Combining Ontology Alignment Metrics Using the Data Mining Techniques

Babak Bagheri Hariri and **Hassan Sayyadi** and **Hassan Abolhassani** [1]

Abstract. Several metrics have been proposed for recognition of relationships between elements of two Ontologies. Many of these methods select a number of such metrics and combine them to extract existing mappings. In this article, we present a method for selection of more effective metrics – based on data mining techniques. Furthermore, by having a set of metrics, we suggest a data-mining-like means for combining them into a better ontology alignment.

1 Introduction

Ontology Alignment is an essential tool in semantic web to overcome heterogeneity of data, which is an integral attribute of web. In [2] Ontology Alignment is defined as a set of correspondences between two or more ontologies. These correspondences are expressed as mappings, in which *Mapping* is a formal expression that states the semantic relation between two entities belonging to different ontologies.

There have been several proposals for drawing mappings in Ontology Alignment. Many of them define some metrics to measure similarity or distance of entities which form one possible basis for finding existing mappings [4].

A *Similarity* $\sigma : O \times O \rightarrow \mathbf{R}$ is a function from a pair of entities to a real number expressing the similarity between two objects such that:

$$\begin{array}{ll} \forall x, y \in O, \sigma(x, y) \geq 0 & (positiveness) \\ \forall x \in O, \forall y, z \in O, \sigma(x, x) \geq \sigma(y, z) & (maximality) \\ \forall x, y \in O, \sigma(x, y) = \sigma(y, x) & (symmetry) \end{array}$$

For example one popular metric is *Edit Distance* [7] measuring String Similarity between entities under consideration. Also there exist metrics named *Resnik Similarity* [16] and *Upward Cotopic* distance [8] that measures Linguistic and Structural similarities and distances, correspondingly. Most of the Ontology Alignment methods select or define some metrics and combine them to recognize existing mappings. In the extraction phase, in most of these methods, couples having Compound Similarity higher than a predefined threshold – after applying a number of constraints – are selected as final mappings. A number of such methods are explained in the next section. [4] contains a more complete list.

In this article a method based on data mining techniques is proposed to select more appropriate metrics among an initial set and then calculate amount of *Compound Similarity* for them. In fact this article proposes a *Super Metric* to find mappings, opposed to those proposing a new metric. It can be adjusted in future when more effective new metrics are introduced such that we achieve a better combination of metrics for the alignment problem.

It should be mentioned that this article only deals with the problem of selecting most effective metrics and combining them. The succeeding phase, mapping extraction, is left to the user. For that purpose extraction methods like what reported in [4] can be used.

The rest of this article is organized as follows. In section 2, a review of related works in evaluation of existing methods together with popular methods for combining several metrics and computing compound similarity from them, are given. Section 3 reports our proposed method. In section 4 an example of applying this method is shown. Finally in section 5, conclusion and discusses on its advantages and disadvantages and future works are explained.

2 Existing Works

As noted before, the proposed method tries to select more appropriate metrics among a set of them. Therefore we need to review two groups of related works in this section. The first one is the existing methods for ontology alignment evaluation. The second one consists of methods for aggregating results attained from existing metrics to define a new compound similarity metric.

2.1 Ontology Alignment Evaluation

In this section existing methods for evaluation of ontology alignment are explained.

2.1.1 Precision and Recall

Many of the algorithms and articles in Ontology Alignment context uses *Precision* and *Recall* or their harmonic mean, referred to as *F-Measure*, to evaluate the performance of a method [14]. Also in some articles, they are used to evaluate alignment metrics[13]. In such methods after aggregation of results attained from different metrics, and extraction of mappings – based on one of the methods mentioned in [4] – the resulting mappings are compared with actual results. Precision and Recall values are computed as below:

$$precision = \frac{\text{True Founded Mappings}}{\text{All Founded Mappings}} \quad (1)$$

[1] Semantic Web Laboratory, Computer Engineering Department, Sharif University of Technology, Tehran, Iran, email: {hariri,sayyadi}@ce.sharif.edu, abolhassani@sharif.edu

$$recall = \frac{\text{True Founded Mappings}}{\text{All Existing Mappings}} \quad (2)$$

2.1.2 Accuracy

This value is also proposed for evaluation of automatic ontology alignment [9]. This quality metric is based upon user effort needed to transform a match result obtained automatically into the intended result. This value can be represented as a combination of Precision and Recall:

$$Accuracy = Recall \times (2 - \frac{1}{Precision}) \quad (3)$$

2.2 Methods for Calculation of Compound Similarity

In this section works related to calculation of compound similarity from a set of metrics is explained.

