Web Semantics: Science, Services and Agents on the World Wide Web 9 (2011) 52–70

Contents lists available at ScienceDirect

Web Semantics: Science, Services and Agents on the World Wide Web

journal homepage: www.elsevier.com/locate/websem

Enhanced geographically typed semantic schema matching

Jeffrey Partyka[a], Pallabi Parveen[a], Latifur Khan[a,*], B. Thuraisingham[a], Shashi Shekhar[b]

[a] Department of Computer Science, University of Texas at Dallas, 800 West Campbell Rd., Richardson, TX 75080-3021, USA
[b] Department of Computer Science, University of Minnesota, 4-192 EE/CS Bldg, 200 Union St. SE, Minneapolis, MN, USA

ARTICLE INFO

Article history:
Received 15 April 2010
Received in revised form 9 November 2010
Accepted 26 November 2010
Available online 3 December 2010

Keywords:
Schema
GIS
Gazetteer
Geocoding
Geotypes
Geosemantics

ABSTRACT

Resolving semantic heterogeneity across distinct data sources remains a highly relevant problem in the GIS domain requiring innovative solutions. Our approach, called GSim, semantically aligns tables from respective GIS databases by first choosing attributes for comparison. We then examine their instances and calculate a similarity value between them called entropy-based distribution (EBD)[1] by combining two separate methods. Our primary method discerns the geographic types from instances of compared attributes. If successful, EBD is calculated using only this method. GSim further facilitates geographic type matching by using latlong values to further disambiguate between multiple types of a given instance and applying attribute weighting to quantify the uniqueness of mapped attributes. If geographic type matching is not possible, we then apply a generic schema matching method, independent of the knowledge domain, which employs normalized Google distance. We show the effectiveness of our approach over the traditional approaches across multi-jurisdictional datasets by generating impressive results.

1. Introduction

The amount of geospatial data that is accumulating in gazetteers, geodatabases and many other geographic data sources continues to increase at a very fast pace. One of the results of this is the proliferation of independent and heterogeneous data repositories of geospatial data accumulated by an increasingly disparate set of processes. For instance, unmanned aerial vehicles may take snapshots of a land area to analyze its transformation over a period of time [39], and sensor networks commonly are employed to measure the water level of a river to analyze its potential for producing flooding conditions [40].

Because of this, questions regarding the feasibility and potential applications for integrating geospatial data in these repositories have arisen. These questions are some of the most crucial questions regarding information integration, which extends far beyond the geospatial domain. It has been explored in the form of semantic similarity research from cognitive science, information retrieval and artificial intelligence conducted over the past few decades [41–43]. With regard to the geospatial domain, geospatial data inherently possess vagueness, uncertainty and varying levels of granularity [36]. Different sets of data modeling the same geographic location may be represented by differing file formats, type representations, coordinate reference systems, projections, natural language text descriptions, and much more. As a result, measuring the semantic similarity of geospatial data is a uniquely challenging problem that will continue to require innovative solutions that are increasingly sophisticated as the unique properties of geospatial data become better understood.

Semantic similarity in the geospatial domain has been successfully applied to numerous information retrieval and ranking problems, including geolocation [29], text classification [30], geospatial tagging, land cover similarity [34], ontology alignment [4,24,25,31,38,44], recreational tasks like route planning for mountain climbing [20], more serious tasks like emergency response decision making [22] and much more. Furthermore, the success of the geospatial Semantic Web depends very much on semantic similarity algorithms being able to determine commonalities and differences between geospatial data and their data models [28]. Efforts such as the Data Web and LinkedGeoData [19] represent transitional efforts progressing towards a geospatial Semantic Web, as they connect disparate geospatial datasets to better facilitate geographic information retrieval and semantic similarity.

The main focus of our research is determining the semantic similarity between geospatial data within compared schemas. Research into the problem of schema matching within the geospatial domain would seem to be integral to the above efforts. Relatively speaking, though, it has not received very much attention. A number of research efforts [1,2,47] have focused on instance-based schema matching methods that depend on the semantics embedded in

* Corresponding author. Tel.: +1 972 883 4137; fax: +1 972 883 2349.
E-mail addresses: jlp072000@utdallas.edu (J. Partyka), pxp013300@utdallas.edu (P. Parveen), lkhan@utdallas.edu (L. Khan), Bhavani.thuraisingham@utdallas.edu (B. Thuraisingham), shekhar@cs.umn.edu (S. Shekhar).

[1] EBD = entropy based distribution; GT = geographic type; NGT = non-geographic type; GD = normalized Google distance.

1570-8268/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.websem.2010.11.002

structured information, such as a domain ontology, to identify correct correspondences. However, if a domain ontology is not available, or if it is not designed well (either because it is incomplete or subjective), then these methods will not work very well.

In this paper, we introduce GSim, an information-theoretic algorithm used to measure instance similarity between compared attributes in geospatial schemas. Unlike the above methods of geospatial schema matching, GSim does not require any structured information for assistance in deriving attribute matches. At a high level, it works by first comparing tables. This is done by first determining pairs of attributes between the tables that are to be compared. Comparing all attributes of the compared tables and their instances against one another would result in a significant time penalty. Therefore, as a preprocessing measure, we use attribute name and data type matching to reduce the space of possible attribute mappings. Second, for each pair, we examine the respective attributes' instance data using two separate instance-similarity methods. Third, we determine corresponding attributes across tables based on semantic similarity scores. Combining all of the scores from aligned attributes will determine similarity between the tables as a whole.

GSim's primary approach for examining instance data determines the geographic types (GT) over the instances associated with compared attributes. This is done by leveraging an external data source known as a gazetteer [12,13] this might happen if the feature has a common name, such as "Johnson", as there might be "Johnson Road", "Johnson River", etc. A more advanced geotyping algorithm, which GSim features, is able to identify exactly one GT for any instance recognized by a gazetteer with the help of latlong values. Latlong values help in disambiguating several instances with the same name, such that the proper GT may be associated with the instance in the schema. Of course, the effectiveness of this approach depends on whether the feature-type thesaurus in the gazetteer contains a set of types that is able to represent all of the instances from our data. Thus, we have made an assumption in this paper that any instance in our data set that can be identified by a gazetteer has a type which can be represented by that gazetteer.

Whenever possible, GSim calculates similarity between attributes using GTs alone. If the dataset contains latlong values associated with each instance, then based on our type assumption we made above, it is possible to guarantee a 1:1 mapping between each instance and its GT as identified by a gazetteer. However, due to the great variability in how geographic data is stored and represented, not all geographic instances necessarily come with latlong values. Thus, when the instances are fed to a gazetteer, we may derive more than one GT for certain instances. As a result, the best that can be done is to derive 1:N mappings of these instances to their respective sets of GTs. Subsequently, similarity is calculated using these mapped GTs after applying a pruning algorithm for disambiguation purposes.

In the case where too many instances within the compared attributes lack GT information, then GSim resorts to its secondary approach, which uses a generic schema matching algorithm based on a semantic distance measure known as normalized Google distance (GD) [23]. GD, combined with K-medoid clustering of the instances of an attribute, yields a set of non-geographic types specific to that attribute which are then used to compute semantic similarity with another attribute in a different table. This method is generic because it is not dependent on geographic types at all. Despite the utility of GD, solely relying on it to determine similarity is unwise, particularly in the GIS domain. The reason is that a number of situations exist where the instances are determined to be similar due entirely to their close geographic proximity. One such situation is depicted in Section 4.2. GD, which depends on the Web pages indexed by the Google search engine, was chosen because of its effective coverage of the GIS domain. This is in contrast to an external knowledge source such as WordNet [48], a lexical database of English containing over 117,000 synsets and over 200,000 word-sense pairs. While the coverage of WordNet is quite extensive for various domains, for the GIS domain, it is not very extensive at all. Once the instance type has been determined, similarity is calculated by considering the collection of types extracted from instances between the compared attributes. It is based on an information-theoretic measure known as entropy-based distribution (EBD), which is defined as the ratio of the conditional entropy within each type over a pair of compared attributes with the entropy taken over all types for that same pair. An EBD value has a range from 0 to 1, with 0 indicating no similarity whatsoever between the attributes, and 1 indicating identical attributes. The more similar that (1: the sets of GTs between the compared attributes are, and (2: the number of instances representing identical types between the compared attributes are, the higher the EBD will be, and vice versa. A formal definition of EBD is given in Section 3.3.

The major advantage of using of an information-theoretic measure over other semantic similarity measures is its versatility and lack of constraints. Other similarity methods, such as those that use description logic (DL) [31], NLP [35] or network based matching [25], require a strictly defined set of relationships between concepts and attributes. For example, calculating the difference in depths between two concepts (as one would do in network-based approaches), or determining a common parent between two concepts (as one might do in DL approaches) is only possible if the concepts are represented in a hierarchy, such as an ontology. NLP approaches are dependent upon the relationships between the words in a natural language description of a geographical concept. In turn, this depends on the presence of natural language text in a geographic concept, which is not guaranteed, the use of a language for which part of speech tagging can be confidently applied, etc. On the other hand, information-theoretic measures like EBD do not require that data in attributes or concepts be organized in any way. In fact, a flat structure of attributes and concepts is not a problem for an information-theoretic measure. The only requirement that it has is for a probabilistic model to be applicable to the data being compared [37]. If this is the case, then information-theoretic measures can also be combined with any other semantic similarity technique, and it can be applied to various data models. For instance, although we applied the EBD measure in GSim to geodatabases, it can also be applied to ontology matching for Semantic Web applications. Since GSim uses instance-based matching to align attributes between tables in a 1-1 fashion, it can also be used to align the properties associated with the concept instances in a 1-1 fashion. This results in the alignment of concepts between ontologies.

Regardless of whether GSim uses GT matching or GD to match instances, it performs matching at the concept/attribute level by examining instances belonging to those compared attributes/concepts. We consider every attribute/concept matched by GSim to consist of a set of one or more instances (in our experiments, it should be noted that every attribute contained $\geq$24 instances). Therefore, if the similarity between the instances of compared attributes/concepts is high, then this implies that the attributes/concepts themselves share this similarity. On the other hand, if the similarity between the instances of compared attributes/concepts is low, then this again implies that the attributes/concepts themselves share this similarity.

In this paper, we compare three pairs of geospatial data sources using their respective table instances in an effort to determine their similarity; the first pair contains tables describing similar models of transportation networks over multiple jurisdictions, the second pair contains tables detailing varying geographic features beyond road networks, and the third pair contains a mixture of road network data and POI data. The data sources contain large variations

in the geographic areas covered, the number of attributes and the number of instances.

The challenges that we will address in this paper are as follows. First, the derivation of attribute mappings between two compared tables, along with the similarity calculations for each attribute mapping, may be accomplished in many different ways. We intend to clearly distinguish our method from others applied to schema matching. Second, only in the ideal case does the gazetteer match one specific GT for each of the instances. In reality, some instance names, such as "Clinton", are very common, and as a result, the gazetteer is likely to return several GTs. Thus, the challenge of handling multiple possible GTs for a given instance is addressed. The third challenge addressed by our research is the problem of determining the most uniquely relevant attributes within a particular table. It is possible for two tables to share a high similarity score based on matching attributes which are not relevant to the concept that the tables represent. Additionally, these attributes may be commonly occurring relative to other attributes within tables of the same data source. To remedy this, GSim applies attribute weighting to measure table similarity by placing more weight on those attributes which are more relevant to their respective tables and more unique, relative to all other attributes in the data source. This way, the measured EBD value generated for any given table comparison will be based on attributes that represent the essence of the compared tables.

Our contributions in this paper are as follows. First, we describe GSim, a method of aligning geospatial schemas using an information-theoretic measure to determine the semantic similarity of attributes. This is primarily accomplished via GT matching. Second, we propose a method of disambiguating among multiple possible GTs associated with an instance using an associated latlong value. Third, we provide a way to perform attribute matching using non-geographic types, in case insufficient GT information is available. Finally, we introduce a method of attribute weighting that accounts for the uniqueness of the paired attributes relative to all others. This is done in order to improve the accuracy of the semantic similarity value between tables.

The rest of this paper is organized as follows. In Section 2, we discuss an overview of related work. Section 3 states definitions, the problem to be solved and our proposed solution. Section 4 presents in detail the GSim algorithm, detailing both the geographic lookup component as well as the more generic GD component. In Section 5 we present our results generated with GSim and compared them against those generated using N-grams. Finally, in Section 6, we outline our future work.

2. Related work

In this section, we will first present other work related to schema matching. Second, we present work in the GIS domain making use of a gazetteer. Third, we present work making use of reverse geocoding. Fourth, we will present work done regarding attribute weighting. Finally, we contrast our work with another approach used to solve the schema matching problem.

A number of schema matching publications [5–8] tailored to the database community influenced our work. The survey of approaches to automated schema matching by Ralun and Bernstein [5] includes a taxonomy which uses several criteria to categorize matching approaches such as schema and instance based methods, element-level and structure-level methods, and linguistic and constraint-based methods. While this work surveys a wide swath of approaches covered in schema matching literature, it does not present any approaches specifically tailored to the geospatial domain. Matching in the geospatial domain presents unique challenges, due to the properties inherent in geospatial data such as geometry, georeferenced coordinates, variations in formatting and coordinate systems, and much more. The nature of geospatial data is complex enough such that most applications, including our current implementation of GSim, have only addressed a subset of its unique properties. Dai et al. [6] discuss instance-based schema matching using distributions of N-grams among compared attributes. The differences between our work and [6] is discussed later in this section. Bohannon et al. [7] investigate contextual schema matching, in which selection conditions and a framework of matching techniques are used to create higher quality mappings between attributes of compared schemas. Among their methods for deriving selection conditions is the training of a classifier on the attribute values from an attribute involved in a match. This would imply that the values of an attribute can be expressed by a pattern, such as a regular expression. However, this would not work in the geospatial domain because a number of attributes, such as 'City' and 'County' cannot have their attribute values described by a generalized expression. Thus, training classifiers on these attributes would not make a contribution towards a match with other similar attributes. Warren and Tompa [8] propose an iterative algorithm that deduces the correct sequence of concatenations of column substrings in order to translate from one database to another without the use of a set of training instances. While this work addresses some of the same challenges that we do, our work is distinguished by the inclusion of attribute weighting to account for differences in the importance of certain attribute comparisons over others, and also by our use of latlong driven disambiguation as applied to geographic instances identified by gazetteers.

Within the AI community, a number of works in the schema matching area applied machine learning and statistical methods to learn attribute properties from data and examples. Li and Clifton [9] describe a tool known as SEMINT, which uses neural networks to determine match candidates by learning the metadata and data values patterns of attributes. From this, other attributes with similar metadata and data value patterns are sought in order to create 1:1 attribute mappings. However, their methods would not work in many cases for geospatial schema matching because several attributes in this domain share similar metadata and/or data value patterns, yet are completely different. For instance, the attributes "County" and "City" both could be characterized with the same SQL datatype (i.e.: CHAR (40)), and they may even share some identical data values. However, learning these characteristics would never amount to anything, because of the arbitrary nature of the names of counties and cities. Berlin and Motro [10] describe a tool known as Autoplex which uses supervised machine learning techniques such as Naïve Bayes classification for automating the discovery of new content for virtual database systems. While the versatility of the Naïve Bayes approach is widely known, its binary classification methodology is a problem for geospatial schema matching. In the Naïve Bayes approach, an instance either belongs to an attribute or it does not. In geospatial schema matching, a finer grained approach is needed since instances often display degrees of membership to various attributes. GSim takes into the possibility of instances having multiple GTs. It attempts to reduce the number of GTs for an instance to one, but if this is not possible, then it takes into account all possible GTs for that instance into the final EBD calculation. Embley et al. [11] explore both 1:1 and m:n schema mapping techniques by applying knowledge obtained from domain ontology snippets and data frames. However, if this method was applied to geospatial schema matching, then it would fail for the same reasons as [9] would fail. The problem is the assumption that the membership of an instance value to an attribute is based on a data pattern or a regular expression. In the geospatial domain, this is often not the case. Also, the use of domain ontology snippets for schema matching is highly subjective. The structure of the ontology is often dependent on the specific vision of its designers, which might differ from

the vision of those individuals who designed the schemas being mapped. Furthermore, the choice of the ontological snippet to use is inevitably fraught with bias in one form or another.

