Information Theoretic Similarity Measures for Inter-domain Predicate Mapping

Nikita I. Lytkin
Dept. of Computer Science
Rutgers University
nikita.lytkin@rutgers.edu

William M. Pottenger
DIMACS
Rutgers University
billp@dimacs.rutgers.edu

Abstract

The development of similarity functions for first-order logic predicates and argument types is the initial step in the development of techniques for inter-domain predicate mapping. Predicate mappings established across textual data sources can be applied in federated text search during resource selection, and by systems such as Markov Logic Networks for transfer learning.

In this work, we propose similarity functions for mapping predicates and argument types. Each predicate is represented by a mutual information matrix characterizing statistical associations between predicate arguments. Drawbacks of using the Euclidean distance function as a similarity measure are discussed and mitigated in our approach. We also demonstrate that variations in the numbers of groundings of predicates have a significant and undesirable impact on their similarity scores, and propose a normalization scheme to address this deficiency.

Preliminary experimental results on real world datasets collected from the web demonstrate the effects of normalization of mutual information matrices and the resulting invariance of the similarity functions to variations in the numbers of groundings of predicates. The results also show that predicates that encode the same type of relations (e.g., one-to-many) tend to receive higher similarity scores than pairs of predicates where each predicate encodes a different type of relation (e.g., one-to-many and one-to-one).

Overall, our approach to measuring similarity for predicate mapping promises to scale to a number of text mining applications including federated search and retrieval, as well as other domains such as transfer learning.

1 Introduction

Federated text search [1, 3, 5] provides a unified search capability over heterogeneous data sources, and relies on methods of constructing resource descriptions [3, 4] and selecting resources [3, 5, 14] most appropriate for a given search query. One approach to resource description relies on identifying relationships in narrative textual data. This field, known as *relationship extraction*, involves mining textual data for relationships such as `President(‘‘George W. Bush’’, ‘‘United States’’)` [7, 18, 19, 2].

Relationship extraction produces instances of relationships (e.g., `WorkedUnder(‘‘Julia Roberts’’, ‘‘Steven Soderbergh’’)`) that are, essentially, groundings of first-order logic (FOL) predicates (e.g., `WorkedUnder(person,person)`). This inspires an approach to resource description based on the use of a FOL framework for predicate mapping.

We consider domain descriptions in terms of FOL predicates and their groundings. Building on a work [9] on schema matching [15, 17] published in the database community, we propose similarity functions for predicates and argument types. The similarity functions are based on characterizations of predicates by statistical interactions between their arguments. The characterizations are constructed using a measure of mutual information between the arguments.

The development of similarity functions for predicates and argument types is the initial step in the development of techniques for inter-domain predicate mapping. Predicate mappings established across textual data sources can then be applied in federated text search during resource selection performed by the search system while responding to a query.

The ability to automatically establish mappings between predicates across domains has applications in other areas as well. For example, predicate mapping would also allow learning systems, such as Markov Logic Networks (MLNs) [16], to be used in a transfer learning framework.

Transfer learning is concerned with developing machine learning methods of leveraging the knowledge learned in a source domain for improving the accuracy and speed of learning and inference in a target domain. Previous work in the area of transfer learning with MLNs either assumed that predicate mappings were known apriori [13], or relied on an exhaustive search procedure [12] for translating FOL clauses between domains. The apriori assumption limits the applicability of the approach to small domains where predicate mappings can be specified manually. The exhaustive search procedure [12], on the other hand, is not guaranteed to produce consistent predicate mappings across clauses. An approach to predicate mapping based on our similarity functions would promise to resolve these issues in a fashion that is scalable across application domains.

The paper is organized as follows. Related work is described in Section 2. The predicate mapping similarity functions along with illustrative computational examples are presented in Section 3. Experimental results are discussed in Section 4, followed by concluding remarks and an outline of future research directions proposed in Section 5.

2 Related Work

One area of research closely related to predicate mapping is automated schema matching. Automated schema matching has been studied [15, 17, 11, 9] in the database community as one of the key aspects of integration of disparate data sources in order to provide a centralized view of the data.

