

Machine Learning Approach for Ontology Mapping using Multiple Concept Similarity Measures

Ryutaro Ichise
Principles of Informatics Research Division,
National Institute
of Informatics
2-1-2 Hitotsubashi Chiyoda-ku
Tokyo, 101-8430, Japan
ichise@nii.ac.jp

Abstract

This paper presents a new framework for the ontology mapping problem. We organized the ontology mapping problem into a standard machine learning framework, which uses multiple concept similarity measures. We presented several concept similarity measures for the machine learning framework and conducted experiments for testing the framework using real-world data. Our experimental results show that our approach has increased performance with respect to precision, recall and F-measure in comparison with other methods.

1 Introduction

Currently, numerous people use the internet to collect information as a decision making tool. For example, when making vacation plans, users conduct research on the internet for suitable lodging, routes, and sightseeing spots. However, these internet sites are operated by individual enterprises, which means that we are required to check the sites manually in order to collect information. In order to resolve this problem, the Semantic Web is expected to become a next generation web standard that will be capable of connecting different data resources. On the Semantic Web, the semantics of the data are provided by ontologies for interoperability of the resources. However, since ontologies cover a particular domain or use, it is necessary to develop a method to map multiple ontologies in order to increase the coverage of different domains or uses. In this paper, we organize an ontology mapping problem into a machine learning framework. The framework uses a standard machine learning

method with multiple concept similarity measures. If we utilize this framework, we can integrate different types of similarity measures into one standard method without any ad-hoc procedures.

This paper is organized into seven sections. First, we define the problem of ontology mapping that we are tackling. Second, we organize an ontology mapping problem into a machine learning framework. Next, we propose new similarity measures for machine learning frameworks and compare the performance of the proposed method using real Internet data. Then, we discuss the performance and related methods. Finally, we present our conclusions.

2 Ontology Mapping Problem

In this section, we describe the ontology mapping problem that we are undertaking. When we have several instances of objects or information, we usually use a concept hierarchy to classify them. Ontologies are used for such organization. We assume that the ontologies in this paper are designed for such use. The ontology used for our paper can be defined as follows:

The ontology O contains a set of concepts, C_1, C_2, \dots, C_n , that are organized into a hierarchy. Each concept is labeled by strings and can contain instances.

An example of an ontology is shown in the graphic representation on the left side of Figure 1. The black circles represent a concept in ontology and the white boxes represent instances in the ontology. The concepts (black circles) are organized into a hierarchy.

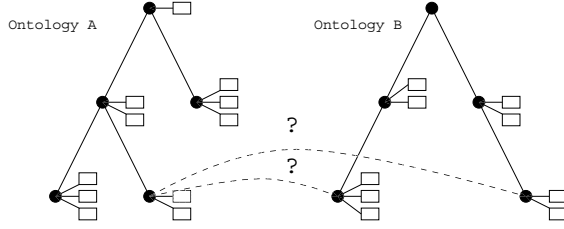


Figure 1. Ontology mapping problem to determine correct mappings of concepts among different ontologies.

The ontology mapping problem can be defined as follows: When there are two different ontologies, how do we find the mapping of concepts between them? For example, in Figure 1, the problem is finding a concept in ontology B that corresponds to the concept in ontology A. For the bottom center concept of ontology A, the possibility of the mapping can be the right bottom concept or the left bottom concept in ontology B, or there may be others. If we find appropriate mappings of the concepts, we could interoperate any information organized with those ontologies. In order to do this, we discuss a method to find the mapping by machines in the following section.

3 Ontology Mapping as a Machine Learning Problem

To solve this problem, we think about the combination of concepts among different ontologies. In this case, the problem can be defining the value of the combination pair. In other words, the ontology mapping problem consists of defining the value of pairs of concepts in a concept pair matrix, as shown in Figure 2. The rows of the matrix illustrate the concepts of Ontology A, that is, C_{a1} , C_{a2} and C_{a3} , and the columns of the matrix illustrate the concept of Ontology B, that is, C_{b1} , C_{b2} and C_{b3} . The values in the matrix represent the validity of the mapping. The value is 1 when the two concepts can be mapped and 0 when the two concepts cannot be mapped. For example, the value in the second row and third column of the matrix represents the validity of mapping for C_{a2} on Ontology A and C_{b3} on Ontology B. This particular mapping is not valid because the value in the matrix is zero. Although we assume that the matrix value is binary in this paper, we propose that a continuous value is more favorable for representing the probability of mapping. This extension is planned for future work.

