Sub-Merge: Diving Down to the Attribute-Value Level in Statistical Schema Matching

Zhe Lim and **Benjamin I. P. Rubinstein***
Department of Computing and Information Systems
The University of Melbourne, Australia
zhe@infinitelooplabs.com brubinstein@unimelb.edu.au

Abstract

Matching and merging data from conflicting sources is the bread and butter of data integration, which drives search verticals, e-commerce comparison sites and cyber intelligence. Schema matching lifts data integration—traditionally focused on well-structured data—to highly heterogeneous sources. While schema matching has enjoyed significant success in matching data attributes, inconsistencies can exist at a deeper level, making full integration difficult or impossible. We propose a more fine-grained approach that focuses on correspondences between the values of attributes across data sources. Since the semantics of attribute values derive from their use and co-occurrence, we argue for the suitability of canonical correlation analysis (CCA) and its variants. We demonstrate the superior statistical and computational performance of multiple sparse CCA compared to a suite of baseline algorithms, on two datasets which we are releasing to stimulate further research. Our crowd-annotated data covers both cases that are relatively easy for humans to supply ground-truth, and that are inherently difficult for human computation.

Introduction

Data integration has enjoyed a remarkable level of attention from the academic communities in databases (entity resolution, noisy joins), IR (information extraction), machine learning & statistics (record linkage), NLP (co-reference resolution), that is matched only by its real-world impact. Starting mid-20th century in statistics, the earliest literature in the area sought to integrate official statistics in health and census data (Dunn 1946). While early database applications helped combine customer records, data integration has seen renewed focus thanks to routine crawls of the deep web, sharing of data by businesses, and open-data initiatives. These modern applications have placed new requirements on data integration, such as scalability to large sources (both in number of records and attributes), to many sources, to sources with heterogeneous schemas, and data that is semi-structured or unstructured altogether. Schema matching (Rahm and Bernstein 2001; Bernstein, Madhavan, and Rahm 2011) was born out of the need to align source schemas—the set of attributes of a dataset. Where schema matching proposes to normalize source structure at the attribute level, say identifying course duration and time in the Stanford and Berkeley catalogs, sub-attribute level differences can remain, *e.g.*, not matching Autumn quarter and Fall semester. It is data cleaning at this important *attribute-value level* that we address in this paper.

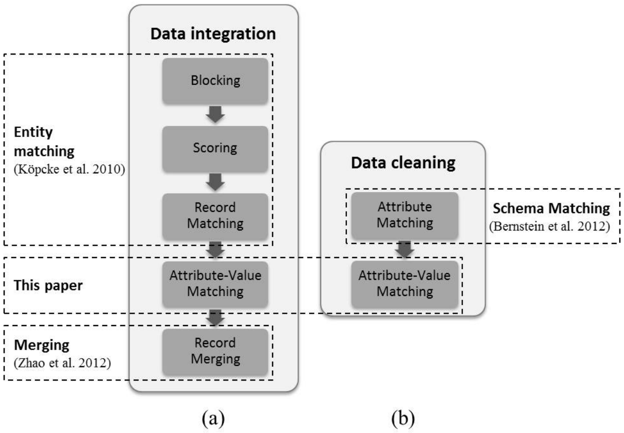

Figure 1: Depictions of how attribute-value matching fits within (a) data integration and (b) data cleaning.

Attribute-value matching fits naturally as a stage within two common scenarios shown in Figure 1. In data integration the stage can make use of partial-matchings of records (Köpcke, Thor, and Rahm 2010; Köpcke and Rahm 2010; Winkler 2006), and provides the normalization required to complete record merging (Zhao et al. 2012). It can also run subsequent to instance-based schema matching (Rahm and Bernstein 2001; Bernstein, Madhavan, and Rahm 2011), as a fellow instance-based method. In both settings, attribute-value matching can improve integration quality. It also endows immediate benefit: we may wish to surface for example genres when browsing movies in an integrated catalog (*e.g.*, streaming on the Xbox); or enable faceted search over many sellers (*e.g.*, Amazon.com), or a product comparison website (*e.g.*, Google Shopping).

