Entity Resolution Evaluation Measures

Hitesh Maidasani, Galileo Namata, Bert Huang, and Lise Getoor
Department of Computer Science
University of Maryland
College Park, MD 20742, USA
hitmai@terpmail.umd.edu, {namatag, bert, getoor} @cs.umd.edu

May 17, 2012

Abstract

Entity Resolution (ER) is the task of finding and merging entities within a single data source or across several data sources that represent the same real world entity. Evaluating ER results is a very important procedure used to ensure how accurate and correct an ER algorithm is. There have been several measures proposed and used for ER evaluation (e.g., pairwise F_1, B^3 F_1, CEAF). With so many new and existing evaluation measures, there is a need to survey, characterize, and evaluate these measures. In this paper, we explore these ER evaluation measures. First, we provide a general definition to the ER problem. Next, we define many of the widely used measures, including a discussion of the different advantages and disadvantages of each measure. We also discuss trends in the use of the ER evaluation measures based across multiple domains. Finally, we provide examples of ER predictions and compute several of the measures we discuss to highlights differences and overlaps in the measures. In doing so, we hope to provide practitioners with a practical guide for understanding ER evaluation.

1 Introduction

Entity Resolution (ER) is the problem of identifying and merging references within text or across several data sources that represent the same real world entity. It is important to properly understand ER since it is such a widely occurring problem

[22, 6, 27]. For example, when a company is merging two databases of customer financial data, the same customer might be represented differently as shown in Table 1.

Table 1: Customer information within two databases

Database	**Name**	**Date of Birth**	**Address**
Database 1	John Doe	11/11/1990	College Park, MD
Database 1	J. Doe	-	Maryland, USA
Database 2	John K. Doe	Nov. 1990	College Park, MD 20742
Database 2	Jon Doe	-	MD, USA
Database 3	J. K. Doe	1990	College Park, MD

In the above example, the first three customer records or references (John Doe, J. Doe, and John K. Doe) actually refer to same person and the last two (Jon Doe, J. K. Doe) refer to a different person. As this example shows, ER can be a very difficult problem. Duplicates exist within the same database and across different databases with varying levels of quality. The amount and quality of available information often make it very difficult to resolve any ambiguities in the data. In this simple example, the ER algorithms must take into account missing data or different ways of representing data (e.g., date of birth, addresses). In practice, however, there also temporal (e.g., transactional) and relational information which also need to be considered.

The Entity Resolution (ER) problem exists in many different domains and is given different names within different domains. In the computer vision and image processing domain, one version of the the problem is called "object identification" where the task is to match each object in a video to a corresponding person [32]. In natural language processing, the problem is called "coreference resolution" where the task is to determine which noun phrases refer to the same entity [6]. In the database domain, the task of removing duplicates while merging two or more databases is called "database merging" or "merge/purge processing", and removing duplicates from a single database is called "deduplication" [6], "data alignment," and "Entity matching" [25]. The machine learning domain uses several of the above names including "entity resolution" [6], "entity matching" [15], and "deduplication" [2].

There have been many different approaches to ER [22, 6, 27]. As the number of these proposed approaches grows, so does the importance of properly evaluating how accurate and complete ER predictions are. This paper provides an in-depth analysis of several ER evaluation techniques. In Section 2, we provide a general overview of the ER problem. We define many of the widely used measures in Section 3, including a discussion of the different advantages and disadvantages of each measure. We also discuss trends in the use of the ER evaluation measures based across multiple domains. Finally, in Section 4 we provide examples of ER predictions and compute several of the measures we discuss to highlight the differences and commonalities in the measures.

2 Entity Resolution Evaluation

2.1 Definition of Entity Resolution

Entity resolution, in its most general sense, involves reasoning over a given a set of ambiguous references $R = \{r_i\}$ to some unknown set of entities. The objective of ER algorithms is to identify, for each $r_i, r_j \in R$, whether r_i and r_j refer to the same real world entity (i.e., coreferent) given the attributes and relationships of the references.

The predictions of ER algorithm can naturally be represented as either "pairs" or "clusters" of references. The pairs of references represent the pairs which the ER models have predicted as coreferent (i.e., they refer to the same underlying entity). We denote pairs by (r_i, r_j), where r_i and r_j are individual references predicted coreferent. In some applications, however, the transitivity of this relation (i.e., if (r_i, r_j) and (r_j, r_k), then (r_i, r_k)) may need to be enforced. In those cases, a more natural representation is a set of references (we refer to as a cluster) whose element references are coreferent to each other. We denote a cluster of nodes as $\{r_1, r_2, ..., r_k\}$. We use "cluster" or "entity" interchangeably as a cluster represents the set of references which are predicted to the same real world entity.

3 Overview of Evaluation Measures

There are several ways of evaluating ER. Evaluation of an ER algorithm involves checking how correct the ER predictions are compared to some previously annotated the ground truth. This comparison is done with an evaluation measure which signify how close the predicted result is to some annotated ground truth of pairs or clusters. We list many of the most commonly used measures below separated into three main categories: pairwise, cluster, and edit distance. We define each of the categories in the following sections. Variants of different measures in sublist of the original measure. In addition, the most commonly used measures, discussed further in the following sections, are shown in bold:

1. Pairwise:
    - **Pairwise precision, recall, $\boldsymbol{F_1}$** [21]
2. Cluster:
    - **Cluster precision, recall, $\boldsymbol{F_1}$** [11] [21]
    - **Closest Cluster precision, recall, $\boldsymbol{F_1}$** [4] [21]
    - **MUC precision, recall, $\boldsymbol{F_1}$** [30]
    - **$\boldsymbol{B^3}$ precision, recall, $\boldsymbol{F_1}$** [3]
        (a) B^3all precision, recall, F_1 [29] [28] [8]
        (b) B^30 precision, recall, F_1 [29] [28] [8]
        (c) $B^3_{r\&n}$ precision, recall, F_1 [24] [8].
        (d) B^3_{sys} precision, recall, F_1 [8]
    - **Constrained Entity-Alignment F-Measure (CEAF)** [18]
        (a) CEAF$_{r\&n}$ precision, recall, F_1 [24] [8]
        (b) CEAF$_{sys}$ precision, recall, F_1 [8]
        (c) CONE CEAF (Constrained Entity-Alignment F-Measure [17]
    - CONE B^3 precision, recall, F$_1$ [17]
    - Automatic Content Extraction (ACE) evaluation score [9]
3. Edit Distance:
    - **Basic Merge Distance (BMD)** [1] [21]
    - Generalized Merge Distance (GMD) [21]
    - Variation of Information (VI) [20]