The goal of schema matching is to identify viable mappings between elements (e.g. tables and attributes of relational schemas, or XML attributes of XML schemas) of schemas in a collection, where each schema describes a different data source. A survey of automatic schema matching techniques was published in [15].

Two types of schema matching methods could be distinguished — schema-based and instance-based. Schema-based methods, such as those surveyed in [17], use *structures of schemas* in order to produce mappings between their elements. In [11], for example, a pair of schemas being matched is represented as a directed graph. Nodes of the graph correspond to pairs of elements (e.g. attributes of relational tables), each belonging to a different schema. Presence and direction of edges of the graph are determined according to the relations between elements within a schema. Initial similarity scores for the nodes are computed using a string-based similarity measure. The final similarity scores for the schema element pairs are produced by a fixed-point computation that propagates the initial similarity scores over the directed graph.

Instance-based methods, such as [9], rely on the analysis of *data instances* in order to perform schema matching. In the approach proposed in [9], statistical associations between attributes of a relational table were captured using a mutual information matrix. Given a pair of tables, the best matching between their attributes was determined by minimizing a distance function defined over the mutual information matrices.

Measures of association based on mutual information have also been applied in text mining. For example, the pointwise mutual information metric was applied in [6] for unsupervised extraction of facts from textual data on the web. This is related to our approach, which relies on mutual information to assess the similarity of predicates extracted from web data as well.

In another related effort, [8], a modified mutual information metric was used for extraction of highly associated terms (words or phrases) for constructing a product description by attribute-value pairs.

Extraction of functional key words for genes from biomedical literature is yet another example of an application of mutual information in text mining. In [10], mutual information was used as a measure of cluster quality in a method of clustering genes based on the functional key word associations.

3 Similarity Measures Based on Mutual Information

In this section, we describe the proposed similarity functions for mapping predicates and argument types. As the basis for the development of the similarity functions, we used the mutual information measure of attribute associations and the Euclidean distance function described in [9].

In Subsection 3.1, we discuss two drawbacks of using the Euclidean distance function [9] directly as a similarity function. The drawbacks stem from the unboundness of the range of values of the Euclidean distance function and its monotonic growth with the dimensionality of its arguments. In the same subsection, we also describe an improvement scheme that overcomes these drawbacks. An example of predicate similarity computation is given in Subsection 3.2.

In Subsection 3.3, we demonstrate that variations in the numbers of groundings of predicates have a significant and undesirable impact on their similarity scores. In order for the similarity functions to be robust to variations in the numbers of groundings of predicates, we propose a normalization scheme for mutual information matrices.

Subsection 3.4 describes the proposed similarity function for argument types. An example of type similarity computation is given in Subsection 3.5.

3.1 Similarity Measure for Predicates

Given predicates X and Y, we represent each predicate as a relational table whose attributes (columns) correspond to predicate's arguments while records (rows) correspond to the true groundings of the predicate.

Following a mutual-information-based method of database schema matching published in [9], square $n \times n$ matrices $M^K = \|m_{ij}^K\|$ are computed for each $K \in \{X, Y\}$, where n is the arity of (i.e. the number of arguments in) predicates X and Y, and m_{ij}^K is the *mutual information* coefficient computed on a pair of attributes $i \in K$ and $j \in K$.

The mutual information of two random variables A and B is a non-negative symmetric quantity defined as

$$M(A,B) = \sum_{a \in \mathrm{dom}(A)} \sum_{b \in \mathrm{dom}(B)} p(a,b) \log \frac{p(a,b)}{p(a)p(b)}. \tag{3.1}$$

The mutual information measures the reduction in entropy of one random variable due to knowledge of the value of the other random variable.

The proposed similarity function is based on the Euclidean distance function defined for a pair of matrices M^X and M^Y as

$$E(M^X, M^Y) = \sqrt{\sum_{i,j} (m_{ij}^X - m_{ij}^Y)^2}. \tag{3.2}$$

Let us note that the value of (3.2) depends on the correspondence between indices of matrices M^X and M^Y. This dependence must be eliminated by the similarity function, because the order of arguments in a predicate may be arbitrary (e.g. `Publication(title,person)` and `Publication(person,title)`).

