the size down.

Traditional approaches employee an indexing technique called blocking, which refers to the process of clustering similar entities into blocks, then performs comparisons only among entities in the same block. In recent years, different indexing strategies have been proposed to learn blocking schemes under varying scenarios [4] [5] [6] [7][6].

The indexing techniques are just as important as the matching process, and they form a field of study themselves. However, they are not the focus of our research. We refer our readers to Christen et al. [7] for a comprehensive survey.

3.2.3 Step 3: Record Pairs Comparison

After the blocking/indexing step, candidate record pairs are now generated. The next step is called "record pairs comparison", in which numerical similarity vectors will be computed. The similarity vectors are defined as follows: Let **R** be a set of records from one or more datasets, each r $\in$ **R** having a set of attributes **A**. Given two records, $r_1, r_2 \in \mathbf{R}$, we have a similarity vector $w = f(r_1.a, r_2.a)$. This value ranges between 0 and 1, 0 being complete non-matches and 1 being perfect matches. f is the similarity function that is used to quantify this similarity. Again, this topic is not the focus of our research and we refer to Christen et al. [7] for a comprehensive survey on this topic.

3.2.4 Step 4: Matching Algorithms

We now arrive at the matching step in the ER task, which is the focus of our research. The traditional idea is to formulate ER as a classification problem, where the basic goal is to classify entity pairs as matching or non-matching. However, it is likely for real-world data to have some dependencies between attributes. For example, records with the same postcode are likely to be in the same city.

A major line of works have been the rule-based methods, in which rules are formulated to determine what constitutes a match. The limitation of rule-based methods is such that, the use of multiple attributes for generating matching rule introduces 2 problems: how do we generate a set of highly accurate rules, while reducing the complexity. In the worst case scenario, we would need to define an explicit rule for every entity.

Given the limitations of rule-based methods, many academics have turned to ML approaches. Decision Tree [8], SVM [9] [10], and Conditional Random Fields [11] classifiers were used. However, there is a crucial difference between record matching and standard classification problems: for ER tasks, we are facing a highly imbalanced dataset, the amount of non-matches significantly outnumber the amount of matches. This imbalance made it very difficult to identify a suitable labeled training set to learn an accurate classifier. We consider active learning as the best approach to solve these issues. In active learning, the learning algorithm itself picks

the examples to be labeled. The most informative examples will be chosen, better classifiers can be learned with less labeling cost. We will do a critical review on the active learning algorithms used for solving ER tasks in Chapter 4.

Chapter 4

Methodology

4.1 Overview of Methodology

During our research, we used the following research methodology:

1. We did a critical review of the active learning algorithms that are being used for entity resolution, and thematised the existing literature.
2. We formalised the core requirements for a practically applicable algorithm.
3. We identified the most representative exemplars for detailed analysis.
4. We made modifications on the algorithm of the identified exemplars, making them more suitable for our evaluation.
5. We analysed the exemplars theoretically and predicted their performances.
6. We identified appropriate datasets to ensure challenging and diverse evaluations.
7. We critically evaluated the identified exemplars both theoretically and empirically.
8. We critically reflected on the findings both theoretically and empirically and provide suggestions to readers looking for approaches to solve their own tasks.
9. We synthesised the exemplars to gain insight on possible improvements.

10. We proposed a new general-based approach and evaluate it both theoretically and empirically.

4.2 What is Active learning (AL) algorithm for Entity Resolution(ER)?

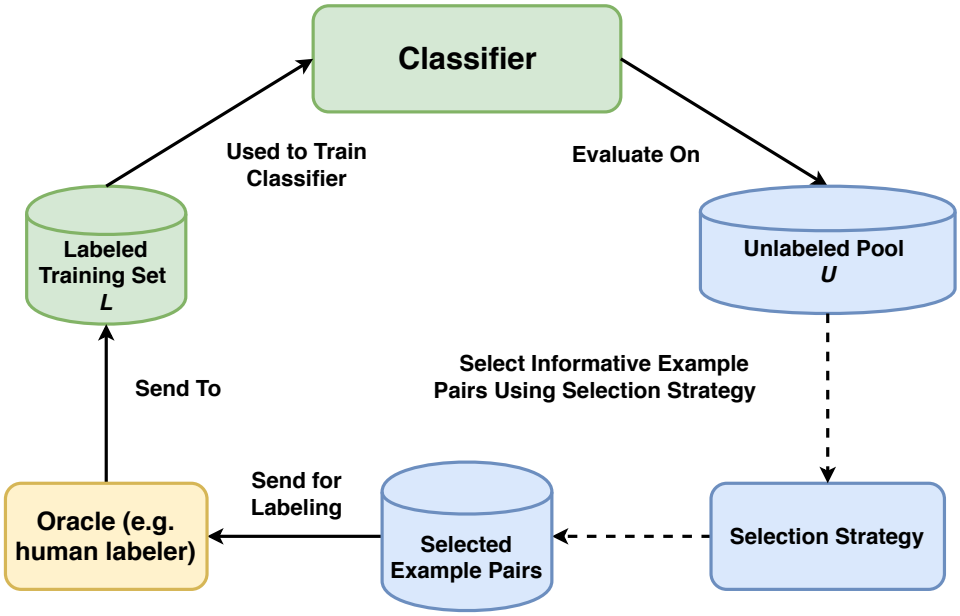

Figure 4.1: Basic Model of Active Learning

As was discussed in Chapter 2, there has been a growing need in using Machine Learning approaches for the ER tasks. However, due to the highly-imbalanced nature of used datasets (non-matches significantly outnumbers matches). Many academics have abandoned the traditional passive learning approaches. Instead, they exploit the idea of active learning.

Active learning, sometimes also called "query learning". It is a learning algorithm to interactively query the user (or some other information source) to obtain the desired outputs at new data points. Figure 4.1 is a basic model for the AL process. Queries are selected from an unlabeled pool and send for labeling to a human oracle. The labeled data will be then send to the training set, which is then used to train classifiers.

The main idea of active learning for ER is to reduce the number of queries

asked to human oracle. This can be done by either actively selecting the most informative examples, or getting as many as labels back as possible using only a few queries.

4.3 Aspects & Themes of AL algorithms for ER

Efficiency of active learning for ER has been addressed from two perspectives: (1) Incorporating blocking or indexing techniques [7]. Christen has done a comprehensive survey on this [2]. (2) Optimising the matching process (in which active learning algorithms are used). In our research, we focus on this second line of works.

We have gone through all the major works in this field and have provided a comprehensive review in Table 4.1. We identified the common **aspects** and **themes** in all ER algorithms for active learning and provided a list of all the works we have considered. This will serve as a guide for any reader looking for a matching algorithm for his own tasks.

		General-based +	Domain-based -	QC +	MA -	RS -	LB +	NO +	Applicable to Crowdsourcing +	Scalable to Large Dataset +
Clustering-based	Dasgupta [12]		Y							Y
	Haffari [13]		Y	Y						Y
	AdInTDS [7]	Y		Y		Y	Y	Y		Y
	CrowdER [14]		Y	Y		Y			Y	Y
	Brew [15]	Y					Y		Y	Y
Conflict-based	**ALIAS [16]**	Y*	Y			Y	Y	Y		Y
	Sholomo [17]		Y							
	Nigam [18]		Y							
	Tejada [19]		Y							
	Fretas [20]		Y							
	Bizer [21]		Y			Y				Y
	ALGPR [22]	Y		Y	Y					Y
	CVHull [23]	Y		Y						Y
	Sheng [24]		Y			Y		Y		Y
	Ipeirotis [25]		Y					Y		Y
	Dal Biango [26]	Y			Y					Y
	Zhang [27]	Y						Y	Y	Y
Uncertainty-based	ADANA [28]		Y						Y	Y
	Donmez [29]		Y					Y	Y	Y
	Yan [30]		Y			Y		Y	Y	Y
	Wu [31]		Y					Y	Y	Y
	Li [32]	Y						Y	Y	Y
	Du [33]	Y						Y	Y	Y
	Lin [34]	Y				Y		Y	Y	Y
	Zhao [35]		Y				Y	Y	Y	Y
	Laws [36]	Y					Y	Y	Y	Y
	Mozafari [37]	Y					Y	Y	Y	Y
	Saar-Tsechansky [38]		Y				Y			Y
	Tsai [39]		Y			Y				Y
*Rule-based	Fisher [40]		Y							
	Qian [41]	Y								Y

Table 4.1: Table (aspects and themes) summarising all Active Learning approaches for ER tasks, with + representing desirable features, - less desirable. **(*In our experiments, we made modifications and improvements on the original algorithm.)**

4.3.1 Aspects: Core Requirements for Practically Applicable Algorithms

We marked out the core requirements for algorithms that is applicable to real world scenarios, by using "+", while "-" representing features that are less desirable.

- **General-based +**

  These algorithms work for all types of classifiers and are applicable across domains. It treats the classifiers as black-box, thus users will not need to alter the internal structure of the classifiers. While internal modification of the classifier structure is possible, it is usually not practical in real-world situations, as state-of-the-art classifiers are rarely straightforward implementation of textbook, and altering them lead to extensive costs.

  We focus on the General-based algorithms in our research.

- **Domain-based -**

  These algorithms are specifically designed for certain types of classifiers and are usually only suitable for a limited type of tasks. However, they sometimes have the advantage of having more accurate results than general-based algorithms under situations of their speciality.

- **QC: Quality Control +**

  A framework is set up for ensuring the learning quality of a given task. The user is asked to specify a precision threshold or error margin, and the classifier is then learned to achieve the maximally possible recall while maintaining a precision greater than the specified threshold. This feature is extremely useful for the imbalanced dataset in an ER task.

- **MA: Monotonicity Assumption -**

  Some works made the monotonicity assumption. According to the assumption, a pair with high similarity has a correspondingly high probability of being a matching pair and vice versa. However, this usually leads to massive computational complexity. Also, the monotonicity assumption does not usu-

ally hold in practice, while the algorithm still guarantees to meet the precision threshold, a low recall is often expected [23].

- **RS: Random Sampling -**
  Random sampling is used in many of the works. It was argued that the use of it guarantees the subset being representative sample for the whole dataset. However, it can lead to unstable results and large variations between multiple runs.

- **LS: Limited Budget +**
  The number of queries can be asked to the oracle is limited. This is usually case for real-world ER tasks, and should be considered by ER algorithms.

- **NO: Noisy Oracle +**
  The labels are send to "non-experts" who do not make perfect labeling. Imperfect labeling is assumed.

- **Applicable to Crowdsourcing +**
  Due to the massive workload, it is common trend for projects nowadays to seek help from crowd sourcing. Imperfect labeling is often expected under these cases.

- **Scalable to Large Dataset +**
  These are the ones have optimised their computational cost, and work under acceptable time for ER tasks with large datasets.

4.3.2 Themes

During our review, we found that most algorithms can be categorised into 3 (+1) approaches:

- **Clustering-based Algorithms**
  This line of work aims to ease fully supervised learning for ER by adopting the clustering structure within the dataset. The idea is as follows: Clusters of pairs will be generated, some sampling will be done to sample a small set

from these clusters. If the sampled subset is found to be majority matches (or majority non-matches), this cluster will be count as all matches (or all non-matches). Labeling effort is minimised by selecting only a subset of pairs to be labeled.

- **Conflict-based Algorithms**

  A committee of classifiers is generated and iteratively refined. The pairs causing the most conflicts between the classifiers within a committee will then be send for labeling. Each pair is reviewed individually, the main idea is to reduce the number of pairs need to be labeled through selecting the most informative examples. The common methods for creating committees are: 1). Making small perturbations on the parameters of the given classifier trained through the given data. 2). Go through a set of classifiers of different types. Despite the common use, committee-based (or conflict-based) algorithms often require a minimum training set at the initial stage and some user-defined thresholds. Thus they usually require considerable labeling efforts.

- **Uncertainty-based Algorithms**

  The idea of Uncertainty-based methods are actually quite similar to that of Conflict-based ones. In both line of works, most informative examples are selected to be labeled. The difference is that, in Conflict-based ones, the examples causing most conflicts between classifiers are selected, while in Uncertainty-based ones, the examples having the highest uncertainty score are chosen. This uncertainty score can be defined and computed in various ways, different algorithms used their own ways of formulation.

- **Rule-based Algorithms***

  This line of work is not directly related to the focus of our research.

  They combines the concept of active learning and rule-based algorithms. However, these type of methods inherit the problems of rule-based algorithms, such that they need explicit rules for each matches and non-matches.

4.4 Identifying The Three Exemplars

For our evaluation, we want to select the three most representative algorithms, out of the three aspects: Cluster-based, Conflict-based and Uncertainty-based.

4.4.1 Cluster-based Algorithms

Most works in this line are domain-based, suitable either specific classifiers [13] [14] or requires the dataset to be in specific structure[12]. Brew et al.[15] proposed a general-based method. However, they did not include lableing budget and noisy oracle which should be essential for an AL algorithm for ER.

We thus select the methods proposed by Christen et al.[7] for the following reasons:

1. AdInTDS is the only method that is general-based and have considered limited budget and a noisy oracle out of all the five methods we have reviewed for this line of works.

4.4.2 Conflict-based Algorithms

ALIAS [16] is the earliest in this line of work, proposed a mechanism for randomising decision tree classifiers. The examples causing the most conflicts between classifiers will be chosen and send for labeling. Like ALIAS, other early approaches for ER also employ a committee of classifiers [17] [18] [19] [20] [21]. But they are either not scalable to large datasets due to their randomness nature, or not general-based thus only work for certain classifiers.

ALGPR [22] and Dal Biango et al. [26] are the most recent conflict-based works. While being both general-based and scalable to large datasets, they adopted a monotonicity assumption, which is a less desirable feature.

Another recent work along this line is CVHull. However, it used an external active learner thus is not suitable to be selected as a representative example.

We want to note that, the original version of ALIAS as stated in the paper was designed specifically for Decision Tree classifiers (It is still general-based as the model is applicable to other classifiers as well). However, we still want to select this for the following reasons:

1. ALIAS is one of the earliest and the most typical Conflict-based algorithms. The later ones either did modifications on it, or includes some external leaner.

2. **We managed to made modifications and improvements on the original algorithm of ALIAS, making it work across multiple classifiers. The details will be discussed in Chapter 5.**

4.4.3 Uncertainty-based Algorithms

While focusing on uncertainty is one of the oldest idea in AL literature, many other approaches either requires a probabilistic classifier or are limited to a specific classifier [28] [29] [30] [31] [35] [38].

Also, many algorithms did not consider both a limited budget and a noisy oracle. For example,[29] [30] [31] [32] only considered the existence of a noisy oracle; While [35] [36] only focused on a limited budget.

Mozafari et al. [37] proposed an active learning algorithm having the advantage of being applicable to both probabilistic and non-probabilistic classifiers. It was shown to perform well in almost all kinds of datasets. Throughout our review, we found that this method was compared against several state-of-the-art AL algorithms for ER. While being generic and widely applicable, they still perform comparably to and sometimes much better than AL algorithms designed for specific tasks [14] [15] [23] [42].

We thus want to select Mozafari et al. for the following reasons:

1. Mozafari is a general-based algorithm and have considered both a limited budget and a noisy oracle.

2. Mozafari was shown to save the labeling cost (number of queries asked to the human oracle) while still achieving good F1-Scores, comparing to many start-of-the-art active learning approaches.

Chapter 5

Critical Analysis of The Three Exemplars

In this chapter, we introduced the 3 general-based active learning algorithms that we have chosen, each one of them is representative for their school of themes, and with different initial assumptions.

5.1 Clustering-based: AdInTDS (& Our Modifications)

This approach exploits the cluster structure in data through active learning. The algorithm selects a subset of similarity score vectors into the training data set by recursively splitting the set of similarity score vectors into smaller subsets until subsets are found to be pure (the majority of similarity score vectors refer to either matches or non-matches, a purity threshold will be used). For example, if a smaller subset is found to be pure matches, the larger subset from where the smaller subset split from, will be classified as pure matches. The optimal number of questions asked to human oracle is then calculated adaptively based on a sampling error margin. The resulting training set can then be used for a user feed-in classifier.

