TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 09/16 pp2468–2484
DOI: 10.26599/TST.2025.9010003
Volume 31, Number 5, October 2026

Research Article

Ontology Alignment Approach Based on Attribute Self-Adaptation Mechanism and Heterogeneous Feature Fusion

Cheng Yang, Chunxia Zhang*, Yihao Chen, Xiaojun Xue, Yizhou Wang, and Zhendong Niu

Abstract: Ontology alignment is a crucial task in the field of knowledge fusion. It provides an essential basis for constructing large-scale high-quality knowledge graphs. However, the previous ontology alignment works face the three problems: lack of implicit semantic within reference mapping (i.e., labeled set), falsely high similarity between class embeddings, and imbalanced training data. To solve these problems, this paper proposes an active learning ontology alignment approach with attribute self-adaptation mechanism and heterogeneous feature fusion (ASHF). The active learning framework and attribute self-adaptation mechanism aim to avoid false positives aligned classes by reconstructing the reference mapping, and to obtain stable performance for sparse ontologies. The heterogeneous feature fusion strategy calculates the similarity between classes by selecting the more distinguished semantic features of classes. Experimental results on eight public datasets show that the proposed model outperforms the state-of-art methods, demonstrating the effectiveness and superiority of the proposed approach in this paper.

Key words: ontology alignment; knowledge graph; active learning; attribute self-adaptation mechanism; heterogeneous feature fusion

1 Introduction

The goal of knowledge graphs is to describe entities and relationships between entities in a graphical structure[1–3]. The knowledge graph has the general and domain knowledge required for artificial intelligence, and connects entities into a knowledge network through relationships. Hence, it is an efficient way to represent, organize, manage, and utilize dynamic and heterogeneous knowledge on the Internet[4]. Knowledge fusion technology aims to build a more complete high-quality knowledge base by integrating the knowledge from different sources[5–10]. Typically, ontology alignment is a crucial task in the domains of knowledge graph fusion and big data knowledge engineering.

The ontology alignment task is to identify ontologies with the same semantics[11]. Specifically, its goal is to identify equivalent relationships among classes, relations, and entities in different ontologies. Technically, ontology alignment focuses on the alignment on the conceptual level, while entity alignment stresses the alignment on the instance level. It provides an essential basis for constructing large-scale high-quality knowledge graphs, and has been widely applied into many downstream tasks including question answering and semantic retrieval. Furthermore, ontology alignment is of great significance to maintaining semantic consistency, knowledge sharing and integration[12–16].

In the literature, ontology alignment methods can be mainly divided into string-based, structure-based, and

• Cheng Yang, Chunxia Zhang, Yihao Chen, Xiaojun Xue, Yizhou Wang, and Zhendong Niu are with School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China. E-mail: chengyang@bit.edu.cn; cxzhang@bit.edu.cn; 3220231215@bit.edu.cn; xiaojunx@bit.edu.cn; yzwang@bit.edu.cn; zniu@bit.edu.cn.
* To whom correspondence should be addressed.
Manuscript received: 2024-02-03; revised: 2024-05-09; accepted: 2025-01-02

deep learning based methods. As a classical string based approach, the ontoEmma model[17] uses four artificially defined features including common names, aliases, definitions, and contexts of entities, and adopts a method to enrich entities in ontology with external entity definitions and contextual information.

For the family of the structure-based techniques, the Rafcom model[18] is a random forest classifier approach, which employs the terminology strategy and the structural information of the ontology. The AML model[19] is an interactive selection algorithm that utilizes alignment results returned by multiple ontology alignment methods to detect possible entity pairs. In addition, the LogMap framework[20] provides problematic alignments to users for verification.

The ontology alignment approaches based on deep learning are to employ deep neural networks to find the equivalent correspondence between entities in different ontologies. DeepAlignment[21] refines the pre-trained word vectors and derives ontology entity descriptions tailored for ontology matching tasks. VeeAlign[22] proposes a dual encoder to encode the text information and path information of a class. Furthermore, BERTMap[23] exploits the textual, structural, and logical information of the ontology to fulfill the ontology alignment task.

The present works have the following problems: (1) Since each sample among the labeled samples introduces different semantic information to the trained model, treating them equally may reduce the performance of the model; (2) Some pre-trained language models, such as BERT, have problems of contextual noise and falsely high similarity between class embedding vectors[24]. The contextual noise means that the same or similar texts in different contexts generate different embeddings, rather than the same embeddings. Specifically, BERT uses the pre-training method "Next Sentence Prediction", which makes it more sensitive to contexts. The randomness in training and the frequency noise in the data lead to the differences of the embedding of the same or similar texts in different contexts; and (3) The over-fitting problems are caused by imbalanced training data, that is, a large gap between the number of positive samples and negative samples.

To solve the problems above, this paper proposes an ontology alignment approach based on attribute self-adaptation mechanism and heterogeneous feature fusion (ASHF). It consists of four modules: attribute self-adaptation learning module, alignment labeling module, embedding generation module, and mapping prediction module. (1) The aim of the attribute self-adaptation learning module is to select the sample data for active learning. This module adopts the active learning framework to augment the semantic information of reference mapping set in our ASHF model. Concretely, it utilizes an improved genetic algorithm based on properly matching strategy to reconstruct the given reference mapping. That is, the algorithm takes into account both the string similarity between class pairs and the number of labels for simultaneously encoding attributes and categories, and adopts the principles of multi-population co-evolution and well-matched pair mating. Thereby, we select high-quality category pairs to be marked and inject the conceptual semantic information into the reference mapping by means of attribute augmentation, which effectively eliminate false positives in mapping, and optimize the results of ontology alignment. (2) The alignment labeling module reselects the reconstructed reference mapping and the semantic information is injected into the reference map. (3) In the embedding generation module, the fine-tuning corpus is generated by means of an unsupervised way, preventing the alignment model from overfitting through data augmentation. (4) Within the mapping prediction module, a pretrained language model robustly optimized bidirectional encoder representations from transformers (BERT) pretraining approach (RoBERTa) is first used to generate embedding vectors of categories, and then a heterogeneous feature fusion strategy is designed to compute the similarity between classes.

The main innovations and contributions of this paper are given as follows:

(1) In view of the fact that the present semi-supervised learning methods can only randomly select limited label pairs while ignoring the semantic information of class pairs to be labeled, an active learning framework for ontology alignment with attribute self-adaptation learning mechanism is proposed to avoid false positives aligned categories by reconstructing the reference mapping, and to obtain stable performance for sparse ontologies for reducing weight fluctuations.

(2) In order to solve the problems about contextual noise and falsely high similarity with class embedding vectors, a heterogeneous feature fusion is constructed

to capture the heterogeneous semantic similarity between class pairs by selecting the more distinguished semantic features of categories, thus significantly improving the accuracy of ontology alignment without recall decreasing.

(3) To address the problem about the imbalanced training data that there are too many non-synonyms and too few synonyms, a data augmentation approach is developed to augment the synonyms set in the corpus by random deletion, random transformation, and back translation of words, which can enhance the balance between data and avoid overfitting.

(4) Extensive experiments are conducted on eight public datasets in biomedical, material, and biology domains. The experimental results show that our proposed ASHF model outperforms the state-of-the-art methods, and the ablation experiments also demonstrate that each component of the ASHF model contributes to the ontology alignment results.

2 Related Work

The existing ontology alignment approaches can be divided into three categories: string-based methods[17], structure-based methods[18−20, 25], and the methods based on deep learning[21−23, 26, 27].

The string-based ontology alignment methods calculate the similarity between entity pairs based on their names or conceptual descriptions, and determine their similarity and correspondence by analyzing the string representation of entity names or identifiers. That kind of methods are usually based on the principles of text similarity and string matching, and mostly require manual feature extraction. The semantic similarity of entities is calculated by using manually selected features, such as entity names, entity definitions, and entity attributes, aiming to identify ontologies with similar names or identifiers in different knowledge graphs. The OntoEmma model[17] uses a technique for enriching entities within the ontology by integrating their external definitions and contextual information. Further, those additional definitions are employed to implement ontology alignment, and a neural architecture is developed to encode those definitions. The string-based methods usually have low computational cost and wide applications. However, they rely on the character level features of strings and are difficult to capture the semantic information of the ontology.

The structure-based methods do not require manually defining features, but encode the entire semantic web and calculate entity similarity. Those methods not only consider the similarity of entity names or identifiers, but also determine the corresponding relationships between entities by analyzing the structural information of the knowledge graph (such as attributes and relationships). The MTransE model[25] first utilizes TransE to encode the entities and relationships within the two ontologies into two different spaces, and obtains the mapping relationship between the two spaces by partially aligned data, thereby learning other alignment relationships. The Rafcom model[18] incorporates ontology structural information while utilizing terminology strategies, which uses three types of features, including selection features, direct similarity features, and context features. Additionally, the AML model[19] employs an interactive selection method that uses alignment results returned by multiple ontology alignment techniques to identify possible entity pairs. The LogMap framework[20] offers problematic alignments to the users for verification, and the verified aligned results are employed to detect conflicts with the found alignments. The structure-based approaches can capture more semantic information than string-based methods. However, these methods depend heavily on the ontology structure, and are not feasible for ontology alignment about sparse ontology.