Invariance of the similarity function to the order of indices of mutual information matrices is achieved by holding the order of indices of one of the matrices fixed, while considering all possible permutations of indices of the other matrix. For each permutation, the Euclidean distance (3.2) is computed. The minimum value of the Euclidean distance function is then used in the similarity function.

Formally, we compute

$$\hat{E}(M^X, M^Y) = \min_{\tau \in \mathcal{I}_Y} E(M^X, M_\tau^Y), \tag{3.3}$$

where $\mathcal{I}_Y$ is the set of all permutations of indices of matrix M^Y, and M_τ^Y is the matrix with rows and columns permuted according to $\tau \in \mathcal{I}_Y$.

In order for the similarity coefficients to be comparable across pairs of predicates, the similarity function must be independent of the size of mutual information matrices, and the range of values

person	person
student265	adviser168
student381	adviser168
student176	adviser407
student352	adviser415
student352	adviser292
...	...

Table 1: A sample of groundings of predicate `AdvisedBy(person,person)`

person	person
Casey Affleck	Soderbergh Steven
Elliott Gould	Soderbergh Steven
Julia Roberts	Soderbergh Steven
Denzel Washington	Pakula Alan J.
Julia Roberts	Pakula Alan J.
...	...

Table 2: A sample of groundings of predicate `WorkedUnder(person,person)`

of the similarity function must be bounded. Function (3.3), however, is unbounded and is monotonically non-decreasing with increasing size n of mutual information matrices.

Therefore, we propose the following exponential similarity function

$$S(M^X, M^Y) = \exp\left(-\frac{1}{n}\hat{E}(M^X, \hat{M}^Y)\right), \quad (3.4)$$

which is independent of the size n of the mutual information matrices, and whose range of values lies in a bounded interval (0,1]. A similarity score of one indicates the maximum similarity possible. Whereas, a similarity score below $\frac{1}{\mathrm{e}}$ suggests that the two predicates are highly dissimilar.

3.2 Example of Similarity Score Computation for Predicates Consider predicates `AdvisedBy(person,person)` and `WorkedUnder(person,person)`. Predicate `AdvisedBy` relates advisers to students, while `WorkedUnder` relates film directors to actors who worked for them. Examples of groundings of the two predicates are shown in Tables 1 and 2. In Table 1, names of advisers and students have been anonymized.

Two mutual information matrices

$$M^A = \begin{pmatrix} 6.22 & 4.39 \\ 4.39 & 4.77 \end{pmatrix} \text{ and } M^W = \begin{pmatrix} 7.43 & 3.77 \\ 3.77 & 4.92 \end{pmatrix} \quad (3.5)$$

are computed over the groundings of predicates `AdvisedBy` and `WorkedUnder`, respectively.

Let set $\mathcal{I}_W = \{(1,2),(2,1)\}$ denote the set of all possible permutations of indices of matrix M^W, where $M^W_{(1,2)} = M^W$ and

$$M^W_{(2,1)} = \begin{pmatrix} 4.92 & 3.77 \\ 3.77 & 7.43 \end{pmatrix}.$$

The Euclidean distance function (3.2) is computed over M^A and M^W_τ for all $\tau \in \mathcal{I}_W$, yielding $E(M^A, M^W_{(1,2)}) = 1.51$ and $E(M^A, M^W_{(2,1)}) = 3.08$.

Therefore, the value produced by function (3.3) is $\hat{E}(M^A, M^W) = E(M^A, M^W_{(1,2)}) = 1.51$, and the resulting similarity score between predicates AdvisedBy and WorkedUnder is $S(M^A, M^W) = 0.471$.