The most closely related work in the GIS domain discusses instance matching over geodatabases, ontologies, thesauri and other geographic data sources. Cruz et al. [4] describe AgreementMaker, a visual tool that provides a user with the ability to perform mappings between ontologies using a multi-faceted strategy involving automated techniques as well as manual specifications. Albertoni et al. [25] devised an instance based similarity measure that matches instances of ontological concepts based on two contextual layers: an ontology context, which is based on a comparison of the concepts' depth in a structured hierarchy as well as the number of attributes and relations they share, and an application context, which uses instance paths and set of predefined comparison operations between concepts to perform a match based on the specific needs of the user. Janowicz and Wilkes [24] describe SIM-DL_A, a DL based instance similarity measure that matches instances from a source concept, specified as a user query, with the instances from all target concepts that can satisfy the query. This is determined with the help of a context concept that is the superclass of all possible target concepts, along with a modified version of the tableau algorithm that is normally used in satisfiability checking. Unlike GSim, each of the above approaches requires that the instances of concepts or attributes belong to a sophisticated ontology replete with numerous relation types between the concepts and/or attributes. In a use case involving matching between two unstructured geospatial data sources, like flat sets of concepts, thesauri or an unstructured folksonomy (which might consist of satellite imagery of a geographic location, along with its keyword annotations) consisting of concepts annotated by a community of users, the methods above which depend on a defined structure of concepts will not be applicable. Karalopoulos et al. [35] outline a method for using POS tagging and subsequent parsing to convert a geographic concept description into a conceptual graph, which could then be used for various purposes including semantic similarity. Though this work does not explore semantic similarity, it also relies on a strict relation structure between the tagged words in the concept definition, as well as a strict grammatical structure of the definition itself. If the concept does not contain any annotations, then this method will not work. Furthermore, the successful creation of a conceptual graph depends on the definition containing an ordered grammatical triple consisting of a genus, differentia and an illustrative example. Obviously, many ontologies exist where concepts are annotated differently. Other work related to instance matching in the geospatial domain is as follows. Ahlqvist and Shortridge [34] introduce semantic variograms, which can be used to determine the semantic similarity of multi-class land areas separated by a series of spatial lags. Paes Leme et al. [1] perform schema matching over GIS databases containing data represented by a dialect of OWL. Brauner et al. [2] perform instance matching over the exported schemas of geographical database Web services and apply their technique over the GeoNames and ADL gazetteers. Brauner et al. [3] leverage instance mapping between distinct terms in feature type thesauri used to classify data in gazetteers, for the facilitation of successful thesaurus migration from one gazetteer to another. The method described in [34] works well for land cover classification, but would not work as well for geospatial schema matching, since its matching criteria only works over ordinal data. The methods outlined in [1–3] use co-occurrence statistics of pairs of keywords or types in order to derive attribute mappings. In many cases, this is an effective method; however, in order for it to work, it relies on a syntactic match between either instance names or the instance types. Often times, the names and properties of geospatial entities contain slight variations which require methods beyond syntactic matching in order to determine a match with another entity. GSim relies on semantic matching by leveraging the GTs and latlong values of compared instances for geographic type matching. If geographic matching is not possible, instances can also be compared using a semantic NGT match via GD and K-medoid clustering.

Much work in the GIS community making use of a gazetteer for information lookup influenced also our work. Zhou et al. [12] apply a deterministic, density-based clustering algorithm to semi-automatically discover gazetteers from users' travel data, as well as disambiguate between uninteresting and interesting results from the gazetteer using temporal techniques. Newsam and Yang [13] integrate a gazetteer with high-resolution remote sensed imagery to automate geographic data management more completely, and they also demonstrate how gazetteers can be effectively used as a source of semi-supervised training data for geospatial object modeling. Pouliquen et al. [14] use a gazetteer lookup, as opposed to linguistic analysis, to search through natural language text and produce geographic maps and animations that represent the area referred to in the text. Despite the novelty of these works, they fail to address the challenges in geospatial matching that GSim is able to meet. The work in [12] and [14] depend on performing exact matches between the user's data and data found in a gazetteer. A sophisticated semantic matching algorithm must discover similarity between heterogeneous sources, whether or not an exact word match exists between the compared data. Thus, the methods outlined in [12] and [14] would be ineffective towards the application of aligning two geospatial ontologies that model the same geographic area, but using different languages. Meanwhile, the work in [13] focuses on using remote sensed imagery as training data in an effort to model geographic objects in a semi-supervised way; since it works with images as opposed to text, it solves a different problem than GSim. However, even if it was applied for semantic matching over compared data sources that also contained representative image data, errors resulting from the variability of images, such as lighting, inclement weather, scale, etc. would cause a fairly high degree of error in identifying objects (or geographic features, in this case) from the images. Using GSim's type matching method, as long as a GT is associated with a geographic feature in a gazetteer, there will be no ambiguity about the type of a feature.

Some work in the GIS community involving reverse geocoding is related to our research. Zhou and Frankowski [15] evaluate the accuracy of personal place discovery using reverse geocoding and clustering through a set of evaluation metrics and an interactive evaluation framework. Joshi and Luo [16] employ reverse geocoding using location coordinates from image data to obtain nearby points of interest connecting an image with its geographic location. Wilde and Kofahl [17] describe the use of reverse geocoding in retrieving location types as an essential component for a geo-enabled Web browser. Our work shares some tangential similarities with the above work (i.e. the use of clustering), but differs fundamentally by using latlong information from gazetteers and attribute weighting to derive a more intelligent means of performing schema matching across data sources in the GIS domain.

Attribute weighting research has mostly focused on applications of machine learning, such as estimation by analogy and query ranking. To the best of our knowledge, it has never been applied to schema matching in the geospatial domain. Li and Ruhe [45] performed a comparative study of five separate attribute weighting heuristics as a means of measuring software effort estimation. The heuristics are based on rough set analysis, which uses the notion of equivalence classes to construct approximations of a given set. This method, as stated in [45], would not apply very well for our purposes to schema matching for two reasons. First, rough set analysis is designed to work with ordinal data, such as a list of categories (i.e.: {Low, Medium, High}). Our data sets consist of non-ordinal data, such as sets of county names or latlong values. Second, the methods described in [45] depend on historical data sets to deter-

mine an analogous weighting scheme suitable to the current data set. However, there is nothing to suggest that these methods can handle new data values that have never appeared in any historical data set. In geospatial schema matching, it is common to encounter entirely new data values with the task of determining their similarity to another data set. Su et al. [46] use attribute weighting to rank a list of results generated from a user query over an e-commerce database without the need for direct user feedback. However, in their approach, while it is true that they do not require a user to provide direct feedback on the attributes most important to him/her, [46] determines the attribute weight largely based on implicit hints provided by the user query. For instance, in a web database of used cars consisting of attributes "Year", "Price", "Mileage" and others, if a user specifies a query, "Year > 2009", then [46] surmises that the user prefers a car with low "Mileage", thus making this attribute more important than others. However, in our experiments, no user feedback whatsoever is available. Also, [46] assumes that the "Price" attribute is always present in a database. For our experiments, we can never assume that a particular attribute is always present.

We seek to compare our schema matching research against the work of Dai et al. [6]. They present a solution to the schema matching problem that makes use of *N*-grams. We argue that GSim features an innovative instance matching algorithm that possesses a number of advantages over the *N*-gram approach, particularly in the GIS domain. An *N*-gram is a substring of length *N* consisting of contiguous characters. So for example, if $N = 2$, then the word 'GSim' has *N*-grams 'GS', 'Si' and 'im'. First, GSim determines GTs for instances via a gazetteer as part of the process of determining an overall semantic similarity value between attribute pairs containing those instances. Because GSim uses domain-specific information to determine the GT for a given instance, it is better equipped than the *N*-gram approach to solve the information integration problem among geodatabases. *N*-Grams cannot take advantage of domain knowledge, since they are only parts of words. Second, GSim can retrieve missing instance values in geodatabases by using associated latlong values to perform reverse geocoding. This ability is not available using solely the *N*-gram approach, because they cannot have latlong values associated with them. Third, in case the geographic lookup component is unsuccessful, GSim leverages clustering of types for use on distinct keywords found between compared attributes via GD. This approach is better able to capture the semantics of comparisons between attributes because words contain more implicit semantic information than *N*-grams. Using words, we can reference external data sources that allow for distance metrics to determine word relatedness. Finally, our new instance matching algorithm does not require a syntactic match between its instances, whereas *N*-grams does. For example, for two *N*-gram instances to match, they have to represent the same string (i.e.: "ab"). On the other hand, GT matching in GSim would be able to match instances such as Spring Valley Road and Canyon Creek Drive, based on their common geographic type.

The work presented in this paper is an extension of our previous work [26,27] in the following ways. First, in addition to the identification and leveraging of GTs for the purposes of improving semantic matching outlined in [27], we now further improve our matching results through the comparison of latlong values in the dataset and in a gazetteer. This way, we can guarantee an exact match between a particular instance within a compared attribute and its correct GT, as long as the gazetteer recognizes the instance's feature type. Second, we developed and tested an attribute weighting scheme to allow semantic matching between tables to occur based on the importance of the attributes in the compared tables relative to the subject of the table itself. For instance, if a set of attributes from a table called Road are taking part in a match with attributes from another table known as Street, then an attribute such as "RoadName" would contribute far more to the semantic similarity (or dissimilarity) to the Street table than an attribute like "rID", which might have nothing to do with roads at all (this would be the case if the attribute represented an ID used internally by a geodatabase). In this case, an attribute pairing of "RoadName" from the table 'Road' with "StreetName' from the table 'Street' would effectively be more important for determining the true semantic similarity value between 'Road' and 'Street' than an attribute pairing of "rID" from 'Road' and "sID" from Street. Third, in addition to the *N*-gram method, our work compares the performance of our algorithm, GSim, to two additional widely accepted methods used for determining semantic similarity: Singular Value Decomposition (SVD) and Nonnegative Matrix Factorization (NMF). We show in Section 5 that our algorithm outperforms *N*-grams, SVD and NMF over three different multijurisdictional datasets in the GIS domain.

roadName	City
Johnson Rd.	Plano
School Dr.	Richardson
Zeppelin St.	Lakehurst
Alma Dr.	Richardson
Preston Rd.	Addison
Dallas Pkwy	Dallas

Fig. 1. Sample table containing two attributes and six instances per attribute.

3. Problem statement and proposal

3.1. Definitions

First, we will provide definitions that will assist in defining the problem and describing GSim.

Definition 1 *(attribute).* An attribute of a table T, denoted as att(T), is defined as a property of T that further describes it.

Definition 2 *(instance).* An instance x of an attribute att(T) is defined as a data value associated with att(T).

Definition 3 *(Keyword).* A keyword k of an instance x associated with attribute att(T) is defined as a meaningful word (not a stopword) representing a portion of the instance.

In Fig. 1 above, the two attributes for the given table are roadName and City, two instances from the roadName attribute are "Johnson Rd." and "School Dr.", and the two keywords associated with the instance "School Dr." are "School" and "Dr.".

Definition 4 *(type).* A type t associated with attribute att(T) is defined as a class of related entities grouped together.

We define two kinds of types:

Definition 4a *(Geographic type (GT)).* A geographic type GT associated with attribute att(T) is defined as a class of instances of att(T) that represent the same geographic feature.

Definition 4b *(non-geographic type (NGT)).* A non-geographic type NGT associated with attribute att(T) is defined as a group of keywords from instances of att(T) that are semantically related to each other. An NGT is only derived for an instance when it cannot be associated with any geographic type from a gazetteer.

Definition 5 *(geographic type (GT) vector).* A geographic type vector $\mathcal{T}_x = \{GT_1, GT_2, \ldots, GT_m\}$ associated with an instance x of attribute att(T) is defined as a set of GTs.

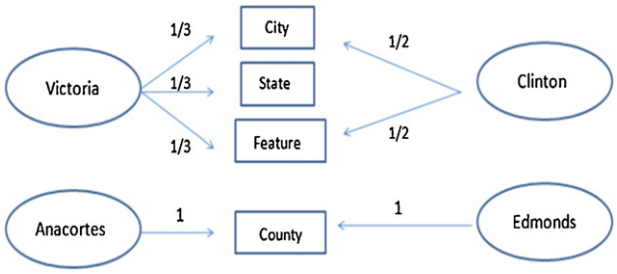

Fig. 2. Sample instances of attribute *att* and their respective sets of GTs.

Definition 6 *(geographic weight (GW) vector).* A geographic weight vector $\mathcal{W}_x = \{w_1, w_2, \ldots, w_m\}$ associated with a GT vector $\mathcal{T}_x = \{GT_1, GT_2, \ldots, GT_m\}$ for an instance x of attribute att(T) is defined as a list of real numbers between 0 and 1 representing the influence of a GT on the instance.

Note that for all i, $GT_i \in T_x$ of any instance x has weight $w_i \in W_x$.

Definition 7 *(geographic type set of attribute ($\mathcal{T}_{att}$)).* A geographic type set of attribute att(T), denoted $\mathcal{T}_{att}$, is the set of GTs derived from the union of the types from all GT vectors for the instances of att(T).

Definition 8 *(non-geographic type set of attribute ($\mathcal{NT}_{att}$)).* A non-geographic type set of attribute att(T), denoted $\mathcal{NT}_{att}$, is the set of NGTs associated with keywords from instances of att(T).

Definition 9 *(geographic type weight list ($\mathcal{W}_{att}$)).* A geographic type weight list $\mathcal{W}_{att}$ associated with attribute att(T) is the total type weights for each type in $\mathcal{T}_{att}$.

In Fig. 2 above, the instances are "Victoria", "Anacortes", "Clinton" and "Edmonds". The GT 'City' represents the instances "Victoria" and "Clinton", The GT vector for "Victoria" = {City, State, Feature} and for "Anacortes", it is = {County}. The GW vector for "Victoria" is {1/3,1/3,1/3}, and for "Anacortes" it is {1}. If these four instances make up the entirety of attribute att, then $\mathcal{T}_{att}$ is {City, State, Feature, County}, and the GT weight list $\mathcal{W}_{att}$ is {1/3 + 1/2, 1/3, 1/3 + 1/2, 1 + 1}, or in simplified form, {5/6, 1/3, 5/6, 2}. The formalized computation of $\mathcal{W}_{att}$ is shown in Section 4.1.2.

As an example of illustrating the weighting of GTs, taking all instances from Fig. 2 into account, the total weighting for the types listed are as follows: "City" = (1/3 + 0 + 1/2 + 0) = 5/6, "State" = (1/3 + 0 + 0 + 0) = 1/3, "Feature" = (1/3 + 0 + 1/2 + 0) = 5/6, and "County" = (0 + 1 + 0 + 1) = 2 (Recall that for "County", 1 is for Anacortes and 1 is for Edmonds).

In Fig. 3 below, given an instance with a value of "Pacific Coast Highway" from attribute *att*, there are two NGTs named generic type 1 and generic type 2. The NGT set $\mathcal{NT}_{att}$ of attribute *att* that contains this instance would have {generic type 1, generic type 2}, as well as other types from other instances of this attribute.

