Improving ontology matchers utilizing linguistic ontologies: an information retrieval approach

Frederik C. Schadd Nico Roos

Maastricht University, P.O. Box 616, 6200 MD Maastricht, The Netherlands

Abstract

Matching ontologies is a crucial process when facilitating system interoperability and information exchange. A reoccurring problem in this process is that names can be ambiguous, yielding uncertainty to whether entities of two heterogeneous ontologies are actually related. Linguistic ontologies provide a clear structure of meanings, rather than names, allowing the quantification of the relatedness of any two given meanings. We propose an approach for the automatic allocation of correct meanings within a linguistic ontology through the use of virtual documents and information retrieval techniques. The benefits of this approach are tested and established using a data set from the Ontology Alignment Evaluation Initiative (OAEI) competition, while further improvements are revealed using a benchmark data set from the same competition.

1 Introduction

Sharing and reusing knowledge is an important aspect in modern information systems. Since multiple decades, researchers have been investigating methods that facilitate knowledge sharing in the corporate domain, allowing for instance the integration of external data into a company's own knowledge system. Ontologies are at the center of this research, allowing the explicit definition of a knowledge domain. With the steady development of ontology languages, such as the current OWL language [10], knowledge domains can be modeled with an increasing amount of detail. Due to the Semantic Web vision [1], information sources on the future World Wide Web will store information in a machine readable way, allowing autonomous agents to collect and interpret information automatically. Just as in current knowledge systems, each information source on the World Wide Web will store its content in a structured way with a publicly available ontology describing the semantics of stored information. Such ontologies are generally developed independently, resulting in many different ontologies describing the same domain. Thus, autonomous agents roaming the Semantic Web need to be able to integrate knowledge of heterogenous sources into their own representation of a specific domain.

Matching heterogenous ontologies has traditionally been done either manually or using semi-automatic tools. However, many research groups have focused their attention on automatic matching approaches, such as Anchor-PROMPT [12] or ASMOV [8]. These tools can also be developed specifically for a certain domain of interest, for example the library domain [7].

Commonly, ontology matching tools combine a variety of similarity measures using advanced aggregation techniques. The focus of this article lies on similarities measures based on linguistic ontologies. More specifically, we investigate the automatic identification of corresponding entries in these ontologies through the use of virtual documents and information retrieval techniques, such that the semantic relatedness of any two ontology entities can be accurately specified. The remainder of this paper is structured as follows. Section 2 will provide the reader with necessary background information while section 3 will explain the details of our proposed approach. Results on two data sets from the OAEI 2010 competition will be presented in section 4. Finally, the conclusion of this paper and remarks on future research are given in section 5.

2 Background

2.1 Semantic Heterogeneity between Ontologies

Ontologies form the basis of present knowledge representations, ranging from data bases to the semantic web. An ontology is an explicit formal specification of a conceptualization. It usually defines concepts (classes), relations between concepts and properties of concepts. More sophisticated ontologies may also define axioms.

Ontologies of the same domain are often developed by groups with different backgrounds and being designed for different purposes. As a result, these ontologies may differ significantly giving rise to the interoperability problem. In essence, because of the differences in representation, information and knowledge exchange is prohibitively difficult. To solve this problem, a mapping between these different ontologies of the same domain is required, identifying all correspondences between the two ontologies. Figure 1 displays an example mapping between two ontologies of the domain of scientific documents.

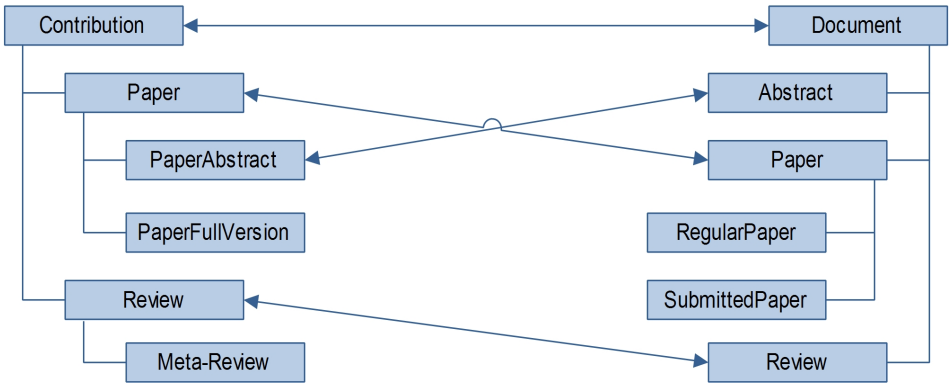

Figure 1: Example of a possible mapping between two ontologies describing scientific documents.