3.3 Normalization of Mutual Information Matrices Consider a *uniformly* distributed random variable X defined over a domain $\mathcal{X}$ of discrete values. The entropy of X

$$H(X) = -\sum_{x\in\mathcal{X}} p(x)\log p(x) = \log|\mathcal{X}|$$

monotonically increases with increasing size of domain $\mathcal{X}$.

Similarly, the sizes of domains of two random variables X and Y affect the value of their mutual information $M(X,Y)$. In the view of our work, this means that the similarity measure (3.4) is affected by the number of groundings of the predicates.

In order to neutralize the effect the numbers of groundings have on the similarity measure (3.4), two normalization schemes were considered:

- by row-vector length

$$\tilde{m}^X_{ij} = \frac{m^X_{ij}}{\|m^X_{i\cdot}\|}$$

- by the sum of row-vector components

$$\tilde{m}^X_{ij} = \frac{m^X_{ij}}{\sum\limits_j m^X_{ij}}, \quad (3.6)$$

where $m_{i\cdot}^X$ denotes the i-th row-vector of a mutual information matrix M^X computed over the groundings of a predicate X.

A distinguishing feature of the proposed normalization (3.6) is that it results in the similarity measure (3.4) being computed based on a *relative* measure of statistical association of the predicates' arguments rather than the *absolute* values of their mutual information. Hence, we employed this normalization scheme in the experiments.

We demonstrate the effect of normalization on the similarity coefficients by extending the example given in Section 3.2. Normalization of mutual information matrices (3.5) of predicates `AdvisedBy` and `WorkedUnder` according to scheme (3.6) yields

$$\tilde{M}^A = \begin{pmatrix} 0.59 & 0.41 \\ 0.48 & 0.52 \end{pmatrix} \text{ and } \tilde{M}^W = \begin{pmatrix} 0.66 & 0.34 \\ 0.43 & 0.57 \end{pmatrix}.$$

The resulting similarity score $S(\tilde{M}^A, \tilde{M}^W) = 0.939$ indicates a much stronger similarity between the predicates than the previously attained score $S(M^A, M^W) = 0.471$.

3.4 Similarity Measure for Argument Types

In this section, we describe a similarity function for predicate argument types. First, let us introduce two auxiliary functions. We define a function

$$\delta(x) = \begin{cases} 1, & \text{if } x = 0 \\ 0, & \text{otherwise} \end{cases}$$

and introduce the following function that computes the normalized Euclidean distance between rows of mutual information matrices of predicates X and Y

$$\begin{aligned} U(i, M^X, p, M^Y) = \quad & \exp\left(-\sqrt{\tfrac{1}{n}\sum_j (m_{ij}^X - m_{pj}^Y)^2}\right) \times \\ & \delta(\tau_i(M^X, M^Y) - p), \end{aligned} \tag{3.7}$$

where $\tau_i(M^X, M^Y)$ is the i-th component of permutation vector

$$\tau(M^X, M^Y) = \arg\min_{\tau \in \mathcal{I}_Y} E(M^X, M_\tau^Y). \tag{3.8}$$

Let $T_k^A = \{(i, M^X)\}$ be the set of all pairs (i, M^X), such that predicate X belongs to domain A and the i-th row of mutual information matrix M^X corresponds to argument type k. The proposed similarity function for argument types k and l of domains A and B, respectively, can now be defined as

$$S_a(k,l) = \frac{\sum\limits_{(i,M^X) \in T_k^A} \sum\limits_{(p,M^Y) \in T_l^B} S(M^X,M^Y) \times U(i,M^X,p,M^Y)}{\sum\limits_{(i,M^X) \in T_k^A} \sum\limits_{(p,M^Y) \in T_l^B} S(M^X,M^Y) \times \delta(\tau_i(M^X,M^Y) - p)}. \tag{3.9}$$

person	position
adviser168	faculty
adviser407	faculty
adviser415	faculty
adviser292	faculty affiliate
adviser349	faculty adjunct
...	...

Table 3: A sample of groundings of predicate `Position(person,position)`

person	gender
Casey Affleck	male
Charlotte Rampling	female
Denzel Washington	male
Julia Roberts	female
Gina Chiarelli	female
...	...