We note that many measures presented above are based on measuring the "precision" and "recall" of the predicted pairs or clusters. While the different measures vary on exactly how each is computed, the general definition for these terms are consistent. Precision is the fraction of the predicted pairs or clusters in the result that "match" the ground truth for various definitions of what a "match" includes. In other words, precision is a measure of the correctness of the predicted results relative to the ground truth. We can formalize precision based with a widely used statistical approach:

$$\text{precision} = \frac{tp}{tp + fp}$$

where tp is short for true positives and fp is short for false positives. True positives are the number of correctly predicted coreferences. False positives are the number of falsely predicted coreferences.

Recall, on the other hand, is the fraction of truths that are successfully "present" in the result. In other words, recall is a measure of the completeness of the predicted coreferences. We can also formalize recall based with a widely used statistical approach:

$$\text{recall} = \frac{tp}{tp + fn}$$

where tp is short for true positives and fn is short for false negative. True positives are defined above. False negatives are the number of true coreferences that have not been predicted.

We note that there are natural trade-offs between precision and recall. For instance, if the result falsely predicts that all the references correspond to the same entity, then the precision would be low as many of these coreferences are false, but the recall would be high because all the coreferences are captured. Inversely, if the result only predicts a very small number of coreferences, then the precision maybe high because the few predicted are correct, but the recall may be low because most coreference would be missed. To capture this trade-off between precision and recall, the F_1 score, the harmonic mean of the two measures, is often used. F_1 gives a single measure consisting of features of precision and recall and is computed as follows:

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

3.1 Pairwise Measures

The pairwise evaluation measures compute all the pairs of references to evaluate ER results. Each pair represents a link between two references. All the pairs represent all the possible combinations between each reference. Result pairs are compared to the truth pairs. Given a truth T and result R set, all pairs between references are computed for both the T and R resulting in pairedR and pairedT. If $T = \{r_1, r_2, r_3\}$, pairedT = $\{(r_1, r_2), (r_1, r_3), (r_2, r_3)\}$.

3.1.1 Pairwise F_1

Pairwise F_1 is one of the oldest ER measures still used today. The reason for this may be the simplicity of computing it. Of the fifteen papers we surveyed, thirteen use a pairwise measure, of which twelve use Pairwise F_1 and one just uses pairwise precision and pairwise recall. As we mentioned, pairwise measures use pairs to represent links between references.

Pairwise precision is the fraction of reference pairs in the result R that are also in the truth T.

$$\text{PairwisePrecision}(T, R) = \frac{|\text{pairedR } \cap \text{ pairedT}|}{|\text{pairedR}|}$$

Pairwise recall is the fraction of reference pairs in the truth T that are also in the result R.

$$\text{PairwiseRecall}(T, R) = \frac{|\text{pairedR } \cap \text{ pairedT}|}{|\text{pairedT}|}$$

Pairwise F_1 is the harmonic mean of the pairwise precision and pairwise recall.

$$\text{Pairwise } F_1 = 2 \cdot \frac{\text{PairwisePrecision } \cdot \text{ PairwiseRecall}}{\text{PairwisePrecision } + \text{ PairwiseRecall}}$$

One of the benefits of pairwise F_1 is the ease of representing each pair as a possible link between references. In addition, a pairwise representation is good way of separating references where transitivity doesn't hold. Pairwise measures are good measures for applications that use pairs, such as active learning. One of the disadvantages is that a pair may not necessarily be the natural way of representing links between references, because of the lack of transitivity between multiple pairs. In other words, in some applications, where transitivity is required, the predicted pairs may not exhibit transitivity. We mentioned above that this way of separating references without transitivity may be an advantage, but many applications consider coreference between multiple references as a transitive relation. Another disadvantage is that pairwise metrics cannot represent singleton entities, which are entities that are mentioned only once [8].

3.2 Cluster Measures

The cluster evaluation measures compares clusters to evaluate ER results. A cluster represents linked references. Result clusters are compared to the truth clusters. Given a truth T and result R set of clusters e.g. $T = \{\{r_1, r_2, r_3, r_4, r_5\}, \{r_6, r_7\}, \{r_8, r_9, r_{10}, r_{11}, r_{12}\}\}$ and $R = \{\{r_1, r_2, r_3, r_4, r_5\},$
$\{r_6, r_7, r_8, r_9, r_{10}, r_{11}, r_{12}\}\}$. A cluster within the set T is denoted by t. A cluster within the set R and is denoted by r. The entry t_i represents a given reference within a cluster t e.g. t is $\{r_1, r_2, r_3, r_4, r_5\}$ and t_1 is r_1. The same holds for r_i.

3.2.1 Cluster F_1

Huang et al. [11] [21] proposed cluster F_1 in 2006. The cluster F_1 compares at the cluster level instead of the reference level and counts the clusters that exactly match [21].

Cluster precision is the fraction of the number of completely correct clusters to the total number of clusters retrieved in the result [11].

$$\text{ClusterPrecision}(T, R) = \frac{|R \cap T|}{|R|}$$

Cluster recall is the fraction of the number of completely correct clusters to the total number of true clusters [11].

$$\text{ClusterRecall}(T, R) = \frac{|R \cap T|}{|T|}$$

Cluster F1 is the harmonic mean between the cluster recall and the cluster precision.

$$\text{Cluster } F_1 = 2 \cdot \frac{\text{ClusterPrecision } \cdot \text{ ClusterRecall}}{\text{ClusterPrecision } + \text{ ClusterRecall}}$$

The advantage of cluster F_1 is that it checks for completely correct clusters and this may be a more reasonable way of checking for coreference, rather than giving credit for partially correct clusters done in pairwise measures. This may also be a disadvantage. Cluster F_1 does not give credit to partially correct clusters that miss a few references, since it checks for completely correct clusters [11]. This makes cluster F_1 more strict and less informative in some cases than the pairwise measures that measure at the reference level [11]. For this reason, cluster F_1 is a good measure for applications that need exact or strict matching.