Comparing to random sampling, AdInTDS has the advantage that, labeling effort is minimised by selecting subsets of examples to form labeled training data, instead of accessing each example individually. However, we noticed three features of this algorithm that is worth mentioning:

- Random Sampling: the method adopts random sampling to form example clusters, this is probably not the best approach as it introduces unstable in its results; Also, as the cluster are sampled randomly, it might sample a smaller subset which is not representative for the larger subset. For example, we have a larger subset of 10 examples, 4 matches and 6 non-matches. Say after calculation based on the sample error margin, we then need to sample a smaller subset of size 3. From random sampling, we can easily end up at sampling the 3 matches, and made the wrong decision that the whole subset of 10 being all matches. The nature of randomness of AdInTDS might lead to poor results occasionally. On the other hand, AdInTDS provides quality control, thus a minimum precision will be guaranteed.

- Constant Sample Error Margin: the method uses a constant user-defined sample error margin (suggested to be e=0.1). However, if we have an initial sample of cluster with a small purity, say to be 0.05. This cluster will be classified as all non-matches, thus the whole dataset will classified as all non-matches and send to training set. Then produce extremely poor results.

- Training Classifiers Using Knowledge Already Known: According to the algorithm, the initial classifier firstly select the matches and non-matches based on its current knowledge. Now Let's consider the case that, the selected non-matches are then send for labeling and is confirmed to be all non-matches, this set of data will be then sent to training set. The training set will then be used for further training this classifier. This is like saying: a kid does a set of practice exams, which is then send to the teachers for marking. The teachers confirms the answers provided by kids are correct, and then use the same set of questions to teach the kid. Here we can start to see the problem: the kid already knows the correct answers, thus he will learn little from the same set of questions (training set). Thus we are not surprised that, in the original paper [43], the algorithm shows a platform-like learning curve for many datasets. On the other hand, though, we understand the benefit of this technique. After sending these "of little use for learning" data to the training set and will be

dealt with computer, it can be made sure that no more of these type of data will be sent to the human oracle for labeling. Thus it aims to save human labeling effort.

Our Modifications:

Instead of using a constant error margin e of a subset, we now made the value of e adaptively change with the purity value of the larger subset it was drawn from. Thus the above stated problem can be avoided. We reached out to the original author, and was confirmed by him this is a possible improvement upon the original method.

We make the following predictions:

1. **AdInTDS will require less labeling cost to reach its peak performance comparing to other methods (reaching their own peak performance).**

2. **AdInTDS will produce platform-like learning curve, showing that the classifier doesn't improve much during AL process.**

5.2 Conflict-based: ALIAS (& Our Modifications)

For a detailed flow chart showing the work flow of ALIAS (modified), see Figure 7.2 in Chapter 7.

Like all committee-based methods as discussed in Chapter 4, in ALIAS, committee of classifiers were built so that each of them is slightly different from each other. A certain duplicate or non-duplicate pair will get the same result for all the classifiers, while the uncertain pairs will get different labels. The examples that causes the most conflict between the committee of classifiers are chosen to be the most informative ones to be labeled.

Our Modifications:

The original method only uses one type of classifier within a committee (Decision Tree), and manually adjust the parameters to create small perturbations within the committee. We found that this technique can be problematic when creating

"slightly different" classifiers, as it hugely depends on manual chosen of parameters. Also, we want to have an algorithm that is general-based, thus applicable to all kinds of classifiers. We thus made some adaptions on the original algorithm on the way of generating committee of classifiers. We abandoned the traditional way of randomising parameters on a single type of classifier. Instead, we now use different type of classifiers to form a committee of classifiers. Each classifier have difference specialities on different datasets, a combination of them will provide more comprehensive information for the trained classifiers.

5.3 Uncertainty-based: Mozafari

For a detailed flow chart showing the work flow of Mozafari, see Figure 7.1 in Chapter 7.

The ranker algorithm is based on nonparametric bootstrap, Each S_i is a bootstrap replicate, and is generated by drawing I.I.D. samples with replacement from raw dataset. This technique has the advantage that, individual bootstraps are independent from each other, and hence can be used to train multiple classifiers..

The key idea is as follows: the examples which is most uncertain (having the highest uncertainty score) for the classifier will be sent for labeling. The intuition is that, the more uncertain the classifier, the more likely it will mislabel the item.

The algorithm bootstraps the current set of labeled data k times, to obtain k different classifiers that are then invoked to generate labels for each item. Let S_i denote the ith bootstrap, and $\theta^{S_i}(u) = l_u^i$ be the prediction of the classifier for u when trained on this bootstrap. Define X(u) = $\sum_{i=MinorityLabel}^{k} l_u^i/k$ (the fraction of classifiers that predict a label of minority for u. We then have the formal notion of uncertainty score:

$$Uncertainty(u) = X(u) \tag{5.1}$$

So, for the case when k=10 (10 classifiers), with 2 classifiers predicting matches, 8 predicting non-matches. The minority label is matches. Thus the uncertainty score is 2/10, which is 0.2.

Mozafari is actually quite similar to the ALIAS approach. They both select the most informative examples for labeling. The only difference is in the way this example is chosen. Other than that, in Mozafari, the whole dataset is split into k bootstraps and reviewed individually; While in ALIAS, the whole dataset is reviewed as a whole. This is like saying: ALIAS is making analysis based on the world population as a whole entity. While Mozafari split the world population into countries, and analyses the population of each country individually. Surely Mozafari will have much more detailed analysis than that by ALIAS. This advantage will start to become obvious when the dataset is of large size and highly-imbalanced.

We thus make the following predictions:

1. **Mozafari will achieve comparable performance to ALIAS, and perform better in highly-imbalanced datasets.**

5.4 Baseline Approach: Random Selector

For our experiments, we also include random selectors: these are the ones without an active learning strategy, the examples sent for labeling are entirely randomly selected. This will be used the baseline of our experiments, and will be compared with the other three methods. We will use these to verify our arguments that active learning algorithms work much better than due to the way it actively selectors the examples for labeling.

5.5 A Synthesis of the Methods

We now synthesise the strategy used by the three methods and their contributions. The key idea of active learning is to exploit the human labelers at most. Thus their major contributions are within the interaction between algorithm and human labelers ("User Interection"). They can be generally split into two types of strategies:

1. "Maximise #Labels Back": AdInTDS

    - Contribution: Maximise the labels it can get from with only a few queries asked to the human lablers. For a set of 50 examples: If a randomly sampled 10 examples are all found to be matches by the human

labeler, it says the remaining 40 examples are true matches as well. Thus it essentially gets 50 labels by only asking 10 queries.

- Drawback: Use of random sampling. It did not consider the option of optimise the queries asked.

2. "Optimise Queries Asked": ALIAS & Mozafari

   - Contribution: Optimise the queries asked to human labelers. In ALIAS, examples causing most conflicts between a committee of classifiers are chosen. Each classifier would draw different conclusions, while they will agree one the easy examples, the example causing the most conflicts is the one hardest to train; In Mozafari, the examples which is most uncertain for the classifier are chosed. The intuition is that, the more uncertain the classifier, the more likely it will mislabel the item.
   - Drawback: Ask one query, get one label. It did not consider the option of maximising the labels by asking fewest amount of queries.

Chapter 6

Evaluating The Three Exemplars

6.1 Algorithmic Preparation

We used Dedupe.io [1] to implement the basic feature engineering (Step 1 to Step 3 in Figure 2.1). We do have the source code for AdInTDS [2] and ALIAS (C++)[3], and we have made adjustments and modifications on that to meet the need of this project. We used 5 basic types of classifiers: DecisionTree, SVM, RF, LogReg, LinReg.

Dataset	**Dataset Type**	**Number of Records**	**Attributes used for Blocking**	**M:N class imbalance**
Abt_Buy	Products	1082 / 1093	name, description, manufacturer, price	1:1076
Walmart_Amazon	Products	2254 / 22074	title, brand, price, shortdescr, longdescr, modelno, dimensions	1:93114
DBLP_ACM	Citations	2617 / 2295	title, authors, venue, year	1:2698
Fodors_Zagats	Restaurants	533 / 331	name, address, city, phone, type, class	1:1574
Music	Songs	1000000 / 1000000	Song_Name, Album_Name, Artist_Name, Copyright, Genre, Price, Released, Time	1:973979

Table 6.1: Characteristics of Datasets used in Experiments

6.2 Identifying Appropriate Datasets

We used five large real world datasets in our experiments. We chose 2 product type datasets, 1 citation type datasets, and 2 other type datasets. Each with different M:N ratios (around $1{:}10^3$, $1{:}10^5$ and $1{:}10^6$).

The product and citation are the most widely used type of datasets for ER tasks. We predict that the product type will be harder than the citation ones as they normally contain more detailed and messier records. We also introduce the restaurants dataset, to see how the three methods perform on unusual datasets that are not usually used in ER tasks. We also introduce the Music dataset, this is the largest dataset we have found and with the highest M:N class imbalance, we hope to set it as a challenge for the three methods.

We expected to see some oscillations throughout the learning curve as the classifier is absorbing new knowledge. In order to present a general picture, we run each method 3 times and take average for all datasets.

We use the parameters as suggested in their original paper that will give the best performance. The batch sizes (and cluster sizes) were adjusted to achieve an acceptable runtime. (See Appendix B).

6.3 Identifying Appropriate Evaluation Metrics

All AL algorithms are evaluated based on the learning curve, which plots the quality measure of interest against the number of data items that are labeled. We now decide on which performance measure wold be best for our evaluation:

- **Precision**, the ratio of true matches to the total number of matches predicted by the model. A high precision would mean that there are fewer incorrect matches between profiles.

- **Recall**, the ratio of true matches discovered by the model to the total number of matches in the labeled data. A higher recall indicates that there are fewer true matches that are missed by the system.

[1]https://dedupe.io
[2]http://users.cecs.anu.edu.au/ christen/
[3]https://www.cse.iitb.ac.in/ sunita/alias/

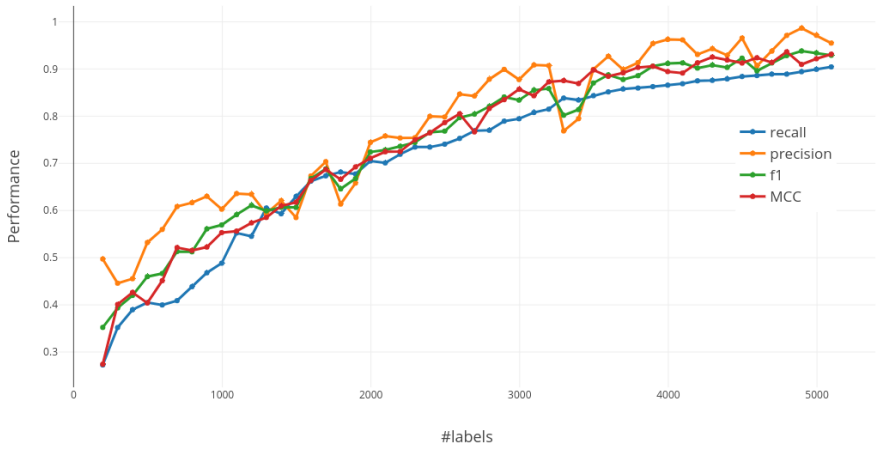

Figure 6.1: Recall, Precision, F1-Score and MCC (Abt_Buy datasets, ALIAS)

- **F1-Score**, the harmonic mean of Precision and Recall to capture the trade-off between precision and recall. The F1 score provides a more balanced view compared to Precision and Recall.

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{6.1}$$

- **MCC (Matthews Correlation Coefficient)**, The range of values of MCC lie between -1 to +1. A model with a score of +1 is a perfect model and -1 is a poor model. This property is one of the key usefulness of MCC as it leads to easy interpretability.

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP+FP) * (TP+FN) * (TN+FP) * (TN+FN)}} \tag{6.2}$$

where TP is true positive, TN is true negative, FP is false positive and FN is false negative.

To choose from them, we experimented with multiple datasets. Here, we only report the results for Abt_Buy dataset on ALIAS. Figure 6.1 shows that Precision and Recall are quite balanced. Also, MCC and F1-Score provide similar performance measurement. Using MCC will not give us revolutionary insights on the

name	description	price
Sony Switcher - SBV40S	Sony Switcher - SBV40S/ Eliminates Disconnecting And Reconnecting Cables/ Compact Design/ 4 A/V Inputs With S-Video Jacks/ 1 A/V Output With S-Video(Y/C)Jack/ 2 Audio Output	$49.00

Table 6.2: a typical record in Abt_Buy

model performance. We decide to use F1-Score as our performance measure for the following reasons:

- It provides a balanced view between Precision and Recall.
- Based on our critical review in Chapter 4, F1-Score is used as a universal measurement within the field. Which would make it much easier when we are trying to make comparisons with other people's works.

6.4 Product Type Datasets

6.4.1 Abt_Buy

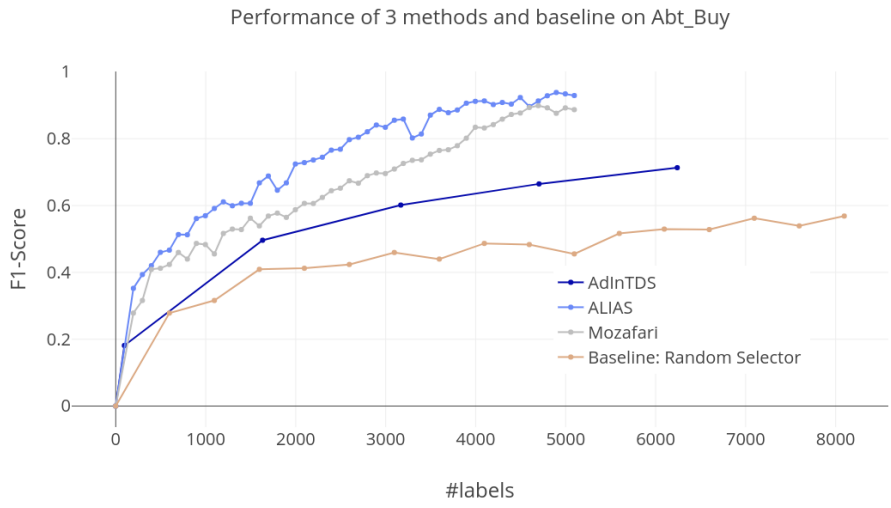

Figure 6.2: Abt_Buy: F1-Score against budget for the three methods

This is the product dataset with the smallest M:N class imbalance. Table 6.2 presents a typical record in the dataset. As can be seen, the difficulty of ER task on product type datasets is that they often contains "description" section, which can be hard to compute accurate similarity vectors.

Larger budgets allow more examples to be labeled, thus leading to a gradual increase in precision, recall and F1-Score for all three methods. The results then reach a platform as they approach the methods' best performance limit.

We evaluate the overall performance of our three methods by comparing their rate of convergence (in #labels) to the peak F1-Score with that of a random selection of the similar number of budgets. AdInTDS and ALIAS showed clear superiority over the random selector. Within 5000 labels, random selector only managed to achieve an F1-Score at 0.5, this confirms our idea that good active learning methods do outperform random selections. ALIAS performs the best and requires least labeling budgets. ALIAS has a straightforward way of picking examples for labeling, and the performance won't be heavily influenced when the dataset is not too highly imbalanced (Abt_Buy Datset has the smallest M:N rario). We will continue to test our predictions on the other datasets which have much higher M:N ratios.

Ideally, for these type of datasets, we want to achieve F1-Score higher than 0.9. AdInTDS and random selector did not reach this threshold within the label budget we used. However, we chose not to use more labels. The point of active learning is to reduce labeling budget through actively selecting the examples. While we can use more labels in this case, a small and (relatively) unbalanced dataset should not need that much labels (more than 5000). The idea of using more label budget to exchange higher performance contradicts the concept of active learning.

6.4.2 Walmart_Amazon

Due to the complex nature of product type datasets, we used another dataset to test the methods again, and see whether the results are still consistent. Walmart_Amazon has a higher M:N ratio and more records pairs. Table 6.3 is a typical record in the dataset. As can be seen, Walmart_Amazon is much more complicated than Abt_Buy and with more attribute features. It also has the key feature of a product type dataset, the "description". In general, Walmart_Amazon is of more records, has larger M:N ratio and more complex attribute features. These should make it harder for AL algorithms.