Table 4: A sample of groundings of predicate `Gender(person,gender)`

3.5 Example of Similarity Score Computation for Argument Types

Consider predicates `Position(person,position)` that relates persons to academic faculty positions, `Gender(person,gender)` that relates film actors and directors to genders, and `WorkedUnder(person,person)`. Sample groundings of these predicates are shown in Tables 3, 4 and 2, respectively.

Suppose that domain A contains predicate `Position` and the associated argument types, while domain B contains the other two predicates and their argument types.

Below, we demonstrate the computation of similarity score for argument type pair (`person`,`person`). In calculations that follow, predicates' names are abbreviated by their first letters. For ease of exposition, *normalized* mutual information matrices computed over the groundings of the predicates are denoted by M^P, M^G and M^W, and are not shown due to space constraints.

Since there is only one predicate in domain A

	person	gender
person	0.895	0.000
position	0.936	1.000

Table 5: An example of similarity scores computed for argument types

and only the first argument of that predicate is of type `person`, set T^A_{person} is comprised of a single element $T^A_{\text{person}} = \{(1, M^P)\}$. Set T^B_{person}, on the other hand, contains three elements $T^B_{\text{person}} = \{(1, M^G), (1, M^W), (2, M^W)\}$, because the first argument of predicate G is of type `person` and both arguments of predicate W are of type `person`.

Permutations, computed by formula (3.8), of indices of mutual information matrices are $\tau(M^P, M^G) = (1, 2)$ and $\tau(M^P, M^W) = (1, 2)$. The similarity score for argument type pair (`person`,`person`) is computed according to (3.4) as

$$S_a(\text{person,person}) = \frac{S(M^P,M^G)U(1,M^P,1,M^G)+S(M^P,M^W)U(1,M^P,1,M^W)}{S(M^P,M^G)+S(M^P,M^W)} = \frac{0.949\times0.93+0.888\times0.857}{0.949+0.888} = 0.895.$$

Similarity scores for all pairs of argument types are shown in Table 5.

4 Results

Data for two domains was used in the experiments. The *academic* domain describes the Department of Computer Science and Engineering at the University Of Washington, and was provided by the authors of [16]. The *film* domain describes the film-making industry and was provided by the authors of [12].

Predicates of the academic domain relate students to their advisers (`AdvisedBy`), faculty members to the types of their positions (`Position`) and authors to publication titles (`Publication`). Predicates of the film domain relate persons to genders (`Gender`), persons to genres (`Genre`), persons to film titles (`Movie`) and actors to directors (`WorkedUnder`). Both domains also contain predicates that indicate "sameness" of a pair of objects (e.g. `SameCourse`, `SameMovie`, etc.). The predicates are listed in Table 6. The number of groundings of each predicate is shown in Table 7.

Academic	Film
AdvisedBy(person,person)	WorkedUnder(person,person)
Publication(title,person)	Movie(title,person)
Position(person,position)	Genre(person,genre)
SameCourse(course,course)	SameMovie(title,title)
SamePerson(person,person)	SamePerson(person,person)
	Gender(person,gender)

Table 6: Predicates from the academic and film domains

Academic		Film	
AdvisedBy	97	WorkedUnder	382
Publication	628	Movie	286
Position	48	Genre	47
SameCourse	122	SameMovie	20
SamePerson	312	SamePerson	268
		Gender	236

Table 7: Number of groundings of each predicate

We distinguish three groups of predicates by the types of relationships they encode. The *first group* consists of predicates `AdvisedBy`, `Publication`, `Genre`, `Movie` and `WorkedUnder` that encode many-to-many relations between objects of the corresponding domains. For example, an adviser may be associated with several students, while a student may have more than one adviser. Similarly, several persons may participate in the creation of a film, while a person may be involved in several films.

The *second group* consists of predicates `Position` and `Gender` that encode one-to-many relations. A person may occupy only one type of a faculty position in the academic domain, but multiple persons may hold the same type of a position. In the film domain, a person is associated with a single gender, but there are multiple persons of the same gender.