A mapping between two ontologies has to address different types of heterogeneities. There are many classifications of the types of heterogeneities that have to be addressed. The two main categories that are often distinguished are syntactic heterogeneity and semantic heterogeneity. Syntactic heterogeneity concerns the different representations of information while semantic heterogeneity concerns the intended meaning of the described information. We assume that syntactic heterogeneity is not an issue due to the use of standard ontology languages, such as OWL, essentially alleviating the problem. Hence, our focus lies on semantic heterogeneities between ontologies. These semantic heterogeneities can be refined into several categories, such as terminological, conceptual and semiotic heterogeneity. Another classification is a refinement into naming conflicts, structural conflicts and data conflicts. These different refinements are by themselves an illustration of semantic heterogeneity.

In this paper we will focus on naming conflicts also called terminological heterogeneity. A naming conflict refers to the use of different names (terms) when referring to the same entity in different ontologies. The difference in names can be the result of different natural language (paper vs artikel), different categories (paper vs memo) and synonyms (paper vs article). It also refers to the use of the same or similar names when representing completely different types of information. Utilizing linguistic ontologies, we will specifically focus on the last two causes of terminological heterogeneity.

2.2 Linguistic Ontologies

A linguistic ontology, also known as a lexical database, is an ontology that is specially built to capture the semantics of a very large set of terms, most commonly all words in a specific language. Since terms can have duplicate meanings, such ontologies usually capture structured meanings rather than structured terms. These ontologies can vary in their complexity. A very simple linguistic ontology, for instance, captures a meaning only by a small written explanation, thus making it equivalent to a dictionary. More advanced ontologies also capture interesting relations that can hold between meanings like synonymity, hyponymy and holonymy, allowing the identification of semantically related terms. A widely popular ontology, which is

also the base for this research, is WordNet [11]. WordNet is a rich ontology, where over 150.000 words of the English language are organized into synonym-sets (synsets), where each synset represents a unique meaning that the words it contains can represent. For these synsets, relations such as hyponymy and holonymy and their opposites are specified. Another example of such an ontology is Cyc [9], which is intended as a large knowledge base facilitating the support of reasoning techniques in varying domains.

2.3 Virtual Documents

The general definition of a virtual document, coined by C. Watters [16], is any document for which no persistent state exists, such that some or all instances of the given document are generated at run-time. Virtual documents can be constructed in a variety of ways. A simple example would be creating a template for a document and, once required, completing the document using values stored in a database. Another possibility is the interactive construction of a virtual document, where computations and visualizations are inserted based on the user's actions. In the context of this research, a virtual document can be constructed using the information residing in the given ontologies, or external knowledge resources. The ontology matching tool Falcon-AO [13] successfully applies this principle, by constructing a virtual document for each entity in two given ontologies, and computing the direct similarities between each document of both ontologies using Information Retrieval techniques.

3 Proposed Approach

Our proposed approach aims at improving matchers applying linguistic ontologies, in this case WordNet. When applying WordNet for ontology matching, one is presented with the problem of identifying the correct meaning, or synset, for each entity in both ontologies that are to be matched. As an example, the word 'house' might have an intuitive meaning for a human, however there are currently 14 different meanings explicitly defined in WordNet for that word. The goal of our approach is to automatically identify the correct synsets for each entity of an ontology using information retrieval techniques. Given two ontologies O_1 and O_2 that are to be matched, where O_1 contains the set of entities $E_1 = \{e_1^1, e_2^1, ..., e_m^1\}$ and O_2 contains the set of entities $E_2 = \{e_1^2, e_2^2, ..., e_n^2\}$, and where $C(e)$ denotes a collection of synsets representing entity e, the main steps of our approach can be described as follows:

1. For every entity e in E_i, assemble the set $C(e)$ with synsets that might denote the meaning of entity e.
2. For every entity e in E_i, create a virtual document of e, and a virtual document for every synset in $C(e)$.
3. For every entity e in E_i, calculate the document similarities between the virtual document denoting e and the different virtual documents originating from $C(e)$.
4. For every collection $C(e)$, discard all synsets from $C(e)$ that resulted in a low similarity score with the virtual document of e, using some selection procedure.
5. Compute the WordNet similarity for all combinations of $e^1 \in E_1$ and $e^2 \in E_2$ using the processed collections $C(e^1)$ and $C(e^2)$.