3.2.2 Closest Cluster F_1

Benjelloun et al. [4] [21] proposed closest cluster F_1 in 2008. The closest cluster F_1 sums the similarities of all "closest" cluster pairs consisting a result cluster paired with a truth cluster. The "closest" cluster pair is found using the maximum Jaccard similarity between all result cluster and truth cluster pairs. Unlike cluster F_1, which checks for completely matching clusters, closest cluster F_1 matches clusters based on Jaccard similarity. In a way, this is similar to the optimal matching done in CEAF, except, here, Jaccard similarity is used. Here, we define Jaccard similarity between two clusters:

$$\text{Jaccard}(r, t) = \frac{|r \ \cap \ t|}{|r \ \cup \ t|}[6]$$

The closest cluster precision is the fraction of the sum of the maximum Jaccard similarity of result cluster and truth cluster pairs and the total number of clusters in the result [21].

$$\text{ClosestClusterPrecision}(T, R) = \frac{\sum_{r \in R} \max_{t \in T} \text{Jaccard}(r, t)}{|R|}$$

The closest cluster recall is the fraction of the sum of the maximum Jaccard similarity of truth cluster and result cluster pairs and the total number of clusters in the truth [21].

$$\text{ClosestClusterRecall}(T, R) = \frac{\sum_{t \in T} \max_{r \in R} \text{Jaccard}(t, r)}{|T|}$$

Closest cluster F1 is the harmonic mean between the closest cluster recall and the closest cluster precision.

$$\text{ClosestCluster } F_1 = 2 \cdot \frac{\text{ClosestClusterPrecision} \ \cdot \ \text{ClosestClusterRecall}}{\text{ClosestClusterPrecision} \ + \ \text{ClosestClusterRecall}}$$

An advantage of closest cluster F_1 is that is fairly easy to compute. A disadvantage of closest cluster F_1 is that it has not been used as much as the other cluster measures as it is a fairly newer measure.

3.2.3 MUC F_1

During the sixth Message Understanding Conference in 1995, Vilain et al. [30] proposed the MUC score, which gets its name from the conference name (MUC-6). The MUC score considers a cluster of references as linked references where each reference is linked to at most two other references. For example, cluster $\{r_1, r_2, r_3, r_4, r_5\}$ would have four links between the five references. MUC measures the number of link modifications required to make the result set identical to the truth set.

We define a function partition(c, S) that takes in a cluster c and a set of clusters S and returns the set of clusters within S that intersect with c.

$$\text{partition}(c, S) = \{s \mid s \in S \;\&\; s \cap c \neq \emptyset\}$$

For MUC precision, we can use $|\text{partition}(r, T)|$ to give us the number of clusters within the truth T that the recall cluster r intersects with. This number gives us the number of missing links for r. MUC precision is a sum for each cluster in the result. MUC precision is the fraction of the difference of the correct links and the missing links in the result cluster and the correct links cluster in the result cluster for each cluster in the result. Basically, MUC precision computes the minimum number of link modifications required to make the result set identical to the truth set.

$$\text{MUCPrecision}(T, R) = \sum_{r \in R} \frac{|r| - |\text{partition}(r, T)|}{|r| - 1}$$

For MUC recall, we can use $|\text{partition}(t, R)|$ to give us the number of clusters within the result R that the truth cluster t intersects with. This number gives us the number of missing links for t. MUC recall is a sum for each cluster in the truth. MUC recall is the fraction of the difference of the correct links and the missing links in the truth cluster and the correct links cluster in the truth cluster for each cluster in the truth. Basically, MUC precision computes the minimum number of link modifications required to make the truth set identical to the result set.

$$\text{MUCRecall}(T, R) = \sum_{t \in T} \frac{|t| - |\text{partition}(t, R)|}{|t| - 1}$$

MUC F_1 is the harmonic mean of the MUC precision and MUC recall.

$$\text{MUC } F_1 = 2 \cdot \frac{\text{MUCPrecision } \cdot \text{ MUCRecall}}{\text{MUCPrecision } + \text{ MUCRecall}}$$

MUC was one of the earliest cluster based measures, and had quite a few flaws. Compared to pairwise F_1 mentioned in Section 3.1.1, MUC F_1 can represent singleton entities [8]. However, MUC doesn't penalize separation of a singleton entity from a linked cluster [3][8]. Another disadvantage is that MUC considers all errors to be equal [3]. We can explain this disadvantage with an example. We are given the truth $T = \{\{r_1, r_2, r_3, r_4, r_5\}, \{r_6, r_7\}, \{r_8, r_9, r_{10}, r_{11}, r_{12}\}\}$ and the first result $R_1 = \{\{r_1, r_2, r_3, r_4, r_5\}, \{r_6, r_7, r_8, r_9, r_{10}, r_{11}, r_{12}\}\}$ and the second result $R_2 = \{\{r_1, r_2, r_3, r_4, r_5, r_8, r_9, r_{10}, r_{11}, r_{12}\}, \{r_6, r_7\}\}$. In R_1, seven references are falsely predicted to be the same entity, while in R_2, ten references are false predicted to be the same entity. We should expect the precision for R_2 to be lower than R_1, but the two results' MUC precisions are the same ($R_1 = 0.90$ and $R_2 = 0.90$) because MUC considers missing links as the same. To summarize, MUC only considers missing links between references, which leads to unintuitive evaluation results.

3.2.4 B^3F_1

B^3F_1 was proposed by Bagga and Baldwin [3] in 1998 to overcome the shortcomings of MUC described above. B^3F_1 is mostly used in the NLP disciple shown by our research in Table 3.

We define a function results(t_i) that takes a reference t_i as an input and returns a set of clusters within the result that contains t_i. We also define n to be the number of total references in the truth.

