Towards High-Precision and Reusable Entity Resolution Algorithms over Sparse Financial Datasets

Douglas Burdick
IBM Almaden Research
650 Harry Road
San Jose, CA 95120
drburdic@us.ibm.com

Lucian Popa
IBM Almaden Research
650 Harry Road
San Jose, CA 95120
lpopa@us.ibm.com

Rajasekar Krishnamurthy
IBM Almaden Research
650 Harry Road
San Jose, CA 95120
rajase@us.ibm.com

ABSTRACT

We describe our approach to the FEIII Data Challenge, which requires matching entities across multiple financial datasets (FFIEC, SEC and LEI). By making use of a high-level language (HIL) that includes constructs for expressing both the matching logic and the policies to avoid or reduce the ambiguities among the matches, we are able to produce highly-accurate results in a sparse context, with only name and location attributes. As part of the high-level specification, we also make use of a Smart-Term Generation (STG) component, which provides us with a sophisticated subroutine for normalizing company names. The high-level specification is reusable, in the sense that the same HIL specification (modulo changing the attribute names) is uniformly applicable not only between FFIEC and SEC, but also between FFIEC and LEI, and between LEI and SEC.

Our approach used only the data provided by the organizers, without analyzing any additional (external) datasets. For the task linking FFIEC records to SEC, we achieved 92.82% precision, 84.32% recall, and 88.38% F1-score. The precision and F1-score were the maximum reported across all participants, and recall was 1.3% less than the maximum 85.63%. For the task linking FFIEC records to LEI, we achieved 99.14% precision, 92.54% recall and 95.72% F1-score, with our F1-score 1.72% less than the maximum reported 97.44%. In this short paper, we provide a description of our method, together with an analysis of our results as well as possible directions for improvement.

1. INTRODUCTION

The objective of the Financial Entity Identification and Information Integration (FEIII) challenge is to create a reference dataset linking financial entity identifiers across multiple heterogeneous datasets. The first step towards this objective motivated four record linkage (or matching) tasks across datasets from FFIEC, SEC and LEI as part of the first FEII Data Challenge. The four tasks involved linking the following datasets: 1) FFIEC to LEI, 2) FFIEC to SEC, 3) FFIEC to LEI & SEC (i.e., find records in FFIEC with matching records in both LEI and SEC), and 4) LEI to SEC.

This problem setting has several characteristics which must be taken into account when designing the actual record linkage solution. First, the datasets are *sparse*: they essentially contain only entity name and location information to identify the entities. The SEC and LEI datasets have attribute sets for two locations (business and mailing location), while the FFIEC dataset has attributes for one location. The location attributes are not always populated; in fact, in some cases, the location information is entirely missing, which adds to the challenge of being able to reliably identify matches across the datasets.

Since the decision for linking records is based on name and location only (unless external data is consulted, which we did not do), the analysis of these fields plays an important role towards good quality matching. In particular, the financial entity name has important structural information that has to be carefully understood. As an illustration, there may be multiple distinct financial entities that have identical locations and *nearly* identical names that should not be linked. Such examples include banks and their parent holding company (e.g., "Isabella Bank", a bank, vs. "Isabella Bank Corp", which is the parent holding company, thus, a distinct financial entity). The presence or absence of a common suffix, such as "Corp" in the preceding example, has significant importance in this setting. (In other settings, "Corp" may be a stop-word which has little or no influence over determining name similarity.) In our approach, the analysis of the structure of the financial entity name is performed via a normalization function that is obtained by instantiating, in a particular way, the Smart-Term Generation component [3] from IBM.

A second characteristic of the problem setting is that we could not assume that the matching has to be one-to-one. In particular, an FFIEC record may seemingly link to multiple SEC records for the same legal entity. As an example, "Zions First National Bank" in FFIEC has multiple corresponding entries in SEC, under similar names: "Zions First National Bank /GFN", "Zions First National Bank /MSD", "Zions First National Bank /TA", all with different SEC ids (CIKs). In this example, the different SEC records correspond to the multiple roles that the bank may play (e.g., "Zions First National Bank /TA" represents a transfer agent for Zions First, while "Zions First National Bank /MSD" represents Zions First as a municipal securities dealer).