3.2. Problem outline

Given two data sources, S_1 and S_2, each of which is composed of a set of tables where $\{T_{11}, T_{12}, T_{13} \ldots T_{1M}\} \in S_1$ and $\{T_{21}, T_{22}, T_{23} \ldots T_{2N}\} \in S_2$, the goal is to determine the similarity between the tables of S_1 and the tables of S_2. This is done by first creating mappings between attributes of compared tables (say T_{11} and T_{21}) such that for every mapping, one attribute of T_{11} is compared against one attribute of T_{21}, until each attribute of T_{11} maps to a single attribute of T_{21}. In the case where T_{11} and T_{21} differ in the number of attributes, we require that the table with the smaller number of attributes has every attribute map to an attribute in the compared table. This means that the table with the larger number of attributes may have one or more attributes not involved in any comparisons. The final similarity value between the tables is taken to be the average similarity values of their attribute mappings. S_1 and S_2 may vary in regards to the number of constituent tables, and the number of attributes and instances within a given table may also vary.

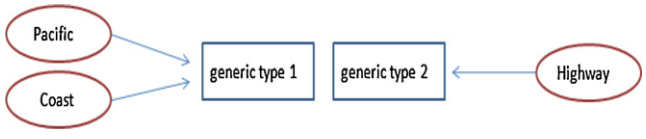

Fig. 3. Sample keywords from an instance of attribute att and their respective NGTs.

3.3. Proposed solution

We present GSim, an instance matching algorithm that generates semantic similarity values between compared attributes in different tables of a geodatabase. The derivation of attribute mappings between a pair of compared tables is created in two separate stages. First, a preprocessing phase based on data type matching and attribute name matching determines the pairs of attributes that are most likely to be similar. These attribute pairs represent the attribute mappings whose collective similarity values will determine the similarity value between their tables. Second, instance-level matching is applied to each attribute pair in order to determine their similarity. Our instance-level matching is based on two separate approaches. The primary approach assigns GTs to instances involved in compared attributes within two tables of the geodatabase with the help of a gazetteer. This results in a pair of GT sets, one for each attribute. The semantic similarity between the compared attributes is then computed using EBD over their respective GT sets. However, since gazetteers will not contain information about every instance, it is possible that attribute matching via geographic-type extraction will be ineffective. In this case, we apply a generic matching method, applicable over any knowledge domain, that is based on the extraction and clustering of instance keywords into NGTs, based on GD. Further details describing GSim in its entirety are described in Section 4.1. It is our intention to clearly show that the use of GSim is better able to capture the true semantics that exist between compared attributes contained within GIS tables as opposed to using *N*-grams.

It is assumed that we perform 1:1 comparisons between attributes from distinct tables and data sources. After calculating a similarity value between compared attributes using EBD, we will repeat the process for all compared attributes between the tables. This results in a set of 1:1 mappings, or alignments, which display the attribute correspondences between the tables. Next, a final similarity value between the tables is calculated by taking the average of the EBD values between all attribute pairs. EBD is based on a comparison of the conditional entropy of the attributes, given a particular type, with the entropy of the attributes over all types:

$$\mathrm{EBD} = \frac{\mathrm{H}(A|T)}{\mathrm{H}(A)} \tag{1}$$

In this equation, A is the attribute, coming from either one table or another (since all table comparisons are 1-1), and T stands for the type of the instances of the attribute. Attributes can also be referred to as 'columns', so in subsequent sentences, H(A) will sometimes be written as H(C) for entropy (where H(A) and H(C) mean the same thing), and H(A|T) will sometimes be written as H(C|T) for conditional entropy. (where H(A|T) and H(C|T) mean the same thing. There may be multiple types per attribute; for geographic matching, T would indicate a GT, such as 'City' or 'County', while for non-geographic matching, T would indicate a given generic type. Intuitively, an attribute A contains a high entropy value if it is impure; that is, the ratios of types (either GT or NGT) making up

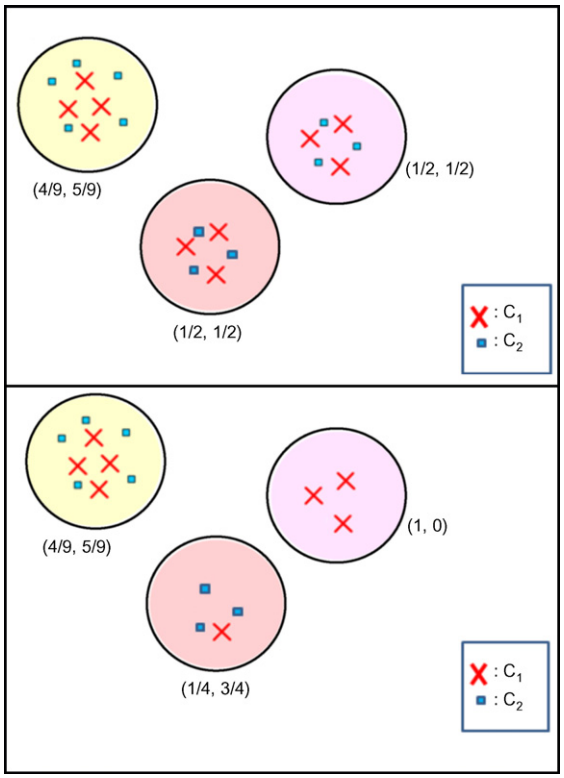

Fig. 4. In (a) on the top, the distribution of types across attributes when EBD is high. H(C) is similar to H(C|T). In (b) on the bottom, distribution of types across attributes when EBD is low. H(C) and H(C|T) have dissimilar values.

A are similar to one another. On the other hand, low entropy in *A* exists when one type exists at a much higher ratio than any other type. As applied to our research, entropy always measures the ratio of the number of instances of an attribute *A* and a compared attribute *A*' regardless of what the GTs or NGTs may be. Conditional entropy, on the other hand, measures the ratio of the number of instances of *A* and *A*', given a particular type (GT or NGT). Fig. 4a and b provide examples to help visualize the concept. In both examples, crosses indicate instances originating from *A*, while squares indicate instances originating from *A*'. Each distinct type (GT or NGT) is represented as a cluster (larger colored circles), each of which contains instances from *A* and *A*' associated with that type. In Fig. 4a, across all types, the total number of crosses = 10 and the total number of squares = 11, which implies that entropy is very high. The conditional entropy is also quite high, since the ratios of crosses to squares within two of the clusters are equal and nearly equal within the other. Thus, the ratio of conditional entropy to entropy will be very close to 1.0, since the ratio of crosses to squares is nearly the same across types and within each type. Fig. 4b portrays a different situation: while the entropy is 1.0 (since the number of crosses is equal to the number of squares overall), the ratio of crosses to squares within each individual cluster varies considerably. One cluster features all crosses and no squares, while another cluster features a 3:1 ratio of squares to crosses. When computing the EBD value for this example, we will derive a value that is lower than the EBD for the first example because H(C|T) will be a much lower value. Intuitively, this makes sense because the ratios of instances associated with a particular type between *A* and *A*' are dissimilar.

4. Overview of GSIM

This section describes GSim, our instance similarity algorithm, and its two components. The first, detailed in Section 4.1, involves the use of a geographic lookup to determine whether the instances of compared attributes between two tables share similar GTs. If so, then an exact match for those instances is made using only GTs. If not, then the second component of GSim, which exclusively relies on a non-geographic measure of semantic similarity between instances of compared attributes, is applied. The rest of the section discusses attribute weighting, a more intelligent method of performing semantic schema matching that relies on the fact that certain attributes contribute more to the meaning of a particular table than others. Section 4.2 describes our justification for using geographic types as our means of applying semantic matching. For our purposes, we use GD as our non-geographic similarity measure, but despite the generalized utility of GD, there are situations when this approach produces inaccurate results. Section 4.2 depicts one such situation. Section 4.3 outlines a proposed solution to problem described in Section 4.2.

We justify our usage of GSim as a semantic similarity metric by comparing it against an alternative semantic similarity metric derived from WordNet, a lexical dictionary for the English language. We decided against using it because of its shallow coverage of concepts relative to that which is covered by the combination of GSim for geographic matching and GD for non-geographic matching. For example, in comparing two street name attributes of the Road-Road table comparison for the GIS transportation dataset (see Section 5 for more information on the table comparisons), GSim + GD was able to compute 4776 out of 4992 (95.7%) distinct pairwise distance values for the extracted keywords between the pair of attributes. For the same attribute pair, WordNet was only able to calculate 2,068 distinct pairwise values, only 43.3% of the number of values calculated by GSim + GD. Additionally, for a comparison of a street name attribute and a port name attribute between the Road table of S_1 and Ferry table of S_2 for the GIS transportation dataset, GSim + GD found 132 out of 161 (81.9%) distinct pairwise values between extracted keywords while WordNet only found 22 out of 161 (16.7%).

4.1. GSim algorithm

4.1.1. Overview

For Algorithm 1 below, the input consists of the attributes $A_1 \in T$ in S_1 and $A_2 \in T'$ in S_2 and gazetteer G. Line 1 initializes T_{gaz}, the set of all GTs recognized by gazetteer G, T_{A1} and T_{A2}, the GT vector lists for A_1 and A_2, respectively, NT_{A1} and NT_{A2}, the NGT vector lists for A_1 and A_2 respectively, and W_{A1} and W_{A2}, the GW vector lists for A_1 and A_2, respectively. Lines 2 and 3 extract the distinct instances from A_1 and A_2. Line 4 determines whether semantic similarity can be performed strictly by relying on GTs, or if GD similarity will be necessary. GT similarity is only possible if a gazetteer is available, and if it contains sufficient GT information about enough of the instances. For our purposes, we established a threshold, t_{min}, which represents the minimum number of instances that contain GT information in G. In our experiements, t_{min} was set to a value of .5. Therefore, if GT information can be found for a number of instances greater than or equal to t_{min} (at least 50% of the instances in the compared columns), then EBD is calculated using only GTs. This process is initiated in lines 5–8, where line 5 retrieves all available GTs, T_{gaz}, recognized by gazetteer G, lines 6–7 derives a GT vector list T_{A1} and its associated GW vector list (W_{A1} in line 6 and W_{A2} in line 7), consisting of GT vectors for each instance of A_1 and A_2. If however, in line 4 if geotypingIsPossible() returns false, then we need to rely on a more generic measure like GD to compute semantic similarity between the compared instances. This is done in line 9. The GD component of GSim will be described in Section 4.1.4. Line 11 calculates the final EBD value between A_1 and A_2 given the combined type vector lists and weight vector lists of A_1 and A_2, and line 12 returns that EBD value.

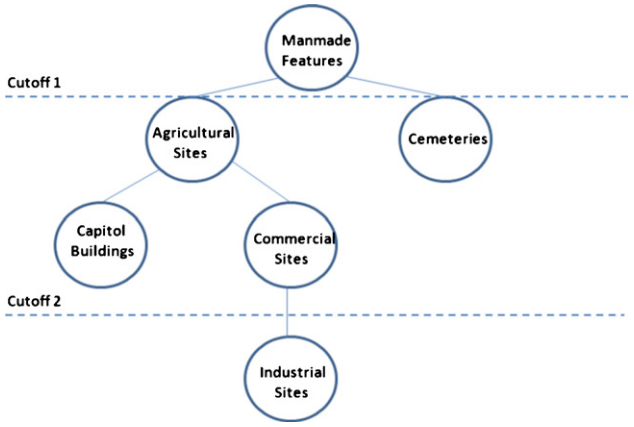

Fig. 5. Segment of the ADL gazetteer's feature type hierarchy for manmade features. The two dashed lines are cutoff points that determine how specific our GTs should be, which effects the final similarity score between compared attributes.

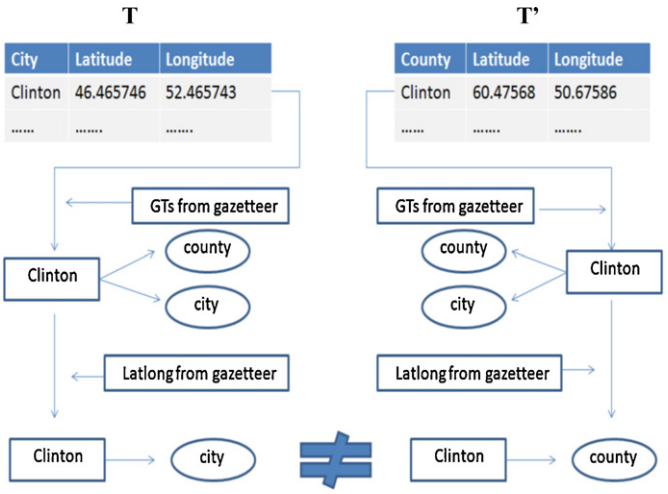

Fig. 6. GSim's use of latlong values associated with an instance allows for further disambiguation among its GTs.

4.1.2. Assigning GTs to instances

We leverage a gazetteer as a way to help determine the GT of an instance. The gazetteer used for our purposes is GeoNames [18], containing information on over 8 million geographic names. The gazetteer classifies locations into different categories, or types. Some examples of GTs include city, county, state and a general feature with several sub-classifications, such as lake, port, school, etc. Instances with more commonplace names are likely to be listed under multiple types in the gazetteer. As a result, a single instance may be associated with a list of GTs = $\{GT_1, GT_2 \ldots GT_n\}$, where n is the number of GTs recognized by the gazetteer. However, as will be described in Algorithm 2, because an instance may have multiple GTs, the weight of that instance for each of those types is divided proportionately. Finally, an EBD calculation over the different GTs is performed.

Formally, let $\mathcal{T}_{gaz} = \{GT_1, GT_2, \ldots, GT_m\}$ be a set of GTs recognized by gazetteer G, with GT_i, $0 \le i \le m$, representing one of these types. For example, GT_i may be a county, city, state, etc. An arbitrary instance x associated with attribute att(T) will be associated with a GT vector $\mathcal{T}_x = \{GT'_1, GT'_2, \ldots, GT'_n\}$, $n \le m$ and $n > 0$. Let $\mathcal{W} = \{w_1, w_2, \ldots, w_n\}$ be a GW vector, where each w_j is associated with each GT'_j in $\mathcal{T}_x$ for instance x, where $|\mathcal{W}| > 0$ and all w_k in $\mathcal{W}$ for x have a value of $1/|\mathcal{W}|$. For example, if x was associated with three GTs, then the weight w_j of each type for x would be 1/3.