We explore a suite of IR- and statistics-based approaches on real-world data, concluding multiple sparse canonical correlation analysis (multiple sparse CCA) to be superior both in terms of matching quality and (surprisingly) run-

*Corresponding author.

time. Machine learning has previously enjoyed applications in many of the stages of data integration, most significantly in combining attribute-level scores when comparing records in record matching (Köpcke and Rahm 2010), and in combining individual schema matchers (Rahm and Bernstein 2001; Bernstein, Madhavan, and Rahm 2011). Matching approaches are typically combinatorial in nature and can make heavy use of linguistic similarities. By contrast multiple sparse CCA directly optimizes what we view as key to good attribute-value mappings based on instance frequencies.

Contributions This paper makes four main contributions

- We demonstrate that multiple sparse CCA effectively solves a variation of schema matching or ontology matching for merging unnormalized attributes: extensive experiments show the superiority of CCA over standard IR techniques; the improvement is most striking under multiple sources, where CCA has better quality and runtime;
- We explore the difficulty of crowd-based matching: while problems exist that are crowd-sourceable, human computation is inappropriate for cases where attribute semantics come from subtle term usage not linguistic similarity;
- To the best of our knowledge, ours is the first application of CCA in database research; and due to the application's importance and the fact that precursor stages in the integration pipeline readily yield parallel datasets, we propose that this problem become a primary application for ongoing research into CCA-type algorithms; and
- To foster further research on this problem, we are releasing with this paper two new manually-labeled datasets[1], constructed by multiple web crawls and crowd-sourced annotation.

Related Work

Our problem resembles schema matching (Rahm and Bernstein 2001; Bernstein, Madhavan, and Rahm 2011) and closely-related ontology matching (Shvaiko and Euzenat 2013), which focus on aligning columns or attributes of data sources but not typically the values within. Like our attribute-value matching, schema matching can be instance-based so that attributes with substantially overlapping values are matched. We discuss as future work, how our approach could be used for schema matching by aligning columns (and simultaneously their values).

The data integration pipeline often ends with record matching (Köpcke and Rahm 2010), also known as record linkage in statistics (Winkler 2006) which can be across multiple sources (Sadinle, Hall, and Fienberg 2011). Partial record-matches seed our approaches with parallel datasets from which to mine attribute-value correspondences. Record merging or truth discovery follows record matching. In our recent VLDB work, Zhao et al. (2012) proposed a Bayesian-statistics merging process that runs after record matching and assumes attribute-values have been somehow normalized. This paper addresses their assumption, filling in an important piece of the data integration puzzle.

Our main approach is based on canonical correlation analysis (CCA) which was first proposed by Hotelling (1936) for two sources. Later in their dissertation, Kettenring (1971) proposed a generalization to multiple sources. While CCA is defined for linear maps from parallel feature spaces to latent semantic space, non-linear transformations are possible via kernel CCA (Lai and Fyfe 2000). See (Hardoon, Szedmak, and Shawe-Taylor 2004) for a good overview. Sparse CCA was first proposed (Hardoon and Shawe-Taylor 2011) then refined (Witten, Tibshirani, and Hastie 2009) for efficient implementation. We make use of many of these improvements to CCA here.

Supervised classifiers have enjoyed huge successes in data integration (particularly scoring & merging *cf.* Figure 1.a) and schema matching (*cf.* Figure 1.b) for combining hybrid matchers. To the best of our knowledge ours is the first application CCA in databases. And while CCA applies to machine translation (Vinokourov, Cristianini, and Shawe-Taylor 2002), multi-modal content-based retrieval (Hardoon, Szedmak, and Shawe-Taylor 2004) and music IR (Torres et al. 2007), this paper shows a particularly compelling application of CCA with natural sources of parallel datasets.

The Attribute-Value Matching Problem

Attribute-value matching exists naturally within data processing pipelines (*cf.* Figure 1; details below). Its role is to normalize columns beyond the traditional focus of schema matching which is to match columns. Going deeper, we wish to match the values expressed within columns.

