FTRLIM Results for OAEI 2019 ⋆ ⋆⋆

Xiaowen Wang[1], Yizhi Jiang[1], Yi Luo[1], Hongfei Fan[1], Hua Jiang[1], Hongming Zhu[1⋆⋆⋆], and Qin Liu[1,2]

[1] School of Software Engineering, Tongji University, Shanghai, China
[2] Tsingtao Advanced Research Institute, Tongji University, Shanghai, China
{1931533,1931566,1731530,1653545,fanhongfei, zhu_hongming,qin.liu}@tongji.edu.cn

Abstract. To achieve better efficiency and feasibility in instance matching between two datasets, we proposed a system named FTRLIM, which is based on the FTRL (Follow the Regularized Leader) model. The FTRLIM system supports the generation of indexes for instances, which enables the system to figure out possible matching instance pairs efficiently. FTRLIM participated in the SPIMBENCH track of OAEI 2019, and obtained the highest F-measure in SANDBOX and almost the highest F-measure in MAINBOX, with the least time cost. The results also provided potential directions for further improvement of FTRLIM.

1 Presentation of the system

1.1 State, purpose, general statement

Researchers have worked a lot on ontology alignment, and early methods mainly focused on matching ontologies based on the schema. Recently, the instance-based matching has gradually become a promising topic.[1] There exists many ontology matching systems that support the solution of the instance matching problem, such as LogMap[2], AML[3], Lily[4], RiMOM-IM[5] and so on. With the rapid growth of data scale, it has become a practical requirement to complete the task of instance matching among large-scale knowledge graphs.

FTRLIM is designed to provide an effective and efficient solution for matching instances among large-scale datasets, whose core functionalities are listed as follows:

1. Build indexes for instances based on textual attributes. Only instances with the same index have the possibility to be aligned.

⋆⋆ This research has been supported by the National Key R&D Program of China (No. 2018YFB0505000), the Science and Technology Commission of Shanghai Municipality (No. 17511107303, No. 17511110202), the National Natural Science Foundation of China (No. 61702374), the Shanghai Sailing Program (No. 17YF1420500) and the Fundamental Research Funds for the Central Universities.
⋆⋆⋆ Corresponding author, email: zhu_hongming@tongji.edu.cn

2. Calculate the similarity between two instances on certain attributes and relationships. Different methods have been used to calculate the similarity according to the data types of attributes or relationships.
3. Generate the train dataset for the FTRL model [6] from the given data automatically. Specific instance pairs are selected as train set during the matching process without manual operations.
4. Aggregate similarities of different attributes and relationships into a similarity score with the FTRL model, which is trained after the generation of the train set.
5. Select aligned instances according to similarity scores between each instance pairs.
6. Customize all procedures based on configuration files.

FTRLIM is a newly developed system and it is the first time that we have participated in the OAEI evaluation. We expect to check the feasibility and efficiency of our system, and thus we rebuilt our system using Java with core functionalities. The complete version of FTRLIM has been developed and deployed on a Spark cluster, which provides the system with ability to deal with large-scale data. The user feedback mechanism has been integrated into the system as well. The system will correct matching results on the basis of feedback. Last but not least, the system also supports merging aligned instances' attributes and relationships.

1.2 Specific techniques used

FTRLIM consists of five major components: Index Generator, Comparator, Train set Generator, Model Trainer and Matcher. The system accepts input instances in OWL format, which are stored in source dataset and target dataset respectively. FTRLIM will find aligned instances between the two datasets. The architecture of FTRLIM is presented in Fig.1.