Algorithm 1. GSim (A_1, A_2, G)

Input:
-attribute $A_1 \in T$ in S_1, attribute $A_2 \in T'$ in S_2, gazetteer G
Output: Semantic similarity value between A_1 and A_2 expressed as EBD
1: $\mathcal{T}_{gaz} = \Phi, \mathcal{T}_{A1} = \mathcal{T}_{A2} = \Phi, \mathcal{NT}_{A1} = \mathcal{NT}_{A2} = \Phi, \mathcal{W}_{A1} = \mathcal{W}_{A2} = \Phi$
2: IL_1 = ExtractInstances (A_1)
3: IL_2 = ExtractInstances (A_2)
4: if (geotypingIsPossible (G,IL_1,IL_2)){
5 : $\mathcal{T}_{gaz}$ = getGezetteerTypes(G)
6 : ($\mathcal{T}_{A1}, \mathcal{W}_{A1}$) = lookupGeoTypes($\mathcal{T}_{gaz}$, IL_1)
7 : ($\mathcal{T}_{A2}, \mathcal{W}_{A2}$) = lookupGeoTypes($\mathcal{T}_{gaz}$, IL_2)
8: } else {
9: ($\mathcal{NT}_{A1}, \mathcal{NT}_{A2}$) = NGDSim($IL_1$, IL_2)
10: }//end if
11: EBD[A_1][A_2]=computeEBD ($\mathcal{T}_{A1}$, $\mathcal{T}_{A2}$, $\mathcal{W}_{A1}$, $\mathcal{W}_{A2}$, $\mathcal{NT}_{A1}$, $\mathcal{NT}_{A2}$)
12: return EBD[A_1][A_2]

Some gazetteers contain a hierarchical feature type thesaurus. One example is the ADL gazetteer [32]; Fig. 5 shows the segment of the feature type hierarchy that represents manmade features. As of now, GSim assigns the most general GT to a given instance. For a gazetteer with a flat feature type system, this is not a problem, as there will be no doubt about the GTs of instances when computing an EBD score between compared attributes. However, for a hierarchical type system, the final EBD score depends upon the specificity of the GT assignments of the instances. For example, in Fig. 5, if instances are assigned GTs that are no more specific than "Manmade Features", as illustrated by Cutoff 1, then any instance that is an "Agricultural Site", "Commercial Site", etc. will have a GT of "Manmade Feature". As a result, the calculated EBD between compared attributes with these instances is likely to be higher. In reality, though, the EBD value is more likely to be overestimated. On the other hand, if we assigned GTs to instances that may be as specific as "Commercial Sites" or "Capitol Buildings", as indicated by Cutoff 2, then it is very possible that many of the instances that were labeled as "Manmade Features" using Cutoff 1 would now be labeled as a more specific GT, such as "Capitol Building" or a "Cemetery". This would result in a final EBD value between compared attributes that is lower than if Cutoff 1 was used. However, in reality, the EBD is likely to be underestimated compared to an EBD score derived from a situation where the user was interested in assigning GTs no more specific than "Agricultural Site". Although this problem is not the focus of our work, we are continuing to study it by carrying out additional experiments. The goal is to determine the cutoff that yields the highest EBD value while sacrificing an acceptable amount of GT specificity.

4.1.3. Using latlong values

GSim also possesses the ability to leverage latlong values for the purposes of improving the accuracy of the semantic similarity measurement between two attributes, and ultimately, between two tables. This is accomplished by comparing latlong values associated with the instances of compared attributes, and comparing them against latlong values for those same instances in the gazetteer. This technique is intended for those instances associated with multiple GTs; using latlong values, it will be possible to identify the correct GT out of many within the GT set for an instance with a common name such as "Clinton". At the same time, latlong values can also help disambiguate among the GTs of instances where the types do not match.

The process of using latlong values for further disambiguation is illustrated in Fig. 6. Here, two attributes are being compared against one another for the purposes of deriving a semantic similarity value. In particular, an instance of the City attribute from table T named "Clinton" is compared against an instance from table T' from the 'County" attribute, also named "Clinton". Since "Clinton" is a common name, a query to a gazetteer by GSim for both instances is very likely to result in the return of >1 GTs.

Without the use of latlong information, we would not be able to definitively pare down the number of GTs for each instance, thus affecting the accuracy of the semantic similarity calculation. In Fig. 6, the instance "Clinton" in table T is associated with both the county and city GTs, and no further disambiguation is possible. The same is true with the instance "Clinton" from table T'. However, the use of latlong information, both from the instance data and the gazetteer itself, allows a comparison of the latlong values to be made so that the correct GT for each instance is chosen in an automated fashion. The end result of this is a more accurate semantic similarity calculation between attributes, and ultimately, between tables. In Fig. 6, using latlong information, it can now be determined unequivocally that the Clinton instances represent different GTs, and thus, should not be matched.

One crucial detail worth mentioning regarding the use of latlong values for GT identification is the natural variation in latlong values displayed by gazetteers. This may come about either because of differing numbers of significant digits in the coordinate values, cartographic projection, or due to differences in the scale, or level of detail, of geographic features. For instance, if our data contains an instance known as "Example Ave.", with a latitude value of "43.24323", and our gazetteer contains this instance at the same level of detail, but with a latitude value of "43.2432332", then in all likelihood, this should be considered a match. To solve this problem, we use a distance tolerance measure that discounts significant digits to the right of the decimal point in the latlong value that are not deemed crucial for the match. The number of significant digits that are discounted depends on the features being matched, and their level of accuracy. Every time an instance in our data set is matched to an instance in the gazetteer, we first determine the geographic type of the instance. Afterwards, we classify the instance match according to 9 possible levels of accuracy, with the lowest level of detail (level 1) being country, and the highest level of detail (level 9) being "premise", which includes building names, property names, shopping centers, etc. We modeled our accuracy hierarchy after version 2 of Google's Reverse Geocoding API [33]. Using the level of detail of the feature data, in addition to the feature type of the instance, we can determine the number of significant digits to discount.

Algorithm 2 below outlines the final geographic type lookup algorithm, including both the naïve geographic type lookup algorithm and the more sophisticated version which matches exactly one type to each instance. It describes the process by which GTs and weights are assigned to instances. The input to Algorithm 2 is the list of available GTs that are recognized by gazetteer G, along with IL, the list of instances associated with a given attribute and the gazetteer G itself, while the output is an ordered pair consisting of the GT vector list and GW vector list for the given attribute. Line 2 begins a loop that considers all instances in IL. Line 3 retrieves the set of GTs from $\mathcal{T}_{gaz}$ that instance x is associated with. Line 4 determines if instance x contains latlong information. If so, then it is possible to prune the number of possible geographic types for instance x to exactly one while assigning a weight of this type to be = 1. This occurs on lines 5 and 6. If x does not contain any latlong information, then Lines 8-10 derive all possible types $\mathcal{T}_x$ for x and assign the weight of each type associated with the current instance. Lines 13–14 aggregate the GT and weight vectors computed for instance x to $\mathcal{T}_{att}$ and $\mathcal{W}_{att}$, respectively. Finally, these vectors are returned as an ordered pair to GSim, which facilitates the EBD calculation between two compared attributes.

4.1.4. Non-geographic matching

If GT matching between compared attributes is not possible, then a non-geographic semantic similarity measure is applied by GSim. The distance metric used for NGT matching is known as the normalized Google distance. The EBD is then calculated by extracting the keywords making up compared instances and assigning them generalized semantic types. These types are represented as clusters of keywords, whose semantic distance from each other is given by GD.

Section 4.1.4.1 below first gives the definition of GD. Section 4.1.4.2 gives an overview of NGT matching. Section 4.1.4.3 provides further details on the K-medoid clustering process, which is instrumental to the success of NGT matching.

4.1.4.1. Google distance. GD is formally defined as follows:

$$\mathrm{GD}(x, y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x, y)}{\log M - \min\{\log f(x), \log f(y)\}} \quad (2)$$

In this formula, $f(x)$ is the number of Google hits for search term x, $f(y)$ is the number of Google hits for search term y, $f(x,y)$ is the number of Google hits for the tuple of search terms xy, and M is the number of web pages indexed by Google. GD(x,y) is a measure for the symmetric conditional probability of co-occurrence of x and y. In other words, given that term x appears on a web page, GD(x,y) will yield a value indicating the probability that term y also appears on that same web page. Conversely, given that term y appears on a web page, GD(x,y) will yield a value indicating the probability that term x also appears on that page.

4.1.4.2. Overview of NGT matching. The algorithm for calculating the EBD between two compared attributes of tables in different data sources using NGT matching is as follows. The input is two compared attributes, with each one originating from a separate table, while the output is an EBD value indicating the semantic similarity between the input attributes. First, the respective keyword lists for each input attribute are extracted. Second, the keyword lists are combined into a single list for the comparison. This list is dubbed as $L_{keywords}$. Third, all pairwise distances between the keywords are computed with the help of an external GD repository, resulting in a pairwise GD dictionary. Fourth, the K-medoid algorithm, which is described in Section 4.1.4.3, is executed, yielding a set of clusters, or NGTs, that represent generic types. Finally, the calculation of EBD proceeds given the NGTs produced by K-medoid.

Algorithm 2. lookupGeoTypes ($\mathcal{T}_{gaz}$, IL)

Input:
-Set of geographic types $\mathcal{T}_{gaz}$ recognized by gazetteer
-List of instances IL associated with attribute att(T)
Output: an ordered pair ($\mathcal{T}_{att}$, $\mathcal{W}_{att}$) across all instances of att(T)
1 : $\mathcal{T}_{att} = \Phi$, $\mathcal{W}_{att} = \Phi$
2: For each instance $x \in$ IL {
3 : $\mathcal{T}_x$ = typeLookup($\mathcal{T}_{gaz}$, x)
4: if hasLatLong(x){
5: prune ($\mathcal{T}_x$)
6 : $\mathcal{W}_x = w_1 = 1$
7: }else {
8: For each $t \in \mathcal{T}_x${
9 : $w_t = 1/|\mathcal{T}_x|$
10 : $\mathcal{W}_x = \{w_1 \ldots w_{last}\}$
11: }//end for
12: }//end if
13 : $\mathcal{T}_{att} = \mathcal{T}_{att} \cup \mathcal{T}_x$
14 : $\mathcal{W}_{att} = \mathcal{W}_{att} \cup \mathcal{W}_x$
15: }//end for
16: return ($\mathcal{T}_{att}$, $\mathcal{W}_{att}$)

4.1.4.3. K-medoid clustering. The algorithm begins by determining the number of clusters K based on the size of $L_{keywords}$ for each pair of compared attributes. Second, exactly one keyword from $L_{keywords}$ is assigned to each of the K clusters in a process called initial seeding. Each of these keywords is then considered a medoid for its particular clustering. Third, we continuously assign each remaining keyword in $L_{keywords}$ that is not a medoid to the cluster

to which it is most semantically related. Once we have assigned all keywords in $L_{keywords}$, the algorithm determines if any cluster medoids need to be recomputed. To do this, we need to use the GD values between the keyword to be assigned to a cluster and all keywords already assigned to that same cluster. A given keyword, k_{new} is assigned to the cluster associated with the smallest summation of the GD values between k_{new} and the cluster's constituent keywords. After all keywords have been assigned to clusters, finally, we determine if the medoid for any cluster needs to be recomputed. This is accomplished by examining each of the keywords in a particular cluster and computing a GD summation between a single keyword in that cluster and all other words in that cluster. The keyword in that cluster that produces the lowest GD summation will be assigned as the new medoid for that cluster. If no medoids have changed in any cluster, then the K-medoid algorithm is finished, and control proceeds to the calculation of the EBD between the compared attributes. However, if at least one medoid has changed in a particular cluster, then we begin a new clustering iteration.

4.1.5. Attribute weighting

GSim also provides attribute weighting capabilities to penalize strong semantic correspondences between tables resulting from attribute mappings where the attributes in the mapped pair commonly occur across all of the tables in their respective databases. Doing this allows us to refine the semantic similarity score generated between tables by focusing on the compared attributes that are unique relative to attributes found throughout all tables. Let $S_1 = (T_{11}, T_{12} \ldots T_{1M})$ be the set of tables belonging to data source S_1, and let $S_2 = (T_{21}, T_{22} \ldots T_{2N})$ be the set of tables belonging to data source S_2, and suppose T_{1J} and T_{2K} are being compared for semantic similarity. Further suppose for the sake of simplicity that pairings between attributes of T_{1J} and T_{2K} have been set such that for all i, attribute i of T_{1J} is matched with attribute i of T_{2K}, and T_{1J} and T_{2K} have the same number of attributes. Before attribute weighting is applied, similarity calculations between attribute i of T_{1J} and attribute i of T_{2K} occur. At this point, the EBD values of each attribute pair have equal weight. Recall that attribute-level EBD tells us which attributes are similar between compared tables. We will designate one such value between two attributes as EBD_{orig} (att(T_{1J}), att(T_{2K})).

Realistically, however, some attribute pairs should be weighted higher than others. For example, given two tables, one called Road and another called Street, if the attribute 'roadType' in the Road table (let us call it Road.roadType) was mapped to an attribute 'streetType' in the Street table (let us call it Street.streetType), then this pair should contribute more substantially to the table similarity between Road and Street than a mapped attribute pair consisting of Road.roadName and Street.streetName. While Road.roadType and Street.streetType are two attributes that are not likely to be found in many other GIS tables, Road.roadName and Street.streetName are indeed likely to appear in other GIS tables, if, for example, these tables describe geographic objects that have some kind of street address such as a school, port or business. After deciding the weights of each attribute pair among a set of mappings across compared tables, the end result will be a more accurate EBD score. This is a result of the discriminative power of attribute weighting in that it can determine the attribute pairs that are most important to the table match.

A successful attribute weighting measure ensures that an attribute pairing att1-att2 between table T and table T' is weighted more heavily than other attribute pairs between T and T' because (1: from the pairing att1-att2, att1 and its instances are *relevant* to table T, and att2 and its instances are relevant to table T' (2: each attribute in the pairing att1-att2 is *unique* to its respective table.

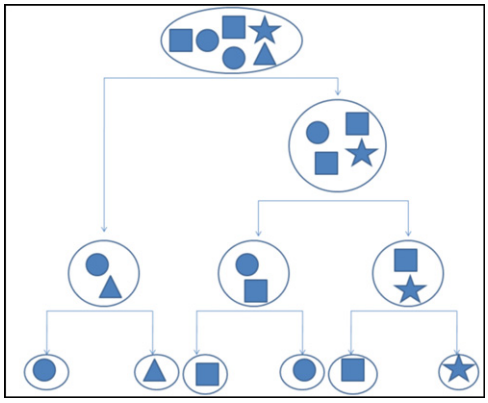

Fig. 7. Conceptual diagram of hierarchical agglomerative clustering.

In other words, for att1-att2 to be weighted more heavily, the frequency by which each individual attribute is found in other tables across both data sources should be small relative to other attribute pairings. In addition, it should be noted that for attribute weighting to be successful, it needs to be executed after deriving EBD measures between all attribute pairings. Section 4.1.5.1 below discusses attribute uniqueness, the main idea behind attribute weighting, while Section 4.1.5.2 discusses the final calculation, determining the weight placed on each attribute pair between compared tables.

4.1.5.1. Attribute uniqueness. The uniqueness of an attribute att1 found within table T for an attribute pairing att1-att2 is known as attribute uniqueness (AU). It is determined by applying hierarchical agglomerative clustering over all attribute names in tables present throughout all data sources. Fig. 7 shows the basic outline of this method of clustering. In the first step, each attribute that takes part in an attribute pairing is contained within its own singleton cluster. Next, two singleton clusters are merged together to form a new one containing two attributes. Each merger of two distinct clusters is known as a cluster iteration (CI). Each subsequent step continues to merge two distinct clusters until ideally, all related attributes across tables and data sources are grouped into distinct clusters.

The quality of the clustering, and thus the accuracy of AU values for any given attribute, depends on two factors: (1: the intercluster distance measure (2: the measure used to determine when to stop the clustering.

The intercluster similarity (ICS) measure used to determine the similarity between any two clusters A and B is expressed as follows:

$$ICS_{AB} \frac{\sum_{a \in A} \sum_{b \in B} (S_N(a, b) + (S_{EBD}(a, b))}{|A| \times |B|} \quad (3)$$

where a is an attribute name belonging to cluster A, b is an attribute belonging to cluster B, S_N is the name similarity between the names of attributes a and b, S_{EBD} is the EBD value generated between attributes a and b, $|A|$ is the number of attributes in cluster A, and $|B|$ is the number of attributes in cluster B. If no attribute pairing exists between attributes a and b, then we assume that the sum of S_N and S_{EBD} in this case is = 0. This measure allows attribute similarity among sets of attributes within clusters to be based not only on the properties of the attribute names themselves, but also on their associated instances.