Definition 1. *Consider sources* $1, \ldots, k$ *which could be databases, tables, crawled datasets, etc., each with a corresponding attribute or column. If the domains of these columns are denoted* $D_1, \ldots, D_k$ *respectively then the goal of* attribute-value matching *is to discover a relation* $R \subset \prod_{i=1}^{k} 2^{D_i}$ *that represents a correspondence between sets of values. Properties of ideal matchings are given below.*

Example 1. *In integrating IMDB and Yahoo! Movies we discover that both sources have* genre *columns that clearly correspond: their domains share many elements. Despite this their domains are not identical in size or in elements. The output of attribute-value matching would identify correspondences including*

IMDB	*Yahoo! Movies*
{*Action, Adventure*}	{*ActionAdventure*}
{*Documentary*}	{*Special Interest*}
...	...

In most data integration (*cf.* Figure 1.a) applications some attributes are not used for matching records, due to inconsistencies (Köpcke and Rahm 2010). For example, after matching movie records on movie title and release year, Zhao et al. (2012) merge directors across matched movie records by taking a consensus of weighted votes. This assumes inconsistent directors for individual matching movies, but fundamentally corresponding director across domains. This is often not the case with genres which can vary in how the genre concept is expressed in each source's genre domain (see

[1] Datasets at http://people.eng.unimelb.edu.au/brubinstein/data

above example, and results below). Attribute-value matching after the record matching stage would lead to successful merging of records on an attribute like genre.

Attribute-value matching can be of benefit outside a full data integration pipeline. When normalizing source structure (*cf.* Figure 1.b), schema matching is first run to align attributes/columns. Since this stage works well with instance-based methods (Rahm and Bernstein 2001) which use potentially matching records to inform similarity of columns, it is natural to consider also an instance-based approach to matching the domains within columns.

These two use cases motivate matching attribute values via an instance-based approach.

Main Assumption. *A partial matching of records between sources is available, based on columns/attributes that have substantially overlapping domains.*

An example is title and release year in movies, which have closely-matching domains. After matching on these, we may wish to merge director, genre and others. Both pipelines of Figure 1 indicate readily-available sources of matched records. An ideal attribute-value matching has:

- **Co-occurence.** A good matching R is one supported by significant co-occurrence in matched records.
- **Multi-valued.** While in some applications a 1-to-1 constraint may apply, in general we are interested in matching multiple values from one source to another at-a-time.
- **Sparsity.** Correspondences in the matching R should be sparse: *small* sets of values should be mapped, since we expect key properties of an entity to be expressed by few terms, no matter the form of expression. A soft limit exists on the multiplicity of the previous property.
- **Transitivity.** In the event that R matches values A_1, B_1 in sources A, B then A_1, C_1 in sources A, C, it follows that B_1, C_1 also match, and it is preferable that the triplet is matched altogether. This is a consistency constraint.

Approach

Definition 1 and our ideal properties suggest an instance-based approach. After building up to a variant of CCA for this problem, we discuss adaptations to the approach for applications in practice.

Canonical Correlation Analysis (CCA)

The key property of co-occurrence states that attribute-values should be matched if they are used by many matched records. Restricting to the two-source case, a natural statistical quantification is correlation. Given the sources have unique views on the attribute, we must transform these views into one space in which correlations can be made.

We take linear transformations of the data before computing correlations as follows. Let $\mathbf{X}_1, \mathbf{X}_2$ be $n \times |D_1|, n \times |D_2|$ dimensional matrices representing n matched record pairs in rows, and the attribute values embedded in columns—known as a *paired dataset*. Then we seek projection directions $\mathbf{w}_1 \in \mathbb{R}^{|D_1|}, \mathbf{w}_2 \in \mathbb{R}^{|D_2|}$ such that projected on $\mathbf{w}_1, \mathbf{w}_2$ respectively, $\mathbf{X}_1, \mathbf{X}_2$ have high correlation:

$$\max_{\mathbf{w}_1,\mathbf{w}_2} \quad \frac{\mathrm{cov}\left(\mathbf{X}_1\mathbf{w}_1, \mathbf{X}_2\mathbf{w}_2\right)}{\sqrt{\mathrm{var}\left(\mathbf{X}_1\mathbf{w}_1\right)\mathrm{var}\left(\mathbf{X}_2\mathbf{w}_2\right)}} \ . \qquad (1)$$