Recently, more ontology alignment approaches have adopted deep learning based strategies. The ontology alignment methods based on deep learning utilize deep neural networks to achieve entity correspondence in different knowledge graphs.

DeepAlignment[21] builds the pre-trained word vectors and obtains ontology entity descriptions tailored for ontology matching tasks. It adopts a reverse fitting strategy to refine word embeddings and better represent class labels. VeeAlign[22] designs a dual encoder to encode the text information and path information of a class, using multi-dimensional contextual representations to calculate the cultural representations of concepts. Then, VeeAlign uses them to discover semantically equivalent concepts, greatly improving the portability of the system. Furthermore, BERTMap[23] utilizes the textual, structural, and logical information of the ontology. At first, BERTMap predicts the mapping using a classifier based on the fine-tuned context embedding model BERT on the textual semantic corpus extracted from the ontology,

and then leverages the ontology structure and logic to refine the mapping through extension and repair. The ontology alignment methods based on deep learning can better handle semantic matching and semantic similarity. However, it is difficult for them to obtain large-scale annotation data to train deep learning models, and they are easily interfered by noisy data.

3 Ontology Alignment Approach Based on ASHF

3.1 Problem formulation

Knowledge graph is defined as a four-tuple $G = (E, R, C, T)$, where E, R, C, and T represent sets of entities, relationships, classes, and tuples, respectively. Each triple is in the form of (h, r, t), where h is a class, t can be an entity or a class, and r describes the relationship between h and t[28].

Ontology alignment in this paper aims to identify equivalent relationships between pairs of cross-ontology classes. Formally, given a pair of knowledge graphs $O_1 = (E_1, R_1, C_1, T_1)$ and $O_2 = (E_2, R_2, C_2, T_2)$, ontology alignment builds an equivalent mapping f from C_1 to C_2, i.e., $f(c_1) = c_2$, where $c_1 \in C_1$ and $c_2 \in C_2$. That is, the class c_1 and the class c_2 have an equivalence relationship.

For example, Fig. 1 shows the class "Bony_part_of_pharyngotympanic_tube" in the foundation model of anatomy (FMA) dataset has the meaning of the bone part of the Eustachian tube, while this class is called "Structure_of_osseous_portion_of_Eustachian_tube" in the SNOMED dataset. In fact, those two classes have an equivalent relationship. Typically, the task of ontology alignment is intended to recognize the equivalent classes in different ontologies.

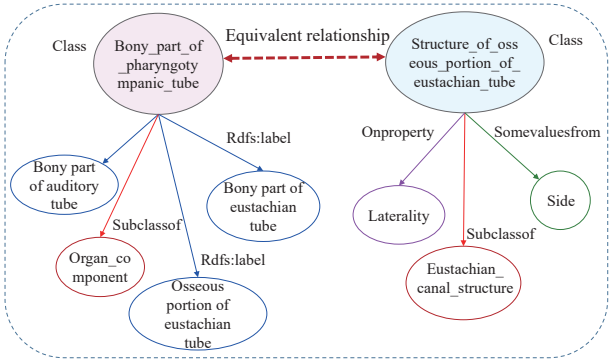

Fig. 1 Example of ontology alignment.

3.2 Technical overview

Figure 2 shows the overall framework of our active learning ontology alignment approach with ASHF. The framework includes four modules: attribute self-adaptation learning module, alignment labeling module, embedding generation module, and mapping

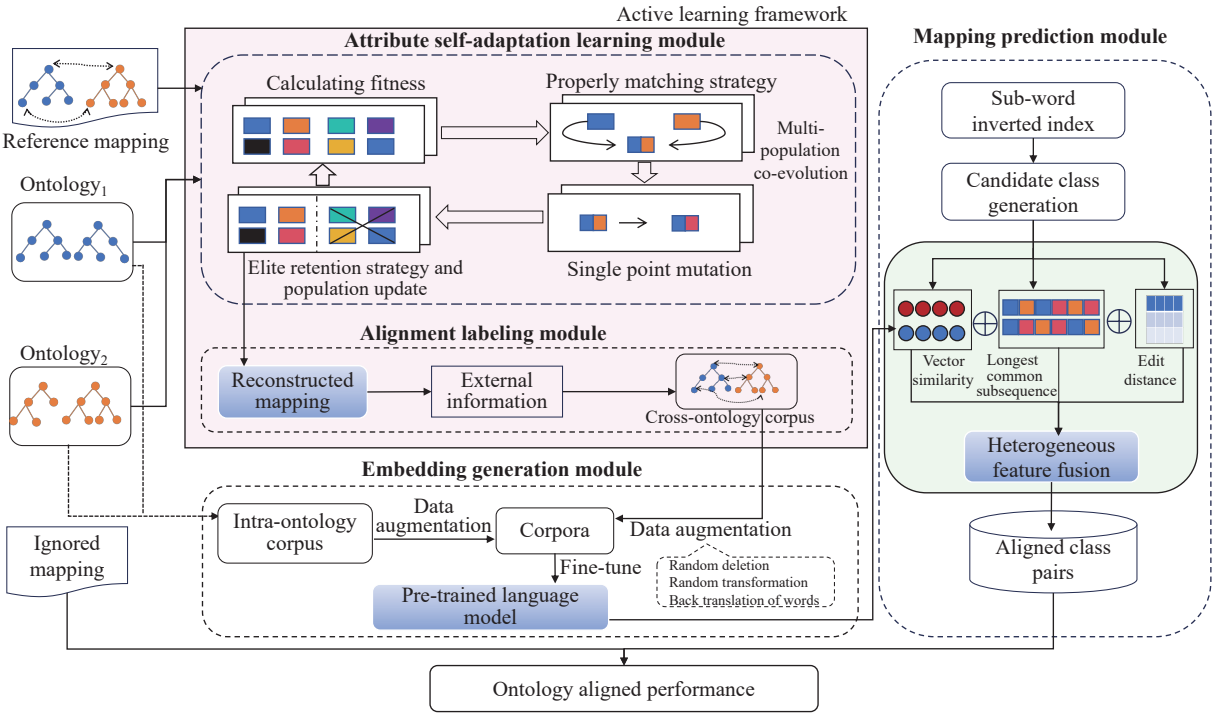

Fig. 2 Overall architecture of ASHF.

prediction module.

The input of our ASHF model is the owl ontology description set and the reference mapping set, and the output is the triple set including equivalent classes. The reference mapping set is composed of the classes in two datasets and their similarity scores.

Typically, the attribute self-adaptation learning module introduces the string similarity between class pairs and their number of labels to reconstruct the reference mapping. That reconstraction strategy augments the semantic information of the classes in the reference mapping, helps the ASHF model to smooth the updating process of weights, and reduces weight fluctuations for improving the stability of our ontology alignment model. Second, the purpose of the alignment labeling module is to choose the reconstructed reference mapping again, while incorporating semantic information into the reference mapping. Third, in the embedding generation module, the construction of the fine-tuning corpus is carried out in an unsupervised manner, and data augmentation is employed to prevent overfitting of the alignment model. Finally, within the mapping prediction module, the initial step involves utilizing a pre-trained language model RoBERTa to generate embedding vectors for each class description. Subsequently, a heterogeneous feature fusion strategy is developed to calculate the similarity between classes.

3.3 Attribute self-adaptation learning module

It has been observed that most of the present ontology alignment methods are semi-supervised ones founded on ontology description. Those methods use limited labeled data to classify and align the entire ontology collection, and do not fully capture the semantic information of class pairs to be labelled. Compared with traditional supervised techniques, those semi-supervised approaches require a smaller number of samples, and have low time complexity.

However, the semantic information provided by different samples in the reference mapping is different, and treating these samples equally will inevitably decline the performance of the ontology alignment model. Thereby, this paper introduces the idea of active learning with an attribute adaptation mechanism to enrich the semantic information by reconstructing the reference mapping. Further, the reconstructing is realized through the attribute information of the class for enhancing the attribute features in the reference map.

The input of the attribute self-adaptation learning module is the reference mapping set, and its output is the reconstructed reference mapping set to be annotated. The idea of active learning is to augment the semantic information of the semi-supervised learning training set[29], and thereby improves the results of model training by reselecting the limited labeled class pairs[30, 31]. Specifically, we reselect the reference mapping to increase the performance of ontology alignment and eliminate false positives. It is noted that only higher quality label pairs are re-selected to improve the quality of the training set, while the number of label pairs does not increase.