The next question is what type of information is available to compose the matrix. According to our definition of ontologies, we can define a similarity measure of concepts, us-

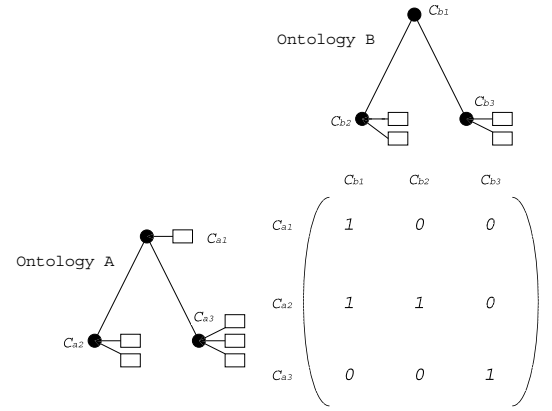


Figure 2. Matrix formulation of the ontology mapping problem.

ing a string-matching method, such as concept name matching, and other methods. However, the single similarity measure is insufficient for determining the matrix because of the diversity of ontologies. For example, we can assume the concept of a “bank” in two ontologies. The concepts seem to be mapped when we use the string similarity measure. However, when one ontology has a super concept of “finance” and another has that of “construction,” these two concepts should not be mapped because each represents a different concept. In such a case, we should also use another similarity measure of concepts. Therefore, it is necessary to use multiple similarity measures to determine the correct mappings.

From the above discussion, the problem in our paper is to define matrix values by using multiple similarity values of the concepts. As a result, we can tabulate the problem as shown in Table 1. The *ID* shown in the table represents a pair of concepts: *Class* represents the validity of the mapping, and the columns in the middle represent the similarity of the concept pairs. For example, the first line of the table represents the ontology mapping for C_{a1} and C_{b1} , and has a similarity value of 0.75 for similarity measure 1. When we know some mappings, such as $C_{a1} \Leftrightarrow C_{b1}$ and $C_{a1} \Leftrightarrow C_{b2}$, we can use the mapping to determine the importance of the similarity measures. Then, we can make a decision on unknown classes such as $C_{a5} \Leftrightarrow C_{b7}$ by using the importance of the similarity measures. The example table shown is the same as the problem in the supervised machine learning framework. Therefore, we can convert the ontology mapping problem into a machine learning problem by using this framework. In addition, we can apply general machine learning methods, such as support vector machines (SVM) [2], decision trees and neural networks for ontology mapping problems.

Table 1. Table formulation of the ontology mapping problem.

ID	Similarity measure 1	Similarity measure 2	...	Similarity measure n	Class
$C_{a1} \Leftrightarrow C_{b1}$	0.75	0.4	...	0.38	1 (Positive)
$C_{a1} \Leftrightarrow C_{b2}$	0.52	0.7	...	0.42	0 (Negative)
...
$C_{a5} \Leftrightarrow C_{b7}$	0.38	0.6	...	0.25	?
...

4 Concept Similarity Measures

In the previous section, we showed the feasibility of the application of the general machine learning framework for the ontology mapping problem. In this section, we discuss the similarity measures which correspond to the attributes on the machine learning framework.

4.1 Types of Concept Similarity Measures

Many similarity measures have been proposed for concept similarities, including the string-based similarity, graph-based similarity, instance classification similarity, and knowledge-based similarity. The string-based similarity is widely used for ontology mapping. We will discuss this similarity later. The graph-based similarity utilizes the similarity of the structures of ontologies. The ontologies are organized as tree structures, so we can calculate the graph similarity of the ontologies: examples include Similarity Flooding [12] and S-Match [8]. Instance classification similarity uses principles that, if the classification of instances is similar to the concepts in different ontologies, the concepts are similar. SBI [9] utilizes this similarity with the calculation of κ -statistics. The knowledge-based similarity utilizes other knowledge resources, such as a dictionary and WordNet [7] to calculate the similarity. We discuss this approach later.

Although there are many similarity measures, we discuss four similarity measures for use in our framework. The similarities are “word similarity,” “word list similarity,” “concept hierarchy similarity,” and “structure similarity.” We will discuss these in this order. It should be noted that our framework using tables of similarity, such as Table 1, is very general, so we can introduce any other similarity measures of concepts not presented in this paper.