In our approach, we make use of a high-level language HIL [1, 2] that includes constructs for expressing both the

DSMM'16, June 26-July 01 2016, San Francisco, CA, USA

DOI: http://dx.doi.org/10.1145/2951894.2951909

matching logic and the policies to avoid or reduce the ambiguities among the matches. As a result, we can write simple entity resolution algorithms that are able to produce highly-accurate results even in a sparse context, where only name and location attributes are available. The high-level specification is reusable, in the sense that the same HIL specification (modulo changing the attribute names) is uniformly applicable not only between FFIEC and SEC, but also between FFIEC and LEI, and between LEI and SEC.

This paper is organized as follows. Section 2 describes the underlying HIL and STG technology used in all four tasks, while Section 3 describes the specific implementation details. Section 4 provides an analysis of results, and we conclude in Section 5.

2. TECHNOLOGY USED

As already mentioned, our solution to the FEIII Data Challenge leverages two technologies developed in IBM Research. First, we make use of HIL to express the actual logic for record linkage. As part of the HIL specification, we make use of a Smart-Term Generation (STG) component, which provides us with a sophisticated subroutine for normalizing company names by analyzing the structure of the name.

HIL [1, 2] is a SQL-like scripting language that combines, within a single high-level framework, multiple types of operations that are needed for *entity integration flows*, that is, flows that create unified clean entities from a variety of raw structured or semi-structured data sources. The operations that HIL provides include: 1) mapping of the data (e.g., from the various raw facts to a target model or any intermediate schema), 2) resolving and linking references to the same real-world entity (i.e., entity resolution or entity linking), 3) fusion, transformation and aggregation of data (including operations for fusion of all the records that were linked as the result of entity resolution).

A salient design feature of HIL is that it shields its users from the lower-level details of the particular runtimes. Thus, HIL allows to decouple the high-level specification of the entity resolution and integration operations from the actual runtime operations. Once expressed in HIL, entity resolution and integration algorithms can then be compiled and optimized for various runtimes for distributed computation, including Map/Reduce and Spark. For the FFIEC data challenge, we used HIL with Map/Reduce runtime.

Regarding the entity resolution fragment of HIL, which we use in this data challenge, an important feature towards obtaining both high recall and high precision is the ability to define and compose multiple rules, where each individual rule expresses various combinations of matching conditions. Additionally, the language includes constructs for expressing 1:1 or 1:N matching constraints, as well as more advanced constructs to disambiguate among multiple matches. In this challenge, as we describe in Section 3, we make use of the 1:1 matching constraint in an essential way that allows us to reliably identify matches even among sparsely populated entities (e.g., financial institutions with missing or incomplete address information).

More complete information about HIL, including the language reference and various use cases can be found at [1].

Smart-Term Generation (STG) [3] is a technology that allows to generate semantic variants for entity names, by accounting for the various ways in which prefixes, suffixes, digits, punctuation marks, etc., can be used in an entity name. STG provides a programmable interface where one can customize and build a suitable semantic term generation for a particular domain (financial institutions in our case).

By using STG, one can search or match for a financial entity name by utilizing all the semantic variants of the name. In our case, the semantic variants for "3rd Federal Savings & Loan Bank" would include: "Third Federal Savings & Loan Bank", "3rd Federal Savings and Loan Bank", and "Third Federal Savings and Loan Bank". Also, the semantic variants for "Bank of America, NA" would include "Bank of America, N.A.", "Bank of America, National Association", "Bank of America NA", "Bank of America N.A.", "Bank of America National Association", which account for the various usages of the "NA" suffix (common for banks), and also for the presence or absence of the comma and dot separators.

For the data challenge, we further customized STG so that it rewrites each entity name into exactly one of its semantic variants (intuitively, a standardized replacement of the input name). Thus, we obtained a *normalize* function for financial institution names, which we then used in the record linkage logic (see next section). As examples of normalization, *normalize*("3rd Federal Savings & Loan Bank") = "Third Federal Savings and Loan Bank", *normalize*("Bank of America, NA") = "Bank of America National Association", and *normalize*("JPMorgan Chase & Co") = "JPMorgan Chase and Company". Two financial institutions will be considered to have the same name if their normalized names are the same (modulo upper case conversion).