An interesting observation is seen between AdInTDS and random selector.

brand	title	price	long description	short description	modelno	dimensions
Epson	Epson 1500 Hours 200W UHE Projector Lamp ELPLP12	438.84	EPSON ELPLP12 1500HRS 200V REPL LAMP FOR LAMP POWERLITE FOR 7700P 5600P 7600 Features Lamp Life 1500 Hour Manufacturer Epson Corporation Compatible Devices LCD Manufacturer Part Number ELPLP12 Manufacturer Website Address www.epson.com Product Name Replacement Lamp Package Type Retail Product Type 200W UHE Projector Lamp Tech Specs Manufacturer Epson Corporation Manufacturer Part Number ELPLP12 Shipping Dimensions 5.25,Depth Manufacturer Website Address www.epson.com Lamp Life 1500 Hour Compatibility Epson Powerlite 7700P Projector Epson Powerlite 7600P Projector Epson Powerlite 5600P Projector Compatible Devices LCD Product Name Replacement Lamp Shipping Weight 1 lb Package Type Retail Product Type 200W UHE Projector Lamp	Epson ELPLP12 Replacement Lamp	ELPLP12	6.75 x 5.75 x 5.5 inches

Table 6.3: a typical record in Walmart_Amazon

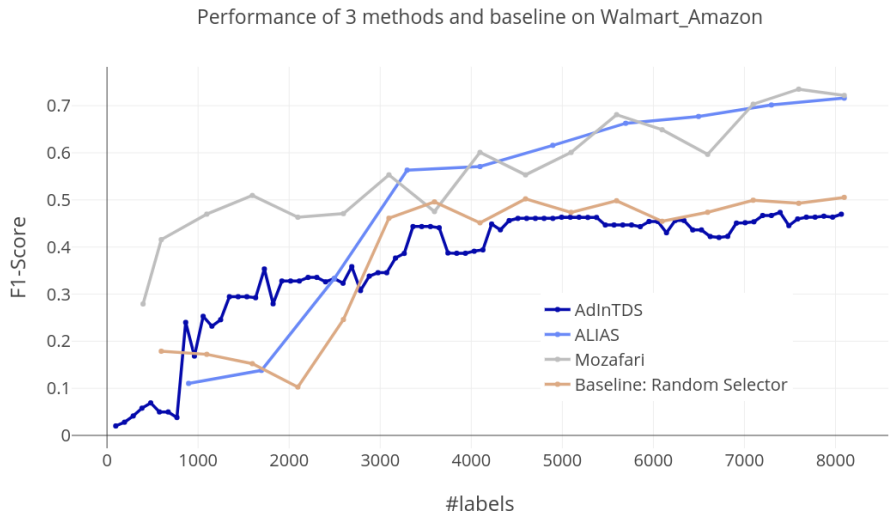

Figure 6.3: Walmart_Amazon: F1-Score against budget for the three methods

Within 2500 labels, AdInTDS has superiority over random selector. After that, however, AdInTDS and random selector showed nearly the same pattern with random selector being slightly better. Does this mean AdInDTS works even worse than a random selector? We performed a another set of 5 experiments, and AdInTDS is always of similar or worse performance than random selector. The reason behind is probably what we have been argued in Chapter 5, the nature of AdInTDS's algorithm does not guarantee the most useful examples will be used as training sets.

title	authors	venue	year
Benchmarking Spatial Join Operations with Spatial Output	Erik G. Hoel, Hanan Samet	VLDB	1995

Table 6.4: a typical record in DBLP_ACM

Also, the use of a constant error margin will lead to bad results.

6.5 Citation Type Datasets

6.5.1 DBLP_ACM

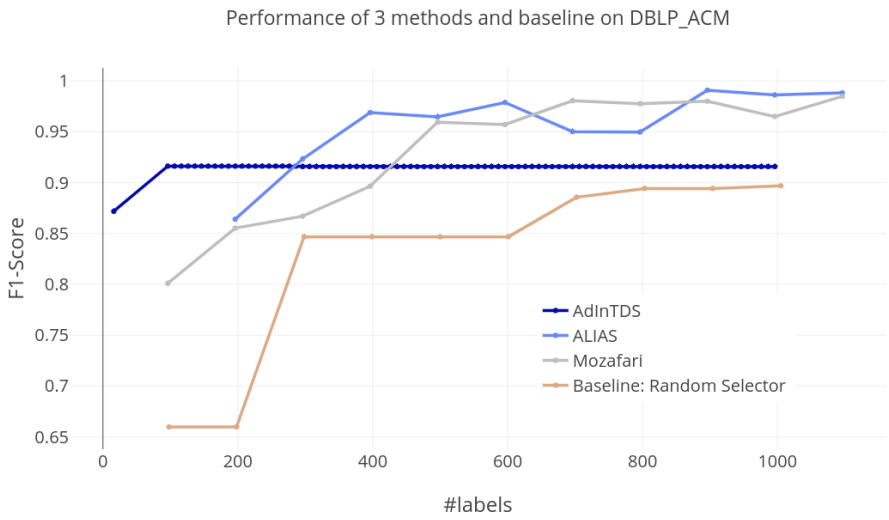

Figure 6.4: DBLP_ACM: F1-Score against budget for the three methods

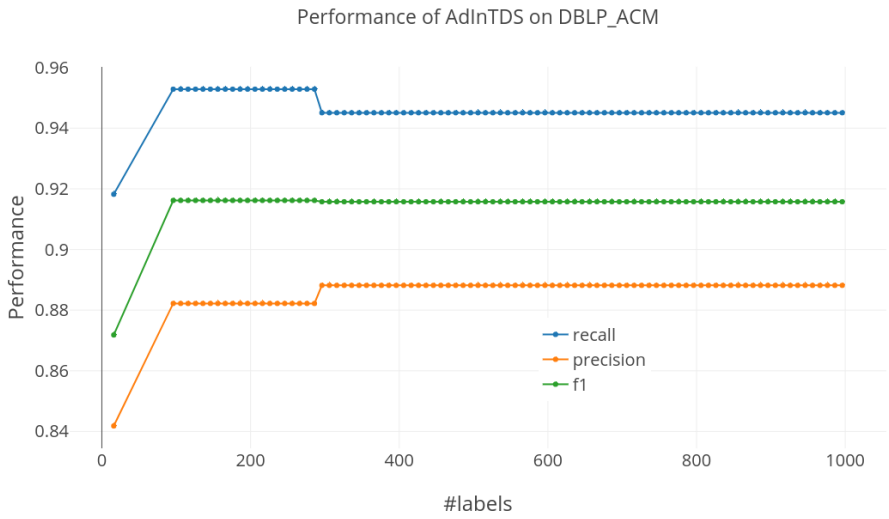

Figure 6.5: DBLP_ACM: performance against budget of AdInTDS

name	address	city	phone	type	class
matsuhisa	'129 n. la cienega blvd.'	'beverly hills'	310/659-9639	asian	14

Table 6.5: a typical record in Fodors_Zagats

This is another common type of datasets that is usually used for ER tasks. The dataset, DBLP_ACM we used here, is of a reasonable size and M:N ratio. Table 6.4 is a typical record in it. In general, citation type datasets are the easiest tasks for entity resolution. They often do not include complex information, also, attributes like authors and venue are usually in standard formats, making it easy for computing accurate similarity vectors.

In general, citation type datasets require less labeling cost than product ones, and achieved better performance. In general, methods used on citation datasets achieve high performance without needing too much budget. All three methods achieve high performance even at the initial phase of training, with very little budget.

AdInTDS shows a particularly interesting pattern, thus we report it here in Figure 6.5: After the initial steep increase, AdInTDS has a near platform performance. Similar platform-like pattern is seen in the original paper as well. Which confirms our analysis and predictions in Chapter 5.

6.6 Other Type Datasets

6.6.1 Fodors_Zagats

Fodors_Zagats is a restaurant dataset. This is the smallest of all our datasets with the lowest M:N ratio. Table 4.6 is a typical records. As can be seen, the attributes are not complex. And features like "city", "phone" and "type" are easy to compute accurate similarity vectors.

All methods managed to achieve an F1-Score at 100%, with the active learning ones being more efficient. It is interesting to see that, ALIAS managed to be achieve 100% performance within just 2 labels.

We do have a question though, whether it is worth using active learning in datasets like this. The main purpose is to actively select the examples for labeling,

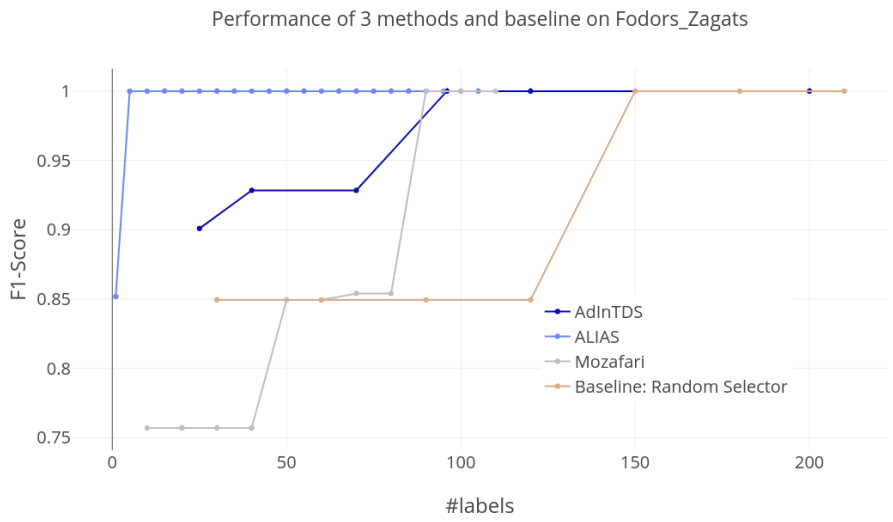

Figure 6.6: Fodors_Zagats : F1-Score against budget for the three methods

Album_Name	Artist_Name	CopyRight	Genre	Price	Released	Song_Name	Time
Welcome to Cam Country - EP	Cam	2015 Sony Music Entertainment	Country,Music, Contemporary Country,Honky Tonk	0.99	31-Mar -15	Runaway Train	3:01

Table 6.6: a typical record in Music

thus save the label budget required. In simple datasets, however, little effort is already needed for the ER task. Thus we do doubt whether it is necessary to adopt active learning algorithms for them. However, the decision very much depends on the label budget one have for their task. And we hope the results we report here will help our readers to make their own decisions.

6.6.2 Music

This is the dataset which sets a challenge for the three methods. As can be seen, all methods perform quite poorly, achieving F1-Score no higher than 0.4. This is due to the nature of the Music dataset being highly imbalanced (most imbalanced of all our datasets).

Once again, AdInTDS performed quite poorly. Mozafari achieved the best performance at 0.4, higher than ALIAS. Also, ALIAS is shown to be in disadvantage comparing to Mozafari for highly imbalanced datasets. This confirms our predictions in Chapter 5.

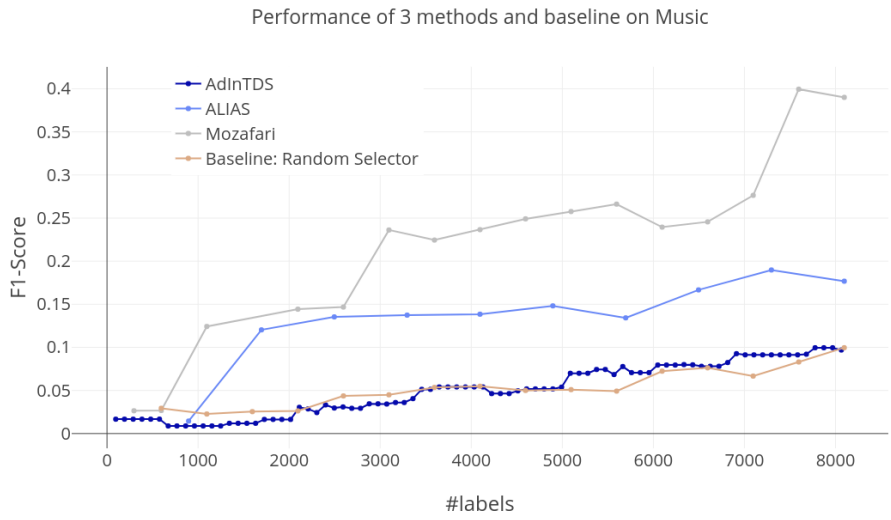

Figure 6.7: Music: F1-Score against budget for the three methods

datasets		95% of peak performance (F1-Score)		
		AdInTDS	**ALIAS**	**Mozafari**
product type	**Abt_Buy**	1.21×	1.67×	1.58×
	Walmart_Amazon	0.97×	1.4×	1.41×
citation type	**DBLP_ACM**	1.02×	1.06×	1.04×
other type	**Fodors_Zagats**	1×	1×	1×
	Music	1×	2×	4×

Table 6.7: Average improvements over baseline approach (random selection) in 95% of their Peak Performance, across all 5 datasets, by the three AL algorithms

6.7 Findings Relating to The Three Exemplars

We have calculated the average improvement of the three methods over baseline approach (random selection) in 95% of their peak performance and budget saved across all 5 datasets in Table 6.7. ALIAS tends to achieve better performance in less imbalanced datasets; Mozafari outperforms ALIAS in highly imbalanced ones.

Figure 6.9 shows the number of queries required (#labels) to achieve 95% of each method's own peak performance. This gives an indication on the approximate learning rate of each method. AdInTDS managed to achieve comparable learning rate to the other methods on most of the datasets, and required less labels on the other datasets. This confirms our analysis in Chapter 5 that, AdInTDS's designing intention was to achieve a fast learning rate and save #labels required.

Figure 6.10 shows the #labels required to achieve 95% of the peak perfor-

mance of the baseline approach (random selector). While AdInTDS requires most labeling cost for most datasets due to its poor performance, it is interesting to see it still achieved satisfying results for simple datasets. Again, ALIAS saved most labeling cost in less imbalanced datasets (< 1: 10^4); Mozafari save most in highly imbalanced ones (> 1: 10^4).

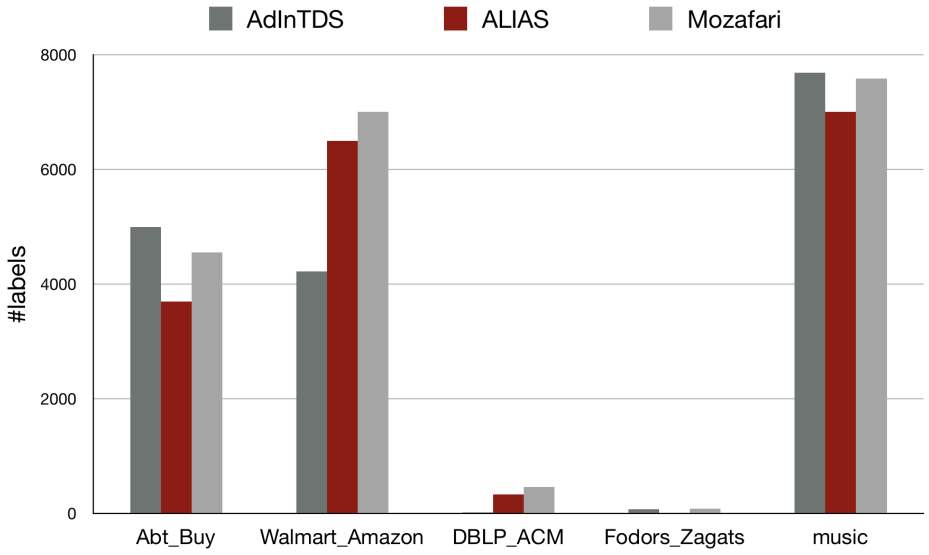

Figure 6.8: #labels (number of queries) required to achieve 95% of each method's own peak performance

The three methods worked in a consistent manner for all 3 types of datasets, which confirms our expectation that for general-based methods, they should be suitable for any type of datasets. Throughout our experiments, we found some consistent pattern that are worth discussing:

- AdInTDS behaved the worst in all our experiments: For highly imbalanced datasets ($> 1 : 10^4$), it have comparable performance to our baseline approach (random selection), although reaching the peak value at a faster rate than that. The reason for this was already discussed for a few times through our "kid learning new knowledge" story: AdInTDS focused on saving label budgets, sending less useful to training sets, thus classifier is repeatedly trained on knowledge it already know, leading to small improvement. Despite that, we can see that for datasets with low M:N ratios ($< 1 : 10^4$), it still performs better than the random approach.