2.2.1 Minkowski Distance

Let O a set of object which can be analyzed in n dimensions, the *Minkowski* distance between two such objects is:

$$\forall x, x' \in O, \delta(x, x') = \frac{\sum_{i=1}^{n} \delta(x_i, x_i')^p}{p} \quad (4)$$

in which $\delta(x_i, x_i')$ is the dissimilarity of the pair of objects along the i^{th} dimension.

2.2.2 Weighted Sum

Let O a set of objects which can be analyzed in n dimensions, the weighted sum (or weighted average)between two such objects is:

$$\forall x, x' \in O, \delta(x, x') = \sum_{i=1}^{n} w_i \times \delta(x_i, x_i') \quad (5)$$

in which $\delta(x_i, x_i')$ is the dissimilarity of the pair of objects along the i^{th} dimension and w_i is the weight of dimension i.

2.2.3 Weighted Product

Let O a set of objects which can be analyzed in n dimensions, the weighted product between two such objects is:

$$\forall x, x' \in O, \delta(x, x') = \prod_{i=1}^{n} \delta(x_i, x_i')^{\lambda_i} \quad (6)$$

in which $\delta(x_i, x_i')$ is the dissimilarity of the pair of objects along the i^{th} dimension and λ_i is the weight of dimension i.

2.2.4 Learning Based Methods

In this group of methods, using machine learning techniques, some coefficients for weighted combination of metrics are attained [3]. Optimal weights in such methods are calculated by defining or proposing some specific measures and applying them on a series of test sets - an ontology couple with actual mappings between their elements.

3 Proposed Method

To use Precision, Recall, F-measure and Accuracy for metrics evaluation, it is needed to do mapping extraction. It depends on the definition of a *Threshold* value and the approach for extracting as well as on some defined constraints. Such dependencies results in in-appropriateness of current evaluation methods, although methods like what defines in [13] used to compare quality of metrics. We propose a new method for evaluation of metrics and creating a compound metric from some them featuring independent of mapping extraction phase. Like other learning based methods, it needs an initial training phase. In this phase a train set - an ontology pair with actual mappings in them - is given to the algorithm. Also a number of metrics with their associated category is considered. Categories are for example *String Metric*, *Linguistic Metric*, *Structural Metric* and so on. Proposed algorithm selects one metric from each category. So to enforce the algorithm to use a specific metric we can define a new category and introduce the metric as the only member of it.

The goal of defining categories and assignment of metrics to them is that in combining metrics usually String and Linguistic based metrics are more influential than others and therefore if we don't use such categorization, and apply the algorithm on a set of un-categorized metrics, most of the selected ones are linguistic which results in lower performance and flexibility of algorithm on different inputs. After categorization and defining metrics, our algorithm selects the best metric from each category and proposes an appropriate method to aggregate them.

3.1 Learning Phase

In our algorithm, selection of appropriate metrics and aggregation of them are done based on *Data Mining*.

3.1.1 Reduction to a Data Mining Problem

To reduce this problem to a data mining problem, for a pair of Ontologies a table is created with rows showing comparison of an entity from first ontology to an entity from the second one. For each metric under consideration a column is created in such a table with values showing normalized metric value for the pair of entities. An additional column with true or false values shows the existence of actual mapping between the two entities is also considered.

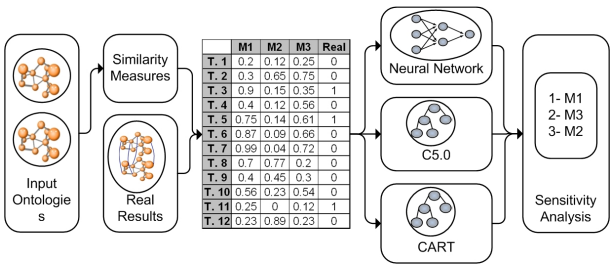

Figure 1. Proposed evaluation technique in detail

One table is created for a pair of Ontologies in the training set. Then all of such tables are aggregated in a single table. In this final table the column representing actual mapping value between a pair of entities is considered as target variable and the rest of columns are predictors. The problem now is a typical data mining and then we can apply classic data mining techniques to solve it. Fig. 1 shows the process.