Example 2. *If two genre attributes are to be mapped, these would be embedded by bag-of-words. Each movie embeds into a row of zeros, with ones only where a genre describes the movie. A projection direction $\mathbf{w}_i$ stores a mixture of syntactic genres appearing in source i that together represent a semantically-meaningful genre. For example $(0.3, 0.2, 0)$ might mean romantic (0.3) and comedy (0.2) but not horror (0.0). The projection $\mathbf{X}_i\mathbf{w}_i$ is a vector in $\mathbb{R}^n$ reflecting to what extent each movie is described by the semantic genre/mixture of source-specific genre tags. If $\mathbf{w}_1, \mathbf{w}_2$ reflect the same concepts, then records should match these concepts in both sources simultaneously, and not match simultaneously, leading to high correlation.*

We can pose the simultaneous optimization for all $p = \min\{rank(\mathbf{X}_1), rank(\mathbf{X}_2)\}$ projection vectors, by rewriting CCA in terms of the distance between projected data matrices as measured by the Frobenius norm. If the data matrices are first column centered, with $\mathbf{C}_{ij}$ denoting the covariance matrices between $\mathbf{X}_i, \mathbf{X}_j$, we arrive at (Kettenring 1971; Gifi 1990)

$$\min_{\mathbf{W}_1,\mathbf{W}_2} \quad \|\mathbf{X}_1\mathbf{W}_1 - \mathbf{X}_2\mathbf{W}_2\|_F \qquad (2)$$

$$\text{s.t.} \quad \mathbf{W}'_m\mathbf{C}_{mm}\mathbf{W}_m = \mathbf{I} \qquad (3)$$

$$\mathbf{w}'^{i}_{m}\mathbf{C}_{ml}\mathbf{w}^{j}_{l} = 0 \ ,$$

where $m \neq l = 1, \ldots, 2$ and $i \neq j = 1, \ldots, p$, the objective penalizes distance between the data projected into latent semantic space, with a constraint for orthogonality of solutions, and a constraint fixing denominators after noting that the quotient in (1) is invariant to scaling of the projection vectors. This convex program can be solved efficiently as a generalized eigenvalue problem, can be regularized, dualized and kernelized (Shawe-Taylor and Cristianini 2004).

Multiple CCA

To generalize CCA to multiple sources, it is not enough to run CCA on each pair of sources individually, as the resulting matchings are likely to violate the transitivity property. It is a simple matter to generalize the distance form of CCA (3) to $k > 2$ sources—see (Hardoon, Szedmak, and Shawe-Taylor 2004) for details:

$$\min_{\mathbf{W}_1,\ldots,\mathbf{W}_k} \quad \sum_{l,m=1}^{k} \|\mathbf{X}_l\mathbf{W}_l - \mathbf{X}_m\mathbf{W}_m\|_F$$

$$\text{s.t.} \quad \mathbf{W}'_m\mathbf{C}_{mm}\mathbf{W}_m = \mathbf{I}$$

$$\mathbf{w}'^{i}_{m}\mathbf{C}_{ml}\mathbf{w}^{j}_{l} = 0$$

where $m \neq l = 1, \ldots, k$ and $i \neq j = 1, \ldots, p$. This is a form of transfer or multi-task learning: matching one pair constrains the matching of others, and will often lead to improved accuracy over sequential bisource approaches.

Algorithm 1 OneVsAll CCA post-processing

```
Require: unmatched value $v$ from source $i^\star$
Phase 1:
    for record $r = 1$ to $n$ do
        if $X_{i^\star}^{rv} \neq 1$ then set $X_{i^\star}^{ru} = 1$ for all $u \neq v$;
    Run CCA;
    if high eigenvalue matches identified then return;
Phase 2:
    for record $r = 1$ to $n$ do
        if $X_{i^\star}^{rv} \neq 1$ then set $X_{i^\star}^{ru} = 1$ for all $u \neq v$;
        if $X_{i^\star}^{rv} = 1$ then set $X_{i^\star}^{ru} = 0$ for all $u \neq v$;
    Run CCA;
    if high eigenvalue matches identified then return;
return no match found;
```

Sparse CCA

Multiple CCA does not yield sparse solutions. Indeed if the attribute is not multi-valued but rather single-valued—each movie is only ever tagged with one genre—the CCA solution can be degenerate. Direction $(\lambda, \ldots, \lambda)$ produces projection λ for every record no matter the active genre, and maximizes correlation. Standard IR baselines explored below are only natural fits for extreme sparsity: relations on singletons.