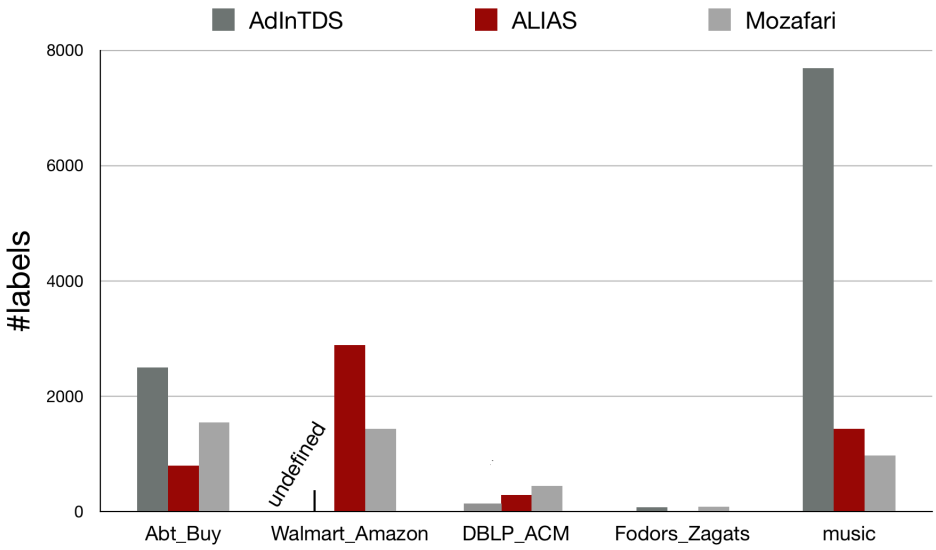

Figure 6.9: #labels (number of queries) required to achieve 95% of baseline approach's (random selection) peak performance ("Undefined" due to algorithm not able to reach the set F1-Score)

- ALIAS and Mozafari often have similar performance peak values and require similar labels budgets. While ALIAS works better on almost all datasets, it can be seen in the Walmart_Amazon and Music dataset, Mozafari outperforms it. This confirms our expectation in Chapter 5: ALIAS is slightly disadvantaged for highly imbalanced datasets. As the classifiers are trained with less details than that in Mozafari.

In general, we would like to give our readers the some suggestions when they are choosing methods for their projects (Table 6.8).

AL Algorithms	Suitable Tasks
AdInTDS	low imbalanced datasets ($< 1 : 10^4$), with simple attributes e.g Citation Type Datasets
ALIAS	almost all datasets, in slight disadvantage for highly imbalanced datasets though (M:N $> 1: 10^4$)
Mozafari	almost all datasets, works better for highly imbalanced datasets (M:N $>1: 10^4$)

Table 6.8: Suitable tasks for the three methods

Chapter 7

A New Method: COMBINEDSEL

Through our theoretical and empirical evaluation of the three exemplars, we looked into the details of these algorithms and consolidated our idea of the new method. Based on our experimental results in Chapter 6, we picked the two most promising ones (ALIAS and Mozafari), and developed our new idea on the synthesis of them.

7.1 Insights From Previous Approaches

7.1.1 Insights from Mozafari

Figure 7.1 shows the work flow of Mozafari. The idea is to bootstrap the labeled training set into k sets Classifier are trained on these k sets individually, so that we can get k different classifier. The k Classifiers will then be applied on to the Unlabeled Pool: for each example pair u, it has k binary predictions (Matched or Non-matched). From which the Uncertainty Score is then calculated. Note that, for k=10, the highest uncertainty score is 0.5. According to the Selection Strategy, the B (= Batch size) example pairs with the highest Uncertainty Score will be send for labeling.

Drawback of this approach:

Take the example of k=10, we now only have five possible uncertainty scores: 0.1, 0.2, 0.3, 0.4, 0.5. There will surely be many duplicates of these values. For the example pairs with the same uncertainty score, say 0.5 (5M, 5N). We now can only select them with random sampling, which is undesirable. An obvious way of solving this problem is to increase k. A larger k will increase the number of possible

uncertainty scores. However, this methods is still limited by the fact that it is only using one type of classifier for each evaluation. (Note that this is still general-based as the model can be used for any type of classifier) Each classifier suits each type of datasets differently. And only using one type of classifier means that all results trained will be naturally effected by the choice of that classifier.

7.1.2 Insights from ALIAS (modified version)

Figure 7.2 shows the work flow of ALIAS (our modified new version). The idea is to calculate the number of conflicts caused by an unlabeled example pair between a committee of classifiers. The ones (B = Batch size) causing the most conflicts will be send for labeling.

Drawback of this approach:

The problem is similar to that in Mozafari. Unlabled example pairs will surely have duplicated #Conflicts, and random sampling would be done in those cases. And we want to avoid the use of random sampling as it introduce varying results. A possible way of solving this is to increase the type of classifiers within the committee. However, this does not help much as the number of available classifiers is limited.

Also, based on the view of Mozafari, the labeled training set is a bit wasted by the ALIAS approach. The detailed analysis was presented in Chapter 5.

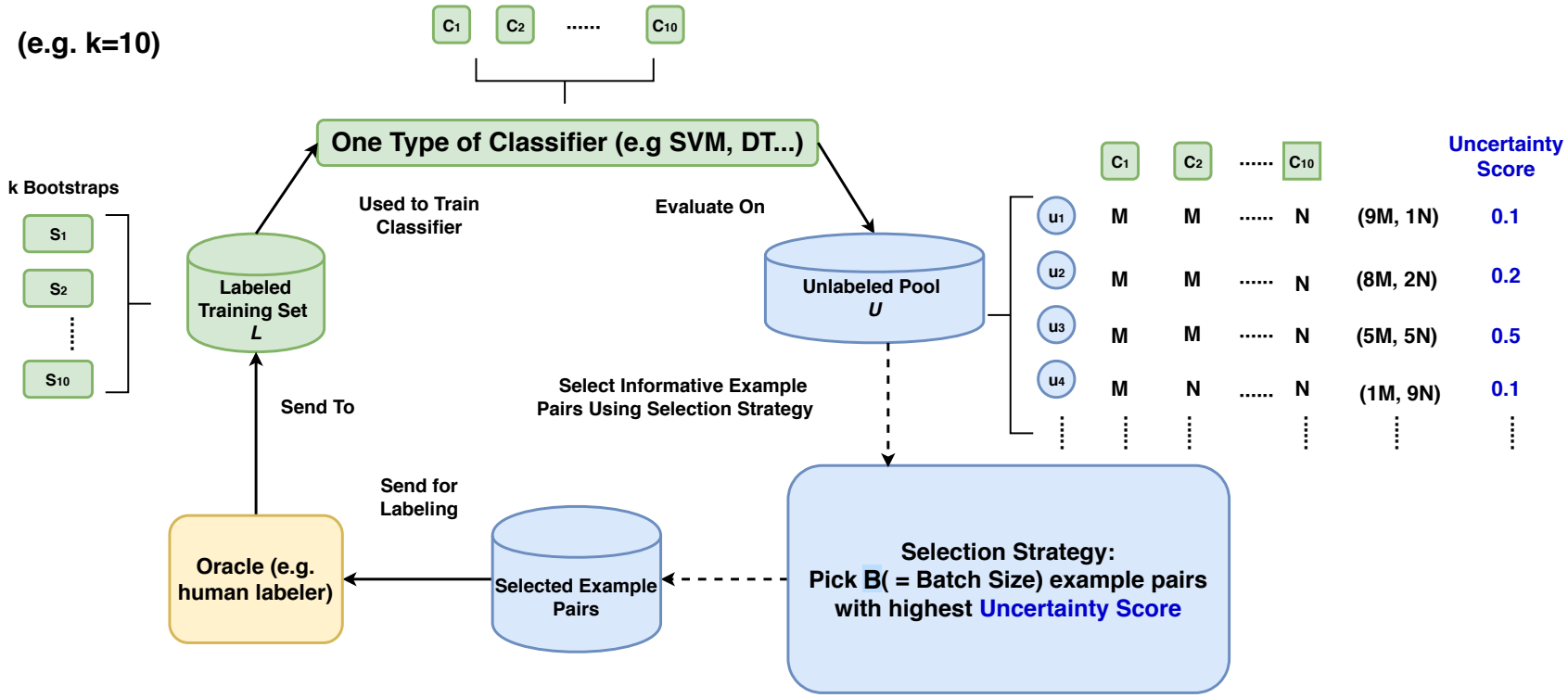

Figure 7.1: Mozafari : A Flow Chart (e.g k=10, M is Matches, N is Non-Matches)

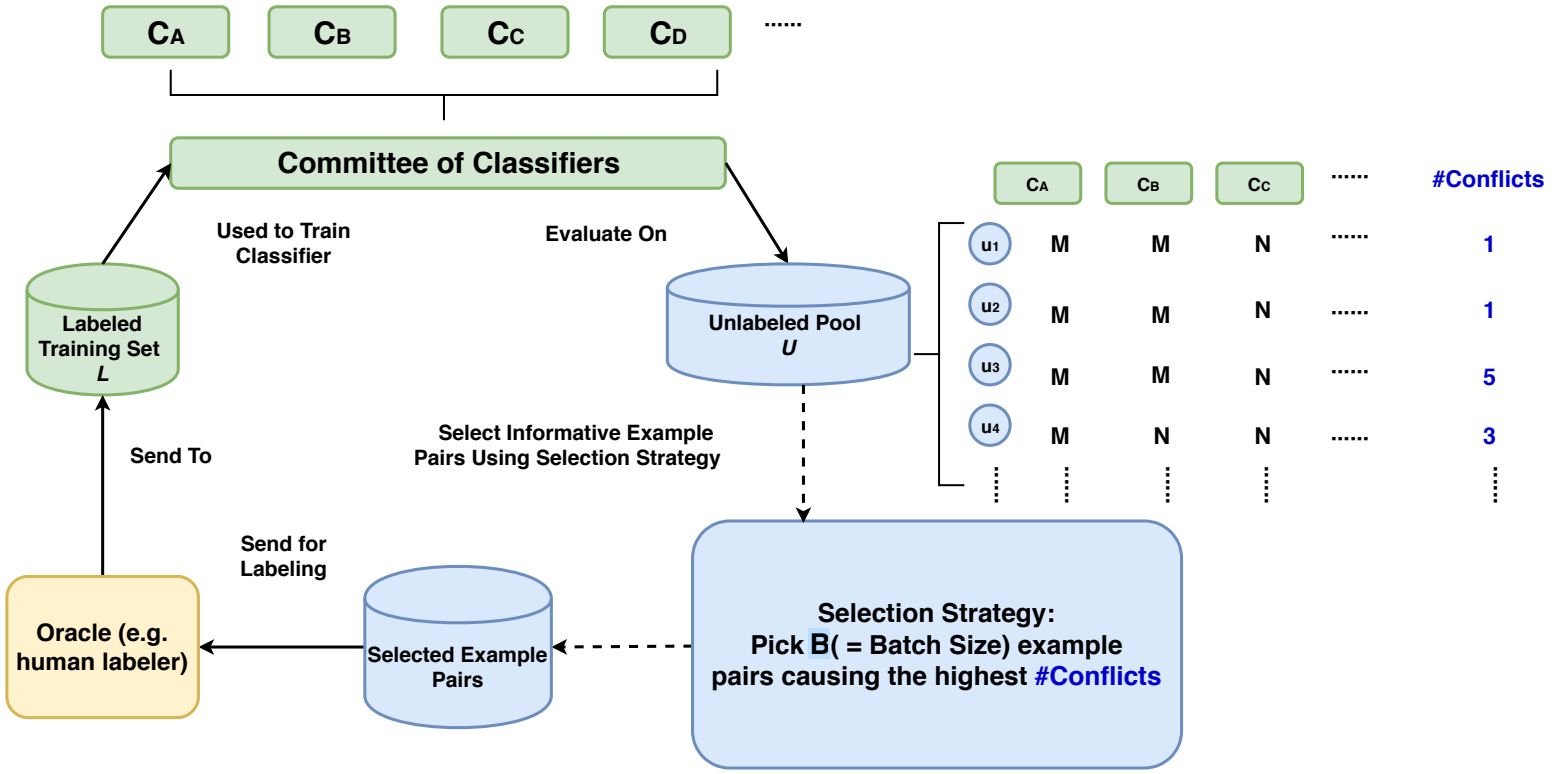

Figure 7.2: ALIAS : A Flow Chart (M is Matches, N is Non-Matches)

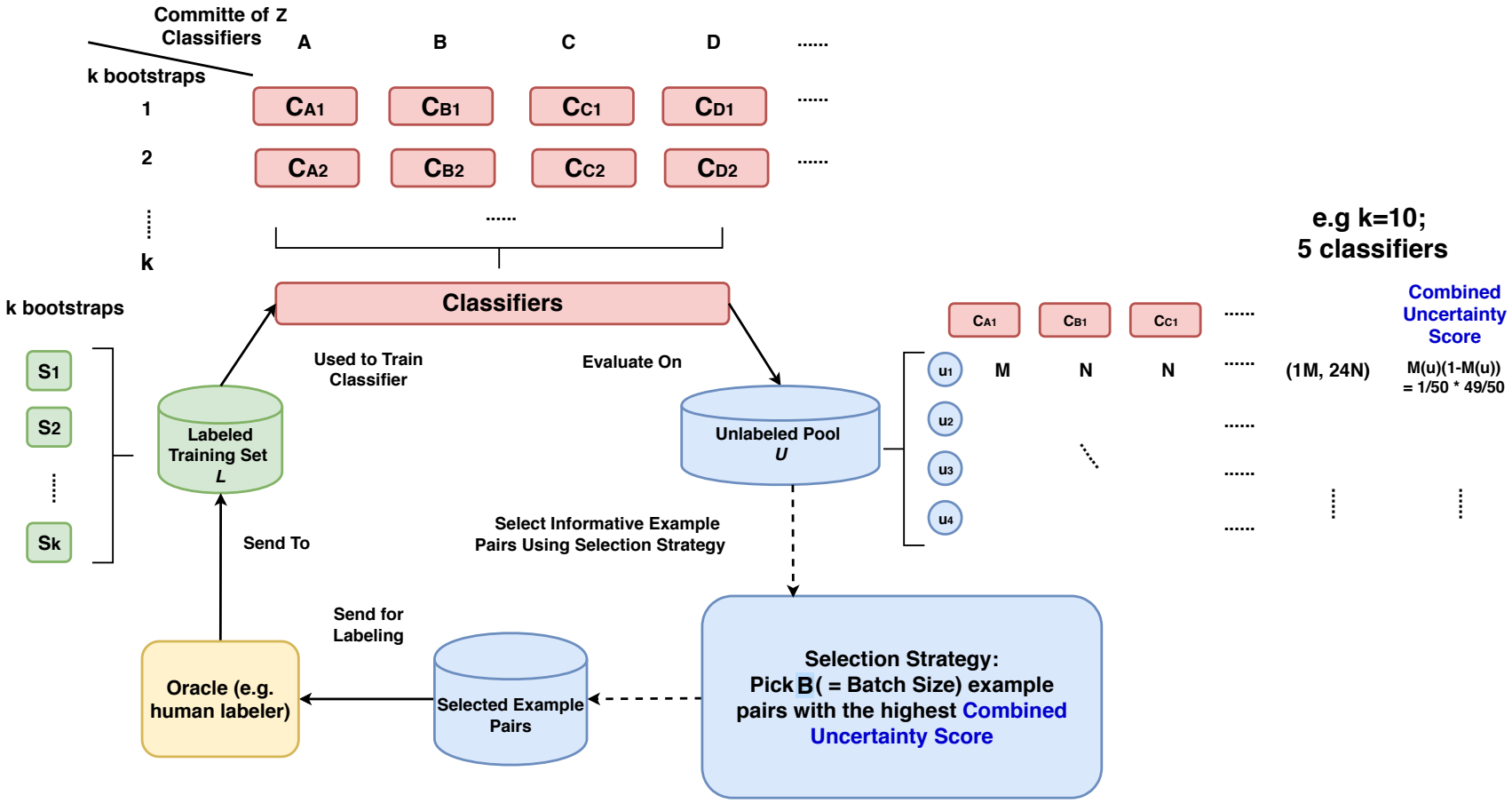

Figure 7.3: COMBINEDSEL : A Flow Chart (M is Matches, N is Non-Matches)

7.2 The COMBINEDSEL Algorithm

We now propose our new method, the COMBINEDSEL (combined selection) Algorithm. The idea is to assign a new combined uncertainty score to each of the unlabeled example pair.