Two popular methods of Data Mining which are used in this article are explained briefly in what follows.

A **Neural Network** consists of a layered, feed forward, completely connected network of artificial neurons, or nodes. The neural network is composed of two or more layers, although most networks consist of three layers: an input layer, a hidden layer, and an output layer. There may be more than one hidden layer, although most networks contain only one, which is sufficient for most purposes. Fig. 2 shows a Neural Network with three layers. Each connection between nodes has a weight (e.g., W_{1A}) associated with it. At initialization, the weights are randomly assigned to values between 0 and 1.

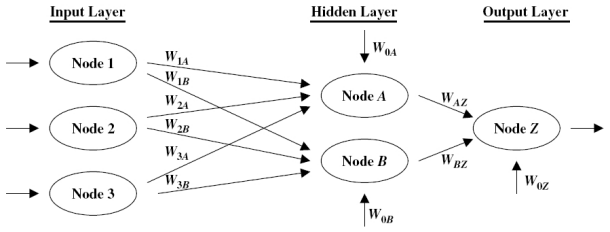

Figure 2. Neural Networks Model

Neural Network works as follows. First, a combination function (usually summation, $\sum$) produces a linear combination of the node inputs and the connection weights into a single scalar value. Thus, for a given node j:

$$net_j = \sum_i W_{ij}x_{ij} = W_{0j}x_{0j} + W_{1j}x_{1j} + \cdots + W_{Ij}x_{Ij} \quad (7)$$

where x_{ij} represents the i_{th} input to node j, W_{ij} represents the weight associated with the i_{th} input to node j and there are $I+1$ inputs to node j. Note that $x_1, x_2, \cdots, x_I$ represent inputs from upstream nodes, while x_0 represents a constant input, analogous to the constant factor in regression models, which by convention uniquely takes the value $x_{0j} = 1$. Thus, each hidden layer or output layer node j contains an *extra* input equal to a particular weight $W_{0j}x_{0j} = W_{0j}$, such as W_{0B} for node B.

Decision Tree is one attractive decision method involves a collection of decision nodes, connected by branches, extending downward from the root node until terminating in leaf nodes. When no further splits can be made, the decision tree algorithm stops growing new nodes. Beginning at the root node, which by convention is placed at the top of the decision tree diagram, attributes are tested at the decision nodes, with each possible outcome resulting in a branch. Two most famous decision trees are CART [ref] and C.5 [ref]. The decision tree construction algorithms grows the tree by conducting for each decision node, an exhaustive search of all available variables and all possible splitting values, selecting the optimal split according to the specific criteria.

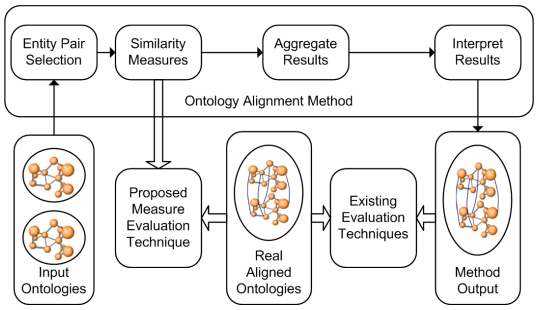

Figure 3. Proposed evaluation technique versus existing techniques

3.1.2 Selection of Appropriate Metrics

Based on the table explained in the previous section the selection of more appropriate metrics is reduced to a data mining problem. In what following, we analysis this problem with Neural Networks as well as $CART$[2] and $C_{5.0}$ decision tress[6]. As mentioned before, columns of the table corresponding to values of metrics are considered as Predictors and the actual mapping value is the target variable. Fig. 1 shows the process while Fig. 3 compares our method with other existing ones.