B^3 precision is the weighted precision for each reference t_i in the truth.

$$B^3\text{Precision}(T, R) = \frac{1}{n} \sum_{t \in T} \sum_{t_i \in t} \sum_{r \in \text{results}(t_i)} \frac{|r \ \cap \ t|}{|r|}$$

B^3 recall is the weighted recall for each reference t_i in the truth.

$$B^3\text{Recall}(T, R) = \frac{1}{n} \sum_{t \in T} \sum_{t_i \in t} \sum_{r \in \text{results}(t_i)} \frac{|r \ \cap \ t|}{|t|}$$

B^3 F_1 is the harmonic mean of the B^3 precision and B^3 recall.

$$B^3 \ F_1 = 2 \cdot \frac{B^3\text{Precision} \ \cdot \ B^3\text{Recall}}{B^3\text{Precision} \ + \ B^3\text{Recall}}$$

Like MUC, B^3 F_1 can also represent singleton entities [8]. The biggest advantage of B^3 F_1, as we mentioned above, is that eliminates the flaws of MUC.

Primarily, that all errors are not considered to be equal. One disadvantage of B^3 is that it assumes that the references in the result to be identical to the truth [8]. B^3 does not deal with references that are not in the truth, called twinless mentions [8] [28]. Twinless mentions can be described with an NLP example, since they arise in NLP applications. For instance, there is an application that is using noun phrases to identify presidents from a documents that talk about presidents and their family life with another document that only talks about presidential campaigns. Assuming that names of family members of presidents are only mentioned in the first document, these names are twinless mentions.

Stoyanov et al. [29] [28] [8] propose B^3all and B^3_0 in 2009 to deal with twinless mentions. B^3_0 discards twinless mentions, while B^3all retains twinless mentions [8]. Rahman and Ng [24] [8] propose $B^3_{r\&n}$ in 2009 to handle twinless mentions based on singletons. Cai [8] proposed B^3_{sys} in 2010 to handle twinless mentions more adequetely compared to the previous variants. Lin et al. [17] proposed CONE B^3 F_1, which is based on approximation algorithms, in 2010.

3.2.5 CEAF

Luo [18] proposed Constrained Entity-Alignment F-Measure (CEAF) in 2005. Luo criticized B^3 because it uses clusters more than once in computing precision and recall [18] [8]. CEAF uses similarity measures to first create an optimal mapping between result clusters and truth clusters. Using the optimal mapping, CEAF computes the precision and recall using self-similarity and one the similarity measures (ϕ) described below:

$$\phi_1(T,R) = \begin{cases} 1, \text{if } R = T \\ 0, \text{otherwise} \end{cases}$$

$$\phi_2(T,R) = \begin{cases} 1, \text{if } R \cap T \neq \emptyset \\ 0, \text{otherwise} \end{cases}$$

$$\phi_3(T,R) = |R \cap T|$$

$$\phi_4(T,R) = \frac{2 \cdot |R \cap T|}{|R| + |T|}$$

In most uses of CEAF, both similarity measures ϕ_3 and ϕ_4 are used to give CEAF-ϕ_3 and CEAF-ϕ_4 respectively.

We define a function $m(r)$ that takes in a cluster r and returns the true cluster t that result cluster r is mapped to, with the constraint that one true cluster can be mapped to at most one result cluster. We assume that if the result cluster r does not get mapped, m(r) returns the empty set. In other words, $m(r)$ returns the optimal mapping for result cluster r in the truth.

CEAF precision is the fraction of the score of the optimal match and the score for mapping the result to itself (or self similarity).

$$\text{CEAF}\phi_{\text{i}}\text{Precision}(T, R) = \frac{\max_m \sum_{r \in R} \phi_i(r, m(r))}{\sum_{r \in R} \phi_i(r,\ r)}$$

CEAF recall is the fraction of the score of the optimal match and the score for mapping the truth to itself (or self similarity).

$$\text{CEAF}\phi_{\text{i}}\text{Recall}(T, R) = \frac{\max_m \sum_{r \in R} \phi_i(r,\ m(r))}{\sum_{t \in T} \phi_i(t,\ t)}$$

CEAF F_1 is the harmonic mean of the CEAF precision and CEAF recall.

$$\text{CEAF}\phi_{\text{i}}\ F_1 = 2 \cdot \frac{\text{CEAF}\phi_{\text{i}}\text{Precision}\ \cdot\ \text{CEAF}\phi_{\text{i}}\text{Recall}}{\text{CEAF}\phi_{\text{i}}\text{Precision}\ +\ \text{CEAF}\phi_{\text{i}}\text{Recall}}$$

As mentioned above, the two most used similarity measures for CEAF are ϕ_3 and ϕ_4. If we look at CEAFϕ_3 and CEAFϕ_4, for both precision and recall, closely, we can see that CEAFϕ_3 is normalized by the number of references while ϕ_4 is normalized by the number of clusters or entities. In other words, CEAFϕ_3 is reference-based and CEAFϕ_4 is entity-based.

An advantage of CEAF over B^3 is that clusters are not used more than once in computing CEAF precision and recall. Rather, the optimal mapping between each cluster is used. In other words, entities won't receive double credit. In addition, this optimal mapping may be a more practical approach for evaluation. One of CEAF's disadvantages is that there is considerably more computation required compared to other measures. Another disadvantage of CEAF is that it does not handle twinless mentions, introduced in Section 3.2.4, since twinless mentions are not mapped to the truth [8].

Rahman and Ng [24] [8] proposed CEAF$_{r\&n}$ in 2009. Cai [8] proposed CEAF$_{sys}$ in 2010 to handle twinless mentions. Lin et al. [17] proposed CONE CEAF, which is based on approximation algorithms, in 2010.

3.2.6 Other Variants

The ACE evaluation score was initially proposed during the Automatic Content Extraction program in 1999 [9]. It is a successor of MUC, but a predecessor of CEAF. Like CEAF, it, too, does optimal mapping which is described above. Once the optimal matching clusters have been found between the result and truth, the precision and recall are calculated, between optimal matches. Unlike CEAF, ACE calculates precision and recall based on the true positive, false positive, false negative approach presented in Section 3. In addition, this approach doesn't normalize the precision and recall in the way CEAF precision and recall do, which is mentioned above.

3.3 Edit Distance Measures

The edit distance measures are based on cluster splits and merges needed to convert the result to the truth. This type of comparison is a natural way of comparing ER results which represents the number of changes required to correct the result. We believe that there is more research required in edit distance approaches for ER evaluation.

