Table of content

Ontology Mapping with Machine Learning Techniques

Soufia Naseri
[redacted]

ABSTARCT

The demand for ontology mapping have been on the rise mostly due to expansion of applications in semantic web. With the expansion of semantic web, it is not possible to meet all the requirements with a single ontology, but many. Therefore, exploring different approaches to combine two or more ontologies by applying specific tools and methods is of great value. In this paper, we review ontology mapping applications via machine learning techniques. Furthermore, we will discuss the advantages of both supervised algorithms, such as SVM, and unsupervised, GLUE.

KEYWORDS

Ontology; Ontology Mapping; Ontology Matching; Machine Learning; Neural Networks; SVM; GLUE

1. INTRODUCTION

A great challenge in Ontology mapping is finding an algorithm which can understand and automatically relate words and sentences together like the human minds work. I have considered machine learning techniques since they can be a huge help in automating processes. There are great algorithms available in machine learning methods. These algorithms and mathematics behind it, have been worked on for a while. Mathematician, scientist, data scientist and engineers have put their minds together to figure out how to use math to solve any existing problem and came up with these algorithms. However, at the end there are some minor error but when applying this to ontology mapping, these minor errors have a significant impact on the final results. Manually mapping of ontologies is not applicable since the available information is enormous. Therefore, the challenge is to make that, an automated process. In this paper, machine learning techniques are considered because the main focus in machine learning is for machines to learn and automate the process. The goal is by decreasing the run time, the accuracy not only stays the same but also increases. Following explains several issues that is possible to improve.

1.1 SVM PROBLEMS

Support vector machine (SVM) is one of the best algorithms available in ML and many papers claim that if they substituted part of their algorithm with SVM they can use SVM advantages fully and better performance can be obtained. SVM training algorithm is very powerful in such a way that with only a few training sets, the learners give accurate results. However, there is a problem which is the overfitting. Due to high dimensional processes in SVM, the least square regression, which helps data to fit, will face issues which end up reducing performance. Overfitting happens when a number of features or attributes in spaces are greater than available entities or classes [1]. Also, it happens when the dataset is very small and perfect matching occurs therefore, causing an overfitting in the algorithm. However, in our case this wouldn't be the reason since our data is massive.

1.2 GLUE PROBLEMS

GLUE has been already discussed previously. It's a tool which is being used for ontology mapping. The accuracy in GLUE is between 66% to 97% [3]. Although it is great performance, there's still room to improve it. GLUE uses a relaxation labeler where a node or entity inherits the features of surrounding nodes. GLUE structure is shown in figure 2. This technique has been used to do local optimization of the output. It has been mostly used in image processing by looking at the surrounding pixels and also NLP by looking at before or after words. Although accuracy is being increased by relation labelling, this algorithm sometimes gets stuck in local maxima and thus prompting inaccurate results.

2. BACKGROUND

Semantic web was proposed to replace World Wide Web. World wide web has created a great area for people to communicate, have access to information and exchange data. In order for them to have access to accurate results, Semantic Web has been taken into consideration. Semantic web is for machines to understand human communication and to consider concepts of words. Ontology is a main part of Semantic Web, since ontology is concepts of words in a hierarchical structural. To combine different concepts from different domains, ontology mapping is used in order for different concepts to share knowledge and build more information [2].

Machine learning algorithms are categorized into two different learners, supervised and unsupervised learning. Unsupervised learners such as Neural Networks have been around longer than supervised such as Support Vector Machine and Naïve Bayes belong. In supervised learning the system model is built on a given training set. Training set is where the input and output is given to the machine and it's expected that machine find the relation between them and learns from them. Most of Supervised learners are known to have a better performance than unsupervised since in unsupervised learners there is no training set, meaning there is no preference or information on the output, therefore, "estimation" should be made [1]. However, sometimes the result in unsupervised learnings are better or at least as good as supervised ones. This of course depends on the dataset that it's being classified. In [4], Prior+ which is a neural network and part of the unsupervised algorithm, claims to have a performance as well as GLUE which uses a supervised algorithm.

3. METHODOLOGY IN DETAIL

To come up with a perfect algorithm which understands the entities, we should study how human brain is able to do so. When we are looking for concepts and relations between entities, we don't think about it since it comes to our mind in an instant. But there's a lot of information processing in that instant. Related neurons to the features of that specific image or word would light up hence sending the information to the brain. As a result, we are able to understand their relations. Our brain works and communicates with neurons and then runs it through a complex algorithm then saves them in memory which has its own set of complicated processes. The brain harmonizes the different layers of algorithms and makes them work with one another.