The first step of the procedure is fairly straightforward, where all corresponding synsets are collected if the complete name of an entity is present in WordNet and string processing techniques such as word stemming or finding legal sub-strings in the name are applied if the complete name is not present in WordNet. The remaining steps of the list are further explained in subsections 3.1, 3.2 and 3.3. Figure 2 illustrates steps 2 - 4 of our approach:

Once the similarity matrix, meaning all pairwise similarities between the entities of both ontologies, are computed, the final alignment of the matching process can be extracted or the matrix can be combined with similarity matrices stemming from other approaches.

3.1 Virtual Document Similarity

The first step of our approach is to create virtual documents of every entity in both ontologies. Here it is important to collect information in such a way, that the resulting document adequately represents the

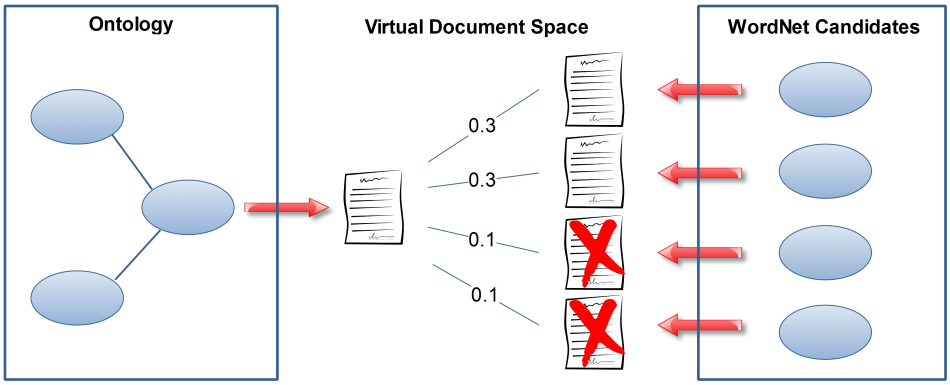

Figure 2: Visualization of the proposed approach.

entity, meaning that every piece of information inside the document is somehow related to the entity. Since ontologies are semantically structured, it is a simple task to accumulate information that is related to a specific entity, especially if an expressive ontology language such as OWL is used. An entity itself can contain useful information such as different labels, properties and comments or other kinds of annotations which can be a written description of the entity. Related entities, such as hypernyms and holonyms, can provide a useful context for the entity in question, hence their information should be included as well.

When an ontology entity is to be compared to any other entity using WordNet, first all the synsets are collected which might denote the entity. The process of virtual document creation is then applied to these synsets and to the entity itself, resulting in a set of documents. On these documents a series of pre-processing techniques is applied, such as stop-word removal and word stemming. The documents can then be transformed into the well known vector space model [14], which is commonly applied in the field of information retrieval [2]. The similarities between the synset documents and the entity document can be computed using the cosine similarity [15].

3.2 Synset Selection

Once the similarities between the entity document and the different synset documents are known, a selection method is applied in order to discard synsets that resulted in a low similarity value. There exist many ways to approach the selection procedure, ranging from very lenient methods, discarding only the very worst synsets, to strict methods, maintaining only the highest scoring synsets. Several selection methods have been investigated for this research, such that both strict and lenient methods are tested. To test lenient selection methods, two methods using the arithmetic and geometric mean as a threshold have been investigated. Two other methods have been tested in order to investigate whether a more strict approach is more suitable. One method consists of subtracting the standard deviation of the similarities from the maximum obtained similarity, and using the resulting value as a threshold. This method has the interesting property that it is more strict when there is a subset of documents that is significantly more similar than the remaining documents, and more lenient when it not as easy to identify the correct correspondences. The other investigated strict method consists of simply selecting the document with the highest similarity value and discarding the rest.

