

Building Ontology-Matching Similarity Features with Adaptive Aggregation of Neural Networks

Mrs.M.Kavitha , Mrs.Lavanya Gullani, Mr.MD.Giasuddin

1,2,3 Assistant Professor

1,2,3 Department of H&S

1,2,3Global Institute of Engineering & Technology, Moinabad, Rangareddy Dist., Telangana State.

Abstract: Ontology is the kernel technique of Semantic Web (SW), which enables the interaction and cooperation among different intelligent applications. However, with the rapid development of ontologies, their heterogeneity issue becomes more and more serious, which hampers communications among those intelligent systems built upon them. Finding the heterogeneous entities between two ontologies, i.e., ontology matching, is an effective method of solving ontology heterogeneity problems. When matching two ontologies, it is critical to construct the entity pair's similarity feature by comprehensively taking into consideration various similarity features, so that the identical entities can be distinguished. Due to the ability of learning complex calculating model, recently, Artificial Neural Network (ANN) is a popular method of constructing similarity features for matching ontologies. The existing ANNs construct the similarity feature in a single perspective, which could not ensure its effectiveness under diverse heterogeneous contexts. To construct an accurate similarity feature for each entity pair, in this work, we propose an adaptive aggregating method of combining different ANNs. In particular, we first propose a context-based ANN and syntax-based ANN to respectively construct two similarity feature matrices, which are then adaptively integrated to obtain a final similarity feature matrix through the Ordered Weighted Averaging (OWA) and Analytic hierarchy process (AHP). Ontology Alignment Evaluation Initiative (OAEI)'s benchmark and anatomy track are used to verify the effectiveness of our method. The experimental results show that our approach's results are better than single ANN-based ontology matching techniques and state-of-the-art ontology matching techniques.

Keywords: ontology matching; similarity feature construction; artificial neural network; ordered weighted averaging; analytic hierarchy process

Introduction

The emergence of Semantic Web (SW) [1] enables machines to understand semantic documents and data, which promotes the interaction and cooperation between intelligent applications. As the kernel technique of SW, ontology is a "formal statement of shared conceptualization of explicit" [2], which formally defines the domain concepts and their relationships. However, with the rapid development of ontologies, different preferences among experts define the same concepts in their own ways, which hampers communications among those intelligent systems built upon these ontologies and leads to the heterogeneity problem. At present, one of the most effective methods to solve this problem is ontology matching [3]. Since Artificial Neural Network (ANN) has the abilities of automatic learning, associative storage and high speed searching for optimal solutions [4], it becomes one of the most popular methodologies for addressing the ontology matching

problem. In recent years, different ANNs, such as Siamese Neural Network [5] and Convolutional Neural Network (CNN) [6], have been used to match ontologies and obtained acceptable results. However, due to the complex intrinsic of ontology matching, different ANN-based matching techniques cannot guarantee the obtained alignment's quality on all matching tasks since they might focus on addressing only one kind of heterogeneity features. To enhance the quality of matching results, this work proposes to adaptively aggregate different ANNs to construct the accurate similarity feature for each entity pairs. Figure 1 shows the framework of aggregating different ANN-based matching techniques.

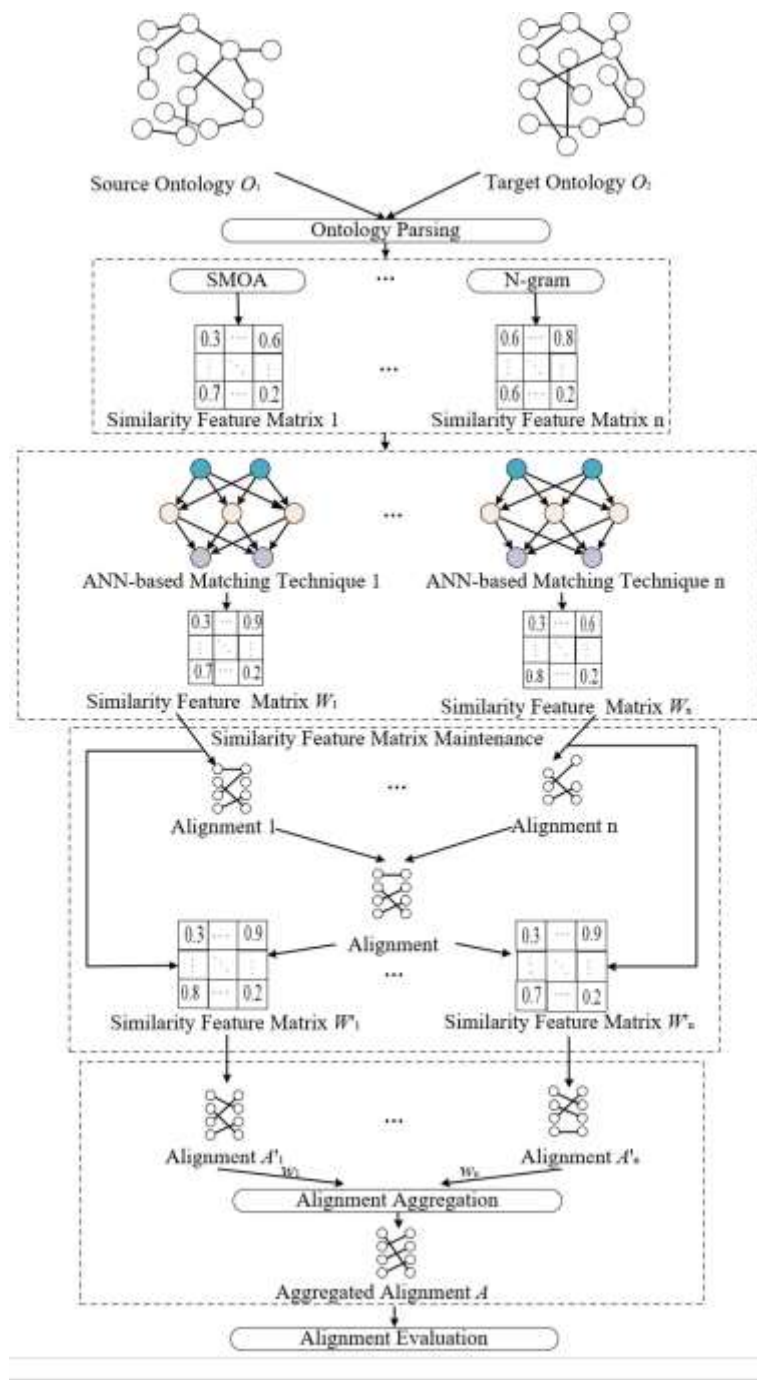


Figure 1. The Framework of Aggregating ANN-based Ontology Matching Techniques.

In the figure, the rounded rectangle represents the specific method or strategy, and the network between the two bold black lines represents the artificial neural network. The whole frame can be thought of as a function: $A = f(M(O_1, O_2)_1, \dots, M(O_1, O_2)_n)$, where A represents the final alignment, O_1 and O_2 represent two different ontologies, and $M(O_1, O_2)_n$ represents the similarity feature matrix obtained using the n -th ANN-based matching technique. The first phase pre-processes two ontologies by parsing them to obtain the entities. Then, different similarity feature values on entity pairs are calculated with different similarity measures, which are then used to construct the similarity feature matrices. Here, one similarity measure corresponding to one similarity feature matrix, whose row and column are respectively two entity sets, and the element are two entities' similarity feature value that determine by this similarity measure. In the second phase, n ANNs are executed in parallel to determine n similarity feature matrices, which is represented with two bold black lines. In the third phase, n similarity feature matrices are maintained to remove incorrect similarity features. The final phase aggregates n similarity feature matrices to obtain

the final one. After that, the quality of the corresponding alignment is measured with the quality evaluation metrics. To implement this framework, we need to answer two questions: (1) since it is not the fact that more ANNs being selected, the better results would be (some totally contradictory results could bring negative impacts), how to choose the suitable ANNs to be aggregated; and (2) how to aggregate the ANNs to make them enjoy the mutual benefits. To answer these two questions, we propose two kinds of ANNs to train three broad categories of similarity features, context-based ANN uses semantic context information to find similarity features, while syntactic ANN mainly uses string information to find similarity features. The two consider different aspects of semantics and have certain complementarity, and use an adaptive aggregating strategy to coordinate the contradictions among different alignments and enhance the alignment quality. In particular, the main contributions made in this work are as follows: (1) a framework of aggregating ANNs to construct the similarity feature matrix is constructed; (2) two ANNs are presented to respectively make use of entities' context and syntax information to determine corresponding feature matrices; (3) the adaptively aggregating strategies OWA and AHP are proposed to determine the final feature matrix; (4) a similarity feature matrix maintaining technique is proposed to improve the similarity feature's quality.

The rest of this paper is briefly described as follows: section "Related Work" provides the latest progress of ANNs. Section "Preliminary" introduces the definition of ontology and ontology matching, the similarity feature used in our work and the evaluation metrics of our approach, and introduces the word embedding ANN. Section "Artificial Neural Network Based Ontology Matching" introduces context-based ANN ontology matching technique, syntax-based ANN ontology matching technique and similarity feature matrix maintenance strategy. Section "Ordered Weighted Average Operator With Analytic Hierarchy Process" introduces the OWA technique, AHP technique, OWA and AHP based ontology matching technique and ontology alignment aggregation strategy. Section "Experiment" shows the experimental configuration and experimental result. Finally, section "Conclusion" summarizes our work and gives the conclusion.