Instead our sparsity property leads us to employ sparsity-yielding techniques related to the lasso. We follow the approach of Witten, Tibshirani, and Hastie (2009) which is to introduce L_1 penalties on the projection directions and to treat the self-covariance matrices as identity matrices (Dudoit, Fridlyand, and Speed 2002). Solution is again via an efficient iterative method; please see references for details.

Practical considerations

Regularization Coefficients We perform binary search to set L_1 penalties for a desired level of sparsity, and cross-validation to model select regularization terms (Witten, Tibshirani, and Hastie 2009).

Stopping Short An important task is to determine the number of principal components. A natural approach is via the scree plot: eigenvalues by rank. The retained components can be set by thresholding the eigenvalues—which correspond to correlations under discovered components—or by identifying a 'knee' in the curve.

OneVsAll Algorithm CCA is limited to a maximum $p = \min_i |D_i|$ projection directions. However the number of true attribute values could be larger if some are unique to a source, or if only some sources represent a value. To find values common to some sources, not identified by a first run of CCA, we propose Algorithm 1. Phase 1 introduces co-occurrence, forcing down correlations with values irrelevant to v; while phase 2's zeroing makes sparsity more achievable.The approach also works for multi-source CCA.

Experiments

Datasets

We constructed datasets in the movie and restaurant domains (*cf.* Table 1). For both, we matched records across sources into a single parallel dataset available online.[1]

Movie Dataset (n = 7,852)		**Restaurant Dataset** (n = 3,120)	
Source	**#Genres**	**Source**	**#Cuisines**
Yahoo! Movies	15	Factual	120
IMDB	26	Google	182
Rotten Tomatoes	22	Yelp	151
The Movie DB	34	Foursquare	136

Table 1: The sources for our movies and restaurants datasets.

Movie Dataset We obtained movie data from four online sources—Yahoo! Movies, IMDB, Rotten Tomatoes & The Movie Database—through a combination of official data dumps and API access. The intersection of these sources includes 7,852 movies (*cf.* entity resolution below). The attribute-values we wish to normalize are for genre.

As expected for databases that have been curated separately over many years, a different number of movie genres is used by each source. Yahoo! Movies uses only 15 genres which is the least amongst the four, while The Movie DB has the most genres at 34. The difference stems from how genres are grouped and their granularity. For example, IMDB has separate Action and Adventure genres but Yahoo! Movies groups these two genres under Action / Adventure.

Restaurant Dataset Our restaurant dataset collects 3,120 restaurants in London represented in each of four online sources—Factual, Google Places, Yelp and Foursquare. In each source, each restaurant is associated with one or more types of cuisine.

The dataset was generated in two steps. First, we queried Factual's API to find a list of London restaurants. Next, we searched for each restaurant in the list by utilizing the respective search APIs of the remaining three data sources. To focus the search, we queried the APIs using not just the restaurant's name but also ancillary information such as the restaurant's longitude, latitude and postal code. We additionally crawled the Google+ page of each restaurant to supplement the Google Places API results which omit the cuisine type.

We aim to find relations between the different sets of cuisine types used by each data source. While the list of restaurants is less than the list of movies, there are substantially more types of cuisines in each set than the genres of movies. This results in less support for each cuisine. The expanded set of attribute-values also motivates the use of an algorithmic approach to finding relations as it becomes more challenging for humans to accomplish.

Human Annotations To augment the two datasets, we used Amazon Mechanical Turk (AMT) to crowd-source matches.We use these annotations (a) as a basis of comparison for CCA generated matches (particularly in the case of movies) and (b) to evaluate the accuracy of human matches when there are a large number of attribute-values (*i.e.*, in restaurants). The crowd-sourcing campaign was run in a multiple choice question format, whereby for each value $u \in D_i$ in each source i, a Mechanical Turk worker is asked to select another value $v \in D_j$ from another source j that best matches u. We collect 10 answers for each u and used the majority vote as the consensus answer.

Entity Resolution

Here we describe the data cleaning and record matching that were performed to generate the parallel movie and restaurant datasets. When matching records, we optimize for correctness over completeness to achieve our goal of producing high-quality parallel datasets that can be used for subsequent CCA research. This means that we err on the conservative side and only match two records if they are highly likely to represent the same entity.