The attribute self-adaption learning module uses an improved genetic algorithm to select classes to be marked by properly matching strategy and the principle of multi-population co-evolution, which improves the convergence speed and stability of our ASHF ontology alignment model. The introduction of single-point mutation aims to ensure the accuracy of ASHF algorithm convergence, while the use of multi-population co-evolution guarantees the convergence speed of the ASHF algorithm. Typically, the implementation of well-matched mating principle is intended to make that superior individuals continue with a high probability.

The attribute self-adaption learning algorithm includes the following phases:

(1) Initialization and calculating fitness

We use binary coding to represent the characteristics of the sample, and generate the initial population in a random way. Each individual represents a candidate feasible solution, which is a string composed of “0” and “1”. Here, “0” means that the pair of this type is not selected, while “1” denotes that the pair of this type is selected. The calculation introduces two factors affecting the similarity between two classes, which include the string similarity and the number of related classes. If a class pair has a high string similarity, but is not essentially an equivalent aligned pair, it is easy for the alignment model to identify false positives. Typically, the related classes about that class pair can help enrich the semantics of the ontology and help the alignment model to identify false positives. Therefore, we design the following method for calculating the fitness function.

Assume that there are a total of N classes in an ontology, for each individual, each class pair is traversed. Here, we use $C[i]$ to represent the i-th class.

The string similarity between class pairs is represented by the string edit distance d between two classes, expressed as follows:

$$\begin{cases} d(s, r) = |s|, & \text{if } |r| = 0; \\ d(s, r) = |r|, & \text{if } |s| = 0; \\ d(s, r) = d(s[1:], r[1:]), & \text{if } s[0] = r[0]; \\ d(s, r) = 1 + \min(d(s[1:], r), d(s, r[1:]), \\ \qquad d(s[1:], r[1:])), & \text{otherwise} \end{cases} \tag{1}$$

where $|s|$ denotes the length of string s, $s[1:]$ is the substring of string s after removing the first character, $s[0]$ is the first character of string s, and $d(s, r)$ represents the edit distance between strings s and r.

The individual fitness function is proportional to the string similarity of each class pair, and also proportional to the element number of related class set C_x of each class pair. Thereby, the individual fitness F is defined as follows:

$$F = \sum_{i=1}^{N} (d_i(s, r) + C[i]_x) \tag{2}$$

where $d_i(s, r)$ represents the string edit distance between the i-th class pair s and r, and $C[i]_x$ is the size of the related class set of the i-th class pair.

(2) Properly matching strategy

In the cross-matching process, we employ the properly matching strategy[32], and the parent individuals are sorted according to the fitness function. Parent individuals with fitness large functions are paired together, while those with small functions are analogously paired. The advantage of that strategy is that it can speed up the convergence of the algorithm. Additionally, restricting pairing between high-fitness and low-fitness parent individuals can prevent rapid spread of genes from local optimal individuals throughout the population, causing the convergence result to fall into local optimality. Suppose there are n class pairs in a reference mapping set, during the crossover process, two chromosomes with similar objective function values are extracted from the chromosome library based on the principle of properly matching strategy, $c_f = w_1 w_2 \cdots w_n$ and $c_m = e_1 e_2 \cdots e_n$, where c_f and c_m represent the chromosomes of the parent class set. The crossover positions u_1 and u_2 are randomly selected, and the two parent chromosomes are partial chromosome exchange between u_1 and u_2,

$$\begin{aligned} c_s &= w_1 w_2 \cdots w_{u_1 - 1} e_{u_1} \cdots e_{u_2} w_{u_2 + 1} \cdots w_n, \\ c_d &= e_1 e_2 \cdots w_{u_1 - 1} w_{u_1} \cdots w_{u_2} e_{u_2 + 1} \cdots e_n \end{aligned} \tag{3}$$

where c_s and c_d denote the chromosomes of the crossoverd offspring class set, and w_i and e_i represents the class pair at the i-th position.

(3) Single-point mutation

Single-point mutation randomly selects the position u_3 of the chromosome mutation, and randomly selects a class pair from the corresponding chromosome to replace the gene at the original position. We randomly select a chromosome mutation position u_1, and randomly choose a class pair from the original reference map to replace the gene at the original u_1 position,

$$\begin{aligned} c_f &= w_1 w_2 \cdots w_{u_3 - 1} w_{u_3} w_{u_3 + 1} \cdots w_n, \\ c_h &= w_1 w_2 \cdots w_{u_3 - 1} e_{u_3} w_{u_3 + 1} \cdots w_n \end{aligned} \tag{4}$$

where c_f is the chromosomes of the parent class set, c_h is the chromosomes of the mutated offspring class set, and w_i and e_i represents the class pair at the i-th position.

(4) Multi-population co-evolution and information exchange

We build multiple populations, and each population evolves independently. Each population performs selection, crossover, and mutation operations according to the standard genetic algorithm process to generate new offspring individuals. Regular information exchange and cooperation among populations promote co-evolution among populations. We can selectively share elite individuals, and migrate a certain number of individuals, and so on, to increase communication and cooperation between populations. Here we design the immigration operator in the following way: (a) obtain the current population number; (b) traverse the population and obtain the optimal class pair set individuals of each population and the worst class pair set individuals of the target population; and (c) convert the current population. The best individual in the population is replaced by the worst individual position of the target population.

(5) Elite retention strategy and population update

Individuals are sorted according to their fitness values, and then the top Q individuals with the best fitness are selected as elite individuals, which are retained in the next generation.

We merge the offspring individuals and elite individuals generated by each population to form a new population. After the population is updated, the fitness values of the individuals in the new population are calculated, and the final elite population is obtained

through artificial selection operators to avoid falling into local optimality. The executing of the artificial selection operator is as follows: (a) obtain the current population number; (b) traverse the population and record the optimal individuals of each population; and (c) extract the optimal individuals of each population to form an elite population. Through iterative evolution, the samples in the population gradually tend to be of better quality. When the termination condition is reached, the iteration stops and the optimal individual in the current population are returned as the selected high-quality sample. The genetic code of the optimal individual represents the selected sample set.

3.4 Alignment labeling module

The alignment labeling module reselects the reconstructed reference map, and adds the augmented semantics into the classes in the reference mapping.

For each class pair cp to be annotated, we recur to the knowledge learned from Wikipedia to determine whether each class pair is a matching equivalence class pair. Concretely, we acquire cp's meaning and description in Wikipedia to identify whether that class pair refers to the same entity. For each class, the texts within the corresponding item in Wikipedia are used to assist in labeling the class pairs to be labeled.

Finally, we build the annotated cross-ontology corpus, and construct the intra-ontology corpus using the method in BERTMap. The final fine-tuning corpora is obtained by adding the cross-ontology corpus and the intra-ontology corpus.

3.5 Embedding generation module

In the embedding generation module, the fine-tuning corpus is constructed in an unsupervised means to fine-tune the RoBERTa pre-trained language model. Thereby, a fine-tuning corpus containing positive samples and negative samples is built. Concretely, for each class in the ontology, we use the words in its label as positive samples, and the class itself is used as its own positive sample, while other classes and labels of those classes are randomly selected as negative samples. In order to prevent random positive samples from being added to the negative samples, we delete the positive samples in the negative sample set after the construction is completed.

In addition, for the labeled class pairs outputted by the attribute self-adaptation learning module, we add the pairs labeled as equivalent classes and the words in their label labels as positive samples into the corpus, and the pairs labeled as non-equivalent classes are augmented to the corpus as negative samples to further expand the fine-tuned corpus.

Now, it is found that for each class, the number of negative samples is far more than the number of positive samples, which would lead to serious overfitting when fine-tuning our ASHF model. In order to avoid this problem, we employ three types of methods to perform data augmentation on the positive samples in the training set, including random deletion, random insertion, and back translation. By expanding the number of the positive samples, a quantitative balance between positive examples and negative samples is automatically achieved, which helps avoiding overfitting.

The process of random deletion is given as follows: Given a word v, we get a new word by randomly deleting some letters within it. Assume that the word v is composed of letters $l_1, l_2, \ldots, l_n$, the existence or absence of each letter w_i can be expressed as a binary random variable z_i. "$z_i = 1$" means existence, and "$z_i = 0$" stands for deletion. The probability of existence is p, and the probability of deletion is $1-p$. Then z_i obeys the Bernoulli distribution with probability p as expressed in $z_i \sim \text{Bernoulli}(p)$.

The steps of random insertion are given as follows: Given a word v, new words are obtained by randomly selecting positions and inserting new letters. Assuming that word v consists of letters $w_1, w_2, \ldots, w_n$, a random variable Y_i can be introduced to represent the probability of inserting a new letter at position i. For each position i, the probability of inserting a new letter is p, and the probability of not inserting it is $1-p$. For each position i, random sampling is performed according to a binary distribution to determine whether to insert a new word at that position. In this way, Y_i obeys the Bernoulli distribution with probability p.