Related Work

Matching ontologies is a complex cognitive process, and thus it is impractical to find out all the correspondences manually, especially when there are many entities in two ontologies [7]. In recent years, various semi-automatic and automatic matching methods have been put forward, and ANN-based matching technique attracts many scholars' attention due to its robustness and high precision. ANN-based ontology matching is essentially to find the alignment by constructing similarity features for various entity pairs through machine learning. Currently, from the perspective of similarity feature construction, ANN-based ontology matching can be divided into two categories, one is to directly build or train a new similarity feature to match different ontologies, and the other integrates existing similarity features. With respect to the first category, Chakraborty et al. [8] proposed using recursive neural networks to construct a structure-based similarity feature to train the

unsupervised model, it can describe the semantic information of a concept, but it lacks the linguistic-based similarity feature. Jiang et al. [9,10] defined the ontology matching problem as the classification problem and proposed using long short-term memory networks to construct a similarity feature to match ontologies, although the paper further enhances the semantic feature by combining the attention mechanism, it lacks the literal-based similarity feature. Zheng et al. used CNN to construct the similarity feature to find the degree of similarity between patients' ultrasonic examination reports. In this paper, the semantic information was obtained by using the graph embedding method and the LIME algorithm was used for feature recognition [11], but these methods are not universal. Feng et al. [6] used convolutional neural network to construct a new similarity feature, which can extract semantic ontology features and integrate semantic ontology to improve the alignment quality, but the newly constructed similarity feature is not applicable to other ontology matching tasks. To solve this problem, we propose an aggregation framework to aggregate three different similarity features. In addition, some scholars focus on the embedding technique of ANN. Using ANN and some strategies to construct similarity feature and match ontologies directly, the underlying embedding mode of these ANNs is generally character embedding. In the latest study, Iyer et al. [5] and Xue et al. [12] proposed using SNN to construct similarity feature to match ontologies, but they adopted different strategies. The former introduces the notion of multi-faceted context and proposes a novel dual-attentional mechanism, the latter proposes a refinement technique, both of which ultimately improve the quality of alignment. However, the bottom layer of these neural networks relies on the character embedding to learn words, which greatly reduces the semantic relevance between words. Although there are techniques available to improve semantic relevance, such as [5] use a label-based similarity feature to enhance the semantic relevance between words, and Chen et al. [13] present the traditional machine learning-based ontology alignment system and use SNN combined with ontology embedding to enrich the semantics of ontologies. It is not straightforward to improve the semantic relevance of words through specific strategies or techniques. Thus, we propose a context-ANN to construct a context-based similarity feature, which not only considers the semantic relevance between terms, but also makes the matching result more accurate, directly and fast. Furthermore, we propose a sentence-preprocessing strategy for the context-based ANN to calculate the similarity features between sentences and further improve the quality of matching.

With respect to the second category, ANN can be used to learn the weights of various similarity features. Different similarity features have different preferences on their applications. Multiple similarity features used together allow multiple aspects of the word to be considered, greatly increasing the number of alignments. Huang et al. [14] first proposed the ontology matching technique of using the ANN to learn weights, which is accurate and efficient. Since then, many

scholars have also carried out research in this direction. Huang et al. [15] propose the use of artificial neural network matching biological ontologies, which learn and adjust these weights to support a new ontology alignment algorithm and improve the quality of matching through using multiple similarity feature. Djeddi et al. [16] aggregate different similarity feature techniques with artificial neural network and applying them to benchmark tests and anatomic tests, the matching quality is very high. Lev et al. combined different matchers with machine learning methods and considered the output of lexical and semantically similar functions as features. Naive Bayes classifier, logistic regression and so on are trained on these feature sets [17]. Similarly, Xue et al. use ANN to integrate features of different similarity and obtain good matching results [18]. In addition, combining different matching systems is also a good innovative method. Khoudja et al. combined the top-ranked matching systems through single-layer perceptron and defined a matching function, generate a better set of alignment between ontologies [19]. These techniques require a lot of calculation. We choose a neural network that aggregates literal and linguistic similarity features without considering structure-based similarity features, which greatly reduces the computation, can compare the string shape between words and strengthen the semantic correlation between words.

Different similarity features might focus on different aspects of semantic context and cannot guarantee their effectiveness in all matching tasks. Ontology matching field has three categories similarity features, but the existing ANN approach only integrates part of the similarity features. To enhance the reliability of the calculated results, we propose constructing and integrating three similarity features through ANN. In addition, a nested parallel integration framework is used, that is, both context-based ANN and syntax-based ANN are used. Then, a similarity feature maintenance strategy for both ANNs is proposed to further enhance the matching quality. Finally, OWA is proposed to integrate the description of the different layers of ontology, and AHP, a widely used and efficient decision-making technique, is used to determine the integration weights of different ANN-based similarity features. Our approach not only solves the problem of word heterogeneity to a certain extent and makes the matching result more accurate, direct and fast, but also enhances the semantic relevance between words and further improves the quality of alignment.

Ontology, Similarity Measure and Ontology Alignment's Evaluation Metric

Ontology and Ontology Matching

Definition1 An ontology O is a triple (C, P, I) [20], where C, P, I are the concept set, property set and instance set, respectively. **Definition2** An ontology alignment is a set of correspondences, and a correspondence is a quad $\langle e, e', H, R \rangle$ [20]. In order to describe ontology and ontology alignment more intuitively, we draw Figure 2, where rounded rectangles represent class i.e., concept, which are connected by thick solid lines with arrows that point from subclass concepts to superclass concepts. For example, "Book" is a subclass of "Product", these classes form set C . The dotted lines with arrows point to the attributes of the concept, such as "Price" is an attribute of the "Product". These properties form set P . The solid, unbolded lines with arrows point from the individual to the concept, for example, "Matching Learning: Zhilua Zhou" is a concrete example of the "book". These instances form set I . The red solid line with a two-way arrow represents the relationship between

two entities e and e' , where $=$ indicates that the two entities are equivalent, and \supseteq indicates that one entity is contained in the other. For example, "Literature" is included in "Book". The symbol R is used to represent the relationship between entities. Furthermore, entities with equivalence relations have a confidence of 1, such as "title" and "title". The symbol H is used to represent the confidence value, which is represented by a real number between 0 and 1. These are collectively called alignments. **Definition3** The process of finding these alignments is the process of ontology matching, which is a function $A = f(O_1, O_2, A', r, p)$ [20]. The flowchart of ontology matching process is shown in Figure 3, where we consider the whole matching system as a function, first input two ontologies O_1 and O_2 and reference alignment A' , adjust external resources r and some necessary parameters p , and finally obtain the alignment A .

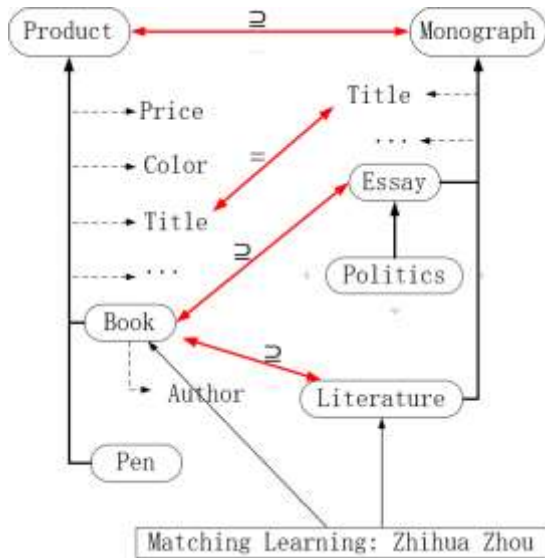


Figure 2. An Example of Two Ontologies and Their Alignment.

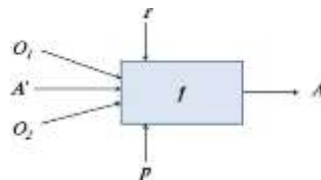


Figure 3. The Flowchart of Ontology Matching Process.

Similarity Measure

Similarity measure is a function whose input are two concepts from two different ontologies, and the output is a real number between 0 and 1. General, similarity measures are divided into three categories, i.e., lexical-based similarity measure, linguistic-based similarity measure, and structural-based similarity measure [18]. Additionally, lexical-based similarity measure calculates the similarity feature by calculating the edit distance between strings of two concepts, linguistic-based similarity measure often computes the similarity feature between two concepts through an external dictionary or corpus, e.g., the WordNet [21]. Structural-based similarity measure calculates the similarity feature by the neighbor entities of the entity. In this paper, we use three similarity measures. N-gram [22] and SMOA [23] are lexical based similarity measures, which mainly calculate the morphological similarity of text content. The Wu and Palmer method (WuP) [24] uses WordNet electronic dictionary to measure the semantic similarity of two words. The definitions on three similarity measures are as follows.

$$N\text{-gram}(s_1, s_2) = \frac{2 \times \text{comm}(s_1, s_2)}{N_{s_1} + N_{s_2}} \quad (1)$$

where s_1 and s_2 are two strings that need to be compared. Usually, we divide the string into three sub strings according to the rule that N is 3 [22]. $\text{comm}(s_1, s_2)$ represents the number of sub strings s_1 and s_2 are the same. N_{s_1} and N_{s_2} represent the number of substrings that s_1 and s_2 are cut into, respectively.

$$\text{SMOA}(s_1, s_2) = \text{com}(s_1, s_2) - \text{dif}(s_1, s_2) + \text{winklerImpr}(s_1, s_2) \quad (2)$$

$$\text{com}(s_1, s_2) = \frac{2 \cdot \sum_i |\text{maxComString}_i|}{|s_1| + |s_2|}$$

SMOA [23] is defined in Equations (2)–(4). $\text{com}(s_1, s_2)$ is a measure of identity between strings s_1 and s_2 . The purpose is to find the largest common substring through recursion, and remove the substring from s_1 and s_2 so that there is no common substring in the two strings. maxComString_i is the largest substring in the i th recursion. $\sum_i \text{maxComString}_i$ represents the sum of the lengths of the common substrings obtained in the previous recursion operations.

$$\text{dif}(s_1, s_2) = \frac{|uL(s_1)| \cdot |uL(s_2)|}{p + (1 - p) \cdot (|uL(s_1)| + |uL(s_2)| - |uL(s_1)| \cdot |uL(s_2)|)} \quad (4)$$

$diff(s_1, s_2)$ is calculated according to the length of the substring that did not match in the first recursive operation, where $uL(s_1)$ and $uL(s_2)$ are the length of unmatched substrings between s_1 and s_2 , respectively. p is a preset parameter, $winklerlmp_r(s_1, s_2)$ is a method of improving the calculation proposed by Winkler.

$$Wup(s_1, s_2) = \frac{2 \cdot depth(LCA(s_1, s_2))}{depth(s_1) + depth(s_2)}$$

Wup similarity measure [24] is defined in Equation (5), where $LCA(s_1, s_2)$ represents the closest common parent concept between the words s_1 and s_2 , $depth(s_1)$ and $depth(s_2)$ represent the depth positions of s_1 and s_2 in the WordNet dictionary, respectively.

Word Embedding ANN

Word embedding refers to the mapping of words from the original space to the new space, that is, the word space conversion to the vector space. We use Word2Vec [25], a word embedding-based ANN, which proposed by Tomas Mikolov et al.. In fact, Word2Vec transforms the words we need into vectors through the semantic information of the context, so it is also a context-based ANN. In addition, due to the unsupervised training of Word2Vec itself, the vectors generated by Word2Vec can better represent the actual meaning of words than the vectors obtained by supervised learning-based ANN to some extent. Word2Vec has two modes: one is CBOW pattern and the other is Skip-gram pattern, each of these modes can be implemented in Hierarchical Softmax or Negative Sampling. This paper adopts Skip-gram + Negative Sampling structure because of its ability to predict the central word through its context. Figure 4 shows the simple model architecture of Skip-gram + Negative Sampling. The right side of the figure shows the hidden layer and output layer drawn to explain what Negative sample Sampling is.