3. CONCRETE TASK IMPLEMENTATION

We describe our solution by starting with Task 2 (matching FFIEC against SEC data), after which we describe how the same approach carries over to the other tasks. The record linkage logic for matching FFIEC records with SEC records is given as the 16 line HIL script in Figure 1. This HIL script consists of two `create link` statements, each encoding one algorithm for creating links between FFIEC ids and SEC ids. The final result for Task 2 is obtained as the duplicate-free union across the two algorithms.

Similar to SQL, the `select` clause in HIL specifies the result tuples (pairs of ids in this case), while the `from` clause specifies the input datasets. The first `create link` statement obtains its result as the union of two matching rules. Both rules check for name matching (as an equality of normalized financial entity names), and for (city, state) matching. The first rule is based on the business address from SEC, while the second rule is the same except that it uses the mailing address from SEC. We do not use street address information, since we found that using just (city, state) matching already gives high precision. (Also, adding a matching condition on street address would reduce the recall.)

While the first algorithm (based on rule1 and rule2) has good precision, we found that there were enough true positive matches that were not discovered by it. In particular, any record with missing city or state information (either from FFIEC or SEC) would not be matched. Furthermore, there were cases of matching entities where the (city, state) information is not the same across the two datasets. As an example, "PNC Bank" is listed in Wilmington, DE in the FFIEC dataset, and in Pittsburgh, PA in the SEC dataset.

To address the above cases, we wrote a second algorithm, which uses only name matching (where the matching condition is the same as in the first algorithm). Since name

```
create link FFIEC_SEC_Links_1 as
select [ ffiec_id: F.IDRSSD,  sec_id: S.CIK ]
from  FFIEC_REF F,  SEC_REF S
match using
   rule1:  normalize(toUpper(F.Financial_Institution_Name_Cleaned)) = normalize(toUpper(S.CONFORMED_NAME))
         and toUpper(F.Financial_Institution_City) = toUpper(S.B_CITY)
         and toUpper(F.Financial_Institution_State) = toUpper(S.B_STATE),

   rule2:  normalize(toUpper(F.Financial_Institution_Name_Cleaned)) = normalize(toUpper(S.CONFORMED_NAME))
         and toUpper(F.Financial_Institution_City) = toUpper(S.M_CITY)
         and toUpper(F.Financial_Institution_State) = toUpper(S.M_STATE);

// A more relaxed algorithm for matching FFIEC with SEC entities just by name
// Used in conjunction with 1:1 cardinality constraint to strengthen its precision
create link FFIEC_SEC_Links_ByName as
select [ ffiec_id: F.IDRSSD,  sec_id: S.CIK ]
from  FFIEC_REF F,  SEC_REF S
match using
   rule1:  normalize(toUpper(F.Financial_Institution_Name_Cleaned)) = normalize(toUpper(S.CONFORMED_NAME))
cardinality ffiec_id 1:1 sec_id;
```

Figure 1: HIL program for linking FFIEC to SEC.

matching alone can be imprecise in the absence of location information, we also added a 1:1 cardinality constraint on the result of this second algorithm. This constraint has the effect that all the "ambiguous" matches (that is, an entity from FFIEC matching two or more entities from SEC, or vice-versa) are dropped from the outcome of this second algorithm. Thus, the result consists of all the "unique" matches that are based on name. As an example, we obtain the "PNC Bank" match mentioned above, since there is only one entry in FFIEC and one entry in SEC.

The combination of the two algorithms has sufficiently high precision (we achieved 92.82%, the highest among the participants), while resulting in higher recall than it would have been possible with each of the individual rules (we achieved 84.32% recall, which was close to the maximum achieved recall of 85.63%). The two algorithms together generate a total number of 261 distinct matches between FFIEC and SEC. (out of 6,652 FFIEC records and 129,311 SEC records).[1] We also note that the first algorithm in Figure 1 does not have a 1:1 cardinality constraint and, as a result, its matches may be one-to-many (or many-to-many, although we did not find such examples). In particular, some FFIEC records may link to more than one SEC record (as discussed in Section 1, the same legal entity may play different roles and be listed under different CIKs in SEC).