Movie Dataset We cleaned the raw movie data by removing (a) movie records with missing genre or release year (b) non-movie records such as TV episodes and (c) outlier genres with fewer than 5 movies.

We then matched movie records by title, disambiguating movies of the same title by release year. We treat release years that differ by at most one as the same to account for varying release schedules worldwide.

Restaurant Dataset Restaurant matching was more involved. We query each source search API with `<name, longitude, latitude, postal code>` to obtain a list of potential matches. Direct name matching is often too restrictive, since many restaurants are listed with slightly different names (*cf.* Table 2). To overcome such ambiguity, we took advantage of a convenient property of the UK postal code system which is granular down to the level of a few buildings. From the search results, we selected the result with a matching postal code and phone number.

Restaurant Name	**Postal Code**
Maze by Gordon Ramsay	W1K 6JP
Maze Grill	W1K 6JP
The White Swan Pub & Dining Room	EC4A 1ES
The White Swan	EC4A 1ES
Il Convivio	SW1W 9QN
Convivio	SW1W 9QN

Table 2: Examples of matching restaurant records.

Baselines For Two Sources

Text Similarity Baselines We evaluated a wide selection of text similarity methods (exact match, match any word, n-gram, edit distance) that match two attribute values based only on their linguistic similarity. This reflects how humans judge matching tasks (Lee, Pincombe, and Welsh 2005).

Probabilistic Baselines We also looked at baselines that ignore the actual attribute value and instead focus on their underlying occurrences. These are:

- **Frequency rank** Assign a rank to attribute values based on occurrence frequency in sources. Values with the same rank in two sources are regarded as the same.

Algorithm	**Accuracy (%) by Judgment Consensus**		
	All	**0.9**	**0.8**
Exact Match	61	75	46
Match Any Word	85	95	**88**
Unigram	76	87	71
Bigram	80	91	78
Edit Distance	70	81	61
Frequency Rank	46	55	41
Frequency Distribution	32	38	29
Most Frequent Class	75	86	75
Cosine Similarity	84	93	85
CCA	**87**	**97**	85

Table 3: Attribute-value matching accuracy over **All** annotated movie data; **0.9** consensus; **0.8** consensus.

- **Frequency distribution** Count occurrences as above but instead of assigning each attribute value to a rank, calculate the % of each value occurring in its source. The attribute value with the closest % in the other source is regarded as the same. This allows gaps/insertions.
- **Most frequent class** For each attribute value, pick an attribute value from another source that it co-occurs with the most.
- **Cosine similarity**: The standard $\mathbf{u}'\mathbf{v}/\|\mathbf{u}\|\|\mathbf{v}\|$ similarity applies to comparing two attribute values: first embedding each as a vector of Boolean values indicating support by each record (a column in our data matrices $\mathbf{X}_1, \mathbf{X}_2$).

Remark 1. *It is notable that cosine similarity is deeply related to CCA: if the column vectors used in cosine similarity were mean-centered, then the cosine similarity exactly corresponds to correlation. In other words, while CCA first projects data onto sparse directions made up of possibly several attribute-values then evaluates correlation, cosine similarity also computes correlation but only for projection directions that involve single attribute-values. We thus expect CCA to be superior whenever multi-valued matchings are desired, but that cosine similarity will often be similar.*

Performance Metrics Under two sources, we compare CCA's accuracy against the baselines using human ground truth. For multiple sources, we use precision, recall and runtime to compare multiple sparse CCA and multi-source extensions of two of the best baseline methods.

Results

Unambiguous Setting: Accuracy to Judgments

We compare CCA and baseline results using the crowd-sourced answers as ground truth, results shown in Table 3. CCA outperforms all methods when judgment consensus is high. The best text similarity method is MATCHANYWORD while the best probabilistic method is COSINESIMILARITY.

When text similarity results are being compared to crowd-sourced answers, we expect them to perform well as they reflect how judges measure similarity (Lee, Pincombe, and Welsh 2005). Conversely crowd-sourced answers with low judgment consensus correspond to attribute values where linguistic similarity is low. CosineSimilarity performed in line with CCA as expected (*cf.* Remark 1).