As shown in Figure 4, where, $W(t)$ stands for the central word. The purpose of this model is to convert words into vectors. Its input and output are one-hot encoded, and the input representing the central word and the output representing the context words. Given an input and output of the model, it will continuously adjust the weight values between the input layer and the hidden layer through gradient descent and back propagation, and the final weight values is the multidimensional vector transformed from the center word, and the number of hidden layer neurons is the dimension of the word vector.

We first regard the neuron of the selected word as the output as 1, and the neurons of the other words that are not selected as the output as 0, and then the neuron which is 0 can be regarded as the negative sample. If we take only a partial negative sample, instead of updating the weights corresponding to all the output neurons, the number of columns in the weight matrix between the output layer and the hidden layer will be greatly reduced. Thus, this reduces the actual computation to a large extent.

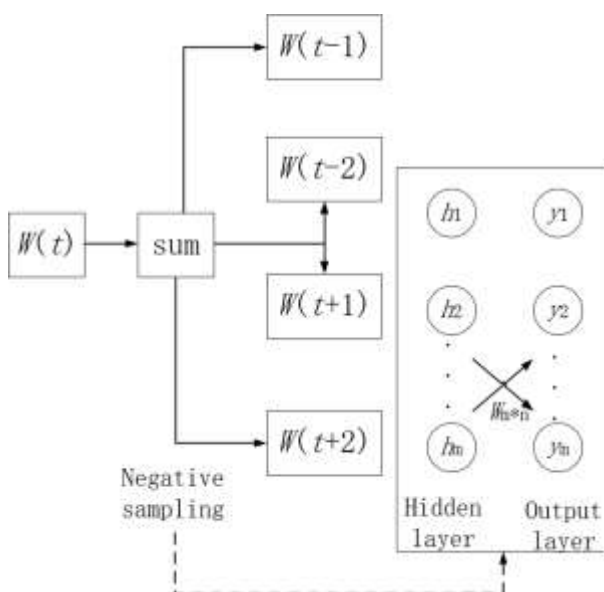


Figure 4. The Structure Of Skip-gram+Negative Sampling.

Evaluation Metrics on Ontology Alignment's Quality

The most common evaluation metrics about ontology alignment quality are precision P , recall R and f-measure F [18]. In particular, precision and recall respectively evaluate the accuracy and completeness of the alignment, and f-measure is the harmonic mean of recall and precision. Formally, they are defined as follows.

$$R = \frac{\text{correct found correspondences}}{\text{all possible correspondences}}$$

$$P = \frac{\text{correct found correspondences}}{\text{all found correspondences}}$$

$$F = \frac{2 \times R \times P}{R + P}$$

$$R + P$$

Artificial Neural Network Based Ontology Matching

The similarity features in ontology matching field can be divided into two categories, i.e., syntax-based and context-based ones, and our work is dedicated to improving the matching result's accuracy by considering both two kinds of similarity features. To this end, we first propose a syntax-based ANN and a context-based ANN to determine the syntax-based and context-based similarity feature values, respectively, and then adaptively aggregate them. In particular, we first use OWA method to aggregate different entity information, and construct the similarity feature matrices with different similarity measures. Then, we propose a similarity feature matrix maintenance strategy to ensure the correctness of the obtained similarity feature matrices. After that, we use AHP method to weight the similarity feature matrices obtained by different ANN-based matching techniques based on their contributions. Finally, we extract alignment and evaluate the results.

Context-Based ANN Ontology Matching Technique

ANN-based ontology matching usually requires an external knowledge base to train the ANN, and the character embedding-based techniques lack the semantic information of words, which might bring negative impact on the results. To overcome this drawback, we propose the Word2Vec [25], a context-based ANN, which directly converts words into vectors without using a specific external knowledge base. In addition, our approach also takes into consideration the semantic information between words. First, we preprocess the entities descriptions, which consists of two steps: (1) all words representing entities are

converted to lower case and all special symbols are removed during parsing, (2) convert the sentence into word sets, turn words into prototypes and all meaningless words were removed, such as: 'not', 'an', etc. After converting the words into vectors, the cosine function is used to calculate their distance. The pseudo-code of context-based ANN ontology matching technique is shown in Algorithm 1.

Algorithm 1 Context-based ANN Ontology Matching Technique

Input: The description of entities: I, L, C

Output: Final similarity feature matrix: W_1

- 1: Cut sentences into words, remove words with no actual meaning;
- 2: Restore each word to its original form;
- 3: Remove special symbols and case conversion operations;
- 4: Get two sets F_1 and F_2 that contains the set of words for the descriptions I or L or C from two different ontologies;
- 5: Each word set in F_1 and F_2 are transformed into the vector sets K_1 and K_2 by Word2Vec;
- 6: **for** $i = 0; i < K_1.size(); i++$ **do**
- 7: **for** $j = 0; j < K_2.size(); j++$ **do**
- 8: **if** $K_1.get(i).size() \neq 1 \quad K_2.get(j).size() \neq 1$ **then**
- 9: Calculate the similarity feature between word sets;
- 10: see also Equations (9) and (10)
- 11: **else**
- 12: Cosine similarity features is measured for entities in two different ontologies;
- 13: see also Equation (9);
- 14: **end if**
- 15: **end for**
- 16: **end for**
- 17: Constructing three similarity feature matrices in terms of three kinds of entity information;
- 18: Merge the similarity feature matrices to obtain W_1 ;

In the algorithm, I, L, C represent the entity's Id, Label and Comments, respectively, whose corresponding

similarity feature matrices are aggregated to determine the final similarity feature matrix W_1 . In addition, $K_1.get(i).size()$ is the number of words, $K_1.get(i).size() = 1$ denotes a word and $K_1.get(i).size() > 1$ denote a phrase or sentence. The process of using context-based ontology matching technique is as follows, we use description I to describe our algorithmic flow: Firstly, we preprocess all word sets accordingly, and take I from different ontologies as the input of Word2Vec, which converts these descriptions into vector sets K_1 and K_2 through the semantic information of the context. Here, three descriptions are used to calculate the cosine similarity features of concepts, cosine similarity feature is defined as follows:

$$\cos(a, b) = \frac{a \cdot b}{\|a\| \times \|b\|}$$

where a and b respectively represent the word vectors of entities from different ontologies, and $\|a\|$ represents the 2-norm of the vector.

If the input is the sentence or phrase, we process sentences or phrases into word sets by removing meaningless words from sentences and phrases. Then the sentences and phrases can be treated as words which are the same situation as $K_1.get(i).size() = 1$. The same as the above that the put is a word, three descriptions are used to calculate the cosine similarity features of all words in the word sets. Then, we need to acquire the similarity features of two different word sets from the features of all the words between the two different word sets, and the similarity feature between two word sets S_1 and S_2 is calculated as follows.

$$sim(S_1, S_2) = \frac{\sum_{i=1}^k s_{ik}}{k} \quad (10)$$

When calculating the similarity feature between two word sets, we need to first calculate the similarity features of all words between two word sets. Then ranking all similarity features in descending order, where i represents the i -th word after descending order, k is the maximum number of matching pairs that can be matched, that is, the number of words in fewer word sets. s_i is the i -th similarity feature after descending order, and the similarity feature $sim(S_1, S_2)$ of the word sets is finally obtained by Equation (10). Finally, we obtain three similarity feature matrices of I, L, C respectively. Specifically, we use OWA technique to aggregate three similarity feature matrices and obtain the similarity feature matrix W_1 .

Syntax-Based ANN Ontology Matching Technique

The purpose of ANN is to train the a variety of similarity features through training samples, and then aggregate similarity features obtained by training, and finally obtain the similarity features between entities. Word2Vec considers the semantic correlation between words through the context based similarity feature, but when it is difficult to address the heterogeneity problems of words such as word misspellings or word abbreviations e.g., the mismatch between “conference” and “conferance”, the mismatch between “semantic net- work” and its abbreviation “SSN”. To enhance the semantic correlation between words and make our proposed approach applicable to all ontology matching fields, we propose the syntax-based ANN. First, ANN is trained by partial reference alignment to obtain weights of the two kinds of similarity features, and then the weights are assigned to the respective similarity features to calculate the final similarity features. In this work, ANN takes into account both literal-based similarity feature and linguistic-based similarity feature, which is able to distinguish the heterogeneous entities in terms of literal and linguistic and enhance the semantic correlation between words. The flow chart of syntax-based ANN ontology matching technique is shown in Figure 5.

As can be seen from the figure, the syntax-based ANN has three layers, the input layer is the similarity feature of the similarity feature matrix obtained by different similarity features. The second layer is linear layer for aggregating similarity features, the output layer is the final similarity feature between entities from different ontologies. We first train the positive and negative samples that are selected in advance, and obtain the weight w_i through gradient descent and back propagation. These weights are the proportions of the various similarity features in the aggregation process. Finally, we aggregate these similarity features through the weight values learned and bias terms learned, where s represents the similarity features calculated by different similarity features, b represents the bias term of the neural network, and s represents the final similarity feature obtained by

^

syntax-based ANN.

In the flow chart of syntax-based ANN ontology matching technique, first, the input comes from two different ontologies, and each entity in the ontology is represented by three descriptions: Id, Label, Comments. Then we preprocess the ontologies, that is, the ontology is parsed into words or sentences that we can understand. Next, to train the syntax-based ANN, we first initialize the necessary parameters: the weights w_1 , w_2 , w_3 of the three similarity features, and the bias term parameter b of the neuron. i is the maximum number of iterations, which is the number of times a neuron is trained. In the training stage, we pick up some correspondences from the reference alignment as positive sample, and meanwhile, we build negative samples, whose number is the same as that of positive sample sets. Respectively, using the above three kinds of similarity feature techniques to train the positive and negative samples. The neural network continuously adjusts the weight values and bias terms of neurons through the back propagation algorithm and gradient descent algorithm, and evaluates these weight values and bias terms by using the Mean Square Error(MSE) loss function. The smaller the MSE is, the more accurate the model can predict the experimental data. If the maximum number of iterations is reached or the value of the MSE loss function is lower than the threshold j , the weight values w_1 , w_2 , w_3 of different similarity features and the bias term b of the neuron can be obtained and the next stage will be carried out, where the choice of j depends on the actual calculation. W_{Id} , W_{Label} , $W_{Comments}$ represent the similarity feature matrix calculated by the same similarity feature. First of all, we calculate the similarity features between concepts from different ontologies. Since entities are composed of three descriptions, we aggregate their corresponding similarity feature matrices through OWA to obtain the similarity feature matrix, OWA techniques are discussed in detail in the next section. After calculating the similarity features between entities using three similarity features and integrating the corresponding W_{Id} , W_{Label} and $W_{Comments}$ of each similarity feature using OWA technique, the three similarity feature matrices are obtained, where k stands for the number of elements in the similarity feature matrix, which is determined by the number of entities in two ontologies to be matched. Then we traverse the similarity feature matrices and integrate the elements of the three similarity feature matrices through the learned w_1 , w_2 , w_3 and b . When all the elements of the three matrices are integrated, the similarity feature matrix W_2 between two different ontologies is finally obtained.