4.2 Word Similarity

In order to calculate the concept similarity, we introduce four string based similarities and also four knowledge based similarities as the base.

The string-based similarity is calculated for words. We utilize the following similarities:

- prefix
- suffix
- Edit distance
- n-gram

The prefix similarity measure is for the similarity of word prefixes such as Eng. and England. The suffix similarity measure is for the similarity of word suffixes such as phone and telephone. Edit distance can calculate the similarity as a count of the string substitutions, deletions and additions. For n-gram, the word is divided into n number of strings, and the similarity is calculated by the number of same string sets. For example, “word” and “ward” similarity is counted as follows: The first word, “word” is divided into “wo, or, rd” for the 2-gram, and the second word “ward” is divided into “wa, ar, rd” for the 2-gram. As a result, we can find the similar string “rd” as the similarity measure for the 2-gram. In our system, we utilize the 3-gram for calculating the similarity.

The knowledge-based similarity is also calculated for words. We use WordNet as the knowledge resource for calculating the similarity. Although a wide variety of similarities for WordNet are proposed, we utilize four similarities:

- synset
- Wu & Palmer
- description
- Lin

The first similarity measure *synset* utilizes the path length of the synset in WordNet. WordNet is organized with synsets. Therefore, we can calculate the shortest path of the different word pairs using synsets. *Synset* similarity measures use the path length as the similarity measure. Wu & Palmer similarity measures use the depth and the least common super-concept (LCS) of words [15]. The similarity is calculated in the following equation:

$$similarity(W_1, W_2) = \frac{2 \times depth(LCS)}{depth(W_1) + depth(W_2)}$$

W_1 and W_2 denote word labels for the concept pair to calculate the similarity, the depth is the depth from the root to the word and LCS is the least common superconcept of W_1 and W_2 . The third similarity measure, *description*, utilizes the description of a concept in WordNet. The similarity is calculated as the square of the common word length in both descriptions of the words. The last similarity measure is proposed by Lin [11]. This measure is calculated using a formula similar to that of Wu and Palmer, except it uses information criteria instead of depth.

4.3 Word List Similarity

In this section, we extend the word similarity measures presented in the previous section. The word similarity measures are designed for words, and the measure is not applicable to a word list such as “Food.Wine.” Such a word list can usually be used as a concept label. If we divide such words using a hyphen or underscore, we can obtain a word list. We define the similarity for a word list in this section. Specifically, we define two types of similarities: *maximum word similarity* and *word edit distance*.

Let us first explain the maximum word similarity. When we use the combination of words in both lists, we can calculate the similarity for each pair of words by word similarity measures. We use the maximum value of the word similarity for word pairs in the word list as the maximum word similarity. In our paper, since we define eight word similarities for words in the previous section, we can obtain eight maximum word similarities by using word similarities.

The second similarity measure, word edit distance, is derived from the edit distance. In the edit distance definition, the similarity is calculated by each string. We extend this method by considering words as strings. Let us assume two word lists, “Pyramid” and “Pyramid, Theory;” the similarity between the two lists is considerably apparent. If we consider one word as a component, we can calculate the edit distance for the word lists. In this case, “Pyramid” is the same in both word lists, so then we can calculate the word edit distance as one. On the other hand, if we assume “Top” and “Pyramid, Theory;” the word edit distance is two. Consequently, we can therefore calculate the similarity by the word edit distance. However, another problem occurs for similar word lists. For example, when we assume “Social, Science” and “Social, Sci” how do we decide the similarity? The problem is the calculation of similarity for “Science” and “Sci”: that is, we have to decide whether the two words are the same word or not. If we decide that the two words are the same, the word edit distance is zero, but if not, the word edit distance is one. In order to calculate the similarity of the words, we employ the word similarity measure with a particular threshold once more. For example, if we use the prefix as the word similarity measure, we can consider the

two words are the same for calculating the word edit distance. However, if we use the synset as the word similarity measure, we cannot consider the two words as the same because “sci” is not in WordNet. From the above discussion, we can define the word edit distance for eight word similarity measures. As a result, we define 16 similarity measures for word lists, consisting of eight maximum word similarities and eight word edit distance similarities.