3.3 WordNet Distance

After selecting the most appropriate synsets, using the document similarities, the similarity between two entities can now be computed using their assigned synsets. This presents the problem of determining the similarity between two sets of synsets, where one can assume that within each of these sets resides one synset that represents the true meaning of its corresponding entity. Thus, if one were to compare two sets of synsets, assuming that the originating entities are semantically related, then one can assume that the resulting similarity between the two synsets that both represent the true meaning of their corresponding entities, should be a high value. Hence, inspecting all pairwise similarities between all combinations of synsets between both sets should yield at least one high similarity value. Likewise, when comparing two sets originating from semantically unrelated entities, one can assume that there should be no pairwise similarity

of high value present. Thus, a reasonable way of computing the similarity of two sets of synsets is to compute the maximum similarity over all pairwise combination between the two sets. However, there is a possibility of false positives, where synsets which do not represent the true meaning of their corresponding unrelated entities do have a high similarity value by chance. A strict enough selection method should discard these unrelated synsets, reducing the possibility of such false positives as much as possible.

There exist several ways to compute the semantic similarity within WordNet [3] that can be applied, however finding the optimal measure is beyond the scope of this research. For this research, a similarity measure with similar properties as the Leacock-Chodorow similarity [3] has been applied. Given two synsets s_1 and s_2 and the distance function $dist(s_1, s_2)$, which determines the distance of two synsets inside the taxonomy, and the overall depth D of the taxonomy, the similarity of these two synsets is computed as follows:

$$sim(s_1, s_2) = \begin{cases} \frac{D - dist(s_1, s_2)}{D} & \text{if } dist(s_1, s_2) \leq D \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

This measure is similar to the Leacock-Chodorow similarity in the sense that it relates the taxonomic distance of two synsets to the depth of the taxonomy. However, in order to ensure that the resulting similarity values fall within the interval of $[0, 1]$ and thus can be integrated into larger matching systems, the log-scaling has been omitted in favor of a linear scale.

4 Experiments

In this section, the experiments that have been performed to test the effectiveness of our approach will be presented. Subsection 4.1 discusses the improvements on the conference test set, originating from Ontology Alignment Evaluation Initiative 2010 (OAEI 2010) competition [5]. Subsection 4.2 establishes the overall performance under different conditions of our approach using the benchmark test set from the OAEI 2010 competition. To complement the similarities calculated using the WordNet distance, they are combined with the Jaro string similarity [4] of the names from the corresponding entities. For the sake of brevity, we will refer to our framework as MaasMatch (MM) with the applied selection procedure denoted behind the name.

When evaluating the performance of an ontology matching procedure, the most common practise is to compare a generated alignment with a reference alignment of the same data set. Measures such as precision and recall, stemming from the field of information retrieval [6], can then be computed to express the correctness and completeness of the computed alignment. Given a generated alignment A and reference alignment R, the precision of the generated alignment A is defined as:

$$P(A, R) = \frac{R \cap A}{A} \tag{2}$$

whereas the recall of the generated alignment A is defined as:

$$R(A, R) = \frac{R \cap A}{R} \tag{3}$$

Given the precision and recall of an alignment, a common measure to express the overall quality of the alignment is the F-measure [6]. Given a generated alignment A and a reference alignment R, the F-measure can be computed al follows:

$$\text{F-measure} = \frac{2 * P(A, R) * R(A, R)}{P(A, R) + R(A, R)} \tag{4}$$

The F-measure is the harmonic mean between precision and recall, hence giving both measures equal importance. Given that these measurements require a reference alignment, they are often inconvenient for large-scale evaluations, since reference alignments require an exceeding amount of effort to create. The used data sets, however, do feature reference alignments, such that the performance of a matching approach can easily be computed and compared.