The aim is to find metrics having most influence in prediction of the target variable using Data Mining Models:

Neural Networks: *Sensitivity Analysis* for any problem is applied after a model has been constructed. With varying the values of input variables in the acceptable interval, the output variation is measured. With the interpretation of the output variation it is possible to recognize most influential input variable. Such analysis is also applied in Neural Networks models. To do it, at first the average value for each input variable is given to the model and the output of the model is measured. Then Sensitivity Analysis for each variable is done separately. To do this, the values of all variables except one in consideration are kept constant (their average value) and the model's response for minimum and maximum values of the variable in consideration are calculated. This process is repeated for all variables and then the variables with higher influence on variance of output are selected as most influential variables. For our problem it means that the metric having most variation on output during analysis is the most important metric.

Decision Trees: For the construction of decision trees, first root node is created and then in each iteration of the algorithm, a node is added. This process is repeated until the expansion of the tree is not possible anymore considering some predefined constraints. Since in each node of the tree a single variable is tested and in algorithms like $C_{5.0}$ [6] selection of a variable as next node in the tree is done based on information theory concepts, then in each repetition a variable with higher influence is selected among candidates. Therefore as a node is more near to the root, its corresponding variable has higher

[2] Classification And Regression Trees

influence on the value of target variable. Hence from the constructed decision tree it is possible to say that the metric in the root node has the highest influence and metrics in nodes with lower depth has more values.

3.1.3 Toward an Appropriate Method for Calculation of the Compound Metric

According to the results from step 3-1 the problem is reduced to a Data Mining problem with the goal of finding an algorithm to compute target variable based on the predictor variables. Saying in other words it is to calculate of mappings based on the values of metrics and category of input ontologies. In the Data Mining area several solutions have been proposed for this kind of problems. Among them we can refer to *CART* and $C_{5.0}$ [6] decision trees, A Priori for Association Rules generation [1] and Neural Networks [6]. Based on experiments and initial results among these methods only Neural Networks has showed acceptable results for mappings. In Decision Trees and Association Rules it is possible to judge that there is no relationships between two entities which is not suitable for the alignment. Neural Networks, on the other hands, have similar behavior with popular Alignment methods and they calculate Compound Similarity in the form of Weighted Sum with the weights is adjusted during learning. Following the detail of the algorithm is given.

After selection of best metrics from each category, with the help of another neural network a combined metric is constructed. Similar to the evaluation method, a table is constructed. As before, columns are the values selected metrics and an additional column records the target variable (0 or 1) showing the existence of a mapping between two entities. Now having such training samples a neural network is built. It is like a combined metric from the selected metrics which can be used as a new metric for the extraction phase.

4 An Example of Using the Proposed Method

In this section with the help of a simple example the proposed method is explained in detail. To simplify the problem only String Based similarity metrics are considered. In this experiment we first select more appropriate metrics and the build a compound similarity metric from them. For the initial set of metrics we consider following metrics: the *Levenshtein* distance [7] which used the *Edit Distance* to match two strings, the *Needleman-Wunsch* distance[11], which assigns a different cost on the edit operations, the *Smith-Waterman* [12], which additionally uses an alphabet mapping to costs, the *Monge-Elkan* [10], which uses variable costs depending on the substring gaps between the words , the *Stoilos* similarity [13] which try to modify existing approaches for entities of an ontology, *Jaro-Winkler* similarity [5, 15], which counts the common characters between two strings even if they are misplaced by a "short" distance, and the *Sub-string* distance [4]which searches for the largest common substring. As explained in section 3 a table is constructed with the values of predictors and a target variable. EON_{2004} data set is used as the training set which is explained below:

1. **Group $\mathbf{1}_{xx}$:** We only use test 103 from this group. This test compares the ontology with its generalization in OWL Lite. Names of entities in this group is remaining without any changing and cause this group not to be a suitable data set for evaluation of string metrics.
2. **Group $\mathbf{2}_{xx}$:** In this test the reference ontology is compared with a modified one. Tests 204, 205, 221, 223 and 224 are used from this group. Modifications involved naming conventions like replacing the labels with their synonyms as well as modifications in the hierarchy.
3. **Group $\mathbf{3}_{xx}$:** The reference ontology is compared with four real-life ontologies for bibliographic references found on the web and left unchanged. We use tests 302, 303 and 304 from this group. This is the only group which contains real tests and may be the best one for evaluation of metrics.
4. **All:** To have a larger test set, we merged all the data from described data sets. The table of results for each data set, are concatenated to each other and form a larger data set.