Manually inspecting the matches, the following matches were easily made by most methods including CCA and the text similarities. This is not surprising, but confirms that CCA discovers matches with high linguistic similarity.

Comedy	↔	Comedy
Music	↔	Musical / Performing Arts
Musical	↔	Musical / Performing Arts

By contrast, CCA triumphs over text-based similarities when attribute values are linguistically dissimilar. It discovers, *e.g.*, two matches that are difficult for some judges:

Documentary	↔	Special Interest
Thriller	↔	Mystery & Suspense

Ambiguous Setting: Accuracy to CCA

For the restaurant domain, we have more than a hundred cuisines per source, many of them without clear boundaries (*e.g.*, Chinese vs Cantonese). A non-expert human can no longer perform the matching task effectively. For example, Bangladeshi cuisine in Factual actually corresponds to Indian cuisine in Google Places, while many AMT judges when asked for a single matching cuisine will select Bangladeshi in Google Places even though it is rarely used in restaurant records. In this case, semantics are defined by co-occurrences rather than linguistic similarity. We therefore compare the baselines and human annotated answers to CCA, with results shown in Table 4. As expected CosineSimilarity highly agrees with CCA, while judges and text-based similarities achieve similar performance.

Algorithm	Accuracy
Exact Match	28
Match Any Word	51
Unigram	52
Bigram	66
Edit Distance	57
Human Annotators	69
Most Frequent Class	74
Cosine Similarity	98

Table 4: Comparison against CCA on restaurants.

Scaling to Multiple Sources

CCA is easily generalized to multiple sources. However, the same is not true for standard bisource approaches.

Accuracy Baseline methods considered, are naturally defined on two sources; searching globally (*e.g.*, to encourage transitivity and higher accuracy) can be achieved generically, with time exponential in the number of sources k. We simply consider all combinations of attribute-values across $D_1, \ldots, D_k$, scoring by summing $\binom{k}{2}$ pairwise similarities.

Figure 2 shows precision-recall curves on the two datasets for all four sources, measured against CCA since manual matching across sources is infeasible.

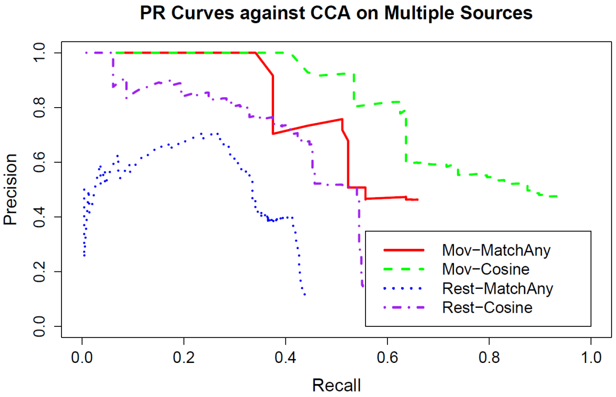

Figure 2: Precision-recall of the best text-based and probabilistic matchers on multiple sources against CCA.

Computational Efficiency We measure runtime on a PC with a 2.3GHz Intel Core i7 processor & 8GB of memory. For each algorithm, we compare the time required in Table 5 to compute baseline and CCA over (a) all pairwise sources and (b) multiple sources. The runtime differences were negligible for movies due to there being few attribute values. The combinatorial explosion becomes apparent for the extended baselines on restaurants where $120 \times 182 \times 151 \times 136 = 450m$ combinations are examined. In this case, multiple CCA is faster by a factor of 10 or more.

Method	Movie		Restaurant	
	Local	Global	Local	Global
Bigrams	0.14	4.36	4.17	8582
Cosine Similarity	8.16	15.0	235	13419
CCA	7.00	8.50	61.0	785

Table 5: Runtimes in seconds when algorithms match pairwise (locally) all on 4-tuples (globally).