We used a similar HIL program for Task 1 (FFIEC to LEI) and also a similar HIL program for Task 4 (LEI to SEC). The logic of the algorithms was exactly the same, but they were instantiated with different field names. We obtained 480 distinct matches between FFIEC and LEI (out of 6,652 FFIEC records and 53,958 LEI records), and 4,325 distinct matches between LEI and SEC (out of 53,958 LEI records and 129,311 SEC records).

As for Task 3, we obtained the FFIEC entities that appear in both LEI and SEC simply as the intersection of the FFIEC ids matched in Task 1 and the FFIEC ids matched in Task 2. We obtained 83 such FFIEC ids.

Finding True Negatives. For Tasks 1 and 2, we were also asked (optionally) to submit the list of FFIEC ids that we were highly confident they do not match with LEI, respectively, SEC entities. We discuss next how we computed a set of FFIEC ids that are unlikely to match with the SEC dataset (i.e., true negatives for Task 2).

Intuitively, the set of FFIEC ids that do not match with SEC entities is the complement of the set of FFIEC ids that do match with some SEC entity. However, for this complement operation, we could not directly use the set of matches that we obtained via the HIL program in Figure 1. That HIL program is targeted towards identifying very strong matches (and our results show, indeed, a precision higher than 90%). This means that there may be other FFIEC ids that may still have a chance to match with some SEC entity. Thus, we first obtained a set of "weak" matches between FFIEC and SEC, by significantly relaxing the conditions in the above HIL program. Concretely, we used a variation of the HIL algorithm 2 in Figure 1 where we dropped the 1:1 cardinality constraint and just used the name matching condition. This resulted in 480 matches, which included as a subset the 261 strong matches that were produced by the full HIL program in Figure 1. We then took the complement between the entire set of FFIEC ids (6,652 of them) and the set of FFIEC ids that participate in any of the 480 matches. This resulted in a set of 6,250 FFIEC ids, which we submitted as our set of true negatives for Task 2.

We performed a similar computation but with LEI in place of SEC and obtained a set of 5,714 FFIEC ids that are unlikely to match with the LEI dataset. This was submitted as our set of true negatives for Task 1.

4. ANALYSIS OF RESULTS

Our analysis of results focuses on Task 1 (FFIEC to LEI) and Task 2 (FFIEC to SEC), since ground-truth data is available for these tasks. For Task 1 (FFIEC to LEI), there were 496 true positive links in the adjudicated ground truth

[1]We note that the overlap between FFIEC and SEC does not appear to be too large.

data, while for Task 2 (FFIEC to SEC) there were 230 true positive links. A summary of results for our submitted approach are given in the table of Figure 2 in rows labeled "Submitted". For comparison, we included maximum achieved among all submitted results in rows labeled " Maximum Reported". As a baseline, our submitted approach for Task 1 had 459 true positive (TP) links and 4 true negative links (TN). The remaining 17 links (recall from Section 3 that the two algorithms generated 480 links for Task 1) were uncertain; they could not be confirmed by the experts as either positive or negative, and they were not taken into account for precision/recall calculation. For Task 2, our submitted approach had 194 true positive (TP) links and 15 true negative (TN) links. The remaining 52 links (out of the total of 261 generated) were uncertain and not considered for precision/recall evaluation.

After performing an analysis of results after the ground-truth data release, we discovered two potential improvements to our approach that we subsequently implemented. Although not submitted for the competition, these results are interesting since they hopefully enhance understanding of these linkage problem settings.

The first improvement involved changing the HIL matching rules to use the "LegalNameCleaned" attribute from the LEI dataset instead of "LegalName" for the Task 1 solution. This change results in 9 additional TP links being identified, and no change in TN count. The updated results are shown in row "Task 1 - Using clean name" of the table in Figure 2. This change is clearly beneficial, since precision, recall, and F1-score all improved.