4.1 Conference Data Set

To test the effectiveness of our approach at resolving naming conflicts, it is reasonable to use a test set that contains a significant amount of these conflicts. The conference test set of the OAEI 2010 competition

serves this purpose well, since it consists of several real independently created ontologies, i.e. no ontologies with artificial impairments, all describing the same domain. Table 1 displays the results of our approach on the conference data set. Note that MM-none denotes our approach where the synset selection in disabled, yielding a WordNet similarity where incorrect synsets are not discarded at all. The other entries denote our approach using the various selection procedures, which are ordered by increasing strictness of their selection.

system	confidence threshold	Prec.	Rec.	FMeas.
MM-none	0.7	23%	56%	33%
MM-Geometric mean	0.7	26%	53%	35%
MM-Arithmetic mean	0.7	28%	53%	37%
MM-Max-STD threshold	0.7	40%	47%	43%
MM-Max synset	0.7	40%	49%	44%

Table 1: Results of our approach on the conference test set from the OAEI 2010 competition.

From Table 1 we can see two notable trends. First and foremost is the observation that the more strict the synset selection procedure is, the higher the overall performance of the matcher is, as indicated by the F-Measure. This is solely due to a steady increase of the precision of the alignments. Secondly, it is notable that the recall of the alignments decreases slightly upon increasing the strictness of the selection procedure. This can be explained by the possibility that during the selection synsets are discarded that more appropriately denote the meaning of a given entity than its similarity value indicates.

system	confidence threshold	Prec.	Rec.	FMeas.
AgrMaker	0.61	53%	68%	58%
AROMA	0.45	37%	50%	42%
ASMOV	0.17	53%	71%	60%
CODI	*	88%	52%	64%
Ef2Match	0.83	63%	61%	61%
Falcon	0.92	80%	52%	62%
GeRMeSMB	0.77	39%	53%	44%
MM-Max synset	0.7	40%	49%	44%
SOBOM	0.37	60%	56%	57%

Table 2: Results of the OAEI 2010 competition on the conference data set, with the results of our approach added for comparison.

Table 2 compares the performance of our best performing framework against the competitors of the OAEI 2010 competition. Note that the confidence threshold for Table 2 denotes the optimal threshold for discarding alignments of lower confidence, such that the average F-Measure is maximized, as determined by the experimenter [5]. The matching framework CODI does not provide confidence measures, hence no optimal threshold can be determined. When comparing the best performing configuration of MaasMatch to the competitors of the OAEI 2010 competition, we can see that it performs well enough, such that its performance is equivalent to the established matchers AROMA and GeRMeSMB. This is especially notable because both AROMA and GeRMeSMB are complete ontology matching frameworks, aimed at resolving all types of heterogeneities, while our approach solely focuses on naming conflicts.

Overall, one can conclude that our proposed approach improves the performance of our otherwise simple matching framework to such an extent, that it can compete with the lower-end of matchers that participated in the OAEI 2010 competition. This performance can possibly be improved upon addressing non-naming conflicts as well or investigating improvements for our current approach.

4.2 Benchmark

The benchmark data set is a synthetic data set, where a reference ontology is matched with many systematic variations of itself. These variations include many aspects, such as introducing errors to or randomizing names, omitting certain types of information or altering the structure of the ontology. Unfortunately, since

a base ontology is compared to variations of itself, this data set does not contain a large quantity of naming conflicts, which our approach is targeted at. However, it is interesting to see how our framework performs when faced with every kind of heterogeneity. Figure 3 compares the performance of MaasMatch on the benchmark data set with the results of the OAEI 2010 competitors.

system	Precision	Recall
refalign	1.00	1.00
edna	0.45	0.58
AgrMaker	0.95	0.84
AROMA	0.94	0.48
ASMOV	0.99	0.89
CODI	0.84	0.44
Ef2Match	0.98	0.65
Falcon	0.82	0.65
GeRMeSMB	0.96	0.67
MapPSO	0.68	0.60
MM-Max synset	0.95	0.44
RiMOM	0.99	0.84
SOBOM	0.97	0.75
TaxoMap	0.86	0.29

Table 3: Results of the OAEI competition on the benchmark test set, with the results of our approach added for comparison.