Conclusion

We address the normalization of attribute values across data sources by canonical correlation analysis (CCA). We demonstrate on two crowd-annotated multi-source datasets, that multiple sparse CCA achieves high quality matches with fast runtime, beating baseline text and probabilistic approaches. The performance of CCA is most striking when scaling to multiple sources and attributes with larger domains. We consider practical issues such as picking the number of CCA components. Finally we demonstrate crowd-based annotations that find ground truth, and a case where human computation is infeasible. To the best of our knowledge, ours is the first application of CCA to databases, which highlights an excellent source of data for future CCA research. We are releasing our datasets to foster research.[1]

For future work we will explore schema matching by CCA: aligning columns by CCA on attribute-values, accepting alignments with scree plot support. We will also explore whether a Bayesian interpretation of CCA (Bach and

Jordan 2005) enables simultaneous attribute-value matching and Bayesian record merging (Zhao et al. 2012).

Acknowledgments

This work was supported by the Australian Research Council (DP150103710).

References

Bach, F. R., and Jordan, M. I. 2005. A probabilistic interpretation of canonical correlation analysis. Technical Report 688, Department of Statistics, University of California, Berkeley.

Bernstein, P. A.; Madhavan, J.; and Rahm, E. 2011. Generic schema matching, ten years later. *Proceedings of the VLDB Endowment* 4(11):695–701.

Dudoit, S.; Fridlyand, J.; and Speed, T. P. 2002. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association* 97(457):77–87.

Dunn, H. L. 1946. Record linkage. *American Journal of Public Health and the Nations Health* 36(12):1412–1416.

Gifi, A. 1990. *Nonlinear Multivariate Analysis*. Wiley.

Hardoon, D. R., and Shawe-Taylor, J. 2011. Sparse canonical correlation analysis. *Machine Learning* 83(3):331–353.

Hardoon, D.; Szedmak, S.; and Shawe-Taylor, J. 2004. Canonical correlation analysis: An overview with application to learning methods. *Neural computation* 16(12):2639–2664.

Hotelling, H. 1936. Relations between two sets of variates. *Biometrika* 28(3/4):321–377.

Kettenring, J. R. 1971. Canonical analysis of several sets of variables. *Biometrika* 58(3):433–451.

Köpcke, H., and Rahm, E. 2010. Frameworks for entity matching: A comparison. *Data & Knowledge Engineering* 69(2):197–210.

Köpcke, H.; Thor, A.; and Rahm, E. 2010. Evaluation of entity resolution approaches on real-world match problems. *Proceedings of the VLDB Endowment* 3(1):484–493.

Lai, P. L., and Fyfe, C. 2000. Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems* 10(05):365–377.

Lee, M. D.; Pincombe, B. M.; and Welsh, M. B. 2005. An empirical evaluation of models of text document similarity. In Bara, B. G.; Barsalou, L.; and Bucciarelli, M., eds., *XXVII Annual Conference of the Cognitive Science Society*, 1254–1259.

Rahm, E., and Bernstein, P. A. 2001. A survey of approaches to automatic schema matching. *The VLDB Journal* 10(4):334–350.

Sadinle, M.; Hall, R.; and Fienberg, S. E. 2011. Approaches to multiple record linkage. In *Proceedings of the 57th Session of the International Statistical Institute*. Invited paper http://www.cs.cmu.edu/~rjhall/ISIpaperfinal.pdf.

Shawe-Taylor, J., and Cristianini, N. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Shvaiko, P., and Euzenat, J. 2013. Ontology matching: State of the art and future challenges. *IEEE Transactions on Knowledge and Data Engineering* 25(1):158–176.

Torres, D. A.; Turnbull, D.; Sriperumbudur, B. K.; Barrington, L.; and Lanckriet, G. R. 2007. Finding musically meaningful words by sparse CCA. In *Neural Information Processing Systems (NIPS) Workshop on Music, the Brain and Cognition*.

Vinokourov, A.; Cristianini, N.; and Shawe-Taylor, J. S. 2002. Inferring a semantic representation of text via cross-language correlation analysis. In *Advances in neural information processing systems*, 1473–1480.

Winkler, W. E. 2006. Overview of record linkage and current research directions. Technical Report Statistics #2006-2, U.S. Census Bureau.

Witten, D. M.; Tibshirani, R.; and Hastie, T. 2009. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics* 10(3):515–534.

Zhao, B.; Rubinstein, B. I.; Gemmell, J.; and Han, J. 2012. A Bayesian approach to discovering truth from conflicting sources for data integration. *Proceedings of the VLDB Endowment* 5(6):550–561.