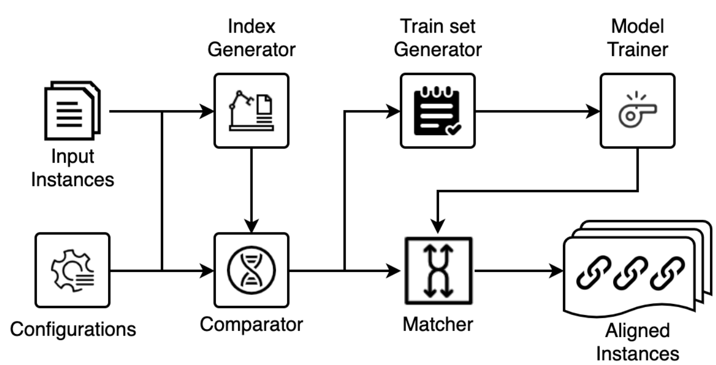

Fig. 1. FTRLIM System OAEI 2019

Index Generator Since the scale of instances that need to be aligned is usually very large, it is very time-consuming and space-consuming to compare all the instances with each other to find aligned instance pairs. FTRLIM uses textual information related to instances to filter out instance pairs that could be aligned efficiently. This work is done by Index Generator. Index Generator plays an important role in FTRLIM. It builds indexes for all input instances based on their attributes. The system first extracts values of a specified instance attribute, then regards each of the values as a document, all of which will constitute a document set. The measurement TF-IDF is used to find keywords for each document. Finally the indexes of an instance are generated from the combination of its keywords. FTRLIM supports users to generate indexes for instances via more than one attribute. In this scenario, different indexes of an instance created referring to different attributes will be concatenated together as the final index. Instances with the same index are divided into the same instance block, and instances from different sources under the same block will form candidate instance pairs. Only when a pair of instances is a candidate pair can it be aligned in the following procedures. When there are only two instances from different data sources in the same block, these two instances will form a unique instance pair[5], which will be regarded as an aligned instance pair directly. Missing value of attributes is taken into consideration to avoid losing candidate instances as far as possible.

Comparator All candidate pairs will be sent to the comparator to calculate similarity. The comparator compares two instances from different aspects. The edit distance similarity is calculated for textual instance attributes, while the Jaccard similarity is calculated for instance relationships. The calculation results will be arranged in order to form the similarity vector. For example, if we compare a candidate pair (x_1, x_2) under two attributes (a_1, a_2) and relationship r_1, the similarities of (x_1, x_2) from each aspect are 0.3, 1 and 0.8, respectively, the similarity vector should be $\langle 0.3, 1, 0.8 \rangle$. All the pairs are compared from identical aspects to ensure that the same dimension of different similarity vectors has the same meaning.

Train set Generator Judging whether a pair of instances is aligned is actually a binary classification problem. We innovatively introduced the FTRL model to solve this problem. The FTRL model has ability to complete the task of classification in large-scale datasets. The model needs to be trained at first. The component, Train set Generator, will generate train set for the FTRL model. The train set is composed of instance pairs' similarity vectors as well as their similarity scores. The Train set Generator regards all unique pairs as aligned pairs. Therefore, it selects all similarity vectors of unique pairs as positive samples, and assigns them with similarity score 1.0. The unaligned pairs are built by replacing one instance of each unique pair randomly. These pairs are assigned with similarity score 0.0 and treated as negative samples in the train set. The input of the FTRL model is the similarity vector, and the output is the similarity score. This component is different from the complete version of FTRLIM, which will be introduced in Section 1.3.

Model Trainer The FTRL model is trained in this component with hyperparameters in configuration files. Benefiting from the FTRL model's feature, the training process won't cost a long time. The trainer plays a greater role in the complete version as well: it can be used to accept the feedback of users and adjust the parameters of the FTRL model. Users are allowed to choose a batch of candidate instance pairs and correct the similarity score, or pick up a certain pair to correct.

Matcher All candidate pairs will obtain their final similarity scores in this component. The trained FTRL model accepts all the similarity vectors and predicts the matching scores of them. Instance pairs with score larger than 0.5 will be regarded as aligned pairs. They will form the final output of aligned instances together with unique pairs.

Configurations FTRLIM is easily to be tailored according to user's requirements. We expect that all matching procedures are under user's control, thus we allow users to customize their own FTRLIM system using configuration files. Users are able to set the attributes for index generation, the attributes and relationships for comparison, the hyperparameters for the FTRL model and many other detailed parameters to get a better result.

1.3 Adaptions made for the evaluation

To participate in the evaluation, we rebuilt the FTRLIM system and replaced some manual operations with automatic strategies. In the complete version, FTRLIM does not regard all unique pairs as aligned pairs directly. It will compute the mean value of similarity vectors' elements as the raw score for each instance pairs. Then it will select a batch of instance pairs that have raw scores higher than a threshold as positive samples, as well as the same amount of instance pairs whose raw scores are lower than the threshold as negative samples. Users will determine the similarity score by themselves to generate the train set. In the version developed for OAEI, this procedure is changed as we mentioned in 1.2. We excluded the non-core functionalities of the system, and made the ways of input and output suitable for the evaluation.

1.4 Link to the system and parameters file

The implementation of FTRLIM and relevant System Adapter for HOBBIT platform can be found at this FTRLIM-HOBBIT's gitlab page.[3]

[3] https://git.project-hobbit.eu/937522035/ftrlimhobbit

2 Result

In this section, we present the results obtained by FTRLIM in the OAEI 2019 competition. FTRLIM participated in the SPIMBENCH track, which aims at determining when two OWL instances describe the same Creative Work. The datasets are generated and transformed using SPIMBENCH[7]. We are the latest team to join this track. Our competitors are LogMap[2], AML[3] and Lily[4], who have participated in this track for many years. The results are published in this OAEI 2019 result page[4].

2.1 SPIMBENCH

The SPIMBENCH task is executed in two datasets, the SANDBOX and the MAINBOX, of different size. The SANDBOX has about 380 instances and 10000 triplets, while the MAINBOX has about 1800 Create Works and 50000 triplets.