Each data set contains some entities. Name of each entity is compared with the names of all other entities. Each comparison of two strings is assigned a similarity degree. Every entry for a string contains a key which is purposed for the identification of the correctness of a pair. After collecting output for each metric, we evaluate them for each data set as it is described in Sect. 2.

Test	Lvs.	Ndl.	Smt.	Jrw.	Mng.	Stl.	Sbs.
103	0.30	0.17	0.14	0.01	0.00	0.04	0.20
204	0.20	0.15	0.23	0.09	0.07	0.08	0.18
205	0.46	0.020	0.13	0.080	0.05	0.02	0.04
221	0.30	0.17	0.16	0.01	0.01	0.02	0.22
222	0.27	0.16	0.08	0.01	0.01	0.05	0.17
223	0.21	0.00	0.03	0.03	0.01	0.03	0.10
224	0.29	0.16	0.15	0.01	0.01	0.04	0.28
302	0.59	0.31	0.07	0.10	0.01	0.05	0.06
303	0.26	0.03	0.30	0.04	0.03	0.04	0.15
304	0.34	0.12	0.02	0.01	0.01	0.06	0.17
All	0.36	0.18	0.25	0.12	0.01	0.02	0.05
Lvs.=Levensteine, Ndl.=Needleman-Wunsch, Smt.=Smith-Waterman,Jrw=Jaro-Winkler, Mng.=Mong-Elkan, Stl.=Stolios, Sub.=SubString							

Table 1. Related importance of metrics using Neural Networks

Table 1 displays results of applying similarity analysis on each test set using Clementine 3 tool. In this table each row shows the relative importance of metrics used in the corresponding data set. Fig 4 shows the results after normalization. As it is clear from the results, Levenshtein similarity is the most important one in predicting the relation of entities. Fig. 3 shows the relative importance of metrics in each data set, by normalizing the results of Table 1.

As explained in the section 3 besides Sensitivity Analysis, decision trees models are also used to confirm the results. So we have other evaluations based on decision trees. In Table 2 we compare results of these techniques. As it is clear, all of three tests agree about importance of Levenshtein similarity on the test set. Neural Network chooses *Levenshtein* while $C_{5.0}$ and *CART* select it as second suitable metric. This data set shows *Jaro-Winkler* as second suitable string similarity for ontology alignment. According to the presented algorithm and considering the fact that only one category is introduced as input, only Levenshtein is selected. In a more real situa-

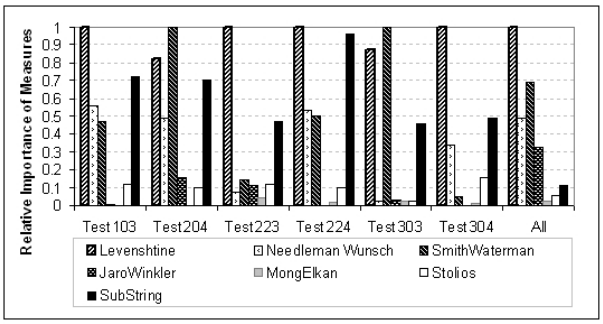

Figure 4. Evaluation of string metrics using Neural Networks

Neural Network	CART	C5.0
Levenshtein	Jaro-Winkler	Needleman-Wunsch
SubString	Levenshtein	Levenshtein
Smith-Waterman	Monge-Elkan	Jaro-Winkler
Needleman-Smith	Stoilos	Monge-Elkan

Table 2. Most 4 important metrics

tion the above steps are done for each category and one metric from each category is selected. To explain the next step, *Levenshtein* and *Jaro-Winkler* are selected (from two imaginary categories). The aim is to select appropriate weights for them (in a combined metric) . A neural network with 3 layer is constructed by *Clementine* [3] tool. In the combined metric as expected *Levenshtein* has more weight than *Jaro-Winkler*.

5 Conclusions and Future Works

In this paper the main problem of existing evaluation frameworks, which we name as indirect evaluation, for ontology alignment has been discussed and it is claimed that the results of such evaluations are influenced by underlying framework itself and need correct threshold for each metric. To overcome these problems, a new evaluation framework which is based on data mining techniques is proposed, featuring independence of alignment framework.