4.4 Concept Hierarchy Similarity

In this section we discuss the similarity for the concept hierarchy of an ontology. As discussed in Section 2, ontologies are organized as concept hierarchies. In order to utilize the similarity of a concept hierarchy, we introduce concept hierarchy similarity measures for concept hierarchies. The concept hierarchy similarity measure is calculated for the path from the root to the concept. Let us explain using the example shown in Table 2. We assume the calculation of the path “Top / Social_Sci” in ontology A and “Top / Social_Science” in ontology B. For calculation of the similarity, we divide the path into a list of concepts, as shown in the middle column of Table 2. Then the similarity can be calculated by the edit distance if we consider the concept as a component. For example, the concept “Top” is the same in both ontologies, but the second concept is different. Then, we can calculate the edit distance for the path. However, how do we decide whether the concept is the same or not? To calculate this, we divide the concept into the word list for calculating the similarity by using the word list similarity. In this case, if “Social_Sci” and “Social_Science” are considered to be a similar concept using the word list similarity, the edit distance is zero; if the two concepts are not considered as a similar concept using the word list similarity, the edit distance is one. In other words, we calculate the edit distance with the right-hand lists in Table 2. As a result, we can calculate the concept hierarchy similarity by using the edit distance of the path. Because we can use any word list similarity measures for deciding the similarity of word list, we obtain sixteen concept hierarchy similarity measures.

4.5 Structure Similarity

In this section, we define the similarity measures using the structure of ontologies. In the previous section, we defined the similarity using the concept hierarchy. However, the similarity presented above cannot handle the similarity of graphical structures. In order to use graphically close concepts, we utilize the parent concept label for calculating the similarity. Because the similarity is calculated by the word list similarity, we can obtain 16 similarity measures for parents. This similarity can be seen as one of the variations of graph similarities.

Table 2. Example of concept hierarchies for explaining concept hierarchy similarity calculation.

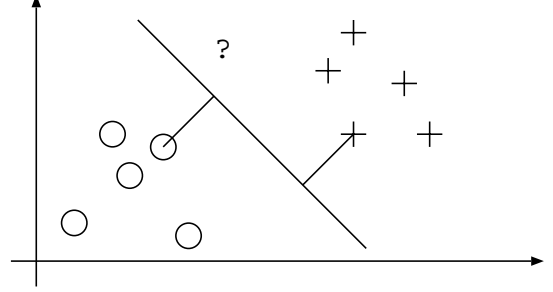
	Path	Path list	Word list
Ontology A	Top / Social_Sci	{Top, Social_Sci}	{Top}, {Social, Sci}
Ontology B	Top / Social_Science	{Top, Social_Science}	{Top}, {Social, Science}

5 Experimental Evaluation

5.1 Experimental Settings

In order to evaluate our framework, we conducted experiment using real internet directory data, which was provided by the Ontology Alignment Evaluation Initiative (OAEI) for the 2007 alignment challenge. The data contains simple relationships of class hierarchies, and is constructed from three Internet directories, Google, Yahoo, and Looksmart. The data includes 4639 pairs of ontologies written in OWL format, with 2265 pairs of the 4639 pairs are correctly matching answers, which are positive examples, and 2374 pairs are incorrectly matching answers, which are negative examples. Unfortunately, since the data has some format errors, we only used 4487 pairs of ontologies, which include 2160 positive examples and 2327 negative examples, for our analysis. We conducted 10-fold cross-validations for the experiment. This means that we randomly divided sets of all examples into 10 sets of examples: nine of these were used for learning and one was used for testing. Then the data set for testing was rotated for 10 times in order. As a result, we can measure the performance for unseen data by this experiment.

Since our proposal uses the general framework of machine learning, we can adapt any machine learning method, such as neural networks, decision trees, and support vector machines. In this paper we utilize the support vector machine (SVM) for the experiments. The SVM method is a machine learning method that can be used to predict both positive and negative examples. The method is regarded as being a state of the art machine learning method because it is capable of predicting both positive and negative examples even if they are not linearly separated. Figure 3 is a schematic diagram of the SVM method. When we have two attributes (similarity measures), we can plot positive examples (correct mappings, which are illustrated by circles) and negative examples (incorrect mappings, which are illustrated by the plus sign) in a two-dimensional field, as in Figure 3. The SVM method determines the separation border to maximize the margin from both examples. As a result, when we have new data with attributes, which is illustrated by a question mark, the system can predict it as a negative example. For the actual SVM method, the method can handle higher attribute space and nonlinear separation problems. For further information regarding this method, please refer to the book by [2]. It should be noted that,

**Figure 3. Schematic diagram of support vector machine (SVM) method.**

although we utilize the SVM method in this paper, when a more powerful machine learning method is invented, we can adopt that method by using the proposed framework.