We add our own contribution to the standard hierarchical clustering technique through a specialized cluster stop criterion. Stopping the clustering at the most appropriate time is based on an intracluster distance measure applied after each cluster iteration over all clusters. We will refer to it as the cutoff point (CP) of the clustering. It is the average summation of the name and EBD similarity between all valid pairings of attributes within a given cluster,

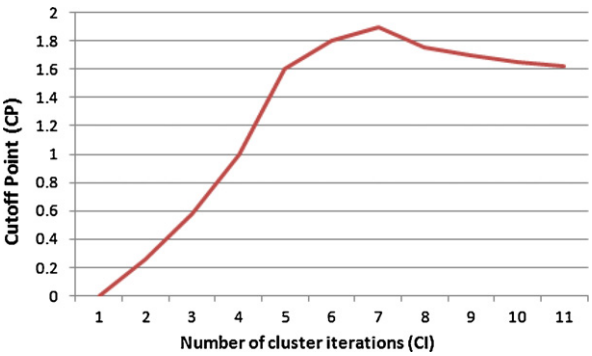

Fig. 8. Cutoff point vs. number of cluster iterations.

taken over all clusters. It is expressed as follows:

$$CP = \frac{\sum_{A \in C}(\sum_{a_1 \in A; a_2 \in A; table(a_1) \neq table(a_2)}(S_N(a_1, a_2)) + (S_{EBD}(a_1, a_2))/\binom{|A|}{2} - \mathcal{K}_j)}{\sum_{A' \in C}|A'|} \quad (4)$$

In Eq. (4), C indicates the set of clusters, A is the cluster in C that contains the attributes a_1 and a_2 which are being considered for comparsion, a_1 and a_2 are distinct attributes within a single cluster A, $|A|$ is the number of attributes in cluster A, A' indicates an arbitrary cluster in C, the binomial coefficient that reads "$|A|$ choose 2" indicates the number of possible subsets of attributes from A that are of size = 2 (in other words, the number of possible pairings of attributes within cluster A), $\mathcal{K}$ indicates the number of attribute pairings within A that are not possible, due to both attributes being from the same table (these do not necessarily include pairings between a_1 and a_2 of different tables that have a value = 0 for $S_N + S_{EBD}$), and $|C|$ is the number of total clusters.

The quantity, $\binom{|A|}{2} - \mathcal{K}$, then, represents the total number of attribute pairings within cluster A in which the attributes in each pair are not from the same table.

The logic behind Eq. (4) is illustrated in Fig. 8. It displays a graph of the relationship between the number of cluster iterations (CI), located on the x-axis, and the cutoff point (CP), located on the y-axis. Once the average summation of S_N and S_{EBD} between all valid attribute pairs over all clusters reaches a maximum value, then the clustering is stopped, as we have attained an optimal clustering. According to Tan et al. [49], typical hierarchical agglomerative clustering cannot be viewed as globally optimizing an objective function. Rather, this type of clustering uses local criteria at each cluster iteration to merge two clusters. While standard hierarchical agglomerative clustering continues to merge clusters until the creation of one final cluster, encompassing all others, our technique uses the CP to stop the clustering prematurely, with multiple clusters remaining. Aside from how the clustering concludes, our clustering technique is identical to standard hierarchical agglomerative clustering. As a result, finding a global maximum for the CP value will be computationally infeasible. Hence, we say that the CP in Fig. 8 represents a local maximum.

One question that naturally arises is the time complexity bottleneck that occurs as a result of the binomial coefficient term. However, since this process is executed offline, and because of the iterative algorithm reported by Manolopoulos [50], we implemented this term to run in $O(\min(k, n-k))$. In our case, $n = |A|$, and $k = 2$, making the execution time polynomial in the size of the cluster. Thus, for the reasons described above, the time complexity of this step is not a bottleneck.

Once we have completed the clustering, the attribute uniqueness AU_{att} of a given attribute is as follows:

$$AU_{att} = 1 - \left(\frac{|A| - 1}{\sum_{A' \in C}|A'|}\right) \quad (5)$$

AU_{att} always takes on a value between 0 and 1, with 0 indicating no attribute uniqueness, and 1 indicating the highest attribute uniqueness. A high AU_{att} value is achieved when attribute *att* appears infrequently across the tables of S_{att}, while a low value of AU_{att} occurs for an attribute that is commonly occurring across the tables of S_{att}. An AU_{att} value of 1 indicates that an attribute is unique (in its own cluster by itself), while an AU_{att} value approaching 0 means that an attribute is one of many attributes in its own cluster. Note that an AU_{att} value for an attribute value *att* that has a value of 1 indicates that *att* has no other matching attribute in its cluster. As a result, *att* should not be involved in any match.

Recall that a single EBD value is between two attributes, and thus, to measure pairwise uniqueness, we need a measure that accounts for the AU_{att} value for both attributes in a pair. This measure is called pair uniqueness and designated as $PU_{att1,att2}$. It may be calculated by taking the arithmetic mean of the AU_{att} values for each attribute in a pair, the minimum AU_{att} value out of the pair, the maximum AU_{att} value out of the pair, and in a number of other ways. For our purposes, we achieved the most promising results when calculating $PU_{att1,att2}$ as the average of AU_{att1} and AU_{att2}. Like AU_{att}, the range of possible values for $PU_{att1,att2}$ is that between 0 and 1, since it is based on AU_{att1} and AU_{att2}, both of which have values between 0 and 1.

4.1.5.2. Deriving a final weighting. Pair uniqueness is then multiplied by the EBD_{orig} value produced by the pair to give a corrected value called EBD_{corr}:

$$EBD_{corr}(att1, att2) = EBD_{orig}(att1, att2) \times PU_{att1,att2} \quad (6)$$

Note that EBD_{corr} must be less than or equal to than EBD_{orig}, because $PU_{att1,att2}$ takes on a value in the range [0,1]. The difference between $EBD_{corr(att1,att2)}$ and $EBD_{orig(att1,att2)}$, called pairwise semantic disparity ($PSD_{att1,att2}$), is then found between att1 and att2, and for all pairs of matching attribute pairs between two compared tables:

$$PSD_{att1,att2} = EBD_{orig(att1,att2)} - EBD_{corr(att1,att2)} \quad (7)$$

Next, the arithmetic mean of the PSD values, dubbed PSD_{avg}, among all of the attribute pairs for a table comparison is found. An attribute pair with a PSD value greater than PSD_{avg} indicates that a greater discrepancy exists between EBD_{orig} and EBD_{corr} relative to other attribute pairs. As a result, this pair should have the weight of its EBD_{orig} value reduced. In contrast, an attribute pair with a PSD value below PSD_{avg} indicates that relative to other pairs, its EBD discrepancy was less, and because of this, its attributes are more unique. Thus its EBD_{orig} value should contribute more substantially to semantic similarity between the tables. The new weight assigned to the attribute pair depends upon how far above or below the PSD value is relative to PSD_{avg}. For instance, an attribute pair that produces a PSD value that is .06 below PSD_{avg} is more unique than an attribute pair that produces a PSD value that is .03 below PSD_{avg}. Conversely, an attribute pair that produces a PSD value that is .06 above PSD_{avg} is less unique than an attribute pair that produces a PSD value that is .03 above PSD_{avg}.

Attribute weighting, as described above for a single table comparison, is illustrated in Algorithm 3 below. Line 1 stores the

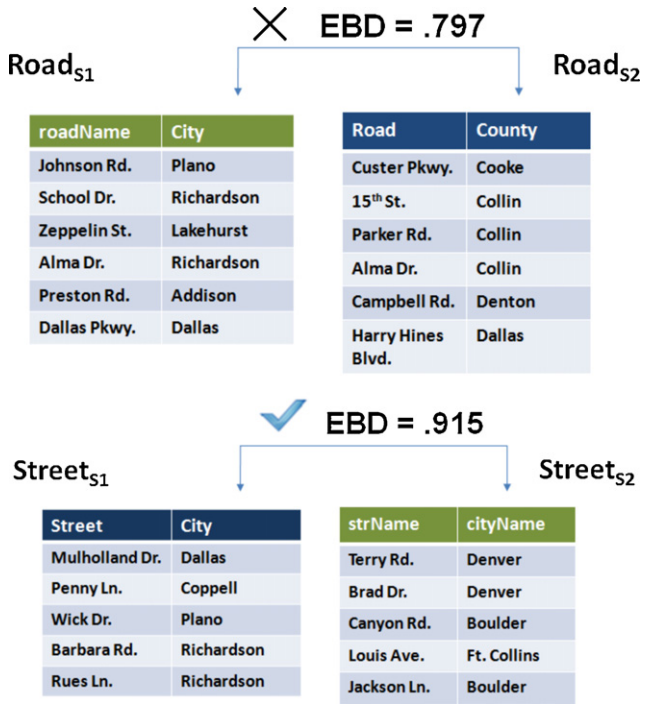

Fig. 9. (a) (top) is an example of how GD can produce an incorrect attribute mapping based on a high semantic similarity score if the instances being compared are geographically proximate. (b) (bottom) shows a situation where a high semantic similarity score from GD produces a correct mapping.

attribute mappings that were generated by GSim. Line 2 performs the hierarchical agglomerative clustering described in section 4.1.5 and assigns the derived set of clusters and their associated attributes taken from $\mathcal{M}_{att(T),att(T')}$ to C. Lines 3–9 analyze each attribute mapping in $\mathcal{M}_{att(T),att(T')}$ and ultimately calculate the PSD value between the attributes in the given mapping. Lines 4–5 calculate AU_{att1} and AU_{att2} for an attributes att1 and att2, respectively, and line 6 calculates the pairwise uniqueness between AU_{att1} and AU_{att2}. Line 7 calculates the corrected EBD value, EBD_{corr} for the pair att1-att2, and this value is used in line 8 to calculate the pairwise semantic distance, or PSD, for the pairing att1-att2. Line 10 determines the average of the PSD values taken over all attribute pairs. Lines 11–15 compare PSD_{avg} against the PSD value generated for a given attribute pair. If the PSD_{avg} is a higher value, then this means that the disparity in EBD values for this pair was less than the average, thus indicating that the pair is unique relative to other pairs. This results in the pair's EBD value having a higher weight relative to other pairs in its table. On the other hand, if the PSD value generated between the attribute pair is higher, then the disparity of EBD values for this pair was more than average, indicating that the pair is not unique relative to other pairs. This results in a deduction of weight for the pair's EBD value relative to other pairs in the table. Finally, line 17 returns the weights of all attribute pairs as a vector.

4.2. Problem with using GD

Despite the utility of GD over a number of domains, it tends to produce inaccurate results with regards to the GIS domain when the compared instances are geographically proximate, despite being completely different types. Fig. 9a describes one particular example of this phenomenon. It serves as a justification of why GT matching is performed.

Algorithm 3. attributeWeighting (T,T')

Input: Tables T and T', which are being semantically compared
Output: A weight vector $W_{att(T)\text{-}att(T')}$ containing normalized weights for each attribute pair among T and T'.
1: $\mathcal{M}_{att(T),att(T)'}$ = getattributeMappings(T, T')
2: $\mathcal{C}$ = performClustering ($\mathcal{M}_{att(T),att(T')}$)
3: For each attribute pair (att1(T),att2(T')) ε
For each attribute pair (att1(T), att2(T')) $\in \mathcal{M}_{att(T),att(T')}$ {
4: $AU_{att1(T)}$=calculateAU(att1(T), $\mathcal{C}$)
5: $AU_{att2(T)}$=calculateAU(att2(T'), $\mathcal{C}$)
6: $PU_{att1(T),att2(T')} = (AU_{att1(T)}+AU_{att2(T+)})/2$
7: $EBD_{corr}(att1(T), att2(T')) = EBD_{orig}(att1(T), att2(T')) \times PU(att1(T), att2(T'))$
8: $PSD_{att1(T),\ att2(T')} = EBD_{corr}(att1(T), att2(T')) - EBD_{orig}(att1(T), att2(T'))$
9: }//end for
10: PSD_{avg} = computeAvg ($\mathcal{M}_{att1(T),\ att2(T')}$,$PSD_{att1(T),\ att2(T')}$)
11: For each attribute pair (att1(T), att2(T')) $\in \mathcal{M}_{att1(T),\ att2(T')}$ {
12: if ($PSD_{att1(T),att2(T')}$-PSD_{avg} > 0)
13: $W_{att1(T),att2(T')}$ = reduceWeight(att1(T),att2(T'))
14: else
15: $W_{(att1(T),att2(T'))}$ = increaseWeight(att1(T),att2(T'))
16: }//end for
17: return $W_{att(T),att(T')}$

The attribute "City", associated with table $Road_{S1}$ is compared against the attribute "County" from table $Road_{S2}$. Although the instances are of different types, they are geographically proximate, as both the cities from "City" and the counties from "County" both describe the Dallas-Fort Worth area. As a result, even though the types are totally different, the exclusive usage of GD for NGT matching will deem that the "City" attribute is semantically similar to the "County" attribute. This happens because GD, by definition, is computed based on the probability of the co-occurrence of search terms *x* and *y* on a given web page indexed by the Google search engine. In many situations, a high probability of co-occurrence between *x* and *y* indicates that the terms are likely to be semantically similar to one another. However, as Fig. 9a shows, co-occurrence does not always imply similarity.

4.3. Proposed solution to GD inaccuracies

We propose a solution to overcome the matching problem inherent in the GD method outlined in Section 4.2.

The proposed idea can be split into two separate parts. First, we try to resort to alternative means of acquiring the GT of an instance, if we cannot determine its type from GeoNames. We may use any number of other gazetteers to directly acquire the type from their type thesauri, use Wikipedia to determine the type based on the Wikipedia category associated with the instance, or retrieve the top *M* highest-ranking Web pages from Google, where M is a threshold indicating a maximum number of Web pages, and use geotagging on the names of the instances. We could also integrate this step as part of our GT matching algorithm; this way, if we need to resort to NGT matching, then we know that we have tried all possible geographic repositories to make GT matching work.

The second part of the solution would be executed if GT similarity was attempted, but was not able to determine the types of a sufficient number of instances (In our experiments, 50% of the total number of instances between the compared attributes having GTs is sufficient for GT matching). In this case, we resort to NGT matching and group the instances of the compared attributes into NGTs based on GD. Each NGT would be represented as a cluster of semantically related instances from both attributes. Among these instances in each cluster, some would have GTs that were explicitly determined from the previously attempted GT matching, and some would not have any GTs. During each 1-1 attribute mapping over NGTs, we would be able to use the instances with GTs from the previously attempted GT matching to verify whether GD has correctly clustered instances together, and thus, if NGT matching has produced a correct attribute match.

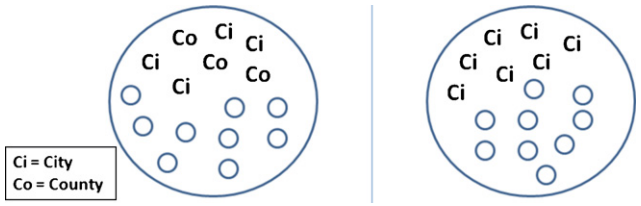

Fig. 10. NGTs containing instances whose GTs were explicitly determined, and instances whose GTs are unknown. The left NGT corresponds to Fig. 9a, an incorrect mapping. The right NGT corresponds to Fig. 9b, a correct mapping.

For each NGT, we are using those instances with GTs to guide us in determining its quality. Informally, if an NGT contains mostly instances associated with > 1 GT, then the NGT is deemed impure. However, if an NGT contains mostly instances with associated with a single GT, then the NGT is deemed pure. If an acceptable number of the instances throughout all of the NGTs have been deemed pure (equal to or exceeding a predefined threshold), then the attribute match is verified to be correct. However, if too many instances across all NGTs have been deemed impure (below a threshold value), then the attribute match is verified as being incorrect. The result of this is a readjustment of the final EBD score between the attributes by changing the contribution that each NGT makes.