The process of back translation involves translating a word v from a source language to the target language using a translation model, and then utilizes another translation model to translate the resulting word a back to English to get the back-translated word v'. Two different translation models may introduce the translation noise and variation. Hence, the generated back-translated vocabulary is slightly different from the source language vocabulary v'.

RoBERTa[33] is a pre-trained language model based on Transformer architecture. It maps text sequences

into high-dimensional vector representations. RoBERTa dynamically focuses on tokens at different positions in the text sequence and uses a feed-forward neural network (FFN) for non-linear mapping, as shown the following:

$$\mathrm{FFN}(x) = \max(0,\ xW_1 + b_1)W_2 + b_2 \tag{5}$$

where x is the input vector, W_1 and W_2 are weight matrices, and b_1 and b_2 are bias vectors. It uses residual connections and layer normalization (LN) to alleviate the gradient disappearance problem during training, as expressed the following:

$$\mathrm{LayerNorm}(x) = \mathrm{LN}(x,\ \gamma,\ \beta) = \gamma \odot \left(\frac{x-\mu}{\sqrt{\sigma^2 + \epsilon}} \right) + \beta \tag{6}$$

where μ is the mean, σ is the standard deviation, γ and β are learnable scaling and offset parameters, respectively. In downstream tasks, "CLS"-tagged embedding vectors are extracted for tasks such as classification.

When a corpus including positive and negative samples is built, we chose to fine-tune the RoBERTa pre-trained language model instead of pre-training from scratch. As for the optimization, we adopt the default AdamW optimizer in the Transformers library for parameter optimization. That approach can make full use of RoBERTa's pre-trained knowledge and transfer it to our ontology alignment task for improving performance. AdamW combines the fast convergence of the Adam optimization algorithm and the regularization effect of weight decay to better optimize the neural network model and prevent overfitting[34].

3.6 Mapping prediction module

Within the mapping prediction module, a heterogeneous feature fusion strategy is developed to measure the similarity between classes to output the aligned class pairs. Previous methods usually use an $M \times N$ matrix A to represent the ontology mapping result, where M is the number of classes in the source ontology O_1, N is the number of classes in the target ontology O_2. Each element in the matrix $A[i][j]$ denotes the similarity score between the i-th class in the source ontology O_1 and the j-th class in the target ontology O_2. That representation approach has high computational complexity, and also produces a huge two-dimensional matrix on a large-scale dataset, which greatly increases the space complexity of the algorithm. The ASHF method of this paper first selects candidate classes to be matched for each class of the source ontology, and then uses the fine-tuned RoBERTa model to fuse heterogeneous features to assign a score to each possible mapping.

(1) Candidate class generation

In order to avoid a large amount of time overhead due to a large number of vector similarity calculations when generating mappings, for each class in the source ontology, we do not calculate the similarity with all classes in the target ontology. However, a certain number of classes are selected as candidate classes from the target ontology in advance. Further, we only compute the similarity between the source ontology class and these candidate classes, and the highest one is selected as the outputted mapping.

Before selecting candidate classes, we have the assumption that the aligned equivalence classes or their sub-labels all contain at least one identical sub-word. Previous methods usually choose the inverted index to build candidate classes[14, 17, 35, 36]. We use BERTMap's reverse sub-word index method when building candidate classes. That approach can capture various word forms without additional processing and employ a more detailed sub-word index, without using the inverted index of word-level. We use the WordPiece word segmentation method[37], which iteratively divides the words in the vocabulary to obtain smaller sub-word units. The division rules are based on the frequency statistics of the corpus. In this process, common sub-word combinations are retained, while uncommon combinations are split into smaller units. Eventually, the words in the vocabulary will be further divided into smaller sub-units word units.

To further improve retrieval efficiency by removing stop words from phrases, the option of removing stop words is added. At the same time, a minimum threshold h for the generation of candidate classes is constructed. If the vector similarity between the candidate class and the class to be matched is less than 0.5, the candidate class will be discarded. Therefore, for each class to be aligned, the above method is used to select K candidate classes for each class. Traditional candidate classes often use scores based on reverse document frequency to rank candidate pairs, and then select K class pairs with the highest scores as candidate pairs. This paper further optimizes the calculation speed by setting a threshold.

(2) Heterogeneous feature fusion

It is found that the generated embedding of the well matching class pairs by using RoBERTa may reduce

the accuracy. Hence, we propose a heterogeneous feature fusion strategy that not only introduces the semantic similarity between classes, but also adds string similarity as a correction. That method includes three types of features: the feature about string edit distance, the features of longest common subsequence, and the feature of vector cosine similarity.

The mapping score calculation based on heterogeneous feature fusion strategy is expressed as follows:

$$S_{\text{map}} = \begin{cases} 1, & \text{if } \Omega(c) \cap \Omega(c') \neq \varnothing; \\ \max S_f(\Omega(c), \Omega(c')), & \text{otherwise} \end{cases} \tag{7}$$

where $\Omega(c)$ denotes the set of class c and all its labels in the source ontology, and $\Omega(c')$ is the set of class c' and all its labels in the target ontology. $\Omega(c) \cap \Omega(c') \neq \varnothing$ represents that the source class set and the target class set have at least one same class and a completely matching string. At this time, the similarity is directly set to 1 for saving computing resources. Otherwise, we construct a Cartesian product of the source set to the target set. For each string pair, we calculate their similarity S_f based on heterogeneous feature fusion, expressed as follows:

$$S_f = \alpha \cdot \text{Levenshtein}(\text{st}, \text{st}') + \beta \cdot \text{LCS}(\text{st}, \text{st}') + \gamma \cdot S_R(\text{st}, \text{st}') \tag{8}$$

where st and st′ denote the string belonging to the source ontology and the string belonging to the target ontology in each string pair in the Cartesian product, respectively; Levenshtein(st, st′) represents the string edit distance between the two strings; LCS(st, st′) denotes the longest common subsequence between the two strings; and S_R(st, st′) is the cosine similarity of the vectors obtained by embedding the two strings into RoBERTa. The weights satisfy the conditions expressed as follows:

$$\alpha + \beta + \gamma = 1 \tag{9}$$

In this way, we get the fused heterogeneous features S_f of the two classes, which takes into account the string similarity and vector similarity between classes at the same time, and captures well the label information.

(3) Aligned class pairs output

For each class C_i in the source ontology dataset, the method of generating candidate classes above is used to generate candidate classes. To each candidate class, the similarity based on heterogeneous feature fusion between it and the source class is calculated, and we select the one with the highest score. In this way, all classes in the source ontology dataset are traversed to obtain a set of aligned class pairs.

4 Experiment

4.1 Experiments settings

4.1.1 Datasets

We conduct ontology alignment experiments on eight datasets to evaluate the performance of the proposed ASHF method. This paper follows the dataset composed of FMA, SNOMED, and NCI from the Ontology Alignment Evaluation Initiative (OAEI) track in 2022[11], and also uses the expansion of SNOMED in BERTMap to obtain the SNOMED+ dataset. Thereby, we obtain three biomedical datasets FMA-SNOMED, FMA-SNOMED+, and FMA-NCI to verify the performance of the ontology alignment models. In addition, we use the three material datasets from the Material Sciences and Engineering track in OAEI 2022[38], including MatOnto-Reduced MaterialInformation (abbreviated as MatOnto-RMat), MaterialInformation-MatOnto (abbreviated as Mat-MatOnto), and EMMO-MaterialInformation (abbreviated as EMMO-Mat), and also utilize the two biology datasets from the Biodiversity and Ecology track in OAEI 2018[39], including FLOPO-PTO and ENVO-SWEET. They have high quality ontology collections and alignment standards. The details of these datasets are shown in Table 1. For each dataset, we divide the dataset into a training set, a validation set, and a test set in a ratio of 2:1:7. The sub-datasets used for training, validation, and testing are disjoint from each other.

4.1.2 Baselines

To demonstrate the effectiveness of the proposed model, we compare it with various mainstream

Table 1 Overall statistics of the datasets.