Denote the total number of bootstraps by k, the total number of classifiers within a committee by Z. Let S_i denote the i^{th} bootstrap, and $\theta^{S_i}(u) = l_u^i$ be the prediction of our classifier for u when trained on this bootstrap. Let C_j denote the j^{th} classifier within a committee of classifiers, and $\theta^{C_k}(u) = l_u^j$ be the prediction of that classifier. The prediction of classifier j on the i^{th} bootstrap is thus $\theta^{C_j,S_i}(u) = l_u^{j,i}$. Define $M(u) := \sum_{i=1,j=1}^{k,Z} l_u^{j,i}/(k \times Z)$, i.e. the fraction of classifiers that predict label of 1 for u. Since $l_u^{j,i} \in 0,1$, the combined uncertainty score for instance u is given by its variance, which can be computed as:

$$CombinedUncertainty(u) = Var[\theta^{C_j,S_i}(u)] = M(u)(1 - M(u)) \tag{7.1}$$

As can be seen in Figure 7.3, k $\times$ Z classifiers are now trained. Our proposed method solved the above stated problems: many unlabeled example pairs having duplicate scores and random sampling would need to be applied. For the case of k=10 bootstraps, and a committee of Z=5 classifiers: before there are only 5 possible uncertainty scores (0.1, 0.2, 0.3, 0.4, 0.5). We now have 50 (k $\times$ Z) trained classifiers, thus 25 possible combined uncertainty scores. Note that the notion we used here is a bit different from that in Mozafari, here we used the variance as notion for uncertainty. For the case of (1M, 24N), M(u) is now 1/50, thus the combined uncertainty score is calculated to be (1/50 $\times$ 49/50). However, there are still only 25 possible numbers of scores.

On top of that, we managed to improve the algorithms intuitively as well. Let's think of the algorithms as the police trying to identify the suspects. In Mozafari, one policeman is given 10 sets of clues (say the testimony, past criminal records etc.), and he identified 10 suspects based on each of that clue. However, the problem is that, no matter how many clues he was given, all his decisions will always inherit

his natural personal bias. In ALIAS, a group of police is now solving the case, and they each have different bias and speciality on different cases. All of them are now given one large set of clues, and they are asked to each make their own decisions. This is quite a waste of resource as they are only allowed to make one decision each. In our new method, we now have a group of police, who all looked at the 10 clues at the same time, and evaluating each clue individually. The decisions they made now are much more comprehensive and combined all sources of information.

7.3 Experimental Evaluation

7.3.1 Experimental Setup

The set up of this experiment is the same with the previous experiments (Chapter 6) in general. The algorithm is tested on the 5 datasets. Throughout this section, we repeated each experiment 3 times and reported the average results. In order for better comparison, we used the same sets of parameters as before. The classifiers within the committee are Decision Tree, SVM, RF, LogReg and LinReg. The number of bootstraps k=10, error margin e=0.1. We continued to use F1-Score as our measure of performance.

7.3.2 Comparing with Previous Approaches

Figure 7.4 to 7.8 shows the comparison of the performance between ALIAS (modified version, throughout the experiments we always meant the modified version unless otherwise stated), Mozafari and CombinedSEL. We include error bars for CombinedSEL for better comparison, the error bars are calculated as $st.E = \sigma/\sqrt{s}$, where σ is the standard deviation between multiple runs, and s is the number of runs (s=3 in our case). In general, error bars are larger at the initial stage of learning, the reason for which was discussed in Chapter 6; error bars tends to be quite small around the peak value, which suggests CombinedSEL has reached a stable performance.

It can be seen for all 5 datasets, CombinedSEL managed to outperformed Mozafari, and produced results comparable or even better than that of ALIAS. It is interesting to see for product type datasets (Abt_Buy and Walmart_Amazon), Com-

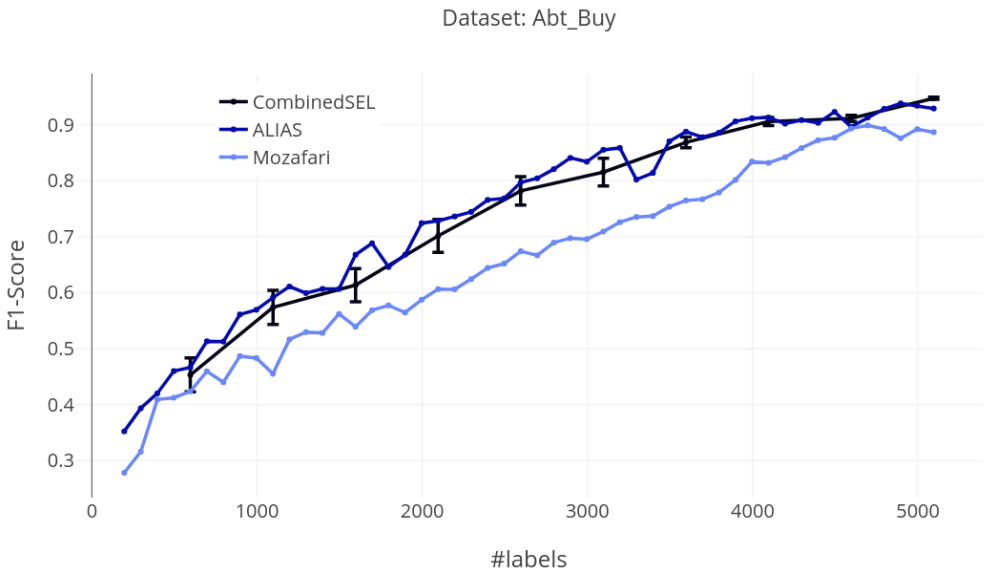

Figure 7.4: Performance of ALIAS, Mozafari and CombinedSEL on Abt_Buy

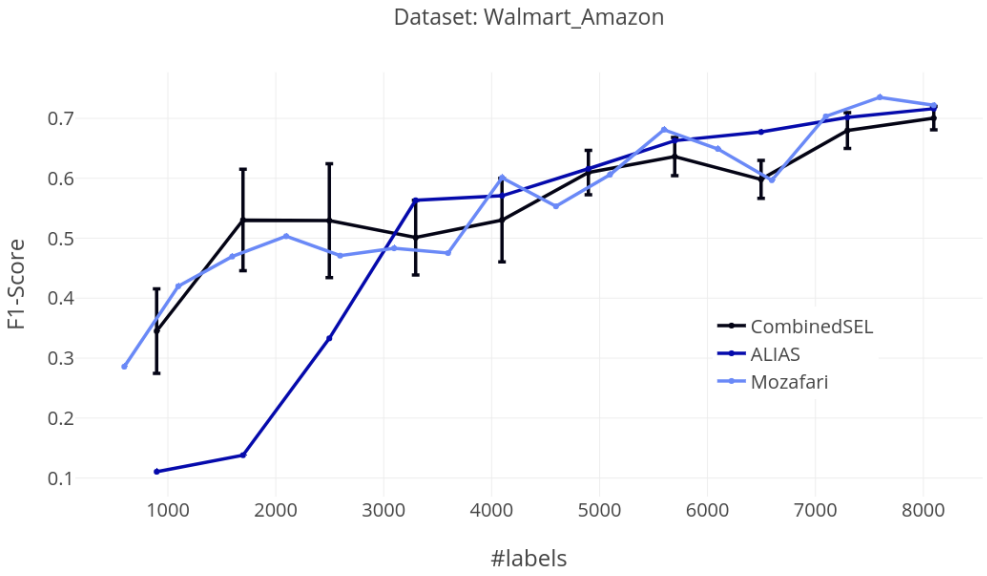

Figure 7.5: Performance of ALIAS, Mozafari and CombinedSEL on Walmart_Amazon

binedSEL achieved comparable results with ALIAS; While for citation type datasets and other type datasets, CombinedSEL managed to outperform ALIAS. The reason behind this is probably within our choice of classifiers within the committee. We are excited to see that, CombinedSEL managed to outperform both algorithms by a considerable amount on the Music dataset (which was set to be a challenge). This confirms the superiority of CombinedSEL over ALIAS and Mozafari as was dis-

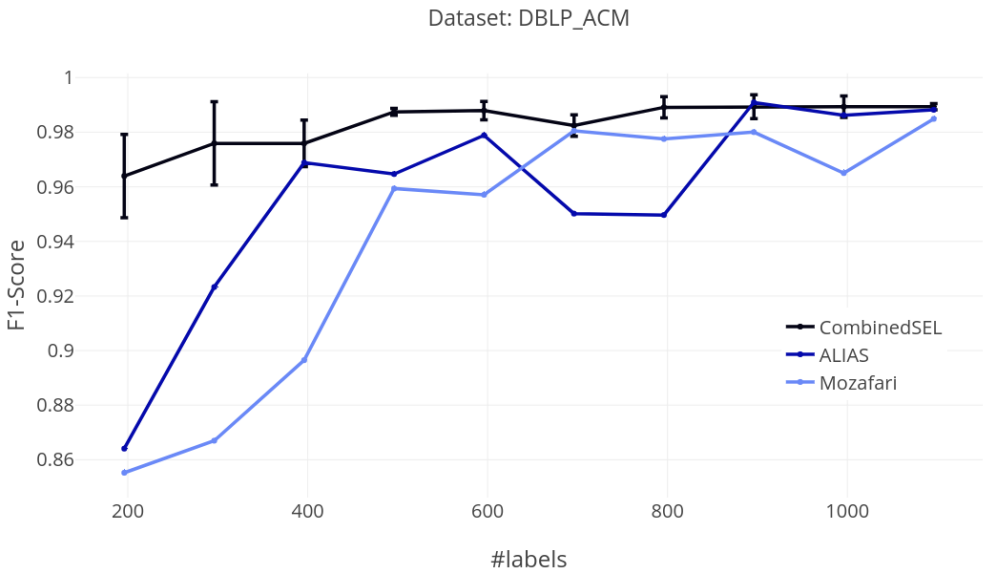

Figure 7.6: Performance of ALIAS, Mozafari and CombinedSEL on DBLP_ACM

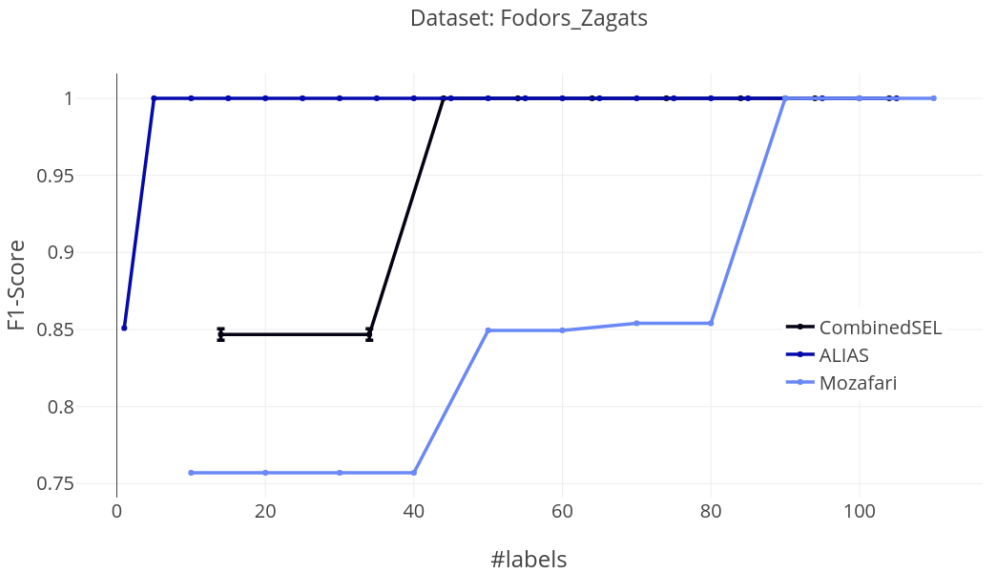

Figure 7.7: Performance of ALIAS, Mozafari and CombinedSEL on Fodors_Zagats

cussed theoretically before.

We did an overall comparison between CombinedSEL and the previous approaches (AdInTDS, ALIAS, Mozafari and Random Selector) in Figure 7.9 and 7.10. As can be seen in Figure 7.9, CombinedSEL required less labels than all other methods to achieve 95% of each method's own peak performance. This suggests CombinedSEL have the fastest learning rate (in terms of #labels). Also, in

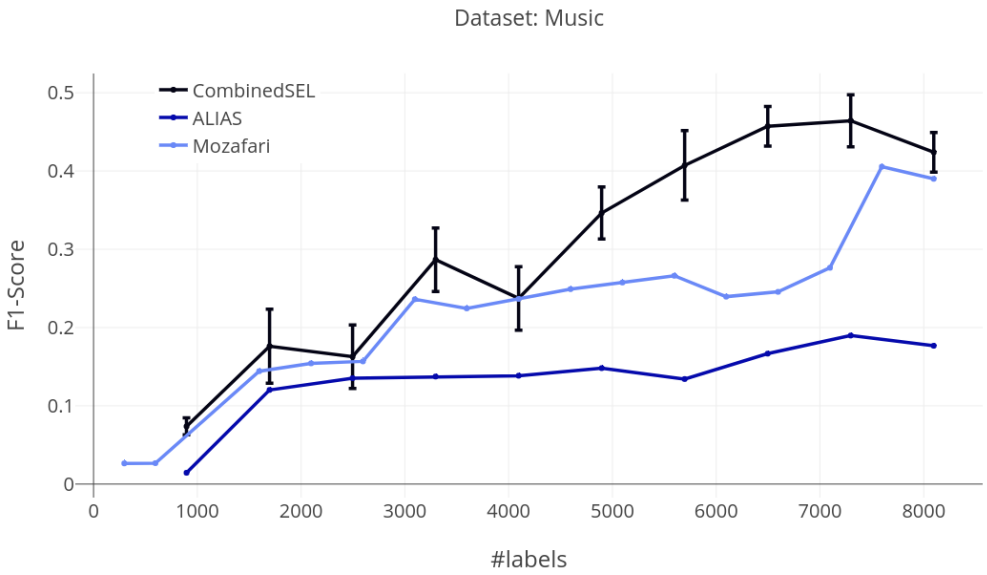

Figure 7.8: Performance of ALIAS, Mozafari and CombinedSEL on Music

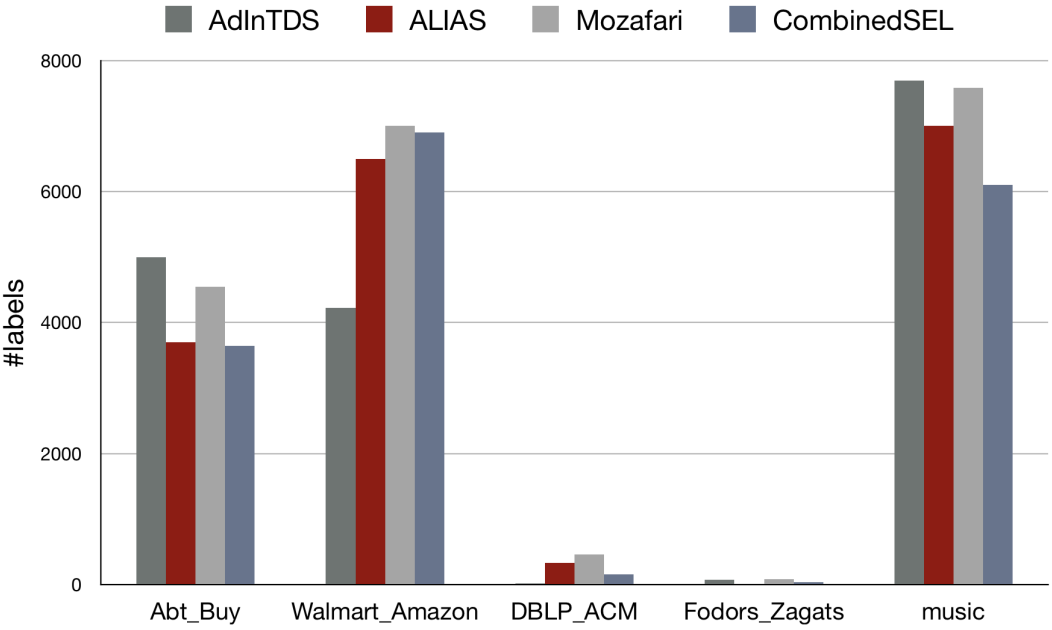

Figure 7.9: #labels (number of queries) required to achieve 95% of each method's own peak performance

Figure 7.10, we see that CombinedSEL required the less #labels than Mozafari on all datasets, and less #labels than ALIAS in all datasets except for Abt_Buy (in which comparable #labels was required). This again confirmed the superiority of this method.