Fig. 10 illustrates an impure NGT (on the left) and a pure NGT (on the right). As can be seen, each NGT represents instances of an attribute comparison between two attributes. The left NGT is derived from the attribute comparison depicted in Fig. 9a, while the right NGT is derived from the attribute comparison depicted in Fig. 9b. We will assume that in Fig. 9a, any instances in the City attribute with GTs have a GT of type "City", while any instances in the County attribute with GTs have a GT of type "County". In Fig. 9b, we will assume that all instances from both attributes that have GTs are of type "City". In both NGTs of Fig. 10, an instance labeled by a gazetteer with "Ci" represents the GT "City", while an instance labeled by a gazetteer with "Co" represents the GT "County". The empty white circles indicate instances whose GT could not be determined explicitly by GSim. The NGT on the left, which results from the attribute comparison of Fig. 9a, is impure. To understand this, we first can see four city instances from the "City" attribute and three county instances from the "County" attribute. We also have a number of instances from both attributes whose GT cannot be determined. Since the instances collectively refer to more than one GT, we can infer that the NGT is impure. We may infer this even if the GD similarity between the two says otherwise. As a result of the impurity of the NGT, we may lower its weight in the EBD calculation between the attributes. For the NGT on the right produced from the attribute comparison of Fig. 9b, all instances whose types are known share a single GT of type "City". Thus, the NGT is pure. If this is the only NGT between the compared attriburtes, we would conclude that the mapping between attributes in this case is correct.

5. Experiments

We now present six separate experiments that we conducted regarding matching between distinct data sources in the GIS domain. The first experiment measured GSim's ability to compute semantic similarity between two pairs of GIS databases. The second experiment applied the use of latlong techniques to disambiguate between the GTs of instances in an attempt to improve our results. The third experiment illustrates GSim's NGT matching component, in a situation where GT matching is not possible, and we compare the results generated with those from the prior GT matching experiments. The fourth experiment illustrated GSim's attribute weighting feature, which gives it the ability to penalize table matches involving commonly occurring and irrelevant attributes found through the GIS database and reward table matches containing attribute pairs that were unique and relevant to their respective tables. The fifth experiment illustrates the results generated by GSim when all of its approaches are applied to tables of a particular dataset, one at a time. Doing this more clearly shows the contribution that each individual matching method makes towards the generated final similarity value between compared tables. The final experiment compares the results of GSim against two other popular methods used in the data mining community to determine semantic correspondence between data sources: nonnegative matrix factorization (NMF) and singular value decomposition.

Table Name	No. of Attributes	Areas Modeled	No. of Instances
Road(S_1), Road(S_2)	5,5	Fort Collins, CO Dallas, TX	1970, 1045
Ferry(S_1), Ferry(S_2)	4,3	Seattle, WA	24,42
Traffic Area(S_1), Traffic Area(S_2)	3,3	Virginia	329,108
Residential Area(S_1), Residential Area(S_2)	5,4	New Jersey, Texas	852,424

Table Name	No. of Attributes	No. of Instances
Flight Schools(S_1), Flight Schools(S_2)	7,8	930, 930
Schools(S_1), Schools(S_2)	3,3	11435, 11890
Piers(S_1)	6	1232
Indian Lands(S_1)	3	3160
Ports(S_2)	5	907
NavWaterways(S_2)	4	1377

Table Name	No. of Attributes	No. of Instances
Hospitals(S_1), Hospitals(S_2)	4,4	829 (for both)
Schools(S_1), Schools(S_2)	4,4	5768 (for both)
Streets1(S_1), Streets2(S_2)	6,6	1384 (for both)

Fig. 11. Description of GIS transportation dataset (top), GIS location dataset (middle) and GIS POI dataset (bottom).

5.1. Dataset details

Fig. 11 above lists the details of three separate datasets to which we applied the GSim algorithm, along with some baseline methods of calculating semantic similarity. In Fig. 11, tables from different data sources are listed either individually or in pairs. When they are listed in pairs, this implies that the tables are semantically similar, whereas if a table is listed individually (such as the table 'Indian Lands' in the GIS Location Dataset), then this implies that the table does not semantically match with another table. Also, for each table(s), the number of attributes and instances reflects the number involved in semantic matching, as opposed to the actual number of attributes or instances that exist within the table(s). Most of the attributes for each table remained unused either because they did not contain string data (and thus were not eligible for a match), or because they were not relevant enough to be used in our semantic matching experiments. Now the details of each data set will be described. The first dataset, which we dubbed

the GIS Transportation Dataset, was created from instance data of the Road and Ferries package of a GIS data model known as GDF (Geographic Data Files) [21]. The tables vary in regards to number of attributes (with the smallest being in Ferry(S_1), 3, and the largest being in Traffic Area(S_2) with 5), number of instances (smallest being Ferry(S_1) with 24, and the largest being Road(S_1) with 1970), and in regards to geographic area (data models six different states spread across the lower 48 states). We preferred data featuring a wide geographic dispersion with no shared instances. Therefore, similarity between tables would only be possible via a semantic match, as opposed to simple keyword matching. Furthermore, we considered this dataset to be multijurisdictional. The second dataset, which we dubbed the GIS Location Dataset, details a wider assortment of location features across the United States and their associated data beyond merely transportation networks. Some of the location features in this dataset include flight schools, piers, navigable waterways and Indian lands. As with the GIS Transportation Dataset, the number of attributes and instances vary; for example, in the GIS location dataset, the Flight Schools table for S_2 has the largest number of attributes taking part in matching (8) and the both Schools tables and the Indian Lands table has the fewest (3). In regards to instances, Schools(S_2) contains 11,890 instances, the most in the dataset, whereas Ports(S_2) contains the fewest number of instances at 907. As with the GIS Transportation Dataset, the instances in the tables of this dataset are multijurisdictional in nature. The third dataset, which we dub the GIS Point of Interest (POI) Dataset, contains instances that extend beyond road networks and which are multijurisdictional in nature, much like the GIS Location dataset. The number of instances and locations modeled vary widely, which results in a dataset that requires semantic methods by an algorithm for any meaningful schema matching to occur.

5.2. Similarity without using latlong values

5.2.1. Measurements and parameters

The results of the alignment of S_1 and S_2 of the compared tables for both the transportation dataset and the GIS location dataset using GSim and the *N*-gram method are shown in Fig. 12a and b, respectively. For each table comparison, there are four values. From left to right, the first two are the precision and recall (denoted as P and R, respectively) produced using *N*-grams between an attribute from a table in data source S_1 and an attribute from a table in data source S_2. The last two values are the precision and recall values produced by GSim between an attribute from a table in data source S_1 and an attribute from a table in data source S_2. As an example, for the comparison of Road from S_1 and Ferry from S_2 in Fig. 12a, the precision and recall generated using N-grams are 0 and 0, respectively, while the precision and recall generated for GSim is .50 and 1.00, respectively. Also, for each cell containing a precision or recall value, there is a ratio. For precision, the top number of the ratio indicates the number of correct attribute mappings between the compared tables that were identified by the similarity method, while the bottom number indicates the total number of attribute mappings (both correct and incorrect) between the compared tables identified by the matching method. For recall, the top number of the ratio indicates the number of correct attribute mappings between the tables that were returned by the similarity method, while the bottom number of the ratio indicates the total number of correct attribute mappings that exist between the tables. For instance, in Fig. 12a, for the comparison of the Road table from S_1 with the Road table from S_2, the ratio in the cell for the precision of the *N*-gram method is "1/2", meaning that the *N*-gram method returned two attribute mappings between these two tables, but only one was correct. The cell to its right, which is the recall value produced by the *N*-gram method for the Road-Road table comparison, reads "2/4". This means that two correct attribute mappings were returned by the *N*-gram method, while there exists a total of 4 correct attribute mappings between the tables.

t ε S_1	t ε S_2	P (N-gram)	R (N-gram)	F-measure (N-gram)	P (GSim)	R(GSim)	F-measure (GSim)
Road	Road	.50 (1/2)	.25 (1/4)	.20	.50 (2/4)	.50 (2/4)	.50
Road	Address Area	.25 (1/4)	1.00 (1/1)	.40	1.00 (1/1)	1.00 (1/1)	1.00
Road	Enclosed Traffic Area	.33 (1/3)	.50 (1/2)	.40	.50 (1/2)	.50 (1/2)	.50
Road	Ferry	0 (0/2)	0 (0/1)	0	.50 (1/2)	1.00 (1/1)	.67
Residential Area	Road	.25 (1/4)	1.00 (1/1)	.40	.50 (1/2)	1.00 (1/1)	.67
Residential Area	Address Area	.50 (2/4)	.50 (2/4)	.50	.75 (3/4)	.75 (3/4)	.75
Residential Area	Enclosed Traffic Area	.25 (1/4)	.50 (1/2)	.33	.50 (1/2)	.50 (1/2)	.50
Residential Area	Ferry	------ (0/0)	0 (0/1)	0	1.00 (1/1)	1.00 (1/1)	1.00
Traffic Area	Road	.33 (1/3)	.50 (1/2)	.40	.50 (1/2)	.50 (1/2)	.50
Traffic Area	Address Area	.50 (1/2)	.50 (1/2)	.50	1.00 (1/1)	.50 (1/2)	.67
Traffic Area	Enclosed Traffic Area	.50 (1/2)	.50 (1/2)	.50	1.00 (2/2)	1.00 (2/2)	1.00
Traffic Area	Ferry	.50 (1/2)	1.00 (1/1)	.67	1.00 (1/1)	1.00 (1/1)	1.00
Ferry	Road	.50 (1/2)	1.00 (1/1)	.67	1.00 (1/1)	1.00 (1/1)	1.00
Ferry	Address Area	1.00 (1/1)	1.00 (1/1)	1.00	1.00 (1/1)	1.00 (1/1)	1.00
Ferry	Enclosed Traffic Area	.50 (1/2)	1.00 (1/1)	.67	1.00 (1/1)	1.00 (1/1)	1.00
Ferry	Ferry	.50 (1/2)	.33 (1/3)	.4	.67 (2/3)	.67 (2/3)	.67
AVERAGE VALUES		.38	.52	.44	.70	.72	.71

t ε S_1	t ε S_2	P (N-gram)	R (N-gram)	F-Measure (N-gram)	P (GSim)	R(GSim)	F-Measure (GSim)
Flight Schools	Flight Schools	1.00 (2/2)	.29 (2/7)	.45	1.00 (7/7)	1.00 (7/7)	1.00
Flight Schools	Schools	------ (0/0)	0 (0/3)	0	1.00 (2/2)	.67 (2/3)	.80
Flight Schools	Ports	------ (0/0)	0 (0/3)	0	.33 (1/3)	.33 (1/3)	.33
Flight Schools	NavWaterways	------ (0/0)	0 (0/3)	0	.33 (1/3)	.33 (1/3)	.33
Schools	Flight Schools	------ (0/0)	0 (0/3)	0	1.00 (3/3)	1.00 (3/3)	1.00
Schools	Schools	.66 (2/3)	.33 (1/3)	.44	1.00 (3/3)	1.00 (3/3)	1.00
Schools	Ports	------ (0/0)	0 (0/3)	0	1.00 (1/1)	.33 (1/3)	.50
Schools	NavWaterways	------ (0/0)	0 (0/3)	0	1.00 (3/3)	1.00 (3/3)	1.00
Indian Lands	Flight Schools	------ (0/0)	0 (0/3)	0	.33 (1/3)	.33 (1/3)	.33
Indian Lands	Schools	------ (0/0)	0 (0/3)	0	1.00 (1/1)	.33 (1/3)	.50
Indian Lands	Ports	------ (0/0)	0 (0/3)	0	1.00 (1/1)	.33 (1/3)	.50
Indian Lands	NavWaterways	------ (0/0)	0 (0/2)	0	1.00 (2/2)	1.00 (2/2)	1.00
Piers	Flight Schools	------ (0/0)	0 (0/5)	0	1.00 (2/2)	.40 (2/5)	.57
Piers	Schools	------ (0/0)	0 (0/3)	0	1.00 (1/1)	.33 (1/3)	.50
Piers	Ports	------ (0/0)	0 (0/3)	0	.50 (1/2)	.67 (2/3)	.57
Piers	NavWaterways	------ (0/0)	0 (0/2)	0	.67 (2/3)	1.00 (2/2)	.80
AVERAGE VALUES		.80	.06	.09	.80	.61	.68

Fig. 12. Precision, recall and *F*-measure values between tables of S_1 and S_2 using N-grams and GSim relative to ground truth for (a: transportation dataset (top) (b: GIS location dataset (bottom) These do not use latlong values.

The values produced by both methods depend on a reference alignment, or ground truth, which contains the attribute pairs that are supposed to be semantically similar. The ground truth for both datasets was created by human experts knowledgeable in the area of GIS. For our experiments, we set two standards that affected the results. First, we decided that whenever an attribute pair produced a similarity value (an EBD value) measured to be greater than or equal to .6, then the method determined that pair to be a match. Second, we set *N*-grams to be of size = 2, since any size > 2 would increase the number of possible *N*-grams by a margin significant enough such that the precision and recall values would almost always be too low to meet the match threshold for any dataset, thus rendering this method virtually useless as a semantic similarity measure for our experiments. Overall, the ground truth for the transportation dataset contained 29 correct mappings across all table comparisons, while the ground truth for the GIS location dataset contained 52 correct mappings across all table comparisons.

It should also be noted that in our experiments, valid attribute mappings were found even between tables that do not naturally

correspond. For instance, in the GIS POI dataset, valid attribute mappings exist between disparate tables like Streets1 and Schools2 (the City attribute, in this case). These mappings, and others which exist among the other datasets that we experimented upon, were included in our reference alignments.

5.2.2. Analysis of results

Fig. 12a shows the comparison of precision, recall and *F*-measure values using both GSim and the *N*-gram method for the transportation dataset. Note that the precision and recall values generated by GSim are never lower than those produced by N-grams for any table comparison. In total, the average precision produced by GSim was .70, and its average recall was .72. In contrast, the average precision of N-grams was .38, and its average recall was .52. GSim achieved a 32% improvement over N-grams in precision, and a 20% improvement in recall. Fig. 12b depicts even more dramatic improvements made by GSim. The precision and recall values for GSim are always higher than those produced by the *N*-gram method for any table comparison. In total, the average precision produced by GSim was .80, and its average recall was .61. In contrast, while the average precision of *N*-grams is .80, the average recall is a staggeringly low value of .06. In fact, the reason why *N*-grams' precision was able to match GSim's precision was due to the extremely low recall. The reason for the low recall value was primarily due to the lack of identical instances between the compared attributes. As a result, most of the comparisons using the *N*-gram method were not able to reach the .60 threshold in semantic similarity. We did not lower the match threshold below .60 because we felt that a match threshold of a value that was lower, such as .50, would not be a realistic match threshold for determining whether two schemas were similar or not. The reason is that at lower thresholds, the precision and recall values generated by sophisticated and simplistic algorithms alike are not significantly different. As a result of the higher threshold, GSim more clearly illustrates its more sophisticated semantic capabilities, largely resulting from GT extraction. This allows it to achieve a 55% improvement on recall versus a syntactic method such as *N*-grams.

In Fig. 12a, the only reason why *N*-grams even performed somewhat competently was because of the large number of identical instances between many attribute pairs that happened to be similar. For the *N*-gram method to derive an attribute mapping, the instances between the compared attributes must share strings of length *N*. As an example, with $N = 2$, the instances "**Pa**sadena" and "El **Pa**so" (from different attributes) would share a single 2-gram match on "Pa". Given enough matches of this sort between instances of two compared attributes, the *N*-gram method will derive a similarity score that meets the threshold of .60, registering as an attribute mapping. In table comparisons where the *N*-gram method derived a precision or recall value that was extremely low, the instances in the compared attributes shared few strings. This is what makes the *N*-gram method a syntactic method, as opposed to GSim. In Fig. 12b, the *N*-gram method creates very few mappings that even reach the threshold of .60, because the GIS location dataset contains very few shared strings between valid attribute mappings of two tables.