Dataset	Number of classes in source dataset	Number of classes in target dataset	Number of reference mappings	Number of ignored mappings
FMA-SNOMED	10 157	13 412	6026	2982
FMA-SNOMED+	10 157	13 412	6026	2982
FMA-NCI	3696	6488	2686	338
MatOnto-RMat	89	933	23	0
Mat-MatOnto	1036	933	302	0
EMMO-Mat	1036	500	63	0
FLOPO-PTO	24 204	1509	244	0
ENVO-SWEET	7081	6764	666	0

ontology alignment methods. The baseline methods also adopt a semi-supervised approach. We compare our model with the baseline methods as follows:

- String-match determines whether the ontology is aligned through string comparison[40].
- Edit-similarity scores ontology similarity by calculating string edit distance[41].
- LogMapLt[42] is a lightweight version of LogMap. It relies only on efficient string matching techniques.
- LogMap[20] provides alignments to users for verification, and the validated aligned results are utilized to identify conflicts with the identified alignments.
- AML[19] uses an interactive selection algorithm that leverages alignment results from multiple ontology alignment methods to identify potential entity pairs.
- LogMap-ML[26] is an extension of traditional ontology alignment systems based on machine learning, using remote supervision for training, ontology embeddings and Siamese neural networks to incorporate richer semantics.
- Truveta Mapper[27] uses a multi-task sequence-to-sequence transformer model to achieve alignment across numerous ontologies in a zero-shot, integrated, and end-to-end fashion.
- BERTMap[23] first uses BERT to predict the mappings, and then utilizes the ontology structure and logic to enhance. Finally, it rectifies the mappings through extension and repair.

In addition, we also compare the ASHF model with the models from the OAEI track 2022[38], including A-LIOn, LogMap, LogMapLight, and Matcha, and the models from the OAEI 2018[39], including AML, Lily, LogMap, LogMapBio, LogMapLite, POMap, and XMap.

4.1.3 Implementation details

The RoBERTa pre-trained language model is used to encode the classes, and the batch size and dropout rate are set to 32 and 0.1, respectively. That model is trained using the AdamW optimizer with an initial learning rate of 5×10^{-5}[34], the warm-up ratio is set to 0, and the weight decay is set to 0.001. Additionally, the ratio of positive and negative samples is set to 2:8, and for each class we extract two positive samples and eight negative samples. The size of the candidate mapping is set to 400.

4.1.4 Evaluation metrics

The precision, recall, and F1-score are utilized to evaluate our ontology alignment model, as shown in the following:

$$\text{Precision} = \frac{P_{\text{out}} \cap \text{Ref}_m - \text{Ref}_{\text{ign}}}{P_{\text{out}} - \text{Ref}_{\text{ign}}} \tag{10}$$

$$\text{Recall} = \frac{P_{\text{out}} \cap \text{Ref}_m - \text{Ref}_{\text{ign}}}{\text{Ref}_m - \text{Ref}_{\text{ign}}} \tag{11}$$

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{12}$$

where P_{out} is the mapping output of the ontology alignment models, Ref_m is the reference mapping of the ontology aignment models, and Ref_{ign} is the given ignore mapping. To improve the precision of the evaluation, OAEI specifies a set of ignore mappings. Mappings in an ignored mapping set are regarded inaccurate, incomplete, or irrelevant. OAEI measures a model's performance by excluding these mappings from the evaluation[38]. It is necessary to delete the repeated parts in the ignored mapping after taking the intersection of the output mapping and the reference mapping. At the same time, it is also necessary to delete the repeated parts of the reference mapping in the denominator part.

4.2 Experimental results

4.2.1 Performance comparison

Tables 2–4 show the overall comparison results obtained by different ontology alignment models on the eight datasets. Here, we can observe the following facts. Our ASHF model outperforms existing mainstream methods on the most cases, which shows the effectiveness of our method. Specifically, the F1-score of ASHF is increased by 0.015, 0.043, and 0.009 compared with BERTMap on FMA-SNOMED, FMA-NCI, and FMA-SNOMED+ datasets, respectively. Additionally, compared with LogMap, the F1-score of ASHF is raised by 0.046 and 0.003 on MatOnto-RMat and Mat-MatOnto, respectively, while the F1-score is increased by 0.059 on FLOPO-PTO. Furthermore, our ASHF obtains the best precision on EMMO-Mat. The string-match method has the highest precision on the three datasets of Table 2, and that method only determines whether the ontology is aligned based on whether the strings are equal. However, this method also has the lowest recall rates. This may be due to the fact that the dataset has many congruent ontology mappings. In Table 2, AML achieves the highest recall in FMA-NCI dataset, and Logmap obtains the highest recall in FMA-SNOMED+ dataset. However, their inadequacy in calculating the initial similarity probably

Table 2 Comparative results on FMA-SNOMED, FMA-NCI, and FMA-SNOMED+ datasets in biomedical domain. The best results are bolded.

Model	FMA-SNOMED			FMA-NCI			FMA-SNOMED+		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
String-match[40]	**0.983**	0.192	0.321	**0.972**	0.747	0.845	**0.972**	0.665	0.790
Edit-similarity[41]	0.963	0.208	0.343	0.970	0.774	0.861	**0.972**	0.724	0.830
LogMapLt[42]	0.956	0.204	0.336	0.953	0.812	0.877	0.940	0.709	0.808
LogMap[20]	0.918	0.681	0.782	0.922	0.897	0.909	0.838	**0.868**	0.852
AML[19]	0.865	0.754	0.806	0.919	**0.898**	0.909	0.868	0.825	0.846
LogMap-ML[26]	0.928	0.208	0.340	0.959	0.714	0.819	0.942	0.700	0.803
Truveta Mapper[27]	0.947	0.738	0.830	–	–	–	–	–	–
BERTMap[23]	0.892	**0.786**	0.836	0.959	0.813	0.880	0.908	0.852	0.879
ASHF (ours)	0.931	0.784	**0.851**	0.955	0.894	**0.923**	0.963	0.823	**0.888**

Table 3 Comparative results on MatOnto-RMat, Mat-MatOnto, and EMMO-Mat datasets in material domain. The best results are bolded.

Model	MatOnto-RMat			Mat-MatOnto			EMMO-Mat		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
A-LIOn	0.130	**0.130**	0.130	0.387	**0.209**	0.217	0.000	0.000	0.000
LogMap	**1.000**	0.043	0.083	**0.881**	0.195	0.320	0.946	**0.841**	**0.891**
LogMapLight	0.400	0.087	0.143	0.851	0.189	0.309	0.911	0.810	0.857
Matcha	0.000	0.000	0.000	0.000	0.000	0.000	0.500	0.032	0.060
ASHF (ours)	0.959	**0.130**	**0.229**	0.876	0.198	**0.323**	**0.952**	0.806	0.873

Table 4 Comparative results on FLOPO-PTO and ENVO-SWEET datasets in biology domain. The best results are bolded.

Model	FLOPO-PTO			ENVO-SWEET		
	Precision	Recall	F1-score	Precision	Recall	F1-score
AML	0.880	**0.840**	0.860	0.776	**0.926**	**0.844**
Lily	0.813	0.586	0.681	0.866	0.641	0.737
LogMap	0.817	0.787	0.802	0.839	0.738	0.785
LogMapBio	0.803	0.787	0.795	0.839	0.724	0.777
LogMapLite	**0.987**	0.661	0.855	0.732	0.817	0.772
POMap	0.663	0.709	0.685	0.839	0.738	0.785
XMap	**0.987**	0.619	0.761	**0.868**	0.716	0.785
ASHF (ours)	0.979	0.769	**0.861**	0.845	0.825	0.835

results in relatively low accuracy in Table 2. The reason may lie in that those two methods introduce manual supervision when generating alignment results, which leads to their higher recall rates. In summary, the reasons why our method outperforms other models on these eight datasets are as follows: (1) The attribute self-adaptive learning method is introduced to reconstruct the reference map set, enhancing the semantic information it implies; (2) Heterogeneous feature fusion is designed to integrate multiple features when calculating the similarity of class pairs to avoid the problem of falsely high similarity of the BERT model; and (3) The data augmentation method is used to expand the positive samples for model fine-tuning, solving the overfitting problem caused by the imbalance of positive and negative samples.

In order to evaluate the effectiveness of different pre-trained models, we have conducted the experiments without per-training models, and using BERT, GPT-2, and RoBERTa. Tables 5−7 report the experimental results using different pre-trained models on eight datasets. We can see from Tables 5−7 that our ASHF model without the per-training model obtains the best precision with 0.996, 0.983, 1.000, 0.879, 0.975, 0.987, and 0.874 on seven datasets, respectively. The ASHF model using RoBERTa achieves the best recall and F1-score on eight datasets. Specifically, the best F1-score are 0.851, 0.923, 0.888, 0.229, 0.323, 0.873, 0.861, and 0.835 on eight datasets, respectively. In summary, the performance of the ASHF model using RoBERTa is superior than those of non-pre-trained model, BERT and GPT-2.

4.2.2 Ablation study

To illustrate the contribution of different modules to our method, we conduct the following ablation experiments. Seven different settings are established.

Table 5 Ablation experiment about different pre-trained models on FMA-SNOMED, FMA-NCI, and FMA-SNOMED+ datasets in biomedical domain. The best results are bolded.