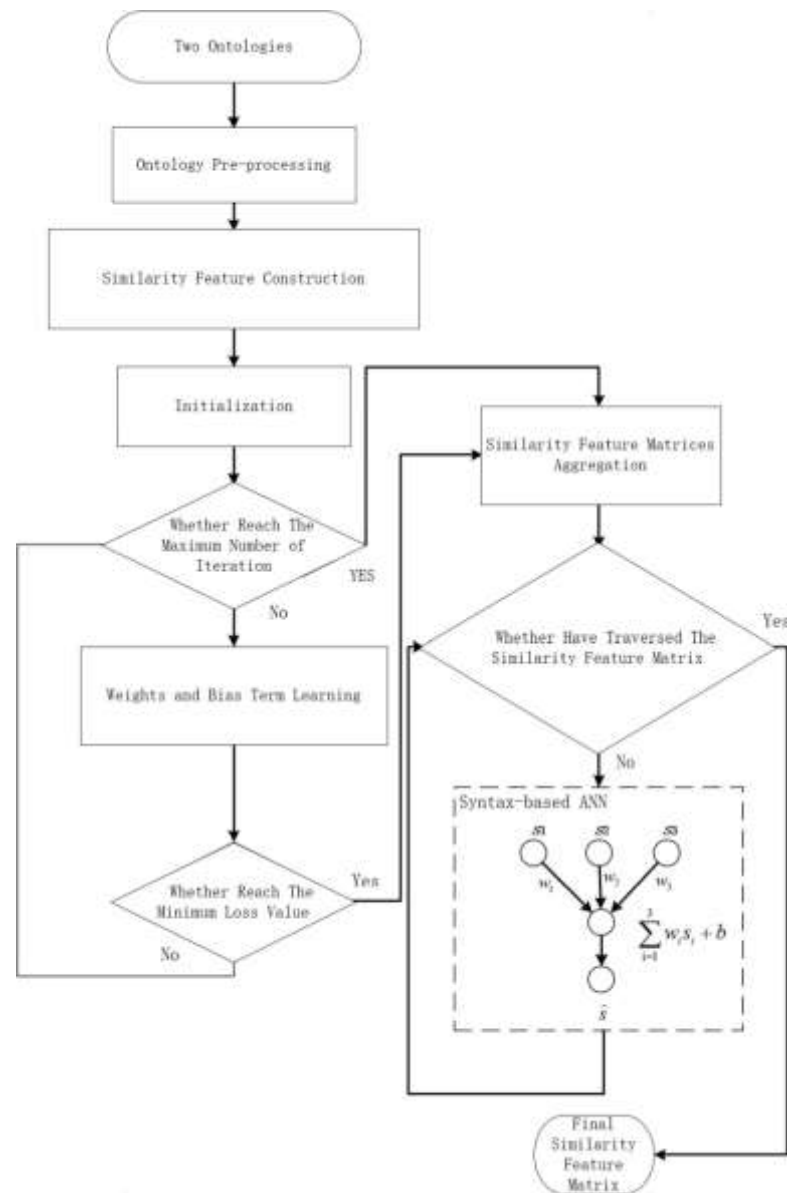


Figure 5. The Flow Chart of Syntax-based ANN Ontology Matching Technique.

Similarity Feature Matrix Maintenance

similarity feature matrices are generated by context-based ANN and syntax-based ANN, respectively. Elements with higher values in the similarity feature matrix represent the high confidences in the alignments. Since these two ANNs find alignments based on different semantic contexts, there is a high probability that they will find error alignments. To further maximize the alignment quality, we use the similarity feature matrix maintenance strategy to ensure the rationality of the matrices. At the same time, the irrationality of thematrices will lead to unreasonable alignments, as shown in Figure 6.

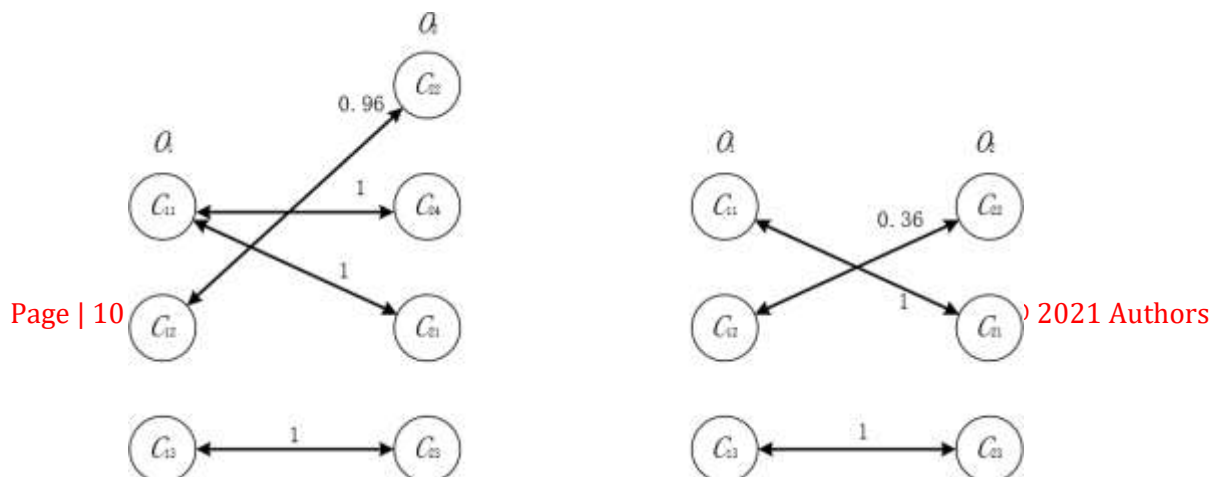


Figure 6. The Examples of Irrationality Alignments.

As shown in the Figure 6, the circle represents the concept to be matched, and the two-way arrow connects the matched concepts. Each pair of matched concepts has a similarity score, and the range of similarity score is $[0, 1]$. For example, the similarity score of c_3 and $c_{.3}$ is 1. To enhance the alignments' quality, we need to select alignments where only one ANN can produce a high similarity score, and remove alignments produced by the other ANN if the similarity score is below a certain threshold value t . For example, if an alignment $(c_2, c_{.2})$ produced by an context-based ANN has a high similarity score, and the alignment $(c_2, c_{.2})$ produced by a syntax-based ANN has a low similarity score, the alignment will be removed. In addition, a concept finds multiple alignments, such as $(c_1, c_{.1})$ and $(c_1, c_{.4})$, and they have the same similarity score in one ANN. The alignment is selected depends on the similarity score calculated by another ANN, and the one with the higher similarity score is selected as the final alignment. Finally, after the alignments are integrated using the OWA-AHP technique, a threshold is set and the alignments below that threshold are removed. The similarity feature matrix maintenance strategy is shown in Algorithm 2, where n and m represent the number of alignments produced by the two ANNs, respectively. Furthermore, w represents a threshold, and alignments with similarity scores less than w are deleted and t represents the a high confidence value that prove the alignment is correct.

Algorithm 2 Similarity Feature Matrix Maintenance

Input: Alignments are generated by W_1 , the set $A = (c_1, c_{.1}), (c_2, c_{.2}), \dots, (c_n, c_{.n})$, alignments are generated by W_2 , the set $B = (c_1, c_{.1}), (c_2, c_{.2}), \dots, (c_m, c_{.m})$

Output: Calculation results of three evaluation indexes

- 1: The alignment set C is obtained by extracting the correct alignment in A and B
- 2: **for** $i=0; i < C.size(); i++$ **do**
- 3: Set all rows and columns of the corresponding elements in W_1 and W_2 for the alignment $C[i]$ to 0
- 4: **end for**
- 5: **for** $i=0; i < W_1.length; i++$ **do**
- 6: **for** $j=0; j < W_1[i].length; j++$ **do**
- 7: **if** $W_1[i][j] > t$ and $W_2[i][j] > t$ **then**
- 8: Get the alignment corresponding to $W_1[i][j]$ and $W_2[i][j]$
- 9: Add the alignment to the set C
- 10: **else**
- 11: Alignment was further extracted using the OWA and AHP integration strategy
- 12: **if** the alignments' confidence value $> w$ **then**
- 13: Add the alignment to the set C
- 14: **end if**
- 15: **end if**
- 16: **end for**
- 17: **end for**
- 18: **if** There are one-to-many alignments in the set C **then**
- 19: Calculate their similarity scores i_1 and i_2 in the context-based ANN and calculate their similarity scores i_3 and i_4 in the syntax-based ANN
- 20: **if** $i_1 + i_3 > i_2 + i_4$ **then**
- 21: The alignment corresponding to i_2 is removed from the set C
- 22: **else**
- 23: The alignment corresponding to i_1 is removed from the set C
- 24: **end if**
- 25: **end if**
- 26: The evaluation indices of the proposed approach are calculated by Equations (6)–(8)

Ordered Weighted Average Operator With Analytic Hierarchy Process

In this work, we first use OWA to integrate different similarity feature matrices generated by various similarity measures, and then AHP is used to aggregate two ANNs' matching results. The motivation behind using OWA lies on the fact that linear combination method, e.g., the weighted average aggregation strategy, treats each ANN's matching results as a black box and use a uniform weight to aggregate all the mappings' similarity feature values inside it, which ignores entity mappings' preferences on different matchers and reduce the final alignment's quality. To further enhance the final result's quality, we use AHP to aggregate different ANNs by taking into consideration their relative contributions.

Ordered Weighted Average Operator

The use of a single similarity feature is not suitable for all ontology matching tasks, OWA is originally used for ontology matching. In order for all the similarity features to be used, this approach combines three different similarity features for matching. The OWA method assigns weights to descriptions based on the importance of their different semantic descriptions, that is, OWA assigns different weights to various similarity features to obtain the final similarity feature. We use AHP adaptive ontology alignment aggregating technique to aggregate different neural networks. Next, we will introduce OWA in detail.