Two advantages of the evaluation method are the metrics do not need threshold in this method and the uniform treatment of Similarity and Distance metrics so that we don't need to differentiate and process them separately. This is because in Data Mining evaluation methods such as Sensitivity Analysis there is no difference between a variable and a linear form of it. The main advantage of the compound similarity method is its independency to the specific metrics and therefore it is possible to extract mapping by several metrics. The method can be improved when new metrics are introduced in such cases it is only needed to add some new columns and do learning to adjust weights. Also with the experiments and researches in evaluations and selection of metrics, many researches have concluded that for different categories of ontologies metrics has varying values. Therefore most of the researchers have emphasized on clustering and application of metrics for clusters as their future works. Another advantage of this method is that we can add cluster value as a new column to influence its importance for combination of metrics. The main disadvantage of this method is the need for a suitable training set. If the training set is not constructed carefully with appropriate size the results are not acceptable. Also if we want to add category of ontologies as main factor during learning a larger set of training is needed.

Alongside the future works, a broader framework will be studied aiming the leverage of a vaster range of metrics, and enabled for categorized-learning - based on above-explained method. We will scrutinize further the role of some other data mining methods such as association rules and clustering in both the compound similarity calculation phase and mapping extraction phase.

[3] http://www.spss.com/clementine

REFERENCES

[1] R. Agrawal, T. Imielinski, and A. Swami, 'Mining association rules between sets of items in large databases', in *ACM SIGMOD Intl. Conf. Management of Data*, (May 1993).

[2] Paolo Bouquet, Marc Ehrig, Jerome Euzenat, and eds., 'Specification of a Common Framework for Characterizing Alignment', Technical Report deliverable 2.2.1, Knowledge Web, Statistical Research Division, Room 3000-4, Bureau of the Census, Washington, DC, 20233-9100 USA, (February).

[3] Anhai Doan, Pedro Domingos, and Alon Halevy, 'Learning to match the schemas of data sources: A multistrategy approach', *Machine Learning*, **50**(3), 279–301, (2003).

[4] J. Euzenat, T. Le Bach, J. Barrasa, P. Bouquet, and eds., 'State of the Art on Ontology Alignment', Technical Report deliverable 2.2.3, Knowledge Web, Statistical Research Division, Room 3000-4, Bureau of the Census, Washington, DC, 20233-9100 USA, (August).

[5] M. Jaro, 'Probabilistic Linkage of Large Public Health Data Files', *Molecular Biology*, **14**, 491–498, (1995).

[6] Daniel T. Larose, *Discovering Knowledge In Data*, John Wiley and Sons, New Jersey, USA, 2005.

[7] Vladimir Levenshtein, 'Binary Codes Capable of Correcting Deletions, Insertions and Reversals', *Soviet Physics-Doklady*, **10**, 707–710, (August 1966).

[8] Alexander Maedche and Valentin Zacharias, 'Clustering ontologybased metadata in the semantic web', in *Proceedings of the 13th ECML and 6th PKDD*, (2002).

[9] S. Melnik, H. Garcia-Molina, and E. Rahm, 'A versatile graph matching algorithm', in *Proceedings of ICDE*, (2002).

[10] Alvaro E. Monge and Charles P. Elkan, 'The Field-Matching Problem: Algorithm and Applications', in *Proceedings of the second international Conference on Knowledge Discovery and Data Mining*, (1996).

[11] S.B. Needleman and C.D. Wunsch, 'A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of two Proteins', *Molecular Biology*, **48**, (1970).

[12] T.F. Smith and M.S. Waterman, 'Identification of Common Molecular Subsequences', *Molecular Biology*, **147**, (1981).

[13] Giorgos Stoilos, Giorgos Stamou, and Stefanos Kollias, 'A String Metric for Ontology Alignment', in *Proceedings of the ninth IEEE International Symposium on Wearable Computers*, pp. 624–237, (October 2005).

[14] Y. Sure, O. Corcho, J. Euzenat, and T. Hughes, eds. *Proceedings of the 3rd Evaluation of Ontology-based tools*, 2004.

[15] William E. Winkler, 'The State Record Linkage and Current Research Problems', Technical report, U. S. Bureau of the Census, Statistical Research Division, Room 3000-4, Bureau of the Census, Washington, DC, 20233-9100 USA, (1999).

[16] J. Zhong, H. Zhu, Y. Li, and Y. Yu, 'Using information content to evaluate semantic similarity in a taxonomy', in *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, (1995).