Given a truth T and result R set of clusters e.g. $T = \{\{r_1, r_2, r_3, r_4, r_5\}, \{r_6, r_7\}, \{r_8, r_9, r_A, r_B, r_C\}\}, R = \{\{r_1, r_2, r_3, r_4, r_5\}, \{r_6, r_7, r_8, r_9, r_A, r_B, r_C\}\}$. A cluster within the set T is denoted by t. A cluster within the set R is denoted by r. t_i represents a given reference within a cluster t e.g. t is $\{r_1, r_2, r_3, r_4, r_5\}$ and t_1 is r_1. The same holds for r_i.

3.3.1 Basic Merge Distance (BMD)

Al-Kamha et al. [1] [21] proposed Basic Merge Distance (BMD) in 2004. Basic Merge Distance (BMD) counts the number of splits and merges necessary to convert a result to a truth set. Below, we formalize BMD with a simple equation.

$$\phi_2(T, R) = \begin{cases} 1, \text{if } R \cap T \neq \emptyset \\ 0, \text{otherwise} \end{cases} \text{(same as above)}$$

$$\text{BMD}(T, R) = \frac{2 \cdot \sum_{t \in T} \sum_{r \in R} \phi_2(t,\ r) - |T| - |R|}{|T| - 1}$$

The denominator in the above equation ($|T| - 1$) is used to normalize BMD. The reason for this normalization is that a result with a smaller number of clusters would require less splits and merges to fix compared to a result with more clusters. We have formalized BMD above, but BMD actually involves a path of splits and merges. We define a split that takes in a cluster c and returns two split clusters that do not overlap.

$$\text{Split}(c) = \{c_i, c_j \mid c_i \cap c_j = \emptyset, c_i \cup c_j = c, c_i, c_j \neq \emptyset\}[21]$$

We define a merge that takes in two clusters c_i, c_j and returns a single combined cluster.

$$\text{Merge}(c_i, c_j) = \{c \mid c = c_i \cup c_j\}[21]$$

BMD is the number of splits and merges required to fix a result. BMD has the restriction that splits must occur before merges. BMD is normalized by the number of clusters in the result. In our examples described below, we represent BMD to be $1-$BMD in order to get a better comparison with the other measures.

3.3.2 Other Variants

Variation of Information (VI) was proposed by Meila [20] [21] in 2003. VI is a distance measure between two clusterings. It measures information lost and gained while converting one clustering to the other [21]. Menestrina and Whang [21] proposed Generalized Merge Distance (GMD) as a better edit distance approach to ER evaluation. GMD computes the shortest edit distance from an ER result to a truth using merges and splits, similar to BMD, on clusters [21]. An advantage of GMD is that merge and split costs can be configured based on cluster sizes [21]. Menestrina and Whang propose that GMD can be standard way of evaluating ER because of its similarity to pairwise F_1, VI and BMD.

4 Analysis and Discussion

4.1 Survey of Measures

We have analyzed several major ER papers from different domains. We have analyzed the ER measures they use to get an insight of trends of use of ER measures. Our survey consists of twenty-one ER papers and this analysis is shown in Table 2. We have analyzed a distribution of major papers from the major domains that evaluate ER. These domains are databases, vision, natural language processing, and machine learning. The table shows the individual ER measures used in each paper.

We see that pairwise measures are one of the most used measures. As seen in our survey, pairwise F_1 is the only measure found in all four of the domains we

Table 2: Table of measures

Paper	**Domain**	**Year**	pF_1	cF_1	ccF_1	**MUC** F_1	B^3F_1	**CEAF**	**BMD**
[16]	Databases	2008	✓	✓	-	-	-	-	-
[31]		2009	✓	-	-	-	-	-	-
[21]		2010	✓	✓	✓	-	-	-	✓
[14]		2010	✓	-	-	-	-	-	-
[25]		2011	✓	-	-	-	-	-	-
[12]	Vision	1999	✓*	-	-	-	-	-	-
[5]		2006	✓*	-	-	-	-	-	-
[10]		2008	✓*	-	-	-	-	-	-
[32]		2011	✓	-	-	-	-	-	-
[17]	Natural Language Processing	2010	-	-	-	-	✓	✓	-
[26]		2010	✓	-	-	-	✓	✓	-
[28]		2010	-	-	-	✓	✓	✓	-
[23]		2010	✓	-	-	✓	✓	-	-
[8]		2010	-	-	-	✓	✓	✓	-
[27]		2011	✓	-	-	-	✓	-	-
[19]	Machine Learning	2000	✓	-	-	-	-	-	-
[7]		2003	✓	-	-	-	-	-	-
[6]		2007	✓	-	-	-	-	-	-
[2]		2009	✓*	-	-	-	-	-	-
[15]		2010	✓	-	-	-	-	-	-
[13]		2011	✓	-	-	-	-	-	-

Note: For the measures, we use pF_1 to denote pairwise F_1, cF_1 to denote cluster F_1, ccF_1 to denote closest cluster F_1, BMD to denote Basic Merge Distance.
✓ denotes that the associated paper uses the associated measure
* uses pairwise precision and recall, but does not specifically use pairwise F_1

surveyed. Pairwise F_1 is also primarily the evaluation method used in Machine Learning and Vision. This is likely due to the advantages we mentioned in Section 3.1.1. Also, we have noticed that the vision domain performs ER in terms of pairs of images or image frames. Because the application is in terms of pairs, pairwise measures seem to be the most natural approach for evaluation.