The *third group* consists of predicates `SameCourse` and `SamePerson` that encode one-to-one relations (assuming that there is only one object signifying a particular course, or a person).

Since similarity measure (3.4) is solely based on statistical properties of the relationships the predicates encode, we expect the pairwise similarity scores to be higher for predicates in the same group than for predicates that belong to different groups.

Similarity scores for predicates in the first and second groups are presented in Tables 8 and 9. The former contains similarity scores computed over non-normalized mutual information matrices,

	Genre	Movie	WorkedUnder	Gender
AdvisedBy	0.076	0.389	0.471	0.040
Publication	0.045	0.546	0.579	0.038
Position	0.405	0.058	0.066	0.309

Table 8: Similarity scores, for the first and second predicate groups, computed using non-normalized mutual information matrices

	Genre	Movie	WorkedUnder	Gender
AdvisedBy	0.858	0.950	0.939	0.806
Publication	0.908	0.961	0.993	0.841
Position	0.892	0.893	0.888	0.949

Table 9: Similarity scores, for the first and second predicate groups, computed using normalized mutual information matrices

	SameMovie	SamePerson
AdvisedBy	0.376	0.040
Publication	0.142	0.051
Position	0.065	0.002
SameCourse	0.074	0.321
SamePerson	0.019	0.803

Table 10: Similarity scores, for the third predicate group, computed using non-normalized mutual information matrices

	SameMovie	SamePerson
AdvisedBy	0.939	0.939
Publication	0.889	0.889
Position	0.799	0.799
SameCourse	1.000	1.000
SamePerson	1.000	1.000

Table 11: Similarity scores, for the third predicate group, computed using normalized mutual information matrices

while the latter demonstrates the effects of normalization scheme (3.6).

In both cases, predicates `AdvisedBy` and `Publication` received higher similarity scores with predicates of the same group (i.e. `Genre`, `Movie` and `WorkedUnder`) than with `Gender`. By considering the two tables column-wise, it can be seen that `Movie` and `WorkedUnder` also attained higher similarity scores with predicates of the same group (i.e. `AdvisedBy` and `Publication`). Predicate `Gender` received higher similarity score with `Position` than with any other predicate.

When no normalization was used (Table 8), predicate `Genre` attained its highest similarity score with `Position`. Moreover, `Position` attained its highest similarity score with `Genre`, and not with `Gender` – the other predicate in the second group.

Once the normalization was employed (Table 9), `Position` and `Gender` became more similar to each other than to any other predicate of the first group. `Genre` became most similar with `Publication`. Although, `Genre` maintained a higher similarity score with `Position` than with `AdvisedBy` (as was the case when no normalization was used).

The effect the number of groundings of a predicate has on its similarity scores with other predicates is made apparent by Table 10. Columns titled `SameMovie` and `SamePerson` correspond to the two predicates that encode one-to-one relations in the film domain. Statistical patterns encoded by these predicates are identical. However, the similarity scores for these predicates vary widely when no normalization is used. For instance, `SameCourse` and `SameMovie` received the similarity score of 0.074 while `SameCourse` and `SamePerson` received 0.321 simply due to the variability in the numbers of groundings of these predicates.

The normalization (Table 11) removed the variability in similarity scores due to varying numbers of groundings of the predicates. As a result, pairs of predicates of the third group received the maximum possible similarity score of one. Similarity scores for pairs where only one of the predicates was from the third group were strictly smaller than one (due to space limitations, only a representative sample of these scores is shown in Table 11).

The effects of normalization can also be observed by considering similarity scores presented in Tables 12 and 13 for argument types.