From Table 3 we can see that the overall performance MaasMatch resulted in a high precision score and relatively low recall score when compared to the competitors. The low recall score can be explained by the fact that the WordNet similarity of our approach relies on collecting synsets using information stored in the names of the ontology entities. The benchmark data set regularly contains ontologies with altered or even scrambled names, making it extremely difficult to extract synsets that might denote the meaning of an entity. These alterations also have a negative impact on the quality of the constructed virtual documents, especially if names or annotations are scrambled or completely left out, resulting in MaasMatch performing poorly in benchmarks that contain such severe alterations. However, despite suffering these drawbacks, it was possible to achieve results similar to established matchers that address all types of heterogeneities, such as AROMA and CODI. Thus, given the results of the benchmark, the performance of MaasMatch can be improved if measures are added which tackle other types of heterogeneities, especially if such measures increase the recall without impacting the precision.

5 Conclusion

In this paper, we proposed a method for establishing correct meanings of ontology entities, by identifying similar entities within a linguistic ontology, such as WordNet, such that the relatedness of any two entities can be specified more accurately. This is achieved through the application of virtual documents and information retrieval techniques. Experiments show that our approach increases the overall performance of our matching framework to such an extent that its performance is on a similar level than existing matching frameworks, encouraging further research. However, experiments on the benchmark data set reveal that the approach relies on the presence of adequate concept names and descriptions, resulting in low recall values if these are not present.

Thus, the most promising improvement of our approach would be increasing the robustness against disturbances in the names and descriptions of entities, allowing the assembly of potential meanings of entities that have these impairments. Since the recall of the computed alignments slightly decreases if our approach is applied, indicating that occasionally the correct meaning of an entity is not established, a further improvement would be the use of more advanced information retrieval techniques such that the computed document similarities more accurately reflect the correct meanings. Also, since our approach is mainly targeted at the linguistic features of an ontology, the overall performance of our framework can be increased if other features, such as structural similarities, are incorporated as well.

References

[1] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.

[2] M. W. Berry, Z. Drmac, and E. R. Jessup. Matrices, vector spaces, and information retrieval. *SIAM Rev.*, 41:335–362, June 1999.

[3] A. Budanitsky and G. Hirst. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *Workshop on WordNet and other lexical resources, second meeting of the North American Chapter of the Association for Computational Linguistics*, 2001.

[4] W. W. Cohen, P. Ravikumar, and S. E. Fienberg. A comparison of string distance metrics for name-matching tasks. pages 73–78, 2003.

[5] J. Euzenat, A. Ferrara, C. Meilicke, J. Pane, F. Scharffe, P. Shvaiko, H. Stuckenschmidt, O. Svab-Zamazal, V. Svatek, and C. Trojahn. First results of the ontology alignment evaluation initiative 2010. In *Proceedings of ISWC Workshop on OM*, 2010.

[6] F. Giunchiglia, M. Yatskevich, P. Avesani, and P. Shvaiko. A large dataset for the evaluation of ontology matching. *Knowl. Eng. Rev.*, 24:137–157, June 2009.

[7] A. Isaac, S. Wang, C. Zinn, H. Matthezing, L. van der Meij, and S. Schlobach. Evaluating thesaurus alignments for semantic interoperability in the library domain. *IEEE Intelligent Systems*, 24:76–86, March 2009.

[8] Y. R Jean-Mary, E. P. Shironoshita, and M. R. Kabuka. Ontology matching with semantic verification. *Web Semant.*, 7:235–251, September 2009.

[9] C. Matuszek, J. Cabral, M. Witbrock, and J. DeOliveira. An introduction to the syntax and content of cyc. *AAAI Spring Symposium*, 2006.

[10] D. L. McGuinness and F. van Harmelen. OWL web ontology language overview. W3C recommendation, W3C, February 2004.

[11] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38:39–41, November 1995.

[12] N. F. Noy and M. A. Musen. Anchor-prompt: Using non-local context for semantic matching. In *Proceedings of the workshop on ontologies and information sharing at the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 63–70, 2001.

[13] Y. Qu, W. Hu, and G. Cheng. Constructing virtual documents for ontology matching. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 23–31, New York, NY, USA, 2006. ACM.

[14] G. Salton, A. Wong, and C.S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18:613–620, November 1975.

[15] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison Wesley, 1 edition, May 2005.

[16] C. Watters. Information retrieval and the virtual document. *J. Am. Soc. Inf. Sci.*, 50:1028–1029, September 1999.