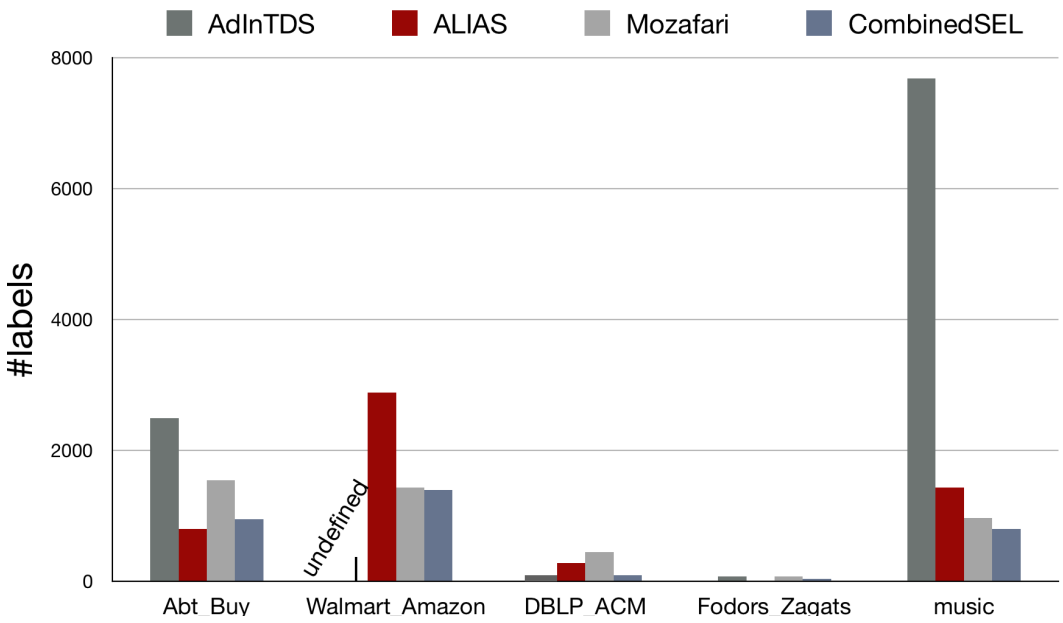

Figure 7.10: #labels (number of queries) required to achieve 95% of baseline approach's (random selection) peak performance ("Undefined" due to algorithm not able to reach the set F1-Score)

7.3.3 Evaluating the Parameters

In Chapter 5, all the experiments we did were using the parameters as stated in their original papers that were shown to give the best performance. And in the previous section, we continue to use these sets of parameters for a clear comparison between our new method and the previous approaches. However, the algorithmic nature of our algorithm is already quite different from the previous approaches, thus the set of parameters we were using before do not necessarily guarantee the maximum performance of CombinedSEL. We now evaluate how parameters would affect the performance of CombinedSEL, in the hope that we can find a set of parameters that will maximise the performance of it. We evaluate the parameters on all 5 datasets, all datasets showed similar results. Thus we only report the results of Abt_Buy datasets here for simplicity.

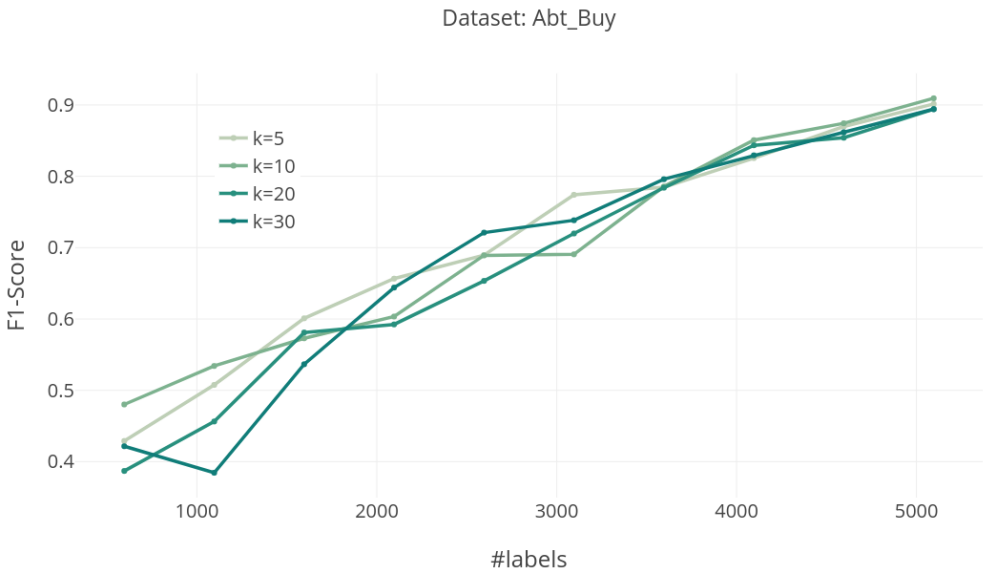

Figure 7.11: effect of adjusting #bootsrtaps k for CombinedSEL (Abt_Buy, w/a budget of 5000 labels, e=0.1, batch size=500)

7.3.3.1 Effect of #bootstraps k

As can be seen in Figure 7.11. The value of #bootstraps does not make much influence on the performance of CombinedSEL. The same pattern was suggested by the original paper of Mozafari as well, "In our experiments, k=100 or even 10 have yielded reasonable results" [37]. This is an expected result. In our experiments, we are using 5 classifiers within a committee. For every increase of 10 for the value of k, the number of individually trained classifiers is increased by 50, thus the number of possible combined uncertainty score is increased by 25. Comparing with the large number of example pairs within the dataset (around 10^6 in Abt_Buy), this increase is negligible. That is, a large number of example pairs within the send batch is still of duplicate combined uncertainty score, thus some extend of random sampling still cannot be avoided. For a detailed calculation, consider the case when k=100 on Fodors_Zagats (smallest of all our datasets, with 165763 example pairs), there are 250 possible combined uncertainty scores. For a dataset of 165763 example pairs, this means that each score is held by 663 example pairs on average. Now, our typical batch size for Fodors_Zagats is 10. It is obvious that many of these 10 pairs will be of duplicated scores. As a result, the performance of CombinedSEL

will not be much effected by the value of k.

7.3.3.2 Effect of error margin e

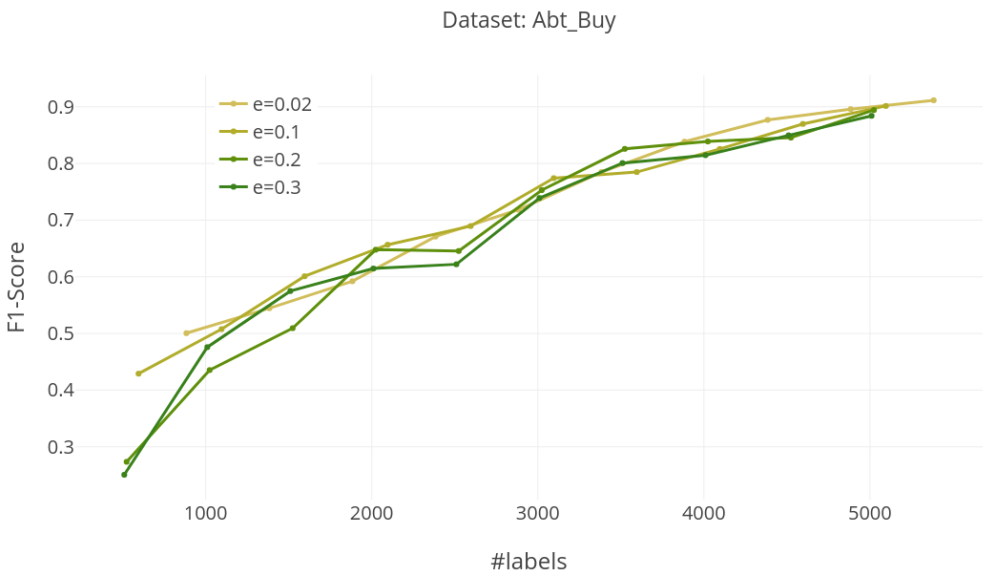

Figure 7.12: effect of adjusting error margin e for CombinedSEL (Abt_Buy, w/a budget of 5000 labels, k=5, batch size=500)

As can be seen in Figure 7.12. The value of error margin e does not effect the performance of CombinedSEL much as well, which is expected. Error margin is a statistic expressing the amount of random sampling error in a dataset. This is used in CombinedSEL to random sample the initial training set, and has no direct influence on the later learning phase. A smaller error margin will lead to a better performance at the initial stage, which is seen in our graph. But it has no impact on the peak performance.

7.3.3.3 Effect of batch size B

We repeated the experiment with the same setting under batch size ranging from 10 to 1000. As can be seen in Figure 7.13, as the batch size gets smaller, the performance gets better. This observation should be expected. Look back to Figure 7.13, batch size is the number of example pairs that will be selected and send to the human oracle. The smaller this batch size, the higher the possibility that less duplicate scores will be selected, thus avoid random sampling. Also, batch size defines how many times the model is retrained. It is interesting to see though,

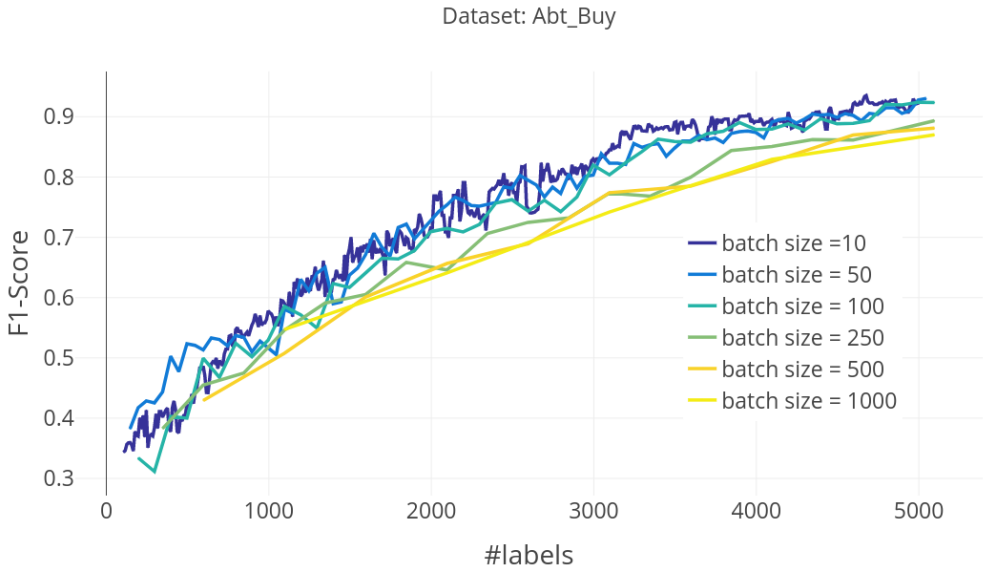

Figure 7.13: effect of adjusting batch size B for CombinedSEL (Abt_Buy, w/a budget of 5000 labels, k=5, e=0.1)

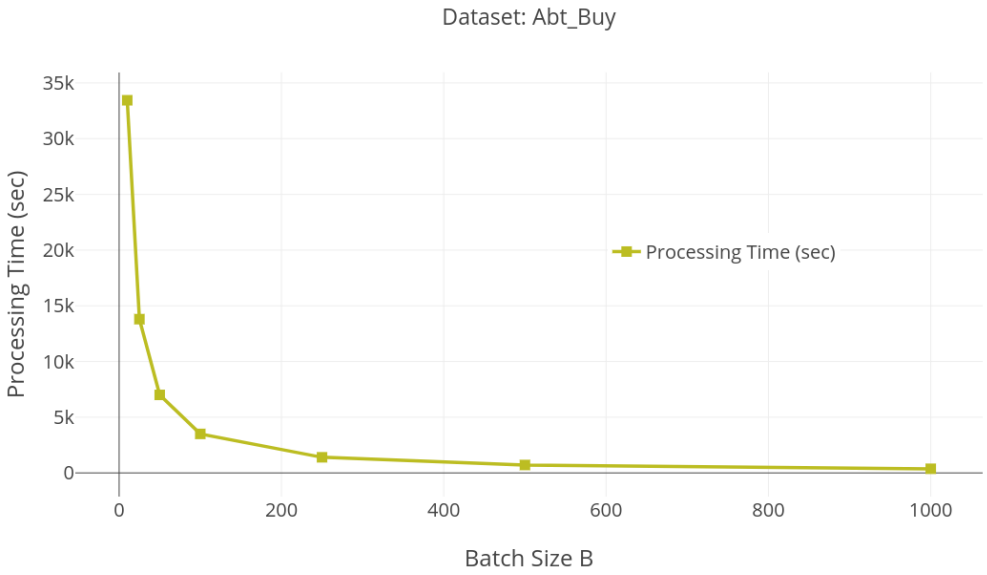

Figure 7.14: effect of adjusting batch size B on CombinedSEL's processing times on linear axis (Abt_Buy, w/a budget of 5000 labels, k=5, e=0.1)

for batch size smaller than 100, the increase in performance gets smaller (nearly negligible). This might give us some hint on the setting of a suitable batch size value.

It is worth noting that: during our experiment, we found that the processing time of CombinedSEL increases as we decrease the batch size. We recorded the

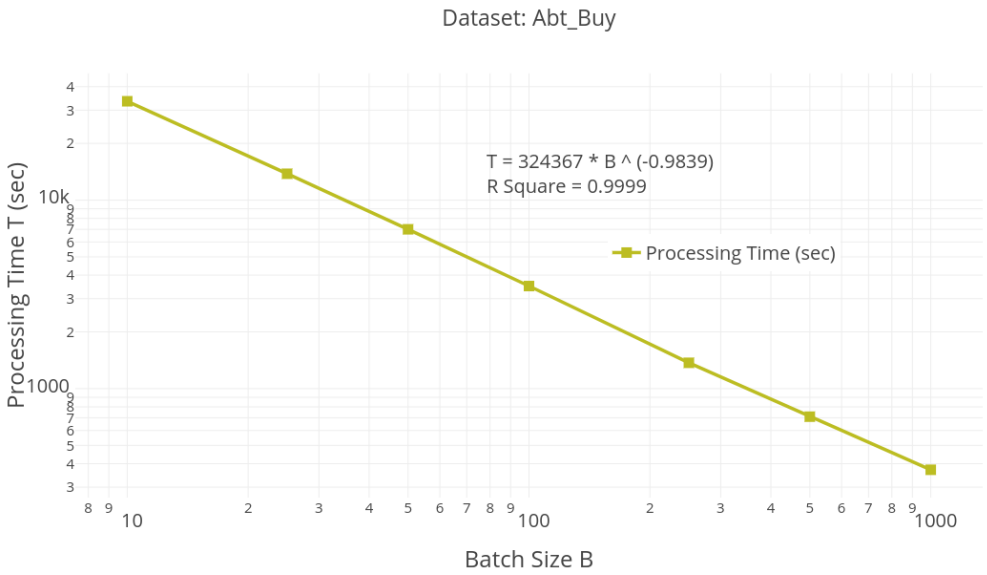

Figure 7.15: effect of adjusting batch size B on CombinedSEL's processing times on log axis (Abt_Buy, w/a budget of 5000 labels, k=5, e=0.1)

processing time (in second) of CombinedSEL when batch size being 10, 25, 50, 100, 250, 500 and 1000. Figure 7.14 shows that processing time ranges from a few seconds to about 35,000 seconds and depends heavily on batch size. We made the hypothesis that there is a inverse proportional relationship between the processing time and batch size. To test our hypothesis, we made another logarithmic graph (Figure 7.15). A perfect linear line was seen, and our hypothesis was confirmed. A function was calculated for processing time T and batch size B

$$T = 324367 \times B^{-0.9839} \tag{7.2}$$

with R-Square being 0.9999. This suggested that the function being a nearly perfect fit. In fact, this equation is essentially saying that:

$$T \propto \frac{1}{B} \tag{7.3}$$

This makes sense as we said before, batch size determines how many times the model is re-trained, the smaller the batch size, the more times the model is retrained.