Given a set $V_1 = (a_1, a_2, \dots, a_n)$, $a_i \in [0, 1]$, $1 \leq i \leq n$, and the set of weights that related to OWA operator is $W = (w_1, \dots, w_n)$. After sorting the elements of the set V_1 in

descending order, we obtain the set $V_2 = (b_1, b_2, \dots, b_n)$, where b_j is the j -th highest value in

V_1 . An OWA operator is a mapping function $F : I_n \rightarrow I$, $I \in [0, 1]$.

$$F(a_1, \dots, a_n) = \sum_{i=1}^n w_i b_i \quad (11) \quad \times$$

where $w_i \in [0, 1]$, and $\sum_{i=1}^n w_i = 1$. Note that a weight w_i is associated with a particular ordered position i of the arguments. w_i is defined as follows:

$$w_i = Q(i/n) - Q((i-1)/n), i = 1, 2, \dots, n \quad (12)$$

where Q is a non-decreasing proportional fuzzy linguistic quantifier [26]. We need to point out that, if $r < a$, $Q(r) = 0$; if $a \leq r \leq b$, $Q(r) = (r - a)/(b - a)$; and if $r > b$, $Q(r) = 1$; $0 \leq a, b, c \leq 1$, where a and b are the predefined thresholds [26].

Analytic Hierarchy Process

AHP approach is often used to solve decision-making problems [27], in simple terms, it is to find the optimal strategy according to the common indicators of multiple strategies. The use of a single similarity feature is not suitable for all ontology matching tasks, to make our framework suitable for all ontology matching techniques, we need to consider not only context-based similarity feature, but also literal and linguistic based similarity features, which can be achieved by integrating ANNs of various similarity features. In this case, so we use AHP method to aggregate different similarity measures according to the degree of their contributions. In this work, AHP is used to assign weights to different ANNs based on their contributions.

In Table 1, we show the provisions of quantitative values between indicators.

Table 1. Regulation Of Quantization Value.

The Degree of Importance of Factor i Relative to Factor j	Quantitative Values
Equally importance	1
A little important	3
Strongly important	5
Very important	7
Extremely important	9
The intermediate value of two adjacent judgments	2, 4, 6, 8
The reciprocal of a_{ij}	$1/a_{ij}$

Then, in order to introduce the importance of indicators and the importance of indicators in a single strategy, we draw matrices between different indicators and different indicators in a single strategy according to Table 1. Then, we normalized the the numbers by column to obtain a new table. Finally, we

take the mean value by row in the matrices to obtain the weight of indicators in the strategy and the proportion of different indicators. In order to make our table more logical, we perform consistency test. If the judgment matrix meets the consistency test, the value of judgment indicator CR will be less than 0.1. In particular, CR , CI and λ are respectively defined as follows, where CR and CI are both indicators of consistency test, and λ is the eigenvalue of the matrix:

$$CR = \frac{CI}{RI}$$

$$CI = \frac{\lambda - n}{n - 1} \quad (13)$$

(14)

$$\lambda = \sum_{i=1}^n \frac{[AW]_i}{nw_i} \quad (15)$$

nw_i

λ is computed to complete consistency check, A represents the matrix which is calculated by Table 1, W represents the column vector of weight, n is the number of indicators, and w_i is the weight of the i -th indicator. RI is obtained mainly by table lookup, which is the indicator of the consistency test. Table 2 shows the RI value used in this work.

Table 2. RI Value.

n	1	2	3	4	5	6	7	8	9	10
RI value	0	0	0.52	0.89	1.12	1.26	1.36	1.41	1.46	0.49

If the matrices all meet the consistency tests, the weights between different indexes and the proportion of different indexes relative to different strategies are reasonable. Finally, we recombine the weights of different indicators and the weights of different indicators under different strategies to synthesize a final matrix, the matrix of each line represent different indicators, each column represents a different strategy, and one column in the matrix represents the proportion of different indicators, we multiply and sum the proportion of indicators in the strategy and the proportion of different indicators respectively, and finally obtain the proportion of different strategies.

Adaptive Ontology Alignment Aggregation Strategy

Context-based ANN actually uses context-based similarity feature, according to the relationship between words to match. Syntax-based ANN, using the literal-based similarity feature and the linguistic-based similarity feature in parallel, which according to string and semantic information to match. The neural network that only have single similarity feature cannot be applied to all ontology matching tasks, in order to propose an ANN-based ontology matching technique with a wide range of applications, we propose to use all three similarity features in the parallel matching framework of ANN-based ontology matching technique, and use AHP aggregation technique to aggregate the ANNs of the three similarity features. This is equivalent to building a comprehensive similarity feature to match.

Ontology matching generally selects the largest element of the three similarity feature matrices corresponding to the three descriptions to aggregate into a similarity feature matrix, which does not consider the important relationship between the three descriptions. To solve this problem, the OWA approach is used to aggregate the three descriptions of Id, Label and Comments, Id is considered to be the most important according to the [26]. According to Equation (12), all three weights are one-third.

AHP approach was used to aggregate the two ANNs. By studying the underlying layers of neural

networks, we selected the number of input neurons(A_1), the number of neurons in the hidden layer(A_2) and the number of iterations of the neural network(A_3) as the indicators of the two networks, these indicators are the basic indicators to configure the neural networks. We list the matrices of the three indicators as follows.

The element in the i -th row and the j -th column in the Table 3 represents the importance degree of the i -th index relative to the j -th index. w_i represents the weight value of the i -th index which was computed. The rightmost column in the table calculates AW_i . Through the Equations (13)–(15) and table lookup, we can calculate that λ is about 3.0093, CI is 0.00465, RI is 0.52, and CR is 0.00894, far less than 0.1. So the table meets the consistency test and is correct. Similarly, we obtained the matrices of different indicators between strategies as follows.

Table 3. The Matrix Between The Three Indices.

Indices	A_1	A_2	A_3	w_i	AW_i
A_1	1	1/4	1/6	0.0893	0.2681
A_2	4	1	1/2	0.3238	0.97445
A_3	6	2	1	0.5869	1.7703

Tables 4–6 also meet the consistency test through calculation. Finally, we multiply and sum the proportion of indicators in the strategy and the proportion of different indicators respectively, and finally obtain the proportion of different strategies, as shown in Table 7.

Table 4. The Proportion Of A_1 In The Strategies.

A_1	Syntax-Based ANN	Context-Based ANN	w_i	AW_i
Syntax-based ANN	1	1/3	1/4	0.5
Context-based ANN	3	1	3/4	1.5

Table 5. The Proportion Of A_2 In The Strategies.

A_2	Syntax-Based ANN	Context-Based ANN	w_i	AW_i
Syntax-based ANN	1	1/3	1/4	0.5
Context-based ANN	3	1	3/4	1.5

Table 6. The Proportion Of A_3 In The Strategies.

A_3	Syntax-Based ANN	Context-Based ANN	w_i	AW_i
Syntax-based ANN	1	6	6/7	12/7
Context-based ANN	1/6	1	1/7	2/7

Table 7. The Proportion Between Strategies.

Indices	The Proportions of Indicators	Syntax-Based ANN	Context-Based ANN
A_1	0.0893	1/4	3/4
A_2	0.3238	1/4	3/4
A_3	0.5869	6/7	1/7

The syntax-based ANN's column represents the proportion of indicator A_i in the syntax-based ANN strategy, and the context-based ANN's column represents the proportion of indicator A_i in the context-based ANN strategy. Through the final calculation, syntax-based ANN accounted for 0.6063, context-based ANN accounted for 0.3937.

Experiment

Experimental Configuration

To evaluate the proposed adaptive alignment integration technique, we use OAEI's benchmark and

Anatomy data sets, and some real sensor ontologies to test our approach. A brief descriptions on the testing data sets are shown in Table 8. OAEI's Benchmark is a famous data set for testing different ontology matching techniques' performance. In addition, those ontologies in the OAEI's Anatomy and real sensor ontology matching tasks are all famous ones in practice. All the ontologies are developed with the English language.

With respect to the configuration of context-based ANN, we empirically set the iteration number to 10, context window length to 1, and the vector dimension to 3. To ensure high alignment quality, t is set to 0.9, the threshold of similarity feature matrix is set to 0.6, and the maximum number of iterations of syntax-based ANN is set to 1000. In addition, we conducted the controlled experiments that only one ANN is used for ontology matching and compare the f-measure values of the proposed method on benchmark with the f-measure values of context-based ANN ontology matching technique and syntax-based ANN ontology matching technique, as shown in Table 9. We need to explain that one single matcher can not ensure its effectiveness in all matching tasks due to the complex intrinsic of heterogeneous contexts, and therefore it is usually necessary to comprehensively consider multiple matchers to enhance the result's confidence. We compared our proposed adaptive integration technique with other traditional integration methods in Table 10. Table 11 shows the comparison of f-measure values between our approach and current advanced methods on the benchmark data sets. Figures 7 and 8 show the comparisons among our approach and other advanced techniques on OAEI's Benchmark and Anatomy track, respectively.

Table 8. A Brief Description On Testing Data Sets.

ID	Simple Description
1XX	Two identical ontologies
2XX	Two ontologies with different lexical, linguistic or structural characters
3XX	Ontologies in reality
mouse	Technical terms of a lot of mouse of the anatomy
human	Technical terms of a lot of human of the anatomy
SSN	Semantic aspects of ontologies are considered
CRISO	Sensors and deployments were described
MMI	The oceanographic devices, sensors and sample were described

Table 9. Comparison With ANN-based Matching Techniques On OAEI's Benchmark.

Testing Case Based	Our Approach	Context-	Syntax-Based	Testing Case	Our Approach	Context-Based	Syntax-
		ANN	ANN			ANN	ANN
101	1.00	1.00	1.00	231	1.00	1.00	0.93
103	1.00	1.00	1.00	232	1.00	1.00	0.93
104	1.00	1.00	1.00	233	1.00	1.00	0.98
201	0.99	0.74	0.92	236	1.00	1.00	1.00
202	0.59	0.04	0.04	237	1.00	1.00	0.88
203	1.00	0.38	0.94	238	1.00	1.00	0.89
204	1.00	0.76	0.93	239	1.00	0.98	0.98
205	0.99	0.74	0.90	240	1.00	0.99	0.94
206	0.97	0.73	0.76	241	1.00	1.00	1.00
207	0.98	0.73	0.78	246	1.00	0.98	0.98
208	0.96	0.35	0.90	247	1.00	0.99	0.94
209	0.50	0.11	0.41	248	0.58	0.04	0.04
210	0.45	0.12	0.36	249	0.58	0.04	0.04
221	1.00	1.00	0.91	252	0.44	0.04	0.03
222	1.00	1.00	0.88	253	0.58	0.04	0.04
223	1.00	1.00	0.89	301	0.88	0.63	0.43
224	1.00	1.00	0.93	302	0.73	0.16	0.57
225	1.00	1.00	0.93	303	0.85	0.64	0.73
228	1.00	1.00	1.00	304	0.97	0.56	0.80
230	1.00	1.00	0.96				

Table 10. Comparison Our Approach With Traditional Aggregation Methods On OAEI's Benchmark.