The second attempted improvement was to strengthen the relaxed matching algorithm (HIL Algorithm 2) which matches two records by the normalized name without considering similarity of location, as long as the matching satisfies the 1:1 constraint. We observed in our submitted results for both Task 1 and Task 2 that most TN links (100% for Task 1, 60% for Task 2) were introduced as a result of this relaxed linking algorithm. We then made the observation that while the same financial entity may have locations with different cities across its records in different datasets, the state is almost always the same. The intuition is that major metropolitan areas tend to have many smaller incorporated areas, and there may be some flexibility with the city name used. To incorporate this observation, we modified HIL algorithm 2 so that matching is done via both the normalized name and the state attribute for either business or mailing address.

The results from this change are in the table in Figure 2, in rows labeled "Modified HIL Alg 2". The change in quality of results appears to be mixed, based on the very slight decrease in F1-score of 0.2%. The increase in precision from the strengthened algorithm came at the expense of decreased recall. There are a number of financial entities with different states that we lose as a result of this variation (e.g., the earlier "PNC Bank" which is listed in DE in FFIEC and PA in SEC). Reliably identifying such links requires further investigation. Additional information or datasets beyond the provided datasets may need to be exploited.

There were three other sources for errors (i.e., true negatives produced, and true positives omitted) in our method. First, from the provided datasets alone there is no obvious way to reliably differentiate entities with exactly the same name and same city and state location. As an example,

Description	Precision	Recall	F1
Task 1 - Submitted	99.14%	92.54%	95.72%
Task 1 - Using clean name	99.15%	94.35%	96.69%
Task 1 - Modified HIL Alg 2	99.36%	93.75%	96.47%
Task 1 - Maximum Reported	99.23%	96.37%	97.44%
Task 2 - Submitted	92.82%	84.35%	88.38%
Task 2 - Modified HIL Alg 2	94.53%	82.61%	88.17%
Task 2 - Maximum Reported	92.82%	85.65%	88.38%

Figure 2: Summary of results.

Eastern Bank in Boston, MA with IDRSSD 128904 in the FFIEC dataset and CIK 1107071 in the SEC dataset refer to two different institutions. This is a situation where additional data sources are required. In our result, we observed 3 such TN links for Task 1, and 5 TN links for Task 2.

Second, there were TP links omitted because either one or both of the name and city/state location were materially different. For example, Customers Bank (IDRSSD 2354985) and New Century Bank (CIK 1478487) in Phoenixville, PA represent the same entity. As before, additional information (e.g, institution history) is required to reliably generate such matches. In our results, we observed 4 such TP links omitted for Task 1, and 3 such TP links omitted for Task 2.

Finally, better processing for name attributes can further improve matching. There are particular normalization procedures for entity name suffixes that appear to be quite specific to this domain. One case, which we discussed and implemented in our method, is being able to understand that "/TA", "/MSD" represent different roles of the same entity, and therefore should not materially influence the name matching logic. Another case is one in which state location information is encoded in the name itself. As an example, in "KEYBANK NATIONAL ASSOCIATION/OH", the state information appears as a suffix. Being able to use this information is useful when the actual location attributes have missing data.

5. CONCLUSION

We described our methodology for the FEIII Data Challenge record-linkage tasks. We used a high-level scripting language, HIL, to express the record linkage logic, and also utilized a name normalization function based on the STG package. As a result, our code base is very compact and easily readable, with no more than 20 lines of high-level code for each task. This facilitated the rapid modification of the linkage logic, in order to transfer the logic from one task to another, and also to experiment with additional improvements.

6. REFERENCES

[1] HIL Reference. http://www.ibm.com/support/knowledgecenter/SSWSR9_11.4.0/com.ibm.swg.im.mdmhs.pmebi.doc/topics/using_hil.html.

[2] Mauricio Hernández, Georgia Koutrika, Rajasekar Krishnamurthy, Lucian Popa, and Ryan Wisnesky. HIL: A High-level Scripting Language for Entity Integration. In *EDBT*, pages 549–560, 2013.

[3] Yunyao Li, Ziyang Liu, and Huaiyu Zhu. Enterprise Search in the Big Data Era: Recent Developments and Open Challenges. *PVLDB*, 7(13):1717–1718, 2014.