We now go back to Figure 7.13 showing the performance of CombinedSEL

under different batch size. We have observed that batch sizes smaller than 100 do not make much improvement on the performance any more. While batch size is inversely proportional to processing time, and a jump from 100 to 50 leads to a huge increase in processing time. It is necessary for us to find a balance between batch size and processing time. In general, we want to find a batch size which is as small as possible while still achieving acceptable processing time. For the case of Abt_Buy, this balance is found when batch size = 100, and processing time being 3500.87138009071 secs (roughly 1 hour).

Chapter 8

Discussion

8.1 Summary

In this dissertation, we focused on using general-based active learning approaches to solve the problem of entity resolution. We have thoroughly gone through the basic concepts within the field of entity resolution and have outlined the general pipeline for ER. We presented the research methodology of this work: starting form a critical review of the works within the field and thematised the existing literature in order to identify the most representative exemplars for evaluation. We then critically analysed the exemplars both theoretically and empirically, reflected on the evaluation. From which we then synthesised a new method. For the critical analysis of the exemplars, we made modifications and improvements upon the original methods, the results among all five datasets were then compared and evaluated, ALIAS and Mozafari were shown to outperformed AdInTDS, while Mozafari works better on highly imbalanced datasets; ALIAS works better for low imbalanced datasets. For our new method, we proposed a new method: CombinedSEL, which was build upon a synthesis of the previous approaches. We evaluated CombinedSEL both theoretically and empirically, and found that CombinedSEL managed to outperform previous approaches in most cases.

8.2 Achievements

We presented to our readers a general pipeline for entity resolution which is clear and easy to interpret. We introduced the idea of active learning used for ER and

managed to critically reviewed and thematise all the important works. To our knowledge, no similar review has been done before, our work filled the gap for that.

We identified three most representative exemplars and critically analysed them both theoretically and empirically. To our knowledge, no such comparisons have been done before, that is, each exemplar represented each of their own theme of methods. Our result offered a new perspective of looking at the algorithms. At the same time, we managed to made improvements upon the original method, which were confirmed by the original author as an improvement of their works. We outlined and analysed the core strategies they each used, which are the main type of strategies used among all works within the field.

We proposed our own new method, based on a synthesis of the identified exemplars. Our results on all five datasets showed that our method managed to outperform the previous approaches. While providing good results, it is also designed to be general-based. We believe that our method will prove to be immensely useful for all types of entity resolution tasks.

8.3 Limitations & Applicability

For our dissertation, we have specifically focused on and chosen the active learning approaches that are general-based, which means that they are applicable to any type of classifiers and any type of datasets. While general-based methods are desirable for their broad applicability, some domain-based algorithms might behave better on the tasks they are specifically designed for.

8.4 Future Works and Concluding Remarks

In the future, we intend to evaluate CombinedSEL on more types of datasets. The datasets we used were of M:N ratio ranging from $1{:}10^3$ to $1{:}10^6$, it would be interesting to see how it performs on datasets out of this range. It is worth noting that all methods (three exemplars and CombineSEL) did not perform well on the Music dataset. While CombinedSEL managed to achieve a considerable improvement over the previous approaches, it still only reached a peak performance around 0.45, which is way below the desirable value (0.7 and above). Below are some possible

starting points for future algorithmic improvements:

1. As we discussed before when we were evaluating the effect of changing #bootstraps k, although we have eased the problem of random sampling from example pairs with duplicated scores, our approach did not solve the problem completely. A convenient improvement can be introducing more type of classifiers into the committee, thus more individual classifiers could be trained.

2. Through our critical analysis of the three exemplars (Chapter 5), we synthesised the approaches into two line of strategies for "User Interaction": "Maximise #Labels Back" and "Optimise Queries Asked". Our CombinedSEL is build upon the synthesis of ALIAS and Mozafari, which all followed the line of "Optimise Queries Asked"; AdInTDS, which followed the line of "Maximise #Labels Back", was abandoned due to its poor performance. However, maximising #Labels back is still a good idea. For the future work, we want to exploit both strategies.

3. It is possible to use k-Fold Cross Validation to maximally exploit our training set. We can thus stop acquiring more data once model performance has reached a reasonable value. However, we first need to confirm that estimated value generated by k-Fold Cross Validation is reasonably close to the true F1-Scores.