Testing Case	Our Approach	Context-Based	Syntax-Based	Testing Case	Our Approach	Context-Based	Syntax-Based
101	1.00	1.00	1.00	231	1.00	1.00	1.00
103	1.00	1.00	1.00	232	1.00	1.00	1.00
104	1.00	1.00	1.00	233	1.00	1.00	1.00
201	0.99	0.98	0.98	236	1.00	1.00	1.00
202	0.59	0.28	0.59	237	1.00	1.00	1.00
203	1.00	1.00	1.00	238	1.00	1.00	1.00
204	1.00	1.00	1.00	239	1.00	1.00	1.00
205	0.99	0.99	0.99	240	1.00	1.00	1.00
206	0.97	0.97	0.97	241	1.00	1.00	1.00
207	0.98	0.97	0.97	246	1.00	0.98	0.98
208	0.96	0.96	0.96	247	1.00	0.99	0.99

Max. Min. Avg. Testing
Our

Table 11. Comparison With OAEI's Participants On OAEI's Benchmark.

Testing Case	Edna TaxoMap	AgrMaker SNN-OM	AROMA Our Approach	ASMOV	CODI	Ef2Match	Falcon	GeRMeSMB	MapPSO	RiMOM	SOBOM			
101	1.00	0.99	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.51	1.00	1.00
103	1.00	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.51	1.00	1.00
104	1.00	0.99	0.99	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	0.51	1.00	1.00
201	0.04	0.92	0.95	1.00	0.13	0.77	0.97	0.94	0.42	1.00	0.95	0.51	0.97	0.99
202	0.03	0.89	0.00	0.88	0.00	0.08	0.00	0.39	0.05	0.81	0.64	0.02	0.00	0.59
203	1.00	0.98	0.80	1.00	0.86	1.00	1.00	0.98	1.00	1.00	1.00	0.49	1.00	1.00
204	0.93	0.97	0.97	1.00	0.74	0.99	0.96	0.98	0.98	1.00	0.99	0.51	0.99	1.00
205	0.34	0.92	0.95	0.99	0.28	0.84	0.97	0.99	0.73	0.99	0.96	0.51	0.98	0.99
206	0.54	0.93	0.95	0.99	0.39	0.87	0.94	0.92	0.85	0.99	0.96	0.51	0.96	0.97
207	0.54	0.93	0.95	0.99	0.42	0.87	0.96	0.96	0.81	0.99	0.96	0.51	0.00	0.98
208	0.93	0.96	0.58	1.00	0.61	0.95	0.98	0.95	0.79	1.00	1.00	0.44	0.00	0.96
209	0.35	0.88	0.37	0.92	0.22	0.47	0.65	0.59	0.16	0.87	0.71	0.14	0.00	0.50
210	0.54	0.93	0.18	0.96	0.24	0.38	0.66	0.58	0.32	0.85	0.82	0.15	0.00	0.45
221	1.00	0.97	0.99	1.00	0.98	1.00	1.00	1.00	1.00	1.00	1.00	0.51	1.00	1.00
222	0.98	0.98	0.99	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	0.46	1.00	1.00
223	1.00	0.95	0.93	1.00	1.00	1.00	1.00	0.96	0.98	0.98	0.99	0.45	1.00	1.00
224	1.00	0.99	0.97	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00	0.51	1.00	1.00
225	1.00	0.99	0.99	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	0.51	1.00	1.00
228	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
230	0.85	0.90	0.93	0.97	0.98	0.97	0.97	0.94	0.98	0.97	0.97	0.49	1.00	1.00
231	1.00	0.99	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.51	1.00	1.00
232	1.00	0.97	0.97	1.00	0.97	1.00	0.99	1.00	1.00	1.00	1.00	0.51	1.00	1.00
233	1.00	1.00	1.00	1.00	0.94	1.00	1.00	0.98	1.00	1.00	1.00	1.00	1.00	1.00
236	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
237	0.98	0.98	0.97	1.00	0.99	1.00	0.99	1.00	0.99	1.00	1.00	0.46	1.00	1.00
238	1.00	0.94	0.92	1.00	0.99	1.00	0.99	0.96	0.97	0.98	0.98	0.45	1.00	1.00
239	0.50	0.98	0.98	0.98	0.98	0.98	1.00	0.98	0.98	0.98	0.98	0.94	0.98	1.00
240	0.55	0.91	0.83	0.98	0.95	0.98	1.00	0.85	0.92	0.94	0.98	0.88	1.00	1.00
241	1.00	1.00	0.98	1.00	0.94	1.00	1.00	0.98	1.00	1.00	1.00	1.00	1.00	1.00
246	0.50	0.98	0.97	0.98	0.98	0.98	1.00	0.98	0.98	0.98	0.95	0.94	0.98	1.00
247	0.55	0.88	0.80	0.98	0.98	0.98	1.00	0.91	0.89	0.94	0.98	0.88	0.00	1.00
248	0.03	0.72	0.00	0.87	0.00	0.02	0.00	0.37	0.05	0.64	0.48	0.02	0.00	0.58
249	0.03	0.88	0.02	0.88	0.02	0.08	0.00	0.35	0.05	0.78	0.64	0.02	0.00	0.58
252	0.02	0.78	0.00	0.86	0.00	0.08	0.00	0.37	0.02	0.68	0.50	0.02	0.00	0.44
253	0.03	0.72	0.02	0.87	0.02	0.02	0.00	0.42	0.07	0.61	0.47	0.02	0.00	0.58
301	0.59	0.59	0.73	0.86	0.38	0.71	0.78	0.71	0.64	0.73	0.84	0.43	0.86	0.88
302	0.43	0.32	0.35	0.73	0.59	0.71	0.71	0.41	0.04	0.73	0.74	0.40	0.75	0.73
303	0.00	0.78	0.59	0.83	0.65	0.83	0.77	0.00	0.00	0.86	0.50	0.36	0.87	0.85
304	0.83	0.86	0.84	0.95	0.74	0.95	0.94	0.77	0.72	0.94	0.91	0.52	0.94	0.97

Testing on OAEI's Data Sets

Figure 7 shows the testing results about recall, precision and f-measure of our approach on the benchmark test sets, where the abscissa represents the test sets of Benchmark, and the ordinate represents the results of three evaluation indexes. Table 9 also shows the results of the controlled experiments. It can be seen that the adaptive ontology alignment aggregation technique is a great improvement over the single ANN-based ontology matching technique on the benchmark set. In particular, the improvement is more obvious in the five test sets of 202,248,249,252,253. This means that the aggregation strategy we put forward has some progressive significance. As can be seen from Table 10, the mean value of our method on the OAEI test set is 0.90, which is better than the maximum method and the average method. In general, the adaptive aggregation technique proposed by us is no worse than the traditional aggregation methods and has certain progress significance. Table 11 compares the results of our approach with some traditional efficient methods on the benchmark test sets [28], where SNN-OM [12] is a Siamese neural network based ontology matching technique, which combines alignment refinement technique to achieve the high quality alignment. It is a relatively advanced approach in the field of ANN-based ontology matching in the latest years. As can be seen from

the table, our approach significantly outperforms other efficient ontology matching methods on the whole benchmark test sets.

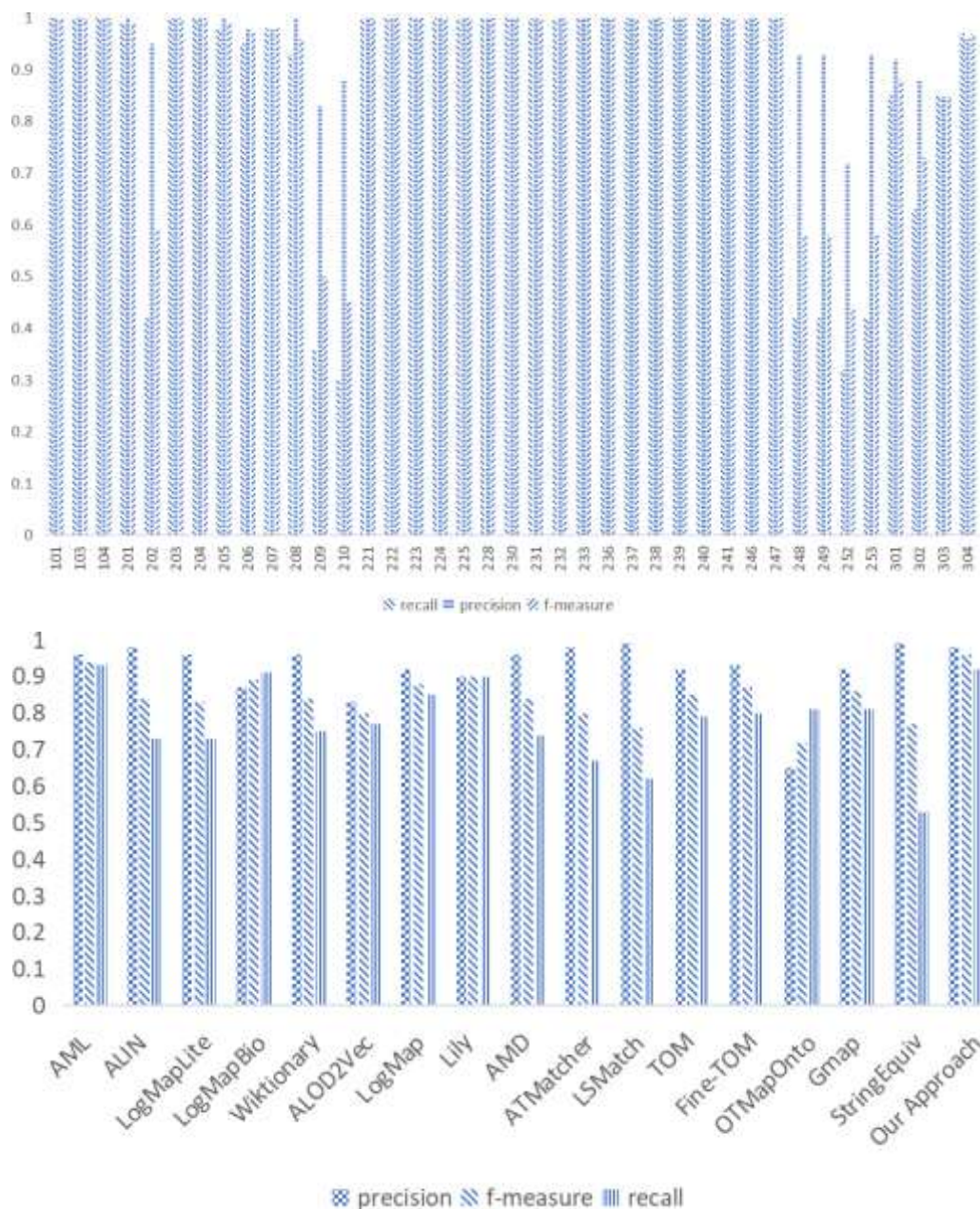


Figure 7. The Testing Results of Our Approach on OAEI's Benchmark.