Types `course` and `gender` had a very small similarity score of 0.002 when no normaliza-

	person	title	genre	gender
person	0.528	0.480	0.063	0.023
title	0.680	0.073	0.000	0.000
position	0.041	0.048	0.391	0.755
course	0.254	0.071	0.006	0.002

Table 12: Similarity scores, for argument types, computed using non-normalized mutual information matrices

	person	title	genre	gender
person	0.892	0.947	0.886	0.972
title	0.910	0.857	0.000	0.000
position	0.966	0.996	0.860	1.000
course	0.880	0.998	0.860	1.000

Table 13: Similarity scores, for argument types, computed using normalized mutual information matrices

tion was used. Once the mutual information matrices were normalized, the (`course`,`gender`) pair received the maximum possible score of one. The reason for this result lies in the relations these argument types participate in, namely `SameCourse(course,course)` and `Gender(person,gender)`.

As was stated earlier, predicate `SameCourse` encodes a one-to-one relation between objects (courses) of the academic domain. Therefore, knowing the value of one of the arguments of `SameCourse` completely determines the value of the other argument. Predicate `Gender`, on the other hand, encodes a one-to-many relation. Hence, knowing the value of the first argument (person) of predicate Gender completely determines the value of its second argument (gender). The converse, however, is not true — knowing a person's gender does not uniquely identify the person.

Once the normalization was used, such patterns of association between argument types resulted in (`course`,`gender`) receiving the maximum similarity score of one, while the pair (`course`,`person`) received a lower score of 0.88.

Type pair (`position`,`gender`) also received a score of one as a result of using the normalization and due to the nature of one-to-many relations encoded by predicates `Position` and `Gender`.

As can be seen in Table 13, the type pair (`person`,`person`) received a lower similarity score than (`person`,`title`), and (`title`,`title`) received a lower score than (`title`,`person`). This result could potentially be improved by introducing constraints on what type of predicates are considered for similarity scoring. Note that computing a similarity score for predicates `SamePerson(person,person)` and `Movie(title,person)` forces the computation of similarity score for type pair (`person`,`title`). In such situations, it may be reasonable to introduce a constraint that would prevent predicates with different *patterns* of argument types from being scored. Under this constraint, predicates `SamePerson(person,person)` and `Movie(title,person)` will not be scored, while `SamePerson(person,person)` and `SameMovie(title,title)` will (and so will predicates `Publication(title,person)` and `Movie(title,person)`).

5 Conclusion and Future Work

In this work, we proposed similarity functions for mapping predicates and argument types in text mining. Mutual information matrices were used for characterizing predicates by patterns of statistical relations between a predicate's arguments.

The Euclidean distance function defined over mutual information matrices was used as the basis in the development of the similarity functions. Unboundness of the range of values of the Euclidean distance function and its monotonic growth with the dimensionality of mutual information matrices rendered the Euclidean distance function inadequate as a similarity function for a pair of matrices.

The range of values of the similarity functions was confined to a bounded interval (0,1] by incorporation of the Euclidean distance function into an exponential function.

The monotonic growth of the Euclidean distance function with the dimensionality of the matrices was circumvented by introducing into the similarity functions a normalization term dependent on the size of the matrices.

We demonstrated that variations in the numbers of groundings of predicates had a significant impact on their similarity scores. In order for the similarity functions to be robust to variations in the numbers of groundings of predicates, a normalization of mutual information matrices was proposed.

Experimental results demonstrated the effects of normalization of mutual information matrices and the resulting invariance of the similarity functions to variations in the numbers of groundings of predicates.

The results also showed that predicates that encode the same type of relations (e.g., one-to-many) tend to receive higher similarity scores than pairs of predicates where each predicate encodes a different type of relation (e.g., one-to-many and one-to-one).

Data used in the preliminary experiments discussed in this paper contained a limited collection of predicate groundings drawn from data available on the web. Mutual information, on the other hand, normally requires large samples of data in order to produce reliable estimates. To this end, it is necessary in future work to evaluate our similarity functions on data sets with large numbers of groundings as well as higher-arity predicates. Resource descriptions based on relationship extraction are an ideal candidate for this future work. Predicate mapping applied in transfer learning forms another candidate for more extensive experimentation with our approach.

Another potential direction for future work lies in an investigation of approaches to combining, in a single similarity function, the analysis of statistical patterns of associations between entities in the domains with a deeper analysis of the textual content of those entities.