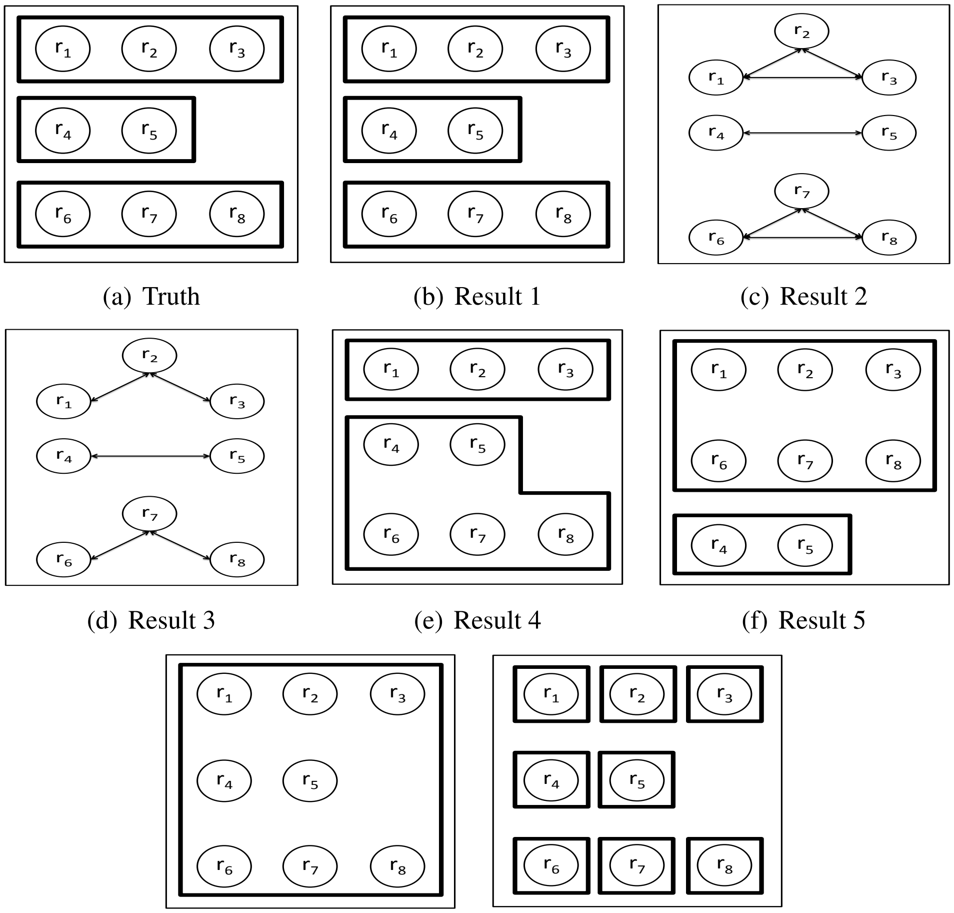
MUC, B^3, and CEAF are mostly used in the Natural Language Processing domain. Precisely, these three measures are defined as measures with links that resemble noun phrase references in the NLP domain. Cluster and closest cluster F_1 are newer measures This survey is only a small sample of the entire ER domain, but it does give us some insight to the use of the measures we have analyzed. In addition, some other measures that we have covered have been recently proposed. Over time, we will know what the response is towards these newer measures.

The database domain also has been using pairwise F_1. However, there has been newer measures in the database domain. These include cluster F_1 and closest cluster F_1. In addition, there has been a push towards edit distance approaches in the database domain. These edit distance approaches include BMD, Variation of Information (VI) (not in Table 2), and GMD (not in Table 2). The proposal of these newer measures by the database domain may show that the pairwise measures may not be a suitable or informative measure for the database domain. However, as we mentioned in Section 3.3, edit distance measures are newer measures that aren't widely used yet, and require more research.

4.2 Comparison of Measures

We have shown that there are several measures used to evaluate ER, and each is a valid way of evaluating ER. In this section, we will show the differences, overlaps, and trends of the measures based on an example shown in Figure 1. We will also show additional disadvantages or advantages of the measures. This example shows the ground truth of the experiment (Figure 1(a)) along with different ER results (Figure 1(b) - 1(h)). The evaluation results using each of the major measures for each of the results in Table 3. Results 1, 2, and 3 essentially represent the same result that are equivalent to the truth. Result 1 is a cluster representation. Results 2 and 3 are the pairwise representations. Results 4 and 5 are representations of different minor false predictions which can lead to different evaluation results. Results 6 and 7 show the two extremes of the results. One extreme is with all the references predicted as the same entity. The other extreme is with all the references predicted as separate individual entities.

Figure 1 describes the main example we will be using to compare the ER measures. Figure 1(a) is the truth and consists of three clusters of references which relate to individual entities. Figure 1(b) is the first result and is identical to the truth. Figure 1(c) is the second result and consists of references that are paired together. This second result is also identical to the truth and, thus, also to the first result. The reason for this is that in the second result, there are pairs between each of the references that relate to the clusters in the first result. As we showed in section 2.1, a cluster is a set of references where each reference have the transitivity relation between them. Because a cluster assumes transitivity between references, in the third result, the pairs or links (r_1, r_2), (r_1, r_3), and (r_2, r_3) can be shown as the cluster $\{r_1, r_2, r_3\}$. Likewise, the second result consists of entities which consists of references that have the transitive relation between them. We can prove that the first result and second result are identical to each other and,



(a) Truth (b) Result 1 (c) Result 2

(d) Result 3 (e) Result 4 (f) Result 5

(g) Result 6 (h) Result 7

Figure 1: Example 1 entities

thus, the truth by showing the results of each evaluation measure evaluated on the results shown in the third and fourth columns of Table 3. As shown, the precision, recall, and F_1 are perfect for each measure. As shown in the results, when the result is completely identical to the truth, each of the three categories of the measures (pairwise, cluster, and edit distance) give identical or perfect results. We will use the first result for comparison with other results since it is a complete match with the truth.

Next, we will show a difference of pairwise measures from the other two types of measures. For now, we can group cluster and edit distance measures because