Figure 8. Comparison With OAEI's Participants On OAEI's Anatomy Track.

The results of our method on testing cases 202, 209, 210, 248, 249, 252 and 253 are not very good. The reason is that the ontologies' heterogeneity characteristics in these tasks require using more information than the semantic information and syntactic information of the entity. Similarly, the results on testing cases 301, 302 and 303 are not the best, which is due to that we dedicate to find the alignment with cardinality one to one, but their results' cardinality is more to more, i.e., one source entity could be mapped with more than one target entities, and vice versa.

Figure 8 shows the comparison between the results of some advanced methods [29] in the data sets of Anatomy and the results of our approach, where the abscissa represents different advanced methods, and the ordinate represents the values of the three evaluation indexes. It can be seen that the testing results of our approach are better than those of other advanced methods, and the values of the three evaluation indexes tested by our approach are all above 0.9. From the above table, we can see that our

approach is a relatively efficient approach.

Testing On Real Sensor Ontologies

In order to further verify the efficiency and practicability of our proposed approach, we use real sensor ontologies to test it. The sensor ontology we use is described as follows, SSN [30] is the most used global reference ontology that has been developed in the domain of sensor networks, which investigated the efficiency of ontology matching technique in semantic level. One of the main purposes of MMI device [31] is developing an extensible ontology of marine devices, the sensor ontology describe the oceanographic devices, sensors and samplers. CRISO [32] describe the sensors and deployments. The advanced sensor ontology matching systems are ASMOV [33], CODI [34], SOBOM [35] and FuzzyAlign [36], and the testing results of our approach and the results of advanced methods are compared as shown in Figures 9–11.

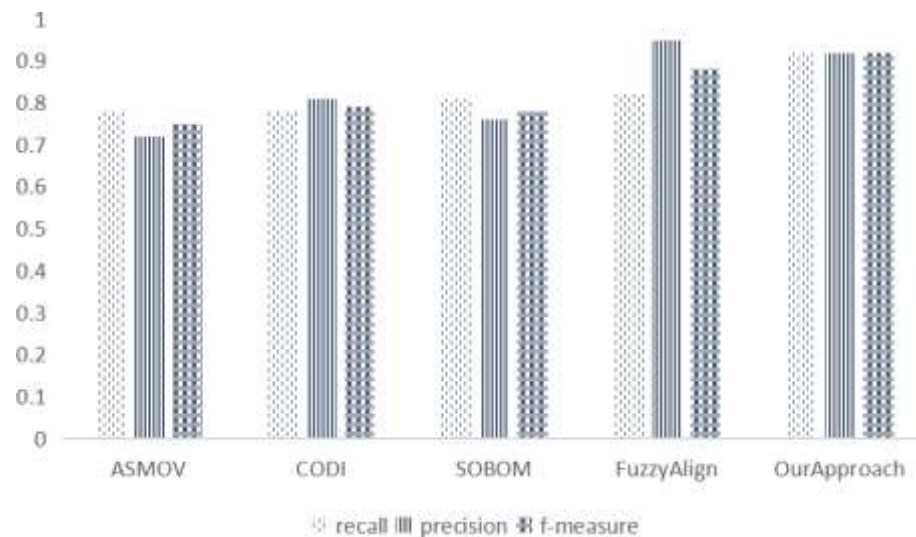


Figure 9. Comparison of Our Approach and the Advanced Sensor Ontology Matching Systems on CSIRO-SSN matching task.

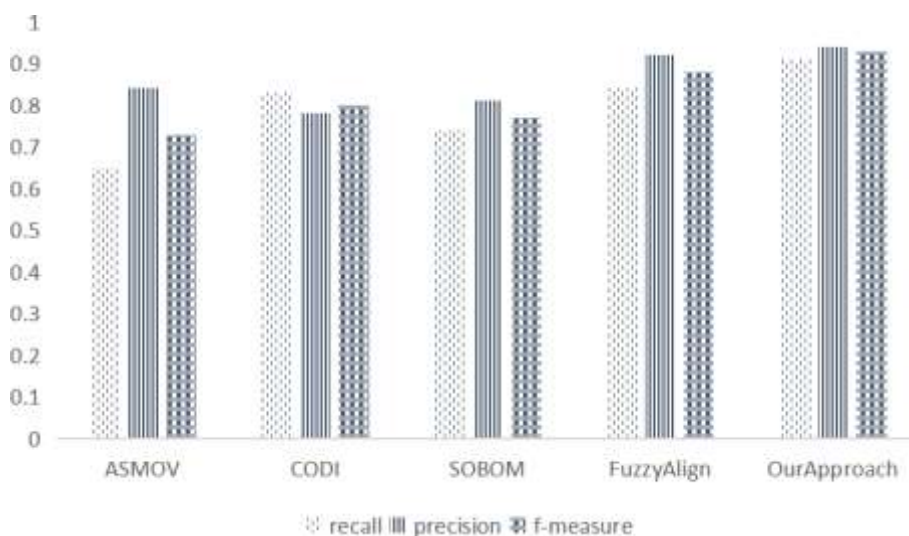
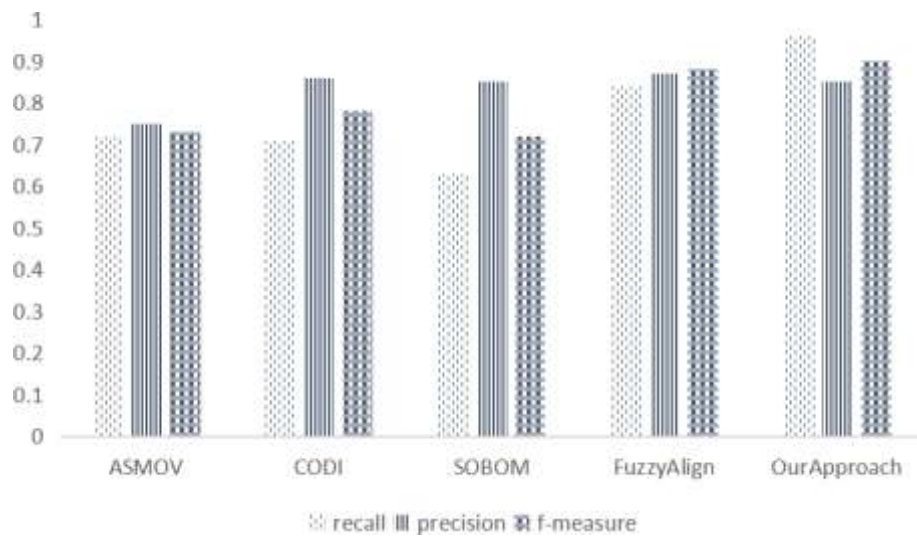


Figure 10. Comparison of Our Approach and the Advanced Sensor Ontology Matching Systems on



MMI-SSN matching task.

Figure 11. Comparison of Our Approach and the Advanced Sensor Ontology Matching Systems on CSIRO-MMI matching task.

It can be seen from these three figures that the recall rate of our approach is much higher than that of other methods, which indicates that the framework we proposed is highly practical. Meanwhile, it can be seen from the comparison results of f-measure value that our approach is superior to other methods. However, the precision of our approach on these real sensor ontology matching tasks is relatively weak, which may be due to the lack of some necessary descriptions in these sensor ontologies, which leads to the failure of context-based neural networks to perform their proper capabilities. For example, the entity whose Id is “ActuatableProperty” in the SSN ontology lacks related Comments, etc., which makes the context-based neural network unable to find the alignment of the entity whose Id is “ActuatableProperty”. In addition, there are some very professional terms in these sensor ontologies, such as “hygrometer” and “humistor”. If we can use a specialized sensor thesaurus to replace WordNet, the results of these three evaluation indicators will be higher. In Figure 12, we show a small fragment of the SSN ontology and the links to the MMI Device and CSIRO ontologies. The dotted line with the arrow connects the alignments of the two sensor ontologies. For example, “system” in the MMI Device and “system” in the SSN ontology is an alignment. Moreover, entities connected by solid arrows in the same sensor ontology represent their relationships with children and parents. For example, in the MMI ontology, the superclass of “System” is “Process”, and the subclasses of Process are “ProcessOutput” and “ProcessInput”.

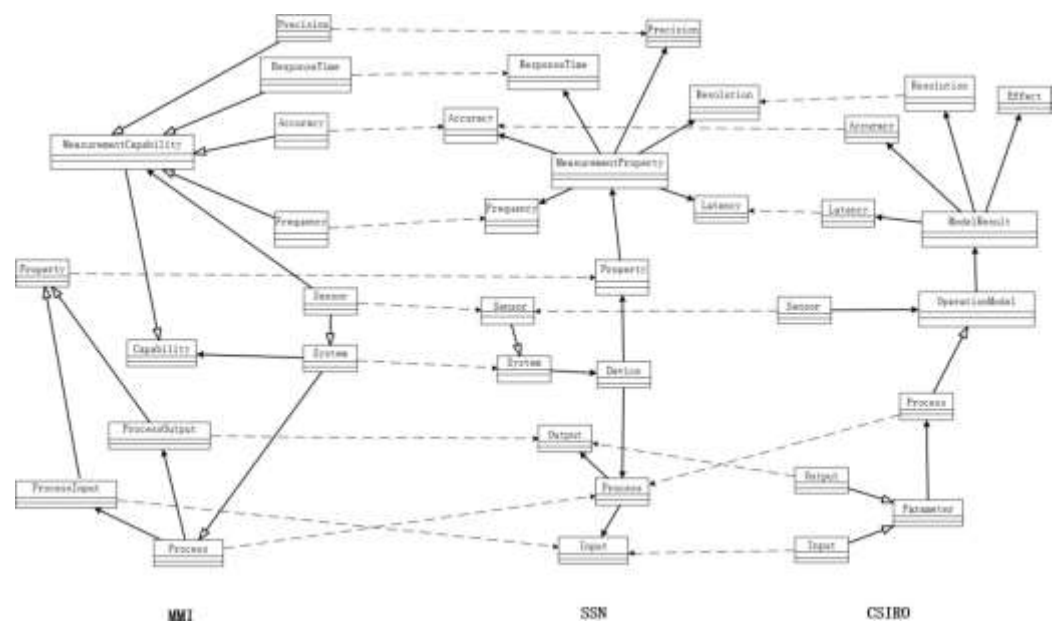


Figure 12. Small fragment of alignments between three sensor ontologies: MMI Device ontology, SSN ontology and CSIRO ontology.