Model	FMA-SNOMED			FMA-NCI			FMA-SNOMED+		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Without pre-training	**0.996**	0.197	0.328	**0.983**	0.742	0.846	0.976	0.682	0.803
With Bert	0.890	0.506	0.645	0.973	0.780	0.866	0.972	0.697	0.812
With GPT-2	0.996	0.187	0.316	0.982	0.743	0.846	**0.977**	0.678	0.801
ASHF(with RoBERTa)	0.931	**0.784**	**0.851**	0.955	**0.894**	**0.923**	0.963	**0.823**	**0.888**

Table 6 Ablation experiment about different pre-trained models on MatOnto-RMat, Mat-MatOnto, and EMMO-Mat datasets in material domain. The best results are bolded.

Model	MatOnto-RMat			Mat-MatOnto			EMMO-Mat		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Without pre-training	**1.000**	0.043	0.082	**0.879**	0.087	0.158	**0.975**	0.634	0.768
With BERT	0.959	0.087	0.160	0.867	0.135	0.234	0.942	0.735	0.826
With GPT-2	0.918	0.043	0.082	0.862	0.178	0.295	0.955	0.701	0.809
ASHF (with RoBERTa)	0.959	**0.130**	**0.229**	0.876	**0.198**	**0.323**	0.952	**0.806**	**0.873**

Table 7 Ablation experiment about different pre-trained models on FLOPO-PTO and ENVO-SWEET datasets in biology domain. The best results are bolded.

Model	FLOPO-PTO			ENVO-SWEET		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Without pre-training	**0.987**	0.543	0.701	**0.874**	0.667	0.757
With BERT	0.935	0.739	0.826	0.870	0.782	0.824
With GPT-2	0.940	0.727	0.820	0.839	0.790	0.814
ASHF (with RoBERTa)	0.979	**0.769**	**0.861**	0.845	**0.825**	**0.835**

(1) "Only attribute self-adaptation learning module" (M_1) indicates that the ASHF model only uses the RoBERTa pre-trained language model that is fine-tuned after constructing the corpus with attribute self-adaptation. (2) "Only heterogeneous feature fusion strategy" (M_2) signifies that the ASHF model uses only heterogeneous features fusion to perform similarity calculation. (3) "Only data augmentation" (M_3) indicates that the ASHF model uses only data augmentation methods to expand the fine-tuning corpus. (4) "Without data augmentation" ($M_4 = M_1 + M_2$) means that the ASHF model uses attribute self-adaptation and heterogeneous feature fusion. (5) "Without heterogeneous feature fusion strategy" ($M_5 = M_1 + M_3$) shows that the ASHF model uses attribute self-adaptation and data augmentation. (6) "Without attribute self-adaption learning module" ($M_6 = M_2 + M_3$) signifies that the ASHF model uses heterogeneous feature fusion and data augmentation. (7) "ASHF" ($M_7 = M_1 + M_2 + M_3$) indicates that the RoBERTa model uses attribute self-adaptation learning module, heterogeneous feature fusion and data augmentation. The results of ablation experiments are given in Table 8.

Table 8 shows that the ASHF model obtains the highest performance when all modules are used. The performance of the ASHF model decreases when every time a module is removed, so each module of the model helps to improve the performance of ASHF. In particular, the precision of methods with heterogeneous feature fusion (M_4, M_6, and M_7) is generally higher than those without heterogeneous feature fusion (M_1, M_3, and M_5), which indicates that heterogeneous feature fusion can effectively alleviate the impact of noise on the RoBERTa model and greatly improve the precision of alignment.

4.2.3 Parameter experiments

In order to analyze the sensitivity of the ASHF model to hyperparameters, we design the following experiments regarding the hyperparameters of the model, including weight decay, learning rate, dropout rate, batch size, warm up ratio, the heterogeneous feature fusion hyperparameter α, β, and γ in Eq. (9), crossover probability, and mutation probability.

Table 8 Ablation experiment. The best results are bolded.

Model	FMA-SNOMED			FMA-NCI			FMA-SNOMED+		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
M_1	0.808	0.775	0.791	**0.971**	0.754	0.849	0.911	0.817	0.812
M_2	0.871	0.735	0.797	0.923	0.851	0.886	0.907	**0.838**	0.871
M_3	0.873	0.735	0.798	0.938	0.842	0.888	0.911	0.833	0.871
M_4	0.881	**0.795**	0.836	0.938	0.846	0.890	0.938	0.825	0.879
M_5	0.871	0.752	0.807	0.936	0.846	0.889	0.932	0.838	0.883
M_6	0.874	0.798	0.834	0.950	0.855	0.900	0.942	0.824	0.879
M_7	**0.931**	0.784	**0.851**	0.955	**0.894**	**0.923**	**0.963**	0.823	**0.888**

The weight decay is set as {0.1, 0.01, 0.001, 0.0001, 0.00001}. Figure 3a demonstrates that the ASHF model achieves the highest F1-scores on three datasets FMA-SNOMED, FMA-SNOMED+, and FMA-NCI, when the attenuation value is 0.001. The learning rate is selected from {0.001, 0.0001, 0.00005, 0.00001, 0.000001}. Based on the results presented in Fig. 3b, it can be observed that the F1-score attains its maximum point on all three datasets when the learn rate is set to 0.00005. The dropout rate is set as {0, 0.1, 0.2, 0.3, 0.4}. Figure 3c demonstrates that the highest F1-score is achieved for all three datasets when the dropout rate is adjusted to 0.1. As the dropout rate increases, the F1-score of the ASHF shows a downward trend. The batch size is assigned the values {32, 64, 128, 256, 512}. The findings from Fig. 3d indicates that when setting the batch size to 128, ASHF gets the highest F1-scores on all three datasets.

The warm up ratio is specified as {0, 0.1, 0.2, 0.3, 0.4}. Figure 4a shows that ASHF achieves the highest F1-scores on all the three datasets when the warm up ratio is 0. The heterogeneous feature fusion hyperparameter α, β, and γ in Eq. (9) are set as {(0.05, 0.05, 0.9), (0.005, 0.005, 0.99), (5×10^{-4}, 5×10^{-4}, 0.999), (5×10^{-5}, 5×10^{-5}, 0.9999), (5×10^{-6}, 5×10^{-6}, 0.99999)}. From Fig. 4b, we can see that the F1-score reaches its peak when the parameters are set to (5×10^{-4}, 5×10^{-4}, 0.999) on all three datasets. The crossover probability is chosen as {0, 0.1, 0.2, 0.3, 0.4}. Figure 4c indicates that the ASHF obtains the highest F1-scores on the three datasets when the crossover rate is 0.3. As the crossover rate gradually increases, the F1-score of the model gradually increases and then decreases. The mutation probability is set as {0, 0.1, 0.2, 0.3, 0.4}. Figure 4d shows that ASHF achieves the highest F1-scores on all three datasets when the mutation rate is 0.2. As the mutation rate gradually increases, the F1-score of the model gradually increases and then decreases.

It can be concluded that the ASHF model is not sensitive to all eight hyperparameter on FMA-SNOMED+ and FMA-NCI datasets. As to FMA-SNOMED dataset, the ASHF model is not sensitive to batch size, crossover probability, mutation probability, and the heterogeneous feature fusion hyperparameter α, β, and γ, while it is sensitive to weight decay, learning rate, dropout rate, and warm up ratio on FMA-SNOMED dataset. In summary, the main advantage of the proposed ASHF model can be highlighted as follows. With the carefully designed active learning framework, technically our ASHF model helps improve the ontology alignment performance, which is hindered by the problems, including the lack of implicit semantic within the reference mapping or labeled set, context noise, and imbalance training data. Extensive experimental results on the eight public datasets also demonstrate that the proposed ASHF approach significantly outperforms the state-of-the-art methods. The limitation of the proposed ASHF model lies in that it does not use structure information. Further, the ASHF relies more on the quality of the attribute information of classes within the data set, such as the richness of attributes of classes.

4.2.4 Running time

The time complexity of our ASHF model for mapping prediction is scale in $O(K \times N)$, where K is the size of the candidate class set, and N is the number of classes in the source ontology. Similarly, the time complexity of the BERTMap model for computing mappings is also $O(K \times N)$[23]. Comparatively, the time complexity of the Truveta Mapper model is $O(N \times \log N)$[27]. The training time of our ASHF model for one epoch is 48 s, 32 s, and 48 s on the FMA-SNOMED, FMA-NCI, and FMA-SNOMED+ datasets, respectively, when the ASHF model is performed on one 4090 GPU.