The attributes are constructed by the method of word list similarity, concept hierarchy similarity and structure similarity, which are discussed in Section 4. We implemented our system called Malfom-SVM (Malfom: Machine learning framework for Ontology Matching using SVM) with Ruby language, SVM_light [10] and the WordNet similarity library [13].

5.2 Experimental Results

The experimental results are shown in Figure 4. The horizontal axis denotes the data set number for the experiments and the vertical axis denotes the percentages of accuracy, precision and recall. The accuracy is the percentage of correctly classified mappings, the precision is the percentage of correct mappings among the mappings which the system judged as correct, and the recall is the percentage of correct mappings which the system found from all the actual correct mappings. As we mentioned in the previous section, since we conducted 10-fold cross-validations, we have 10 data sets. As can be seen from the graph, our system has 56.1% accuracy, 52.5% precision, and 92.5% recall on average, and a stable result among the different data sets. Malfom-SVM has high recall value relative to both accuracy and precision. From this result, it is apparent that our system returned relatively more correct mappings than incorrect mappings.

In order to compare the performance of Malfom-SVM

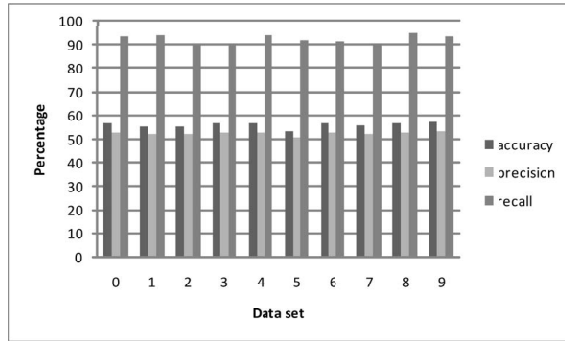


Figure 4. Experimental results of 10-fold cross validation.

with other systems, we created the performance summary in Table 3 by using the report in [6]. The recall by the other seven systems is approximately 46% at the best and 13% at the worst. On the other hand, Malfom-SVM has 92.5% recall, which is twice the performance of the best other system. For the F-measure, Malfom-SVM achieves 67.0% of the performance. F-measure is the harmonic mean of precision and recall. Although the system still has much room for improvement, it has a markedly higher performance than the other seven systems. It should be noted that although the data sets used in experiments for testing the other seven systems were the same¹, the results are not truly comparable because our experimental setting uses a supervised approach, but the others do not. In other words, if we can build a defined number of correct or incorrect mappings, then the results obtained using our method would be relatively more robust. In a comparison of the accuracy obtained using our method with a random method by χ^2 tests, we found that our method was better than the random method at a 1% level of significance. Based on this evidence, we can conclude that our system is highly capable of learning a prediction method for assigning both correct and incorrect mappings.

6 Discussion

The results from the experiments show our system has the ability to effectively produce appropriate mappings. Our framework uses multiple similarity measures. COMA [3] uses a matcher library, which corresponds to our multiple similarity measures. Although COMA uses a combination of similarity measures, it does not use standard machine learning techniques for combinations. GLUE [4] uses machine learning techniques for some steps of ontology mapping; however, it cannot use the similarity measures

¹Data sets of OAEI-2006 and OAEI-2007 are the same for the web directory competition.

of structures and labels in the same manner. APFEL [5] is a very similar approach for our framework. However, our system does not assume other ontology mapping systems because of its treatment of various types of similarity measures, discussed in Section 4. One of the merits of our approach is the separation of the framework and similarity measures. If we design the similarity of concepts using strings, graphs, and other such methods, we can integrate it immediately into our framework. The numerous systems that have been developed for different similarity measures has meant that we can integrate these new technologies into our framework. In addition, our framework is general in the machine learning community. As a result, we can apply any other sophisticated machine learning techniques without ad-hoc integration for the ontology mapping problem.