Appendix A

Precision and Recall Graphs for all datasets

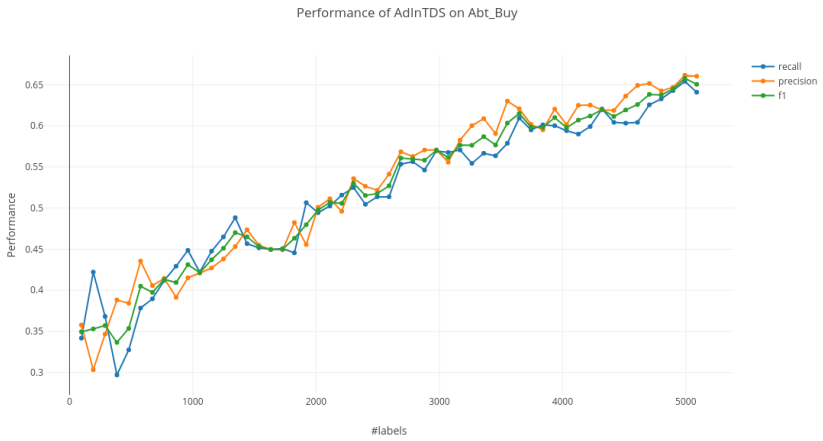

Figure A.1: Abt_Buy: performance against budget of AdInTDS

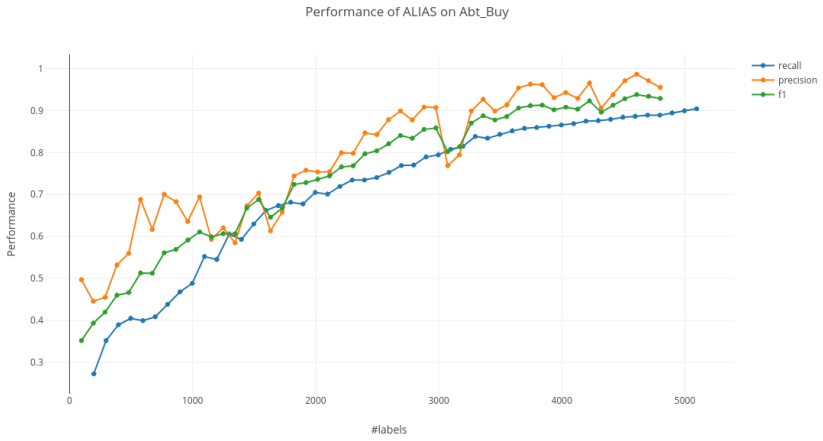

Figure A.2: Abt_Buy: performance against budget of ALIAS

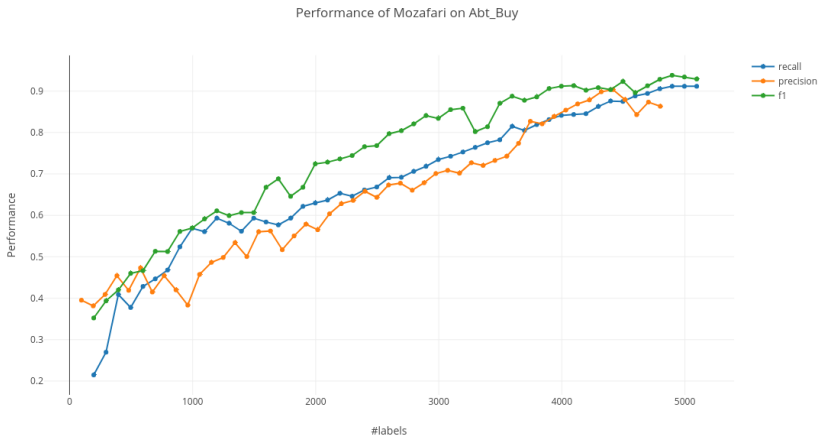

Figure A.3: Abt_Buy: performance against budget of Mozafari

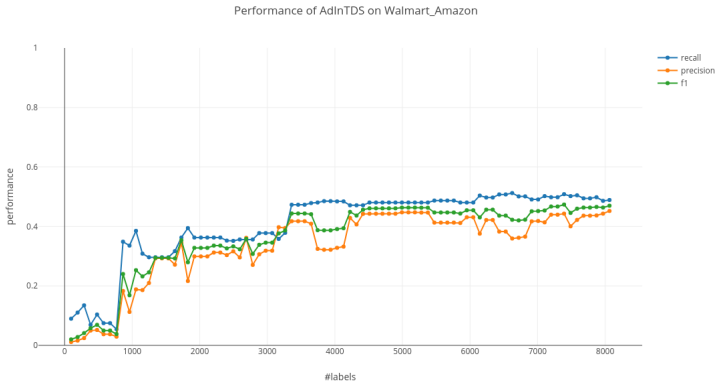

Figure A.4: Walmart_Amazon: performance against budget of AdInTDS

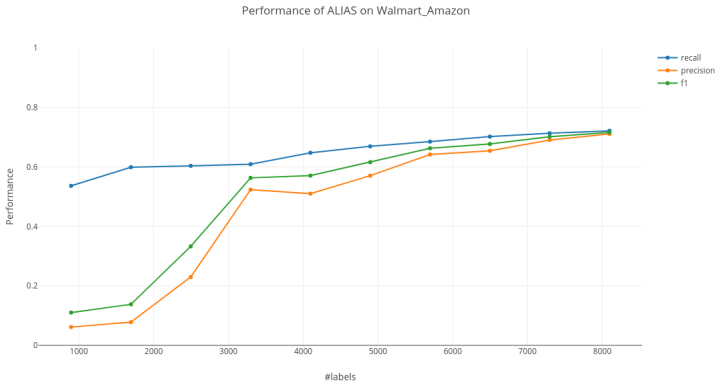

Figure A.5: Walmart_Amazon: performance against budget of ALIAS

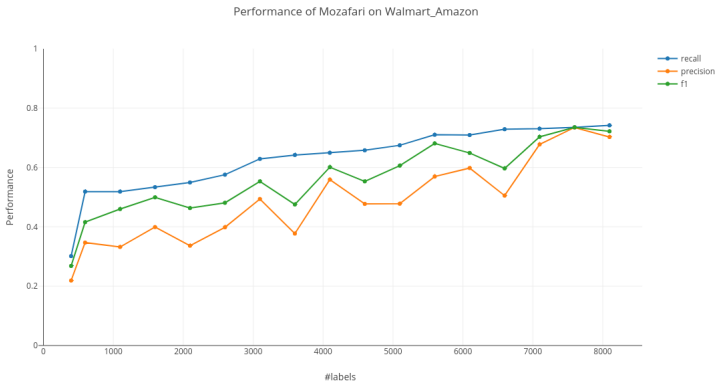

Figure A.6: Walmart_Amazon: performance against budget of Mozafari

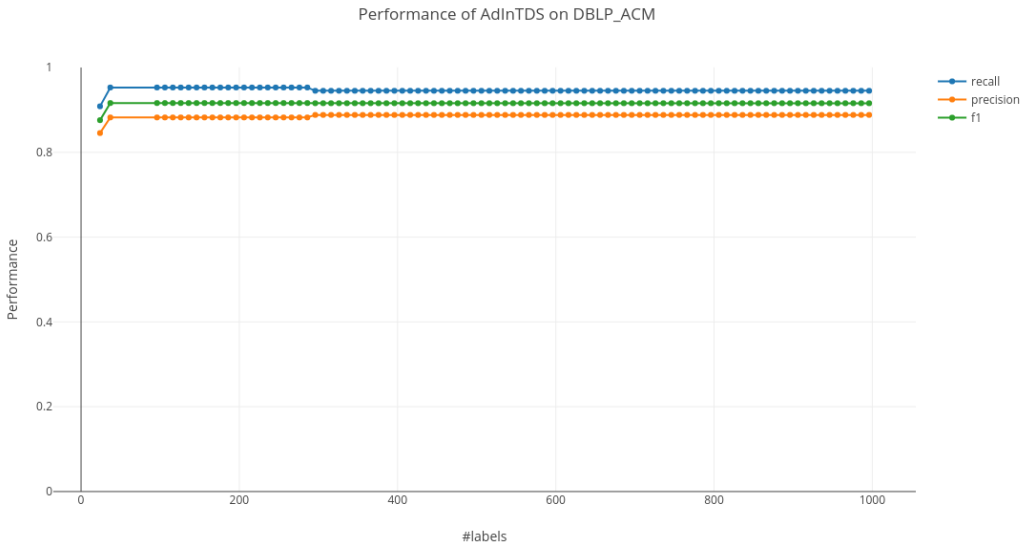

Figure A.7: DBLP_ACM: performance against budget of AdInTDS

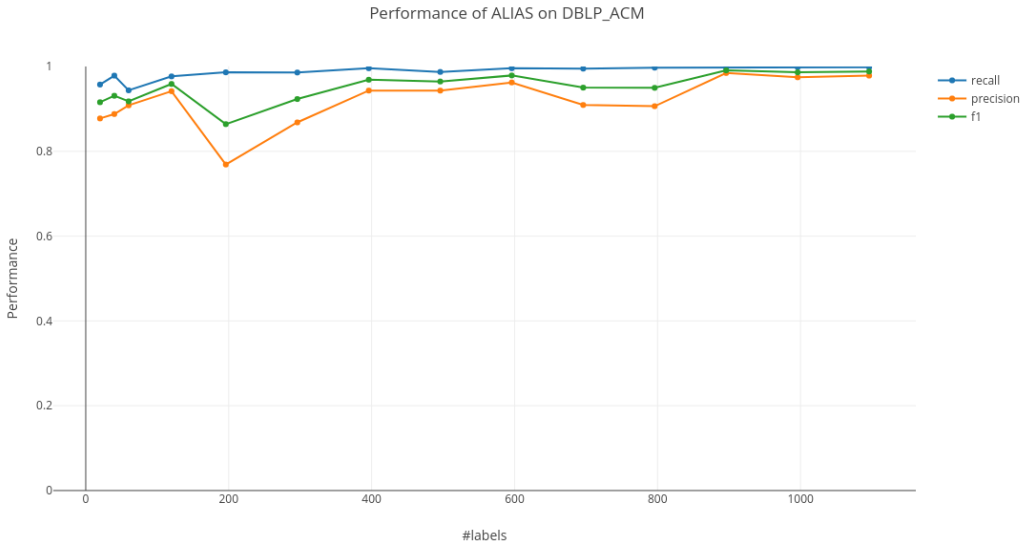

Figure A.8: DBLP_ACM: performance against budget of ALIAS

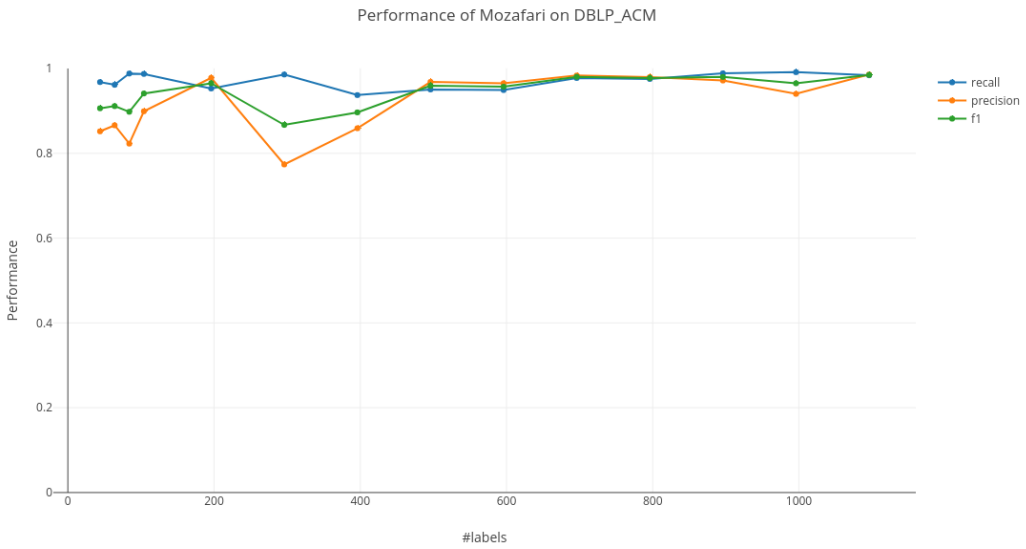

Figure A.9: DBLP_ACM: performance against budget of Mozafari

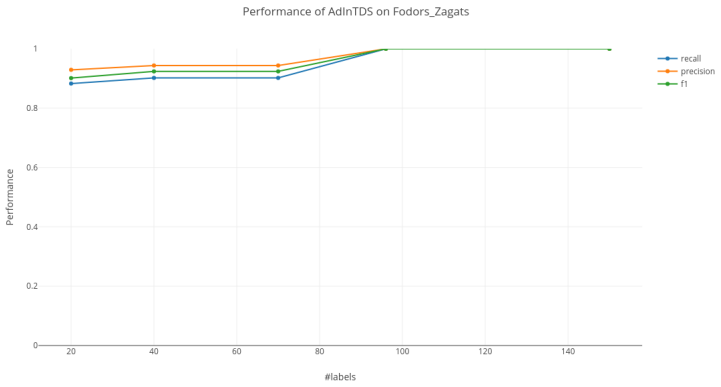

Figure A.10: Fodors_Zagats : performance against budget of AdInTDS

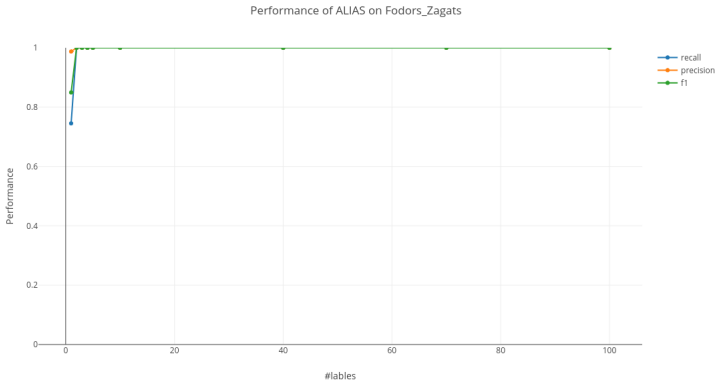

Figure A.11: Fodors_Zagats : performance against budget of ALIAS

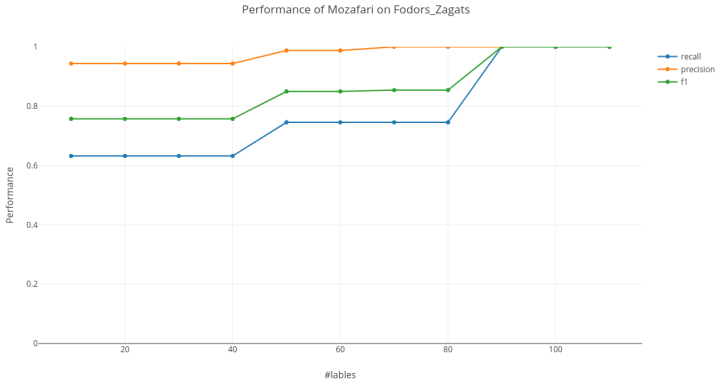

Figure A.12: Fodors_Zagats : performance against budget of Mozafari

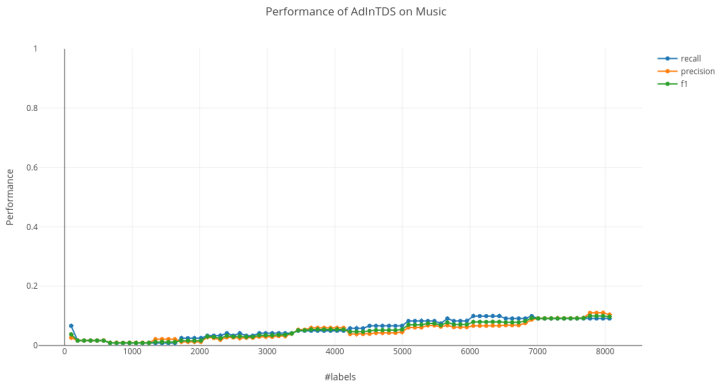

Figure A.13: Music: performance against budget of AdInTDS

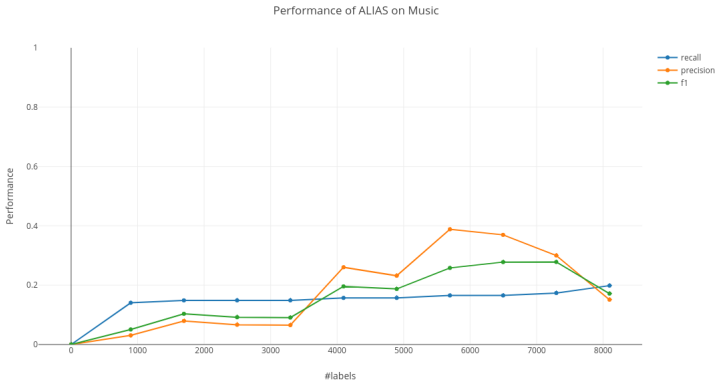

Figure A.14: Music: performance against budget of ALIAS

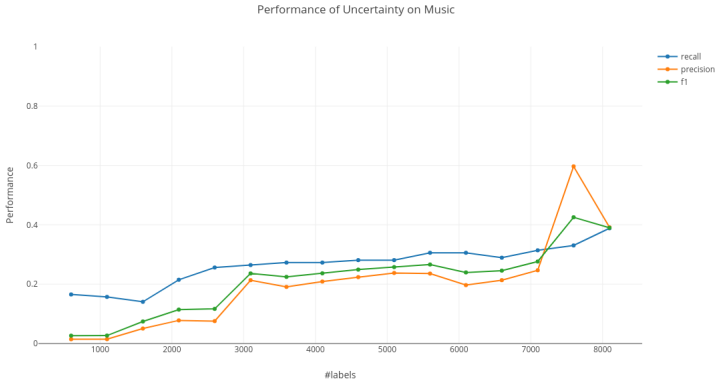

Figure A.15: Music: performance against budget of Mozafari

Appendix B

Parameters Used In Experiments

Dataset	AdInTDS				
	total budget	**min purity**	**sample error margin**	**min cluster size**	**max cluster size**
Abt_Buy	5000	0.95	0.1	20	1000
Walmart_Amazon	8000	0.95	0.1	20	1000
DBLP_ACM	1000	0.95	0.1	20	1000
Fodors_Zagats	100	0.95	0.1	5	20
Music	8000	0.95	0.1	20	1000

(a) AdInTDS

Dataset	ALIAS		
	total budget	**sample error margin**	**batch size**
Abt_Buy	5000	0.1	200
Walmart_Amazon	8000	0.1	800
DBLP_ACM	1000	0.1	100
Fodors_Zagats	100	0.1	5
Music	8000	0.1	800

(b) ALIAS

Dataset	Mozafari			
	#bootstraps	**total budget**	**sample error margin**	**batch size**
Abt_Buy	5	5000	0.1	100
Walmart_Amazon	5	8000	0.1	500
DBLP_ACM	5	1000	0.1	100
Fodors_Zagats	5	100	0.1	10
Music	5	8000	0.1	800

(c) Mozafari

Table B.1: Parameters used in experiments

Bibliography

[1] Howard B. Newcombe and James M. Kennedy. Record linkage: making maximum use of the discriminating power of identifying information. 5:563–566, 11 1962.

[2] Peter Christen. A survey of indexing techniques for scalable record linkage and deduplication. *IEEE transactions on knowledge and data engineering*, 24(9):1537–1555, 2012.

[3] Erhard Rahm and Hong Hai Do. Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull.*, 23(4):3–13, 2000.

[4] Mikhail Bilenko, Beena Kamath, and Raymond J Mooney. Adaptive blocking: Learning to scale up record linkage. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 87–96. IEEE, 2006.

[5] Mayank Kejriwal and Daniel P Miranker. An unsupervised algorithm for learning blocking schemes. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 340–349. IEEE, 2013.

[6] Matthew Michelson and Craig A Knoblock. Learning blocking schemes for record linkage. In *AAAI*, pages 440–445, 2006.

[7] Peter Christen. *Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection*. Springer Science & Business Media, 2012.

[8] Munir Cochinwala, Verghese Kurien, Gail Lalk, and Dennis Shasha. Efficient data reconciliation. *Information Sciences*, 137(1-4):1–15, 2001.

[9] Mikhail Bilenko and Raymond J Mooney. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 39–48. ACM, 2003.

[10] Peter Christen. Automatic record linkage using seeded nearest neighbour and support vector machine classification. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 151–159. ACM, 2008.

[11] Rahul Gupta and Sunita Sarawagi. Answering table augmentation queries from unstructured lists on the web. *Proceedings of the VLDB Endowment*, 2(1):289–300, 2009.

[12] Sanjoy Dasgupta and Daniel Hsu. Hierarchical sampling for active learning. In *Proceedings of the 25th international conference on Machine learning*, pages 208–215. ACM, 2008.

[13] Gholamreza Haffari, Maxim Roy, and Anoop Sarkar. Active learning for statistical phrase-based machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 415–423. Association for Computational Linguistics, 2009.

[14] Jiannan Wang, Tim Kraska, Michael J Franklin, and Jianhua Feng. Crowder: Crowdsourcing entity resolution. *Proceedings of the VLDB Endowment*, 5(11):1483–1494, 2012.

[15] Anthony Brew, Derek Greene, and Pádraig Cunningham. Using crowdsourcing and active learning to track sentiment in online media. In *ECAI*, pages 145–150, 2010.

[16] Sunita Sarawagi and Anuradha Bhamidipaty. Interactive deduplication using active learning. In *Proceedings of the Eighth ACM SIGKDD International*

Conference on Knowledge Discovery and Data Mining, KDD '02, pages 269–278, New York, NY, USA, 2002. ACM.

[17] Shlomo Argamon-Engelson and Ido Dagan. Committee-based sample selection for probabilistic classifiers. *Journal of Artificial Intelligence Research*, 11:335–360, 1999.

[18] Kamal Nigam and Andrew McCallum. Employing em in pool-based active learning for text classification. In *Proceedings of ICML-98, 15th International Conference on Machine Learning*, volume 31, page 32, 1998.

[19] Sheila Tejada, Craig A Knoblock, and Steven Minton. Learning domain-independent string transformation weights for high accuracy object identification. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 350–359. ACM, 2002.

[20] Junio De Freitas, Gisele L Pappa, Altigran S da Silva, Marcos A Gonc, Edleno Moura, Adriano Veloso, Alberto HF Laender, Moisés G de Carvalho, et al. Active learning genetic programming for record deduplication. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8. IEEE, 2010.

[21] Robert Isele and Christian Bizer. Active learning of expressive linkage rules using genetic programming. *Web Semantics: Science, Services and Agents on the World Wide Web*, 23:2–15, 2013.

[22] Arvind Arasu, Michaela Götz, and Raghav Kaushik. On active learning of record matching packages. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 783–794. ACM, 2010.

[23] Kedar Bellare, Suresh Iyengar, Aditya G Parameswaran, and Vibhor Rastogi. Active sampling for entity matching. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1131–1139. ACM, 2012.

[24] Victor S Sheng, Foster Provost, and Panagiotis G Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 614–622. ACM, 2008.

[25] Panagiotis G Ipeirotis, Foster Provost, Victor S Sheng, and Jing Wang. Repeated labeling using multiple noisy labelers. *Data Mining and Knowledge Discovery*, 28(2):402–441, 2014.

[26] Guilherme Dal Bianco, Renata Galante, Carlos A Heuser, Marcos Gonçalves, and Sergio Canuto. A practical and effective sampling selection strategy for large scale deduplication. In *Data Engineering (ICDE), 2016 IEEE 32nd International Conference on*, pages 1518–1519. IEEE, 2016.

[27] Chicheng Zhang and Kamalika Chaudhuri. Active learning from weak and strong labelers. In *Advances in Neural Information Processing Systems*, pages 703–711, 2015.

[28] Xuezhi Wang, Jie Tang, Hong Cheng, and S Yu Philip. Adana: Active name disambiguation. In *2011 11th IEEE International Conference on Data Mining*, pages 794–803. IEEE, 2011.

[29] Pinar Donmez, Jaime G. Carbonell, and Jeff Schneider. Efficiently learning the accuracy of labeling sources for selective sampling. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 259–268, New York, NY, USA, 2009. ACM.

[30] Yan Yan, Romer Rosales, Glenn Fung, and Jennifer G Dy. Active learning from crowds. In *ICML*, volume 11, pages 1161–1168, 2011.

[31] Weining Wu, Yang Liu, Maozu Guo, Chunyu Wang, and Xiaoyan Liu. A probabilistic model of active learning with multiple noisy oracles. *Neurocomputing*, 118:253–262, 2013.

[32] Chaoqun Li, Victor S Sheng, Liangxiao Jiang, and Hongwei Li. Noise filtering to improve data and model quality for crowdsourcing. *Knowledge-Based Systems*, 107:96–103, 2016.

[33] J. Du and C. X. Ling. Active learning with human-like noisy oracle. In *2010 IEEE International Conference on Data Mining*, pages 797–802, Dec 2010.

[34] Christopher H Lin, M Mausam, and Daniel S Weld. Re-active learning: Active learning with relabeling. In *AAAI*, pages 1845–1852, 2016.

[35] Liyue Zhao, Gita Sukthankar, and Rahul Sukthankar. Incremental relabeling for active learning with noisy crowdsourced annotations. In *SocialCom/PASSAT*, pages 728–733, 2011.

[36] Florian Laws, Christian Scheible, and Hinrich Schütze. Active learning with amazon mechanical turk. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1546–1556. Association for Computational Linguistics, 2011.

[37] Barzan Mozafari, Purna Sarkar, Michael Franklin, Michael Jordan, and Samuel Madden. Scaling up crowd-sourcing to very large datasets: a case for active learning. *Proceedings of the VLDB Endowment*, 8(2):125–136, 2014.

[38] Maytal Saar-Tsechansky and Foster Provost. Active sampling for class probability estimation and ranking. *Machine learning*, 54(2):153–178, 2004.

[39] Ming-Hen Tsai, Chia-Hua Ho, and Chih-Jen Lin. Active learning strategies using svms. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pages 1–8. IEEE, 2010.

[40] Jeffrey Fisher, Peter Christen, and Qing Wang. Active learning based entity resolution using markov logic. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 338–349. Springer, 2016.

[41] Kun Qian, Lucian Popa, and Prithviraj Sen. Active learning for large-scale entity resolution. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1379–1388. ACM, 2017.

[42] Sudheendra Vijayanarasimhan and Kristen Grauman. Cost-sensitive active visual category learning. *International Journal of Computer Vision*, 91(1):24–44, 2011.

[43] Peter Christen, Dinusha Vatsalan, and Qing Wang. Efficient entity resolution with adaptive and interactive training data selection. In *Data Mining (ICDM), 2015 IEEE International Conference on*, pages 727–732. IEEE, 2015.