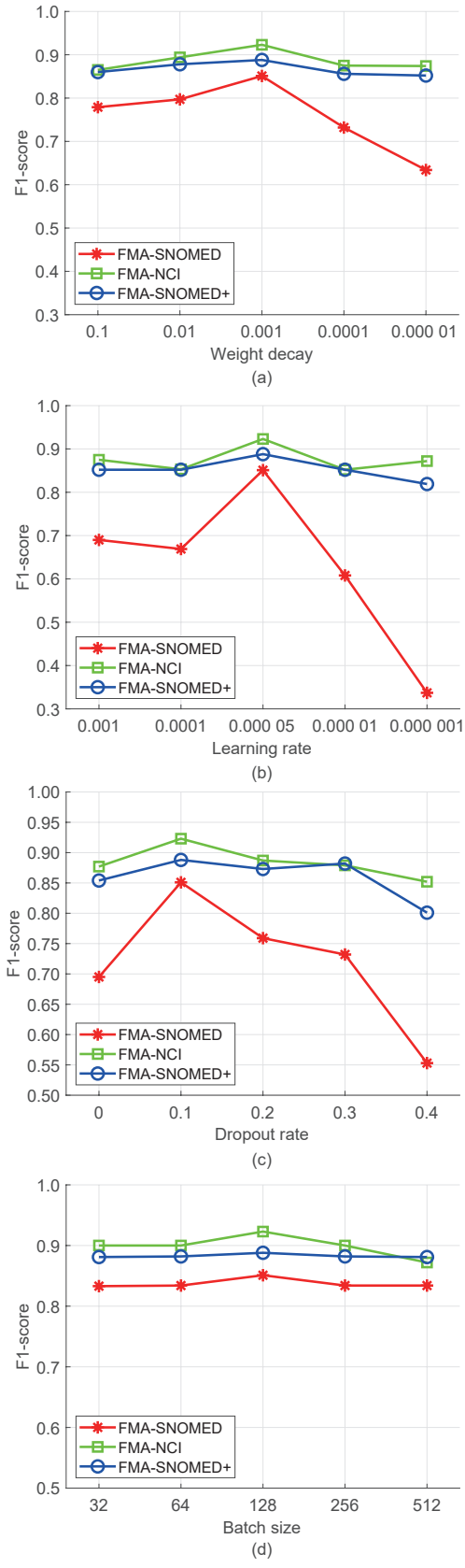

Fig. 3 Hyperparameter sensitivity analysis results about weight decay, learning rate, dropout rate, and batch size.

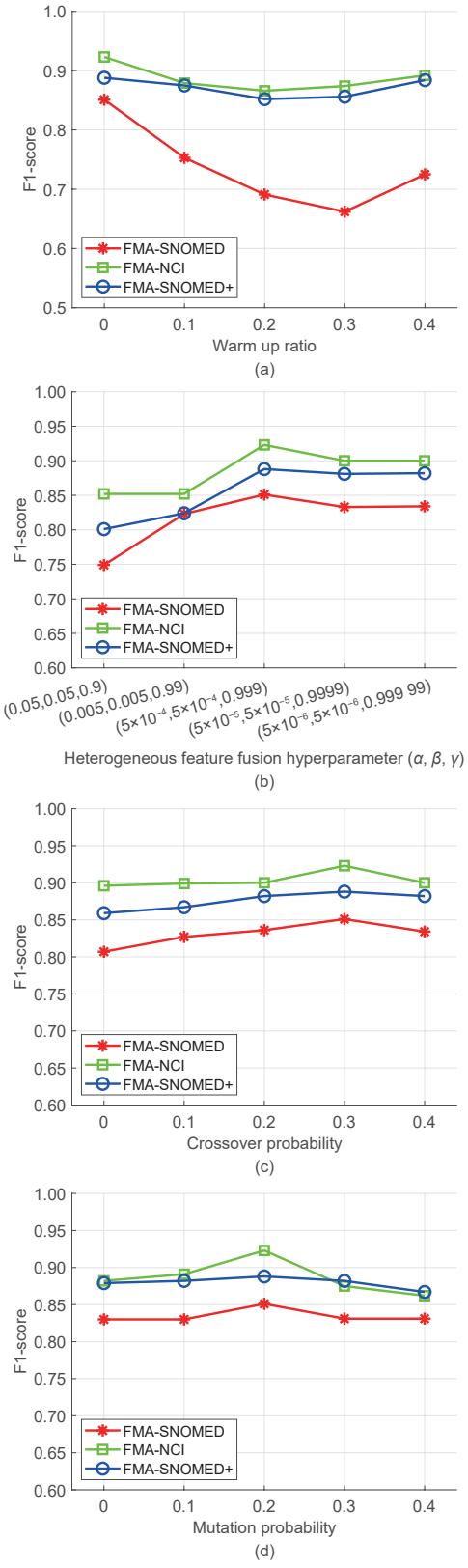

Fig. 4 Hyperparameter sensitivity analysis results about warm up ratio, heterogeneous feature fusion hyperparameter (α, β, γ), crossover probability, and mutation probability.

5 Conclusion and Future Work

In this paper, we propose an active learning framework based on ASHF to solve the ontology alignment task, which can enhance the semantic information in reference mapping and accurately capture the attribute similarity and string similarity between class pairs during mapping. Our heterogeneous feature fusion method can effectively alleviate the context noise problem during BERT fine-tuning and enhance the representation ability of heterogeneous features. Moreover, we use the data augmentation methods to augment the positive samples, which alleviates the over-fitting problem caused by the imbalance of positive and negative samples. In the future, we will like to design a multi-layer graph neural network incorporating multi-scale aggregation to address the problem of entity alignment.

Acknowledgment

The work was supported by the National Natural Science Foundation of China (No. 62072039).

References

[1] X. Zou, A survey on application of knowledge graph, *J. Phys. Conf. Ser.*, vol. 1487, p. 012016, 2020.

[2] D. Sun, Z. Huang, D. Li, and M. Guo, Efficient knowledge graph embedding training framework with multiple GPUs, *Tsinghua Science and Technology*, vol. 28, no. 1, pp. 167–175, 2023.

[3] X. Liu, Y. He, W. Tai, X. Xu, F. Zhou, and G. Luo, Exploring the chameleon effect of contextual dynamics in temporal knowledge graph for event prediction, *Tsinghua Science and Technology*, vol. 30, no. 1, pp. 433–455, 2025.

[4] J. Liu, Y. Lin, Z. Liu, and M. Sun, XQA: A cross-lingual open-domain question answering dataset, in *Proc. 57th Annu. Meeting of the Association for Computational Linguistics*, Florence, Italy, 2019, pp. 2358–2368.

[5] X. Zhao, Y. Jia, A. Li, R. Jiang, and Y. Song, Multi-source knowledge fusion: A survey, *World Wide Web*, vol. 23, no. 4, pp. 2567–2592, 2020.

[6] W. Sun, K. Ren, X. Meng, C. Xiao, G. Yang, and J. Peng, A band divide-and-conquer multispectral and hyperspectral image fusion method, *IEEE Trans. Geosci. Remote Sens.*, vol. 60, p. 5502113, 2022.

[7] K. Ren, W. Sun, X. Meng, G. Yang, J. Peng, J. Huang, and J. Li, CDFSL: Image registration for spaceborne hyperspectral and multispectral data having large spatial-resolution difference, *IEEE Trans. Geosci. Remote Sens*., vol. 61, p. 5515415, 2023.

[8] W. Sun, K. Ren, X. Meng, G. Yang, J. Peng, and J. Li, Unsupervised 3-D tensor subspace decomposition network for spatial-temporal-spectral fusion of hyperspectral and multispectral images, *IEEE Trans. Geosci. Remote Sens.*, vol. 61, p. 5528917, 2023.

[9] T. Yue, R. Mao, H. Wang, Z. Hu, and E. Cambria, KnowleNet: Knowledge fusion network for multimodal sarcasm detection, *Inf. Fusion*, vol. 100, p. 101921, 2023.

[10] S. Cheng, J. Wu, Y. Xiao, and Y. Liu, FedGEMS: Federated learning of larger server models via selective knowledge fusion, arXiv preprint arXiv: 2110.11027, 2021.

[11] Y. He, J. Chen, H. Dong, E. Jiménez-Ruiz, A. Hadian, and I. Horrocks, Machine learning-friendly biomedical datasets for equivalence and subsumption ontology matching, in *Proc. 21st Int. Semantic Web Conf. Semantic Web*, Virtual Event, 2022, pp. 575–591.

[12] I. Horrocks, J. Chen, and L. Jaehun, Tool support for ontology design and quality assurance, in *Proc. ICBO 2020 Integrated Food Ontology Workshop* (*IFOW*), https://foodon.org/icbo-2020-food-workshop/, 2020.

[13] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, et al., Dbpedia-a large-scale, multilingual knowledge base extracted from wikipedia, *Semantic Web*, vol. 6, no. 2, pp. 167–195, 2015.