6 Acknowledgments

This research is sponsored by the Defense Advanced Research Projects Agency (DARPA) and managed by the Air Force Research laboratory (AFRL) under contract FA8750-05-2-0283.

The authors thank Dr. Ilya B. Muchnik for suggestions that helped improve the presentation of the material in Section 3.

Co-author William M. Pottenger thanks His Lord and Savior, Jesus the Messiah, for the love and grace He has shown to him. Thank you, Lord Jesus!

References

[1] T. T. Avrahami, L. Yau, L. Si, and J. Callan. The fedlemur project: Federated search in the real world. *J. Am. Soc. Inf. Sci. Technol.*, 57(3):347–358, 2006.

[2] R. C. Bunescu and R. J. Mooney. Statistical relational learning for natural language information extraction. In L. Getoor and B. Taskar, editors, *Introduction to Statistical Relational Learning*, chapter 19. The MIT Press, November 2007.

[3] J. Callan. *Distributed information retrieval*, chapter 5, pages 127–150. Advances in information retrieval. Kluwer Academic Publishers, 2000.

[4] J. Callan, M. Connell, and A. Du. Automatic discovery of language models for text databases. In *SIGMOD '99: Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, pages 479–490, New York, NY, USA, 1999. ACM.

[5] N. Craswell. Methods for distributed information retrieval, 2000. Ph.D. thesis, The Australian Nation University.

[6] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Unsupervised named-entity extraction from the web: an experimental study. *Artif. Intell.*, 165(1):91–134, 2005.

[7] R. Feldman and B. Rosenfeld. Boosting unsupervised relation extraction by using ner. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, pages pages 473–481, Sydney, July 2006.

[8] R. Ghani, K. Probst, Y. Liu, M. Krema, and A. Fano. Text mining for product attribute extraction. *SIGKDD Explor. Newsl.*, 8(1):41–48, 2006.

[9] J. Kang and J. F. Naughton. On schema matching with opaque column names and data values. In *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 205–216, New York, NY, USA, 2003. ACM Press.

[10] Y. Liu, S. B. Navathe, J. Civera, V. Dasigi, A. Ram, B. J. Ciliax, and R. Dingledine. Text mining biomedical literature for discovering gene-to-gene relationships: A comparative study of algorithms. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 2(1):62–76, 2005.

[11] S. Melnik, H. Garcia-Molina, and E. Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *ICDE '02: Proceedings of the 18th International Conference on Data Engineering*, page 117, Washington, DC, USA, 2002. IEEE Computer Society.

[12] L. Mihalkova, T. Huynh, and R. J. Mooney. Mapping and revising markov logic networks for transfer learning. In *AAAI*, pages 608–614. AAAI Press, 2007.

[13] L. Mihalkova and R. J. Mooney. Transfer learning with markov logic networks. In *Proceedings of the ICML Workshop on Structural Knowledge*

Transfer for Machine Learning, Pittsburgh, PA, July 2006.

[14] H. Nottelmann and N. Fuhr. Evaluating different methods of estimating retrieval quality for resource selection. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 290–297, New York, NY, USA, 2003. ACM.

[15] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *VLDB Journal: Very Large Data Bases*, 10(4):334–350, 2001.

[16] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107–136, 2006.

[17] P. Shvaiko and J. Euzenat. A survey of schema-based matching approaches. *J. Data Semantics IV*, pages 146–171, 2005.

[18] C. A. Thompson and R. J. Mooney. Acquiring word-meaning mappings for natural language interfaces. *J. Artif. Intell. Res. (JAIR)*, 18:1–44, 2003.

[19] A. G. Valarakos, V. Karkaletsis, D. Alexopoulou, E. Papadimitriou, and C. D. Spyropoulos. Populating an allergens ontology using natural language processing and machine learning techniques. In S. Miksch, J. Hunter, and E. T. Keravnou, editors, *AIME*, volume 3581 of *Lecture Notes in Computer Science*, pages 256–265. Springer, 2005.