Conclusions

In order to enhance the quality of ontology alignment, this work proposes a novel ontology matching technique that adaptively aggregate different ANN-based ontology matching techniques. In particular, we first propose a framework of aggregating various ANN-based ontology matching techniques. Then, we propose the context-based ANN ontology matching technique and the syntax-based ANN ontology matching technique to match two different ontologies, and then we use similarity feature matrix maintenance strategy to improve the quality of alignments. After that, OWA and AHP are used to adaptively aggregate two ANN-based ontology matching techniques to further enhancing the quality of final alignments. In the experiment, our approach significantly outperforms the single ANN-based matching technique and other state-of-the-art ontology matching systems on OAEI's Benchmark and Anatomy track, and real sensor ontology matching tasks.

Although the experimental results are promising, there are still some problems that need to be addressed. First, the selection of different indicators of ANN is subjective, and the further study should be made on indicator selection in the future. Second, our approach's f-measure on some test cases of benchmark, three real sensor ontology matching tasks and Anatomy track are less than 1.00, which means its effectiveness can be further improved. To this end, we are interested in introducing the quality-improving strategy, such as the reasoning-based correspondence pruning method that can reduce the error correspondences according to the ontology's concept hierarchy structure. Furthermore, the alignment refining technique [12] is also a feasible method of removing the incorrect correspondences. Further strengthen the entity attributes, instances of matching and find the relationship between entities is also an approach to strengthen the quality of matching. Finally, we are interested in applying our approach in more practical matching tasks, such as biomedical ontology matching [37] and knowledge graph matching [38], to test its robustness.

References

1. Berners-Lee, T., J. Hendler, and O. Lassila. "The semantic web." *Science America*. 2001;284(34):34-43.
2. Gruber, T.R. Toward guidelines for the creation of ontologies used for knowledge sharing? *Int. J. Hum.-Comput. Stud.* 1995,\s43, 907-928.
3. Shvaiko, P., & Euzenat, J. What we know now, and what lies ahead, in the field of ontology matching. *IEEE Trans. Knowl. Data Eng.* 2011, 25, 158-176.
4. Abiodun, O.I.; Jantan, A.; Omolara, A.E.; Dada, K.V.; Mohamed, N.A.; Arshad, H. State-of-the-art in artificial neural network applications: A survey. *Heliyon* 2018, 4, e00938.
5. Iyer, V.; Agarwal, A.; Kumar, H. VeeAlign: Multifaceted Context Representation Using Dual Attention for Ontology Alignment. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online and Punta Cana, Dominican Republic, 7-11 November 2021*; pp. 10780-10792.
6. Feng, Y.; Fan, L. Ontology semantic integration based on convolutional neural network. *Neural Comput. Appl.* 2019, 31, 8253-8266.
7. Portisch J; Hladik M; Paulheim H. Background knowledge in ontology matching: A survey. *Semantic Web* 2022, 2022, 1-55.
8. Chakraborty, J.; Bansal, S.K.; Virgili, L.; Konar, K.; Yaman, B. Ontoconnect: Unsupervised ontology alignment using recursive neural network. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing, Virtual Event, 22-26 March 2021*; pp. 1874-1882.
9. Jiang, C.; Xue, X. Matching biological ontologies with long short-term memory networks. In *Proceedings of the 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Seoul, Republic of Korea, 16-19 December 2020*; pp. 2484-2489.
10. Xue, X.; Jiang, C.; Zhang, J.; Hu, C. Biomedical Ontology Matching Through Attention-Based Bidirectional Long Short-Term Memory Network. *J. Database Manag. (JDM)* 2021, 32, 14-27.
11. Zheng, T.; Gao, Y.; Wang, F.; Fan, C.; Fu, X.; Li, M.; Zhang, Y.; Zhang, S.; Ma, H. Detection of medical text semantic similarity based on convolutional neural network. *BMC Med. Inform. Decis. Mak.* 2019, 19, 1-11.
12. Xue, X.; Wang, Y.; Hou, J. Ontology Alignment based on Instance utilizing NSGA-II. *J. Inf. Sci.* 2015, 41, 58-70.
13. Chen, J.; Jiménez-Ruiz, E.; Horrocks, I.; Antonyrajah, D.; Hadian, A.; Lee, J. Augmenting ontology alignment by semantic embedding and remote supervision. In *Proceedings of the European Semantic Web Conference, Virtual, 24-28 October 2021*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 392-408.
14. Huang, J.; Dang, J.; Vidal, J.M.; Huhns, M.N. Ontology matching using an artificial neural network to

learn weights. In Proceedings of the IJCAI Workshop on Semantic Web for Collaborative Knowledge Acquisition, Hyderabad, India, 7 January 2007; Volume 106.

15.

Huang, J.; Dang, J.; Huhns, M.N.; Zheng, W.J. Use artificial neural network to align biological ontologies. *BMC Genom.* 2008, 9, 1–12.

16.

Djeddi, W.E.; Khadir, M.T. Ontology alignment using artificial neural network for large-scale ontologies. *Int. J. Metadata Semant. Ontol.* 2013, 8, 75–92.

17.

Bulygin, L. Combining lexical and semantic similarity measurements with machine learning technique for ontology and schema matching issue. In Proceedings of the International Conference on Data Analytics and Management in Data Intensive Domains (DAMDID/RCDL), Moscow, Russia, 26–29 October 2018; pp. 245–249.

18. Xue, X.; Jiang, C.; Yang, C.; Zhu, H.; Hu, C. Artificial Neural Network Based Sensor Ontology Matching Technique. In Proceedings of the Companion Proceedings of the Web Conference, Ljubljana Slovenia; 19–23 April 2021; pp. 44–51.

19. Ali Khoudja, M.; Fareh, M.; Bouarfa, H. A novel supervised learning based ontology matching technique employing neural networks. In Proceedings of the International Conference Europe Middle East & North Africa Information Systems and Technologies to Support Learning, Fez, Morocco, 25–27 October 2018; Springer: Cham, Switzerland, 2018; pp. 542–551. \s20. Euzenat, J.; Shvaiko, P. *Ontology Matching*; Springer: Berlin/Heidelberg, Germany, 2007; Volume 18.

21. Fellbaum, C. *WordNet. In Theory and Applications of Ontology: Computer Applications*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 231–243.

22.

Brown, P.F.; Della Pietra, V.J.; Desouza, P.V.; Lai, J.C.; Mercer, R.L. Class-based n-gram models of natural language. *Comput. Linguist.* 1992, 18, 467–480.

23. Stoilos, G.; Stamou, G.; Kollias, S. A string metric for ontology alignment. In Proceedings of the International Semantic Web Conference, Galway, Ireland, 6–10 November 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 624–637.

24. Wu, Z.; Palmer, M. Verb semantics and lexical selection. *arXiv* 1994, arXiv:cmp-lg/9406033. \s25. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimate of word representations in vector space. *arXiv* 2013, arXiv:1301.3781.

26. Ji, Q.; Haase, P.; Qi, G. Combination of similarity measurements in ontology matching using the owa operator. In *Recent Developments in the Ordered Weighted Averaging Operators: Theory and Practice*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 281–295.

27. Saaty, T.L. Decision making—the analytic hierarchy and network processes (AHP/ANP). *J. Syst. Sci. Syst. Eng.* 2004, 13, 1–35. \s28. Acampora, G.; Loia, V.; Vitiello, A. Enhancing ontology alignment by a memetic aggregation of similarity measurements. *Inf. Sci.* 2013, 250, 1–20. \s29. Wang, P.; Zou, S.; Liu, J.; Ke, W. Matching biological ontologies using GCN-based feature propagation. *Math. Biosci. Eng. MBE* 2022, 19, 8479–8504. \s30. Compton, M.; Barnaghi, P.; Bermudez, L.; Garcia-Castro, R.; Corcho, O.; Cox, S.; Graybeal, J.; Hauswirth, M.; Henson, C.; Herzog, A.; et al. The SSN ontology of the W3C semantic sensor network incubation group. *J. Web Semant.* 2012, 17, 25–32. \s31. Rueda, C.; Galbraith, N.; Morris, R.A.; Bermudez, L.E.; Arko, R.A.; Graybeal, J. The MMI device ontology: Enabling sensor integration. In Proceedings of the AGU Fall Meeting Abstracts, San Francisco, CA, USA, 13–17 December 2010; Volume 2010, p. IN44B–08. \s32. Compton, M. The Semantic Sensor Network: Ontology A Generic Language to Describe Sensor Assets. In Proceedings of the AGILE 2009 Pre-Conference Workshop Challenges in Geospatial Data Harmonisation, Hannover, Germany, 2 June 2009. \s33. Noessner, J.; Niepert, M.; Meilicke, C.; Stuckenschmidt, H. Leveraging terminological structure for object reconciliation. In Proceedings of the Extended Semantic Web Conference, Crete, Greece, 30 May–3 June 2010; Springer: Berlin/Heidelberg, Germany, 2010; pp. 334–348. \s34. Jean-Mary, Y.R.; Shironoshita, E.P.; Kabuka, M.R. Ontology matching with semantic verification. *J. Web Semant.* 2009, 7, 235–251. \s35. Xu, P.; Wang, Y.; Cheng, L.; Zang, T. Alignment Results of SOBOM for OAEI 2010. *Ontol. Matching* 2010, 203, 7–11. \s36. Fernandez, S.; Marsa-Maestre, I.; Velasco, J.R.; Alarcos, B. Ontology alignment architecture for semantic sensor web integration.

Sensors 2013, 13, 12581–12604. \s37. Faria, D.; Pesquita, C.; Mott, I.; Martins, C.; Couto, F.M.; Cruz, I.F. Tackling the issues of matching biological ontologies. J. Biomed. Semant. 2018, 9, 1–19. \s38. Xu, K.; Wang, L.; Yu, M.; Feng, Y.; Song, Y.; Wang, Z.; Yu, D. Cross-lingual knowledge graph alignment through graph matching neural network. arXiv 2019, arXiv:1905.11605