[14] E. Jiménez-Ruiz and B. C. Grau, LogMap: Logic-based and scalable ontology matching, in *Proc. 10th Int. Semantic Web Conf*., Bonn, Germany, 2011, pp. 273–288.

[15] F. M. Suchanek, S. Abiteboul, and P. Senellart, PARIS: Probabilistic alignment of relations, instances, and schema, *Proc. VLDB Endow*., vol. 5, no. 3, pp. 157–168, 2011.

[16] H. Zhu, D. Liu, I. Bayley, A. Aldea, Y. Yang, and Y. Chen, Quality model and metrics of ontology for semantic descriptions of web services, *Tsinghua Science and Technology*, vol. 22, no. 3, pp. 254–272, 2017.

[17] L. L. Wang, C. Bhagavatula, M. Neumann, K. Lo, C. Wilhelm, and W. Ammar, Ontology alignment in the biomedical domain using entity definitions and context, in *Proc. BioNLP 2018 Workshop*, Melbourne, Australia, 2018, pp. 47–55.

[18] I. Nkisi-Orji, N. Wiratunga, S. Massie, K. Y. Hui, and R. Heaven, Ontology alignment based on word embedding and random forest classification, in *Proc. Joint European Conf. Machine Learning and Knowledge Discovery in Databases*, Dublin, Ireland, 2019, pp. 557–572.

[19] D. Faria, C. Pesquita, E. Santos, M. Palmonari, I. F. Cruz, and F. M. Couto, The AgreementMakerLight ontology matching system, in *Proc. Confederated Int. Conf. Move to Meaningful Internet Systems*, Graz, Austria, 2013, pp. 527–541.

[20] E. Jiménez-Ruiz, LogMap family participation in the OAEI 2021, in *Proc. 16th Int. Workshop on Ontology Matching Co-Located with the 20th Int. Semantic Web Conf*., Virtual Event, 2021, pp. 175–177.

[21] P. Kolyvakis, A. Kalousis, and D. Kiritsis, DeepAlignment: Unsupervised ontology matching with refined word vectors, in *Proc. Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, New Orleans, LA, USA,

2018, pp. 787–798.
[22] V. Iyer, A. Agarwal, and H. Kumar, VeeAlign: A supervised deep learning approach to ontology alignment, in *Proc. 15th Int. Workshop on Ontology Matching Co-Located with the 19th Int. Semantic Web Conf.*, Athens, Greece, 2020, pp. 216–224.
[23] Y. He, J. Chen, D. Antonyrajah, and I. Horrocks, BERTMap: A BERT-based ontology alignment system, in *Proc. AAAI Conf. Artificial Intelligence*, Virtual Event, 2022, pp. 5684–5691.
[24] B. Hättasch, M. Truong-Ngoc, A. Schmidt, and C. Binnig, It's AI match: A two-step approach for schema matching using embeddings, in *Proc. 2nd Int. Workshop on Applied AI for Database Systems and Applications*, Tokyo, Japan, 2020. https://dblp.org/db/conf/vldb/aidb2020.html#HattaschT0B20.
[25] M. Chen, Y. Tian, M. Yang, and C. Zaniolo, Multilingual knowledge graph embeddings for cross-lingual knowledge alignment, in *Proc. 26th Int. Joint Conf. Artificial Intelligence*, Melbourne, Australia, 2017, pp. 1511–1517.
[26] J. Chen, E. Jiménez-Ruiz, I. Horrocks, D. Antonyrajah, A. Hadian, and J. Lee, Augmenting ontology alignment by semantic embedding and distant supervision, in *Proc. 18th European Int. Conf. Semantic Web*, Virtual Event, 2021, pp. 392–408.
[27] M. Amir, M. Baruah, M. Eslamialishah, S. Ehsani, A. Bahramali, S. Naddaf-Sh, and S. Zarandioon, Truveta mapper: A zero-shot ontology alignment framework, in *Proc. 18th Int. Workshop on Ontology Matching Co-Located with the 22nd Int. Semantic Web Conf.*, Athens, Greece, 2023, pp. 1–12.
[28] J. Huang, Z. Sun, Q. Chen, X. Xu, W. Ren, and W. Hu, Deep active alignment of knowledge graph entities and schemata, *Proc. ACM Manage. Data*, vol. 1, no. 2, p. 159, 2023.
[29] D. Cohn, L. Atlas, and R. Ladner, Improving generalization with active learning, *Mach. Learn.*, vol. 15, no. 2, pp. 201–221, 1994.
[30] M. E. Ramirez-Loaiza, M. Sharma, G. Kumar, and M. Bilgic, Active learning: An empirical study of common baselines, *Data. Min. Knowledge Discov.*, vol. 31, no. 2, pp. 287–313, 2017.
[31] G. C. Cawley, Baseline methods for active learning, in *Proc. Active Learning and Experimental Design Workshop, In Conjunction with AISTATS*, Sardinia, Italy, 2011, pp. 47–57.
[32] M. Li, L. Wang, X. Liu, D. Yang, and P. Li, Multi-objective active reconfiguration of distribution network based on the "properly matched marriage" genetic algorithm, (in Chinese), *Power Syst. Prot. Control*, vol. 47, no. 7, pp. 30–38, 2019.
[33] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, RoBERTa: A robustly optimized BERT pretraining approach, arXiv preprint arXiv: 1907.11692, 2019.
[34] I. Loshchilov and F. Hutter, Decoupled weight decay regularization, in *Proc. 7th Int. Conf. Learning Representations*, New Orleans, LA, USA. https://dblp.org/db/conf/iclr/iclr2019.html#LoshchilovH19, 2019.
[35] Z. Wu, S. Li, M. Jiang, and J. Zhang, Ontology alignment method based on self-attention, (in Chinese), *Comput. Sci.*, vol. 49, no. 3, pp. 215–220, 2022.
[36] N. Wu and X. Tang, Ontology matching method based on multi-head self-attention model, (in Chinese), *Radio Commun. Technol.*, vol. 49, no. 6, pp. 1081–1087, 2023.
[37] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al., Google's neural machine translation system: Bridging the gap between human and machine translation, arXiv preprint arXiv: 1609.08144, 2016.
[38] OAEI, Ontology Alignment Evaluation Initiative 2022 Campaign, https://oaei.ontologymatching.org/2022/, 2022.
[39] OAEI, Ontology Alignment Evaluation Initiative 2018 Campaign, https://oaei.ontologymatching.org/2018/, 2018.
[40] S. I. Hakak, A. Kamsin, P. Shivakumara, G. A. Gilkar, W. Z. Khan, and M. Imran, Exact string matching algorithms: Survey, issues, and future research directions, *IEEE Access*, vol. 7, pp. 69614–69637, 2019.
[41] C. Han, L. Li, T. Liu, and M. Gao, Approaches for semantic textual similarity, (in Chinese), *J. East China Norm. Univ. Nat. Sci.*, no. 5, pp. 95–112, 2020.
[42] E. Jiménez-Ruiz, B. C. Grau, and I. Horrocks, LogMap and LogMapLt results for OAEI 2012, in *Proc. 7th Int. Conf. Ontology Matching*, Boston, MA, USA, 2013, pp. 152–159.

Chunxia Zhang received the PhD degree from Institute of Computing Technology, Chinese Academy of Sciences, China in 2005. As an academic visitor, she visited University of Vermont, USA from January 2010 to February 2011. She is currently an associate professor at School of Computer Science and Technology, Beijing Institute of Technology, China. Her research interests include information extraction, social computing, and machine learning.

Cheng Yang received the BEng degree from Northwestern Polytechnical University, China in 2021. He is currently a master student at Beijing Institute of Technology, China. His research interests include deep learning, knowledge fusion, and entity alignment.

Yihao Chen received the BEng degree from Beijing Institute of Technology, China in 2023. He is currently a master student at Beijing Institute of Technology, China. His research interests include deep learning and knowledge fusion.

Xiaojun Xue received the BEng degree from China University of Geosciences, China in 2019. He is currently a PhD candidate at School of Computer Science and Technology, Beijing Institute of Technology, China. His research interests include information extraction, social network analysis data mining, and machine learning.

Yizhou Wang received the BEng degree from Beijing Institute of Technology, China in 2022, where he is currently a master student. His research interests include deep learning and knowledge fusion.

Zhendong Niu received the PhD degree in computer science from Beijing Institute of Technology, Beijing, China in 1995. From 1996 to 1998, he was a postdoctoral researcher at University of Pittsburgh, Pittsburgh, PA, USA, where he has been a joint professor at School of Computing and Information since 2006. He was a researcher and adjunct faculty member at Carnegie Mellon University, Pittsburgh, USA from 1999 to 2004. He is currently a professor at School of Computer Science and Technology, Beijing Institute of Technology, China. His current research interests include informational retrieval, software architecture, digital libraries, and Web-based learning techniques.