Table 1. The result of SANDBOX

	FTRL-IM	AML	Lily	LogMap
Fmeasure	**0.9214175655**	0.864516129	0.9185867896	0.8413284133
Precision	0.8542857143	0.8348909657	0.8494318182	**0.9382716049**
Recall	**1**	0.8963210702	**1**	0.762541806
Time performance	**1474**	6223	2032	6919

Evaluation results of SANDBOX are summarized in Table 1, where the best results are indicated in bold. Compared with AML[3], Lily[4] and LogMap [2], FTRLIM obtained the highest F-measure, highest recall and best time performance, while the precision is 0.08 lower than LogMap that has the best precision.

Evaluation results of MAINBOX are presented in Table 2 with the best results in bold. Our system is approximately 41% faster than Lily and 17 times faster than the slowest one, while the F-measure is only 0.00014 lower than the best one. We obtained the nearly full mark on recall and the second highest precision as well.

Table 2. The result of MAINBOX

	FTRL-IM	AML	Lily	LogMap
Fmeasure	0.9214787657	0.8604576217	**0.9216224459**	0.790560472
Precision	0.85584563	0.8385678392	0.854638009	**0.8925895087**
Recall	0.9980145599	0.8835208471	**1**	0.7094639312
Time performance	**2155**	39515	3667	26920

[4] http://oaei.ontologymatching.org/2019/results

3 General comments

3.1 Comments on the result

FTRLIM has achieved satisfactory performance in both datasets of SPIMBENCH, especially in the SANDBOX. The Index Generator makes a significant contribution to achieving the results. It helps the system filter out instance pairs with a high possibility to be aligned effectively and efficiently. The comparator only needs to compare instances with the same indexes rather than every instance pairs. The datasets of SPIMBENCH contain a wealth of textual information, and there are many attributes that can be used to build indexes or to compare the similarity among instances. The FTRL model trained by the Model Trainer component is as smart as we expect to learn a weight for attributes or relationships and distinguish pairs of instances pointing to the same entity in real world.

Compared with LogMap, the F-measure of FTRLIM is 8-13% higher while the precision is 4-8% lower. This result shows that FTRLIM could still be improved to obtain higher precision. The OAEI version of FTRLIM considers unique pairs as aligned instances unconditionally, which is not always true. One possible way to solve the problem is validating the matching results. This is one of the centers of our future work.

3.2 Improvements

There are still many aspects to be improved in the FTRLIM system. Besides adding validation stage that described in 3.1, we will continue to optimize the algorithm of generating indexes for instances and the matching strategy in following work. More comparison methods and supporting data types should be attached to our system as well. And we are committed to building the GUI for our system. Although FTRLIM is specially designed to solve the instance matching problem, it is also expected to produce meaningful results in other similar tracks in the future.

4 Conclusion

In this paper, we briefly presented our instance matching system FTRLIM. The core functionalities and components of the system were introduced, and the evaluation results of FTRLIM were presented and analyzed. FTRLIM achieved significantly better time performance than other systems in both datasets of SPIMBENCH, and got the highest F-measure in SANDBOX and almost the same F-measure as the best one in MAINBOX. The results proved the effectiveness and high efficiency of our matching strategy, which is important for matching instances among large-scale datasets.

References

1. Otero-Cerdeira, L., Rodríguez-Martínez, F.J., Gómez-Rodríguez, A.: Ontology matching: A literature review. Expert Systems with Applications **42**(2), 949–971 (2015)
2. Jiménez-Ruiz, E., Grau, B.C., Cross, V.V.: Logmap family participation in the oaei 2018. In: OM@ISWC (2018)
3. Faria, D., Pesquita, C., Balasubramani, B.S., Tervo, T., Carriço, D., Garrilha, R., Couto, F.M., Cruz, I.F.: Results of aml participation in oaei 2018. In: OM@ISWC (2018)
4. Tang, Y., Wang, P., Pan, Z., Liu, H.: Lily results for oaei 2018. In: OM@ISWC (2018)
5. Shao, C., Hu, L., Li, J.Z., Wang, Z., Chung, T.L., Xia, J.B.: Rimom-im: A novel iterative framework for instance matching. Journal of Computer Science and Technology **31**, 185–197 (2016)
6. McMahan, H.B., Holt, G., Sculley, D., Young, M., Ebner, D., Grady, J., Nie, L., Phillips, T., Davydov, E., Golovin, D., Chikkerur, S., Liu, D., Wattenberg, M., Hrafnkelsson, A.M., Boulos, T., Kubica, J.: Ad click prediction: a view from the trenches. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD) (2013)
7. Saveta, T., Daskalaki, E., Flouris, G., Fundulaki, I., Herschel, M., Ngomo, A.C.N.: Spimbench : A scalable , schema-aware instance matching benchmark for the semantic publishing domain (2014)