Table 3: Results for Example 1

Measures			**Results**						
			1	**2**	**3**	**4**	**5**	**6**	**7**
pairwise		**precision**	1.000	1.000	1.000	0.548	0.438	0.250	0.000
		recall	1.000	1.000	0.714	1.000	1.000	0.100	0.000
		$\boldsymbol{F_1}$	1.000	1.000	0.833	0.700	0.609	0.400	0.000
cluster		**precision**	1.000	1.000	1.000	0.500	0.500	0.000	0.000
		recall	1.000	1.000	1.000	0.333	0.333	0.000	0.000
		$\boldsymbol{F_1}$	1.000	1.000	1.000	0.400	0.400	0.000	0.000
closest cluster		**precision**	1.000	1.000	1.000	0.800	0.750	0.375	0.375
		recall	1.000	1.000	1.000	0.667	0.667	0.333	0.389
		$\boldsymbol{F_1}$	1.000	1.000	1.000	0.727	0.706	0.353	0.382
MUC		**precision**	1.000	1.000	1.000	0.833	0.833	0.714	1.000
		recall	1.000	1.000	1.000	1.000	1.000	1.000	0.000
		$\boldsymbol{F_1}$	1.000	1.000	1.000	0.909	0.909	0.833	0.000
$\boldsymbol{B^3}$		**precision**	1.000	1.000	1.000	0.700	0.625	0.343	1.000
		recall	1.000	1.000	1.000	1.000	1.000	1.000	0.375
		$\boldsymbol{F_1}$	1.000	1.000	1.000	0.824	0.769	0.512	0.545
CEAF	ϕ_3	**precision**	1.000	1.000	1.000	0.750	0.625	0.375	0.375
		recall	1.000	1.000	1.000	0.750	0.625	0.375	0.375
		$\boldsymbol{F_1}$	1.000	1.000	1.000	0.750	0.625	0.375	0.375
	ϕ_4	**precision**	1.000	1.000	1.000	0.875	0.833	0.545	0.208
		recall	1.000	1.000	1.000	0.583	0.556	0.182	0.556
		$\boldsymbol{F_1}$	1.000	1.000	1.000	0.700	0.667	0.273	0.303
BMD			1.000	1.000	1.000	0.857	0.857	0.714	0.286

Note: The reported BMD scores above are calculated as $1-$ actual BMD to give a more comparative score with the other measures.

they both use clusters of references. Figure 1(d) shows the third result which is similar but not identical to the third result. In fact, the third result is identical to the truth and the first result. Again, this the truth is the cluster form of the third result. Because a cluster assumes transitivity between references, in the third result, the pairs or links (r_1, r_2) and (r_2, r_3) can also be shown as the cluster $\{r_1, r_2, r_3\}$. However, the third result is not identical to the second result. The third result does not have the pairs (r_1, r_3) and (r_6, r_8) that the second result has. We can expect the pairwise results of the third result to be different compared to the second result. This is shown in the fourth and fifth columns of Table 3. Because the the third result does not consist of all pairs that the second result consists of the pairwise F_1 has decreased for the third result. As shown, the cluster and edit distance results are the same as the first two results, because the cluster forms are the same for the three results. This difference shows that pairwise precision, recall, and F_1 is more sensitive to the exact pairs or coreference relations, while the cluster and edit distance measures generalize a cluster to consist of references with the transitive relations.

Next, we will show the differences between the cluster measures. We will first show the difference between cluster F_1 and the other cluster measures. If we compare the fourth result (Figure 1(e)) with the truth or the first result. We can expect all the measures to produce lower results compared to the perfect match since five references are falsely predicted to be the same entity. The cluster F_1 is actually lower than all the other cluster measures along with the other two categories. The same holds for the fifth (Figure 1(f)), sixth (Figure 1(g)), and seventh (Figure 1(h)) results. Cluster F_1 looks for the exact matching clusters in the truth and result. This property makes cluster F_1 the strictest of the measures. However, this can be a disadvantage as mentioned earlier. We can expect the fifth result to have lower results compared to the fourth result because it false predicts six references to be the same entity compared to the five in the fourth result. However, both results have the same cluster F_1 results because both correctly predict one cluster. As shown in the results, most of the other cluster measures produce lower results compared to cluster F_1 in the fifth result compared to the fourth result. The only exception is MUC F_1 of the cluster measures and also BMD. We will talk about these below.

Next, we will show the difference between MUC and B^3 F_1. The first difference is one of the main motivations of the proposal of B^3, which we described in Section 3.2.4. As we just showed that we expect the fifth result to have lower results than the fourth result because six references are falsely predicted to be the same entity compared to five. MUC F_1, like cluster F_1, produces the same results for the two results.

There are some other interesting evaluation results present. If we look at Result 6, most of the measures have low scores, but BMD is still high with 0.714. This is because Result 6 has merged all references together, and would require a few merges. This can be disadvantage of BMD. When many references have been falsely merged together, BMD would only require a few merges which would still give a relatively high score.

4.3 Practical Considerations

Although there is no one perfect measure, we can give insight to which measures to use or which measure not to use. We are not saying that any measure is better than the other. As we have shown in previous sections, each measure has its advantages and disadvantages. We should point out that CEAF requires more computation than other measures as mentioned in Section 3.2.5, while MUC and B^3, for instance, are relatively easy to compute. Pairwise F_1 is especially easy to compute, which may be one reason why it is widely used. Another point is that there seems to be fewer applications of cluster precision, recall, and F_1, because it a very strict measure that looks for completely matching clusters.

As we showed in Section 2, there are many variants of many of the cluster measures (B^3, CEAF) to handle different types of results. This shows that some of these measures are not perfect and there is constantly a need for newer variants, which has lead to recently proposed measures. In our survey, we have seen that pairwise measures are the most used measures. There is an immense amount of new research in ER evaluation which is expected to continue. We expect several new evaluation measures to be proposed. We believe there will not a standard measure for ER evaluation. Rather, there are more application or domain based measures. With more ER measures being proposed, we expect that ER evaluation will be even more application based, and there would be a specific measure for different applications. Also, we have seen that usually more than one measure is used to evaluate ER results. We believe that this method of evaluation will continue even though there will be more application specific measures. The reason for this is that each measure evaluates differently and evaluation shouldn't rely on just one measure.

5 Conclusion

With the increasing amount of research in ER in several domains and many newly proposed ER evaluation measures, there is a strong need of a guide of several widely used and newly proposed ER measures. We have provided an analysis on many major ER evaluation measures in order to guide future ER work in a variety of domains. There have been other surveys of ER measures, but they have been more domain specific (NLP [8] or Databases [21]). We give an overview of the evaluation techniques in the main ER domains to give a total or wider overview of ER. We have not analyzed every ER evaluation measure ever used, but we have analyzed the most used measures and newly proposed variants of those measures. These newly proposed variants are very promising, but we believe that with the increasing number of variants, there will eventually be a range of application specific measures. Nevertheless, some measures, such as GMD, have been proposed to become a fundamental way of evaluating ER, however, we believe that the use of multiple measures is still required in order to analyze evaluation results from different measures.