Notice that when applying GSim to pairs of tables which seem incompatible (i.e.: Road-Address Area), it still yields some attribute matches, as evidenced by nonzero precision and recall scores. This is because valid attribute matches can exist between tables which are not compatible. An example of this is Road(S_1). County-AddressArea(S_2). Areaname – even though these tables are not related, they share this attribute, and thus, a match should exist. GSim is able to identify these kinds of attribute mappings, regardless of whether the compared tables seem compatible or not. This is evidenced by the 1.0 precision values between tables as different as Residential Area (S_1) and Ferry (S_2) in the GIS transportation dataset, and between Schools (S_1) and Ports (S_2) in the GIS location dataset.

$t \in S_1$	$t \in S_2$	P (N-gram)	R (N-gram)	F-Measure (N-gram)	P (GSim) w/o latlong	R(GSim) without latlong	F-Measure (GSim) without latlong	P (GSim) latlong	R(GSim) latlong	F-Measure (GSim) latlong
Streets1	Streets2	.50	.50	.50	1.00	1.00	1.00	1.00	1.00	1.00
Streets1	Schools2	0	0	0	0	0	0	1.00	.50	.67
Schools1	Streets2	0	0	0	0	0	0	1.00	1.00	1.00
Schools1	Schools2	1.00	.50	.67	1.00	1.00	1.00	1.00	1.00	1.00
Schools1	Hospitals2	1.00	.50	.50	1.00	.50	.67	1.00	1.00	1.00
Hospitals1	Schools2	1.00	.50	.67	1.00	.50	.67	1.00	.50	.67
Hospitals1	Hospitals2	1.00	.50	.67	1.00	1.00	1.00	1.00	1.00	1.00
AVERAGE VALUES		.86	.29	.43	1.00	.76	.86	1.00	.90	.95

Fig. 13. Precision, recall and *F*-measure values between tables of S_1 and S_2 in POI dataset generated using N-grams and GSim without latlong values. To the right of these are Precision, recall and *F*-measure values between tables of S_1 and S_2 in POI dataset generated by GSim using latlong values.

5.3. Similarity using latlong values

5.3.1. Measurements and parameters

Fig. 13 above displays precision, recall and *F*-measure values in a dataset known as the GIS POI (point of interest) dataset comparing semantic similarity generated by the baseline *N*-gram method, GSim without the use of latong values, and GSim with the use of latlong values. The GIS POI dataset represents, as the name implies, a multijurisdictional collection of streets, schools and hospitals that are identified as points of interest in GeoNames. As with our previous experiments, the values produced by both *N*-grams and GSim in this dataset depend on a reference alignment which contains the attribute pairs that are supposed to be semantically similar. The ground truth for both datasets was created by human experts knowledgeable in the area of GIS. However, in this experiment, we also directly compare the benefits that latlong values have on deriving similarity.

5.3.2. Analysis of results

As Fig. 13 shows, not only does GSim produce markedly better results versus the *N*-gram approach, but when GSim has access to latlong values for the purposes of further disambiguating between the GTs of instances, the results are even better. As can be seen, GSim without latlong values has an average precision of 1.00, while the average precision value for *N*-grams is .86. This amounts to a 16% improvement in precision by using GSim. As for average recall, GSim without latlong values produces a value of .76, while *N*-grams produces a value of .29. This represents a nearly threefold improvement in recall for GSim over *N*-grams.As for the average *F*-measure, GSim produces a value of .86, while N-grams produces a value of .43. In other words, GSim produces an *F*-measure that is twice as good as the *F*-measure for *N*-grams. In addition to this, Fig. 13 shows that the use of latlong values in GSim produces further improvement. Using GSim with latlong values, average recall is measured at .90, an 18.4% increase over GSim without latlong values (.76). As for average *F*-measure, GSim with latlong values produces a value of .95, a 10.4% improvement over GSim without using latlong values (.86). Before the use of latlong values, a number of instances (especially those with common names) between any two compared attributes might possess GT sets of a size > 1. The end result of this was that instance that were genuinely of the same GT but were tagged with multiple semi-overlapping GTs would have their similarity diminished unfairly, while instances that were genuinely not of the same GT but were tagged with multiple semi-overlapping GTs would have their similarity bolstered unfairly. However, using latlong values, if the instance is recognized by the gazetteer, then

t ε S_1	t ε S_2	P (NGT)	R (NGT)	F-measure (NGT)
Road	Road	.50 (2/4)	.50(2/4)	.50
Road	Address Area	1.00 (1/1)	1.00(1/1)	1.00
Road	Enclosed Traffic Area	.50 (1/2)	.50 (1/2)	.50
Road	Ferry	0 (0/1)	0 (0/1)	0
Residential Area	Road	.33 (1/3)	1.00 (1/1)	.50
Residential Area	Address Area	.50 (2/4)	.50 (2/4)	.50
Residential Area	Enclosed Traffic Area	.50(1/2)	.50 (1/2)	.50
Residential Area	Ferry	1.00 (1/1)	1.00 (1/1)	1.00
Traffic Area	Road	.50 (1/2)	.50 (1/2)	.50
Traffic Area	Address Area	.50 (1/2)	1.00(1/1)	.67
Traffic Area	Enclosed Traffic Area	.50 (1/2)	.50 (1/2)	.50
Traffic Area	Ferry	.50 (1/2)	1.00 (1/1)	.67
Ferry	Road	1.00(1/1)	1.00 (1/1)	1.00
Ferry	Address Area	1.00 (1/1)	1.00 (1/1)	1.00
Ferry	Enclosed Traffic Area	1.00 (1/1)	1.00 (1/1)	1.00
Ferry	Ferry	.50 (1/2)	.33 (1/3)	.40
AVERAGE VALUES		.55	.61	.58

Fig. 14. Precision, recall and *F*-measure values produced by the NGT matching component of GSim.

a 1:1 mapping between it and its correct GT is guaranteed to exist. Because of this, correct correspondences have their score raised, thus explaining the improved scores.

5.4. NGT matching experiment

To illustrate the effectiveness of GSim's NGT matching component and to compare it to its GT matching component, we replaced instances from the GIS transportation dataset that were previously identified by a gazetteer with new instances whose type could not be discerned. Fig. 14 shows the results of NGT matching applied to the GIS transportation dataset. The precision, recall, and *F*-measure values are all better than what the *N*-gram method produced, but they are not as good as the results of GT matching on this dataset, as seen in Fig. 12a. Specifically, the average precision produced by NGT was 45% higher than the precision produced by *N*-grams, but 21% lower than the precision produced using GT matching. The recall produced by NGT was 17% higher than that produced by *N*-grams, but 18% lower than the recall attained by GT matching.

5.5. Attribute weighting experiment

5.5.1. Measurements and parameters

To better illustrate the benefits of attribute weighting on matching tables, we preprocessed the attributes from tables of the GIS Transportation dataset and the GIS Location dataset to optimize GSim's ability to distinguish between commonly occurring attributes and attributes that are more unique. The results of applying GSim's attribute weighting algorithm to the tables from the GIS Transportation dataset and the GIS Location dataset are shown below in Fig. 15a and b, respectively. Fig. 16a below illustrates EBD values produced between tables of the GIS POI dataset where all attribute mappings share equal weight while Fig. 16b illustrates the EBD values produced between these same tables where the attribute mappings now have attribute weighting applied to them. The table names along the vertical axis of the table belong to S_1, while the tables across the horizontal axis of the table belong to S_2.

One last experimental parameter that should be mentioned is an attribute relevance parameter α that was applied to all attributes in tables from S_1 and S_2. Attribute relevance in GSim is executed as a preprocessing step that prevents any attribute that has a name or instance data which is not relevant to its containing table from taking part in a match with an attribute of another table. For instance, if table "Road" from S_1 is being compared with a table "Street" from S_2, then an attribute "Road.roadName", along with instance data

	Road	Address Area	Enclosed Traffic Area	Ferry
Road	**.598**/.553	**.227**/.225	**.290**/.276	**.451**/.503
Residential Area	**.217**/.210	**.583**/.552	**.412**/.433	**.409**/.407
Traffic Area	**.151**/.136	**.206**/.209	**.962**/.958	**.218**/.235
Ferry	**.139**/.127	**.244**/.237	**.433**/.424	**.589**/.564

	Flight Schools	Schools	Ports	NavWaterways
Flight Schools	**.764**/.720	**.622**/.615	**.520**/.532	**.487**/.503
Schools	**.381**/.388	**.791**/.768	**.381**/.395	**.542**/.540
Indian Lands	**.473**/.489	**.506**/.513	**.488**/.486	**.522**/.533
Piers	**.490**/.496	**.469**/.489	**.639**/.633	**.626**/.616

Fig. 15. Two separate EBD values computed between a table from S_1 and a table from S_2 for the (a: GIS Transportation Dataset (top) (b: GIS Location Dataset (bottom) For each cell, the value right of the slash indicates the EBD value produced without attribute weighting, while the bolded value left of the slash is the EBD produced with the help of attribute weighting.

containing road names, would be considered an attribute that is relevant to its containing table "Road". On the other hand, an attribute known as "Road.internalID", along with instances containing ID values of unknown significance, would likely not have any relevance to its containing table, "Road". The enforcement of attribute relevance is accomplished by taking the GD between the attribute name att1 and the name of the containing table T, added to the average GD between N instance values associated with att1 and the name of the containing table T. We set $\alpha = .90$ for all attribute weighting experiments.

	Streets2	Schools2	Hospitals2
Streets1	**.833**/.701	**.140**/.243	**.134** / .197
Schools1	**.173** / .201	**.842** / .735	**.236** / .269
Hospitals1	**.122** / .165	**.237** / .291	**.847** / .721

	Streets2	Schools2	Hospitals2
Streets1	**.862**/.701	**.122**/.243	**.107**/ .197
Schools1	**.151** / .201	**.871** / .735	**.222** / .269
Hospitals1	**.111** / .165	**.216** / .291	**.869** / .721

Fig. 16. (a) (top) depicts the results of executing GSim on the POI dataset with latlong values but no attribute weighting. For each cell, the value in bold (left of slash) is the EBD score produced using latlong values, while the value to the right of the slash does not use latlong values. (b) (bottom) shows the results of executing GSim on the POI dataset with both latlong values and attribute weighting. For each cell, the value in bold (left of slash) is the EBD produced using both latlong values and attribute weighting, while the value to the right of the slash uses neither latlong values nor attribute weighting.

5.5.2. Analysis of results

The results of Fig. 15a and b shows the effect that attribute weighting by itself has on the EBD scores produced between tables in the GIS transportation dataset and GIS location dataset, respectively. The key observation in the results is that while attribute weighting consistently increases the EBD values between corresponding tables, it produces more arbitrary results among tables which do not naturally correspond. For these tables, latlong values in the data were not available, so the improvement in EBD is entirely the result of attribute weighting. In Fig. 15a, the use of attribute weighting increased the EBD between pairs of corresponding tables (Road–Road, Residential Area–Address Area, Traffic Area–Enclosed Traffic Area, Ferry–Ferry) by 8.3%, 5.6%, 0.5% and 4.4%, respectively, when compared against GSim without latlong values and without attribute weighting. In Fig. 15b, the use of attribute weighting increased the EBD between pairs of corresponding tables (Flight Schools–Flight Schools, Schools–Schools, Piers-Ports, Piers-NavWaterways) by 6.1%, 3.0%, 1.0% and 1.6%, respectively, when compared against GSim without latlong values and without attribute weighting. However, in both figures, EBD values neither consistently increased nor decreased when it came to pairs of tables that do not naturally correspond. The best results with attribute weighting were achieved in Fig. 16b with the POI dataset. Here, we also include Fig. 16a and b as a way to compare the improvement in EBD scores that resulted solely from the inclusion of latlong values (16a) and the improvement garnered with the addition of attribute weighting (16b). In Fig. 16a, in each cell, the value in bold, to the left of the slash, indicates the EBD produced when taking into account latlong values only (without attribute weighting), while the value to the right of the slash indicates the EBD produced by GSim without latlong values or attribute weighting. In Fig. 16b, in each cell, the value to the left of the slash indicates the EBD score produced by GSim when using both latlong values and attribute weighting, while the value to the right of the slash, indicates the EBD produced when using neither latlong values nor attribute weighting. As can be seen in Fig. 16a and b, the use of both latlong values and attribute weighting caused the EBD between corresponding tables to be strengthened more significantly and the EBD between dissimilar tables to be weakened consistently. The use of attribute weighting increase the EBD between pairs of corresponding tables (Streets1–Streets2, Schools1–Schools2, Hospitals1–Hospitals2) by 22.9%, 18.5% and 20.5%, respectively, when compared against GSim without latlong values and without attribute weighting. Additionally, the combination of latlong values and attribute weighting was used to reduce the semantic similarity between dissimilar table pairs by an average of 19.1%. In analyzing the sole effects of attribute weighting, we can see that the EBD between Streets1–Streets increased by 3.5%, the EBD between Schools–Schools2 increased by 3.4%, and the EBD between Hosptals1–Hospitals2 increased by 2.6%. Furthermore, it can be seen that attribute weighting by itself also decreased the EBD values between non-corresponding tables in every case; the average reduction in EBD value due to attribute weighting for these tables was 11.7%.

5.6. Results of combining all approaches in GSim

Fig. 17 above shows the progression of EBD scores when all of the approaches available in GSim are applied one at a time over tables of the POI dataset. For each cell (which represents a table comparison) there are four values. The value in the top row left of slash will be designated as (1), the value in the top row right of slash will be designated as (2), the value in the bottom row left of the slash will be designated as (3), and the value in the bottom row right of the slash will be designated as (4). The values are produced in the following ways: (1) GT matching + latlong + NGT matching + attribute weighting, (2) GT matching + latlong + NGT matching, (3) GT matching + latlong, (4) GT matching. It should be noted that typically, GSim only applies NGT matching if insufficient GT information exists within the data. When it does, it is assumed that GT matching will not be applied, and that NGT matching is applied over all of the instances, including those that do possess GT information. However, for this experiment, we have adapted the NGT matching component of GSim such that it applies only to those instances without a GT. Doing this allows NGT matching to be applied directly on top of GT matching in a cumulative way. The cells containing boldface numbers correspond to semantically compatible table comparisons.

	Streets2	Schools2	Hospitals2
Streets1	**.867/.862** **.833/.701**	.122/.230 .225/.243	.107/.133 .134/.197
Schools1	.151/.173 .173/.201	**.872/.871** **.842/.735**	.222/.222 .236/.269
Hospitals1	.111/.124 .122/.165	.216/.216 .237/.291	**.874/.869** **.847/.721**

Fig. 17. EBD scores produced by GSim over the tables of the POI dataset. For each cell, there are four values, with the value in the top row left of slash designated as (1), the value in the top row right of slash designated as (2), the value in the bottom row left of the slash designated as (3), and the value in the bottom row right of the slash designated as (4). The values are produced in the following ways: (1): GT matching + latlong + NGT matching + attribute weighting (2): GT matching + latlong + NGT matching (3): GT matching + latlong (4): GT matching.

As can be seen, taken over all cells, the largest average change in EBD occurs when latlong values are applied to disambiguate between multiple instances of the same name but different GTs. This accounts for an average of 62.2% of the total EBD change from value (4) to value (1) over all cells. Another trend that can be observed in this experiment is that NGT matching is only beneficial when applied to comparisons involving semantically compatible tables. In these cases, NGT matching proves very useful. However, in situations involving semantically incompatible table comparisons, NGT matching either produces no effect, or in some cases, such as Streets1–Hospitals2 and Hospitals1–Streets2, it actually slightly increases the EBD score. We believe that this occurs for two reasons. First, nearly all instances (about 98.3%) in the POI dataset have a GT identifiable by a gazetteer. Out of the three datasets we have experimented on with GSim, the POI dataset is the only one that contains latlong values associated with its instances. The fact that nearly all instances in the POI dataset having GTs and latlong values guarantees that NGT matching cannot make much of a contribution to the final similarity score. Second, in the cases where NGT matching slightly increases the EBD score between incompatible tables, this occurs because of the tendency of NGT matching to group together instances with more semantic disparity between them than GT matching would allow. NGT matching is based on co-occurrence embedded in the formula for GD. As a result, as long as two instances co-occur on a web page, regardless of their actual types, then they will be grouped together as part of the same generic type. Attribute weighting is responsible for 23.7% of the average change in EBD from value (4) to value (1) over all cells.