One of the problems associated with using our approach for a real-world task is the availability of mapping examples, including correct mappings and incorrect mappings. As we discussed in the previous section, our method can be used to obtain considerably improved performance when compared to that of existing systems. However, we need classified examples because our approach uses a supervised machine learning framework. Some examples can be obtained from existing technology, such as the instance based approach. In our framework, if we have more reliable examples, we would have considerably improved mapping results. We need to investigate the trade-off between the man-hours required for making mapping examples with the performance improvement of our system in the future.

7 Conclusions

We presented a new framework of ontology mapping using a machine learning approach. In order to use the approach, we defined various similarity measures in this paper and we also conducted experiments using real-world data to investigate the performance of our proposed system. The experimental results show that our approach has increased performance with respect to precision, recall and F-measure in comparison with other methods. In addition, since the proposed framework is general, we can easily adopt new similarity measures developed in the ontology matching community and sophisticated machine learning methods developed in the machine learning community. Therefore, our proposed framework is a powerful framework for ontology mapping problems.

Although our experimental results are encouraging, considerable work remains. In our future work, we are planning to introduce new similarity measures, such as gloss overlap [1] to improve performance. In addition, we also plan to investigate the best combination of similarity measures by using a non-black-box machine learning technique such as C4.5 [14].

Table 3. Performance comparison of proposed method with other systems.

	hmatch	falcon	automs	RiMOM	OCM	coma	prior	Malfom-SVM
Precision	32.4	40.5	31.1	39.3	33.3	31.2	32.7	52.5
Recall	13.4	45.5	14.6	40.4	15.7	26.8	24.4	92.5
F-measure	18.9	42.9	19.9	39.8	21.4	28.8	28.3	67.0

References

- [1] S. Banerjee and T. Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In G. Gottlob and T. Walsh, editors, *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 805–810. Morgan Kaufmann, 2003.
- [2] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge Univ Press, 2000.
- [3] H. H. Do and E. Rahm. COMA - A system for flexible combination of schema matching approaches. In *Proceedings of the 28th International Conference on Very Large Data Bases*, pages 610–621. Morgan Kaufmann, 2002.
- [4] A. Doan, J. Madhavan, R. Dhamankar, et al. Learning to match ontologies on the Semantic Web. *VLDB Journal: Very Large Data Bases*, 12(4):303–319, Nov. 2003.
- [5] M. Ehrig, S. Staab, and Y. Sure. Bootstrapping ontology alignment methods with APFEL. In Y. Gil, E. Motta, V. R. Benjamins, and M. A. Musen, editors, *Proceedings of the 4th International Semantic Web Conference*, volume 3729 of *Lecture Notes in Computer Science*, pages 186–200. Springer, 2005.
- [6] J. Euzenat, M. Mochol, P. Shvaiko, H. Stuckenschmidt, O. Svab, V. Svatek, W. R. van Hage, and M. Yatskevich. Results of the ontology alignment evaluation initiative 2006. In *Proceedings of International Workshop on Ontology Matching*, 2006.
- [7] C. Fellbaum. *Wordnet: An Electronic Lexical Database*. MIT Press, 1998.
- [8] F. Giunchiglia, P. Shvaiko, and M. Yatskevich. S-match: an algorithm and an implementation of semantic matching. In C. Bussler, J. Davies, D. Fensel, and R. Studer, editors, *Proceedings of the 1st European Semantic Web Symposium*, volume 3053 of *Lecture Notes in Computer Science*, pages 61–75. Springer, 2004.
- [9] R. Ichise, H. Takeda, and S. Honiden. Integrating multiple internet directories by instance-based learning. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, pages 22–28, 2003.
- [10] T. Joachims. Making large-scale svm learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1999.
- [11] D. Lin. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, pages 296–304. Morgan Kaufmann, San Francisco, CA, 1998.
- [12] S. Melnik, H. Garcia-Molina, and E. Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *Proceedings of the 18th International Conference on Data Engineering*, San Jose, CA, Feb. 2002.
- [13] T. Pedersen, S. Patwardhan, and J. Michelizzi. Wordnet::similarity - measuring the relatedness of concepts. In *Proceedings of the 19th National Conference on Artificial Intelligence*, pages 1024–1025, 2004.
- [14] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
- [15] Z. Wu and M. Palmer. Verb semantics and lexical selection. In *Proc. of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 133–138, New Mexico State University, Las Cruces, New Mexico, 1994.