References

[1] Al-Kamha, R., and Embley, D. W. Grouping search-engine returned citations for person-name queries. *WIDM* (2004), 96–??103.

[2] Arasu, A., Ré, C., and Suciu, D. Large-scale deduplication with constraints using Dedupalog. In *Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on Data Engineering* (Mar. 2009), IEEE, pp. 952–963.

[3] Bagga, A., and Baldwin, B. Algorithms for Scoring Coreference Chains. *ReCALL*, 919 (1998), 563–566.

[4] Benjelloun, O., Garcia-Molina, H., Menestrina, D., Su, Q., Whang, S. E., and Widom, J. Swoosh: a generic approach to entity resolution. *The VLDB Journal 18*, 1 (2008), 255–276.

[5] Bertini, M., Bimbo, A. D., and Nunziati, W. Video clip matching using mpeg-7 descriptors and edit distance. *Image and video retrieval* (2006), 133–142.

[6] Bhattacharya, I., and Getoor, L. Collective entity resolution in relational data. *ACM Transactions on Knowledge Discovery from Data 1*, 1 (Mar. 2007), 5–es.

[7] Bilenko, M., and Mooney, R. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, New York, USA, 2003), ACM, pp. 39–48.

[8] Cai, J., and Strube, M. Evaluation metrics for end-to-end coreference resolution systems. *Computational Linguistics* (2010), 28–36.

[9] Doddington, G., Mitchell, A., and Przybocki, M. The automatic content extraction (ace) program tasks, data, and evaluation. *Proceedings of LREC* (2004), 837–840.

[10] HAMDOUN, O., AND MOUTARDE, F. Person re-identification in multi-camera system by signature based on interest point descriptors collected on short video sequences. *Smart Cameras, 2008* (2008), 0–5.

[11] HUANG, J., ERTEKIN, S., AND GILES, C. L. Efficient name disambiguation for large-scale databases. *Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery 1* (2006), 536–544.

[12] JUNKER, M., AND HOCH, R. On the evaluation of document analysis components by recall, precision, and accuracy. *International Conference on Document Analysis and Recognition* (1999).

[13] KOLB, L., KÖPCKE, H., THOR, A., AND RAHM, E. Learning-based Entity Resolution with MapReduce. *Database* (2011).

[14] KÖPCKE, H., THOR, A., AND RAHM, E. Evaluation of entity resolution approaches on real-world match problems. *Proceedings of the VLDB Endowment 3*, 1-2 (2010), 484–493.

[15] KÖPCKE, H., THOR, A., AND RAHM, E. Learning-based approaches for matching web data entities. *Internet Computing 14*, 4 (July 2010), 23–31.

[16] LAENDER, A., GONÇALVES, M., AND COTA, R. Keeping a digital library clean: new solutions to old problems. *Proceeding of the eighth ACM symposium on Document engineering* (2008), 257–262.

[17] LIN, B., SHAH, R., FREDERKING, R., AND GERSHMAN, A. CONE: Metrics for Automatic Evaluation of Named Entity Co-Reference Resolution. In *Proceedings of the 2010 Named Entities Workshop* (2010), Association for Computational Linguistics, pp. 136–144.

[18] LUO, X. On coreference resolution performance metrics. *of the conference on Human Language Technology* (2005).

[19] MCCALLUM, A., NIGAM, K., AND UNGAR, L. H. Efficient clustering of high-dimensional data sets with application to reference matching. *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '00* (2000), 169–178.

[20] MEILA, M. Comparing clusterings by the variation of information. *Learning Theory and Kernel Machines 2777*, 2777 (2003), 173–187.

[21] MENESTRINA, D., AND WHANG, S. Evaluating entity resolution results. *Proceedings of the VLDB 3*, 1 (2010), 208–219.

[22] MONGE, A., AND ELKAN, C. The field matching problem: Algorithms and applications. In *Proceedings of the second international Conference on Knowledge Discovery and Data Mining* (1996), no. Slaven 1992, pp. 267–270.

[23] RAGHUNATHAN, K., LEE, H., RANGARAJAN, S., CHAMBERS, N., SURDEANU, M., JURAFSKY, D., AND MANNING, C. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing* (2010), no. October, Association for Computational Linguistics, pp. 492–501.

[24] RAHMAN, A. Supervised models for coreference resolution. *Proceedings of the 2009 Conference on Empirical*, August (2009), 968–977.

[25] RASTOGI, V., DALVI, N., AND GAROFALAKIS, M. Large-scale collective entity matching. *Proceedings of the VLDB Endowment 4*, 4 (2011), 208–218.

[26] RECASENS, M., AND HOVY, E. Coreference Resolution across Corpora: Languages, Coding Schemes, and Preprocessing Information. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (2010), no. July, Association for Computational Linguistics, pp. 1423–1432.

[27] SINGH, S., SUBRAMANYA, A., PEREIRA, F., AND MCCALLUM, A. Large-Scale Cross-Document Coreference Using Distributed Inference and Hierarchical Models. In *Association for Computational Linguistics: Human Language Technologies (ACL HLT)* (2011), Association for Computational Linguistics, pp. 793–803.

[28] STOYANOV, V., CARDIE, C., GILBERT, N., RILOFF, E., BUTTLER, D., AND HYSOM, D. Reconcile: A coreference resolution research platform. Tech. rep., Lawrence Livermore National Laboratory (LLNL), 2010.

[29] STOYANOV, V., GILBERT, N., AND CARDIE, C. Conundrums in noun phrase coreference resolution: Making sense of the state-of-the-art. *In Proceedings of the Joint Conference of the 47th Annual Meeting of the Associ-*

ation for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing (2009), 656–664.

[30] VILAIN, M., BURGER, J., ABERDEEN, J., CONNOLLY, D., AND HIRSCHMAN, L. *A model-theoretic coreference scoring scheme*. Association for Computational Linguistics, 1995, p. 45.

[31] WHANG, S. E., BENJELLOUN, O., AND GARCIA-MOLINA, H. Generic entity resolution with negative rules. *The VLDB Journal 18*, 6 (Feb 2009), 1261–1277.

[32] ZHANG, L., VAISENBERG, R., AND MEHROTRA, S. Video Entity Resolution: Applying ER Techniques for Smart Video Surveillance. *Pervasive* (2011).