5.7. Comparing GSim to NMF and SVD

We also sought to compare the effectiveness of GSim relative to two other widely accepted methods for determining the semantic similarity of sets of documents (or data sources) using keyword frequency. These methods are known as nonnegative matrix fac-

Datasets	N-grams			SVD			NMF			GSim		
	P	R	F	P	R	F	P	R	F	P	R	F
GTD	.38	.52	.44	.08	.29	.13	.17	.50	.25	.70	.72	.71
GLD	.80	.06	.09	.20	.15	.17	.26	.19	.22	.80	.61	.68
GPD	.86	.29	.43	.22	.66	.33	.35	.80	.49	1.0	.76	.86

Fig. 18. Precision, recall and *F*-measure values between the three datasets in our experiments over N-grams, SVD, NMF and GSim. Here, GTD = GIS Transportation Dataset, GLD = GIS Location Dataset, and GPD = GIS POI Dataset.

torization (NMF) [26] and singular value decomposition (SVD) [27].

NMF is an algorithm in linear algebra where a matrix X is factorized into two matrices, W and H. Formally, this is stated as: NMF(X) = WH. NMF differs from other matrix factorization methods in that all entries of W and H are to be nonnegative; this is especially applicable for applications of semantic similarity via keyword frequency, since the minimum frequency of any given keyword in a data source is 0. In SVD, the equation $M = U\Sigma V^*$ is satisfied, where U is an mxm unitary matrix over a field K, Σ is an nxm diagonal matrix with nonnegative real numbers along the diagonal, and V^* is the conjugate transpose of V, a nxn unitary matrix over the field K. Though SVD has many uses, in regards to semantic similarity, it can be applied towards the implementation of latent semantic indexing (LSI). LSI uses SVD to identify patterns in the relationships between the terms and concepts contained in an unstructured collection of text. LSI is based on the principle that words that are used in the same contexts tend to have similar meanings.

We have applied SVD (Singular value decomposition) and NMF on the same datasets that were being used in our experiments to find out the semantic similarity between the attribute pairs of any two tables. For this, first, we have generated a matrix X_{MxN} with m rows and n columns where the row represents distinct words and the column represents its attribute name from these two tables. We have two different implementations. In the frequency variant, each entry (i,j) of the matrix represents how many times the word i appears under a particular attribute j. On the other hand, in the binary variant, each entry (i,j) of the matrix represents the presence of word i under the particular attribute j. Thus if a word i appears under an attribute j, in the binary case, the value of the entry (i,j) is set to 1, whether word i appears one time or one-hundred times under attribute j.

We have used SVD to reduce the dimension of the matrix from n to k where $k \ll n$. SVD decomposes X_{MxN} into a product of three matrices as $X_{MxN} = USV^T$ where U is an m x n matrix, S is an n x n diagonal matrix, and V^T is also an $n \times n$ matrix.

$$S = diag(\sigma_1, \sigma_2, \ldots, \sigma_{\min(m,n)})\,\sigma_1 > \sigma_{2>} \ldots \sigma_{\min(m,n)} \text{ and } \sigma_j > 0 \text{ for all } j > \text{rank}(X_{M\times N})$$

To reduce the dimension we generate a new matrix S^k by keeping the k largest singular values. Next, we have calculated the cosine similarity between attribute pairs by exploiting vectors in these reduced dimensional spaces. If the similarity is above a threshold (.5 for our experiments), we declare that to be a match for a 1-1 attribute comparison.

We compared the effectiveness of GSim to NMF and SVD over the three datasets we experimented on (GIS Transportation Dataset, GIS Location Dataset, GIS POI Dataset) and obtained the following results. These are displayed in Fig. 18 above. The effectiveness of each semantic similarity measure with respect to a particular dataset was quantified using *F*-measure. Since *F*-measure takes into account both precision and recall, it represents the best overall metric to measure the effectiveness of semantic similarity algorithms over a common dataset.

In Fig. 18, it can be seen that for the GIS Transportation Dataset, the *F*-measure generated by GSim outperforms that from the *N*-gram method by 61% (.71 to .44). The difference is even greater versus SVD and NMF, as GSim outperforms SVD .71 to .13 and outperforms NMF .71 to .25. For the GIS Location Dataset, GSim outperforms *N*-grams in terms of *F*-measure .68 to .09. The stark difference in similarity values between GSim and *N*-grams is a direct result of this dataset not containing any shared syntactic instances. As a result, only a method that can effectively measure semantic correspondences between instances is likely to be successful over this dataset. For this same dataset, GSim outperforms SVD in *F*-measure .68 to .17, and GSim outperforms NMF .68 to .22. As for the GIS POI Dataset, GSim outperforms its nearest competitor, NMF, in *F*-measure .86 to .49. Over the three datasets, GSim outperforms *N*-grams .75 to .32, SVD by .75 to .23 and NMF .75 to .37.

6. Conclusion and future work

In this paper, we described GSim, an algorithm that computes the semantic similarity of two tables belonging to distinct GIS data sources. It computes semantic similarity using two separate approaches. The first uses a gazetteer to extract GTs for all possible instances within the compared attributes. The weights of the GTs taken over all instances results in GT sets and GT weight lists, where each attribute features its own GT set and GT weight list. In the more advanced geotyping algorithm featured by GSim, every instance is associated with exactly one GT by comparing the latlong information of the instance against the latlong values for all matching instances found in the gazetteer. The instance in the gazetteer that yields the smallest difference in latlong values with the instance in the data is selected, and its GT is taken to be the final GT of the instance in the data. The similarity of the GT distributions between compared attributes determines the similarity between the attributes, and the average over all attribute pairs determines the table similarity. GSim also compensates for situations when a lack of GT information for the instances is available by executing a domain independent semantic similarity algorithm leveraging normalized google distance. This results in the extraction of NGTs from the instances of the attributes, and semantic similarity is subsequently computed. Additionally, GSim provides attribute weighting capabilities across tables in a GIS database that penalizes the similarity between table matches involving a high number of commonly occurring attributes and/or irrelevant attributes found throughout the database, while enhancing table matches containing unique and relevant attribute mappings.

Future efforts to improve GSim will focus on the following. First, we will refine our GT extraction techniques. This can be done in two ways. The first is to leverage multiple gazetteers making use of heterogeneous feature type thesauri while enhancing our recall of the correct type information. The second way is to apply pruning techniques to a given EBD calculation between two compared attributes. This way, geographic types represented by a very small number of instances are not considered in the final EBD calculation. The idea behind this is to correlate high EBD scores with high frequencies of instances across all present GTs. Second, we will work on supporting gazetteers, like ADL, that organize their feature type thesauri in an ontological fashion. Third, we plan on extending GSim such that geo-ontologies can be just as easily compared for similarity as geodatabases. To this end, we also plan on adapting suitable algorithms for comparing ontologies, such as structural and neighborhood matching techniques. We would then integrate them into a more sophisticated GT matching algorithm. Fourth, we plan on implementing the algorithm outlined in Section 4.3 to

overcome inadequate attribute mappings produced by NGT matching by using the available GT information from instances. Finally, we plan on expanding our study of attribute weighting to formalize and measure its contribution under a variety of experimental conditions, and in various domains.

References

[1] L.A.P. Paes Leme, M.A. Casanova, K.K. Breitman, A.L. Furtado, Instance-based OWL schema matching, in: ICEIS, 2009, pp. 14–26.
[2] D.F. Brauner, C. Intrator, J.C. Freitas, M.A. Casanova, An instance-based approach for matching export schemas of geographical database web services, in: GeoInfo, 2007, pp. 109–120.
[3] D.F. Brauner, M.A. Casanova, R.L. Milidiú, Towards gazetteer integration through an instance-based thesauri mapping approach, in: GeoInfo, 2006, pp. 189–198.
[4] I.F. Cruz, W. Sunna, N. Makar, S. Bathala, A visual tool for ontology alignment to enable geospatial interoperability, J. Vis. Lang. Comput. 18 (3) (2007) 230–254.
[5] E. Ralun, P.A. Bernstein, A survey of approaches to automatic schema matching, VLDB J. 10 (2001) 334–350.
[6] B.T. Dai, N. Koudas, D. Srivastava, A.K.H. Tung, S. Venkatasubramanian, Validating multi-column schema matchings by type, in: 24th International Conference on Data Engineering (ICDE), 2008, pp. 120–129.
[7] P. Bohannon, E. Elnahrawy, W. Fan, M. Flaster, Putting context into schema matching, in: VLDB, 2006, pp. 307–318.
[8] R.H. Warren, F.W. Tompa, Multi-column substring matching for database schema translation, in Proc, in: VLDB, 2006, pp. 331–342.
[9] W.S. Li, C. Clifon, Semint: a tool for identifying attribute correspondence in heterogeneous databases using neural networks, Data Knowl. Eng. 33 (1) (2000) 49–84.
[10] J. Berlin, A. Motro, Autoplex: Automated discovery of instance for virtual databases, in: Proc CoopIS, 2001, pp. 108–122.
[11] D.W. Embley, L. Xu, Y. Ding, Automatic direct and indirect schema mapping: experiences and lessons learned, SIGMOD Rec. 33 (4) (2004) 14–19.
[12] C. Zhou, D. Frankowski, P.J. Ludford, S. Shekhar, L.G. Terveen, Discovering personal gazetteers: an interactive clustering approach, in: GIS, 2004, pp. 266–273.
[13] S. Newsam, Y. Yang, Integrating gazetteers and remote sensed imagery, in: GIS, 2008, p. 26.
[14] B. Pouliquen, R. Steinberger, C. Ignat, T. De Groeve, Geographical information recognition and visualization in texts written in various languages, in: SAC, 2004, pp. 1051–1058.
[15] C. Zhou, D. Frankowski, P.J. Ludford, S. Shekhar, L.G. Terveen, Discovering personally meaningful places: an interactive clustering approach, ACM Trans. Inf. Syst. 25 (3) (2007).
[16] D. Joshi, J. Luo, Inferring generic activities and events from image content and bags of geo-tags, in: CIVR, 2008, pp. 37–46.
[17] E. Wilde, M. Kofahl, The locative web, in: LocWeb, 2008, pp. 1–8.
[18] http://www.geonames.org, July 2010.
[19] S. Auer, J. Lehmann, S. Hellmann, LinkedGeoData—adding a spatial dimension to the web of data, in: International Semantic Web Conference, 2009, pp. 731–746.
[20] M. Wilkes, K. Janowicz, A graph-based alignment approach to similarity between climbing routes, in: First International Workshop on Information Semantics and its Implications for Geographic Analysis (ISGA '08) at GIScience, 2008.
[21] http://www.ertico.com/en/about_ertico/links/gdf_-_geographic_data_files.htm, May 2010.
[22] C. Rinner, Multi-criteria evaluation in support of emergency response decisionmaking, in: Joint CIG/ISPRS Conference on Geomatics for Disaster and Risk Management, 2007.
[23] R. Cilibrasi, P.M.B. Vitányi, The Google Similarity Distance, CoRR abs/cs/0412098:, 2004.
[24] K. Janowicz, M. Wilkes, SIM-DLA: a novel semantic similarity measure for description logics reducing inter-concept to inter-instance similarity, in: ESWC, 2009, pp. 353–367.
[25] R. Albertoni, M. De Martino, Semantic similarity of ontology instances tailored on the application context, in: OTM Conferences (1), 2006, pp. 1020–1038.
[26] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization. Advances in neural information processing systems, in: Proceedings of the 2000, Conference, MIT Press, Boston, 2000, pp. 556–562.
[27] GSL Team, 2007, Section 13.4 Singular Value Decomposition, GNU Scientific Library, Reference Manual, http://www.gnu.org/software/gsl/manual/html_node/Singular-Value-Decomposition.html.
[28] T. Finin, Z. Syed, Creating exploiting a web of semantic data, in: ICAART, 1, 2010, pp. 7–18.
[29] C. Fink, C.D. Piatko, J. Mayfield, D. Chou, T. Finin, J. Martineau, The geolocation of web logs from textual clues, in: CSE, 4, 2009, pp. 1088–1092.
[30] J. Martineau, T. Finin, A. Joshi, S. Patel, Improving binary classification on text problems using differential word features, in: CIKM, 2009, pp. 2019–2024.
[31] M.A. Rodríguez, M.J. Egenhofer, Determining semantic similarity among entity classes from different ontologies, IEEE Trans. Knowl. Data Eng. 15 (2) (2003) 442–456.
[32] http://www.alexandria.ucsb.edu/gazetteer, June 2010.
[33] http://code.google.com/apis/maps/documentation/geocoding/v2/index.html#GeocodingAccuracy, July 2010.
[34] O Ahlqvist, A. Shortridge, Characterizing land cover structure with semantic variograms, in: 12th International Symposium on Spatial Data Handling, 2006, pp. 401–415.
[35] A. Karalopoulos, M. Kokla, M. Kavouras, Comparing representations of geographic knowledge expressed as conceptual graphs, in: GeoS, 2005, pp. 1–14.
[36] W. Kuhn, Geospatial semantics: why, of what, and how? J. Data Semantics III (2005) 1–24.
[37] D. Lin, An information-theoretic definition of similarity, in: ICML, 1998, pp. 296–304.
[38] J. Euzenat, P. Shvaiko, Ontology Matching, Springer-Verlag, Berlin/Heidelberg/New York, 1998.
[39] F.T. Fonseca, M.J. Egenhofer, P. Agouris, G. Câmara, Using ontologies for integrated geographic information systems, in: T. GIS 6(3), 2002, pp. 231–257.
[40] E. Klien, et al., An architecture for ontology-based discovery and retrieval of geographic information, in: 7th Conference on Geographic Information Science AGILE, 2004, pp. 179–188.
[41] A. Tversky, Features of similarity, Psychological Review 84 (4) (1977) 327–352.
[42] R.M. Nosofsky, Attention, similarity, and the identification-categorization relationship, Journal of Experimental Psychology: General 115 (1986) 39–57.
[43] B Bouchon-Meunier, M. Rifqi, S. Bothorel, Towards general measures of comparison of objects, Fuzzy Sets and Systems 84 (1996) 143–153.
[44] K. Janowicz, M. Raubal, A. Schwering, W. Kuhn, Semantic similarity measurement and geospatial applications, in: T. GIS 12(6), 2008, pp. 651–659.
[45] J Li, G. Ruhe, Software effort estimation by analogy using attribute selection based on rough set analysis, International Journal of Software Engineering and Knowledge Engineering 18 (1) (2008) 1–23.
[46] W. Su, J. Wang, Q. Huang, F.H. Lochovsky, Query result ranking over E-commerce web databases, in: CIKM, 2006, pp. 575–584.
[47] I.F. Cruz, F.P. Antonelli, C. Stroe, AgreementMaker: efficient matching for large real-world schemas and ontologies, in: PVLDB 2(2), 2009, pp. 1586–1589.
[48] C. Fellbaum (Ed.), WordNet: An Electronic Lexical Database, The MIT Press, Cambridge, MA, 1998.
[49] P.-N. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining, Pearson Education, Inc., New York, 2006.
[50] Y. Manolopoulos, Binomial coefficient computation: recursion or iteration? SIGCSE Bulletin 34 (4) (2002) 65–67.