It has taken millions of years for human brain to be this complex and yet harmonized. Though it was our environment and the way of living which made us smarter and smarter everyday. Human brain is always learning and that is what makes us different. So, in order to have an algorithm that works as well as our brain we should first study the development of our brains.

3.1 Taking advantage of already mapped data

Creating an algorithm or a website which gathers all the manually mapped data from owners, servers or other websites with owners' consents. Thus, inputting those mapped data and feeding it to our training set which can be a huge advantage rather than using a knowledge based. These new and additional information make the training algorithm stronger.

3.2 Parallelizing

Parallelizing is one of the proposed method. Evaluating and processing big data in parallel increases run time drastically. When talking about long time for execution time, we are talking about hours or days of computation sometimes even months. In the reviewed papers, surprisingly not many focused-on parallelizing. Therefore, by decreasing the run time, the saved time can be concentrated on increasing accuracy of the algorithms.

3.2.1 On Different Algorithms

For example, in frameworks each matcher with their algorithms can be run parallel since its data would be independent thus separating is easier. There should be a communication between them so when one's task is finished; the system would be aware of it so that critical path will be less. Critical paths are the paths that their output is required for other tasks. They can be input or the conditions of those tasks. Parallelization can be done on different computers (different CPUs) or one computer but e.g. if there are three algorithms to be determined three cores is needed.

3.2.2 On One Algorithm

This method is harder since data are both dependant and independent, so separating those takes are much more complex. However, if the independent data get extracted parallelization is then possible to be done in a timely manner. In the following proposed solution, layering, this can be explained briefly.

3.3 SVM Solution

SVM has some non-negative values to tune and they should be given to the dataset before its training. Also, kernel functions should be chosen carefully. Kernel Functions are the functions that user can feed into the system for better performance. These functions are fed to increase the space (dimension) of the building models. There few great functions available already however, can be also customized. Different kernel functions can be run in parallel and choose the best function with the appropriate tuning values. The parallel computation can be done on several CPUs or GPUs where execution time decreases and it is possible to avoid overfitting.

3.4 GLUE Solution

In [3], it is not clear what portion of data have been used for training set, however, by increasing the dataset, the accuracy would be much greater. If all the available data have been used for training set, the solution would be to have a better knowledge base. Though experts have been relating entities and concepts together for a while, it is not yet enough.

If the results of the matchers have a *lower percentage* than expected, then mapping should be done again but this time plotting the data while running. Moreover, training the dataset for more iterations and don't let the relaxation labeler stops when it sees the first maxima (saves it in a memory so it won't lose this number). Next, letting the data run for a bit longer until finds the global maxima may avoid the algorithm to stuck in local maxima. For this technique, it is possible to design an algorithm where it can make possible error suggestions based on its *experience* with such those data. If local maxima were a possible option, then an alert pops up so that learner use the above method. For estimation part, e.g. two columns table is created one to save data type and the other to save accuracy percentage. Then, running a probabilistic algorithm to make an estimation. This results in, having an expected accuracy results over the expected one from the original algorithm. By comparing them together it would make suggestion and avoid possible errors.

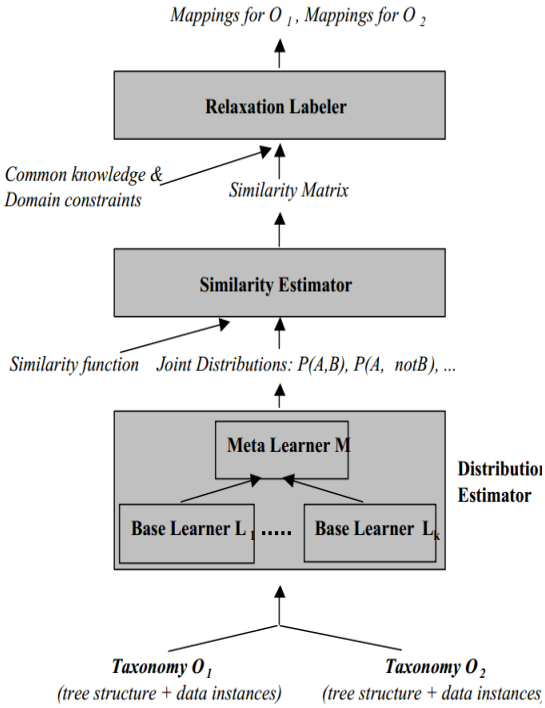

Figure 1. GLUE Structure [3]

3.5 Different Layers of Algorithm

In this part, the proposed solution consists of different layers of machine learning algorithms. Taking advantage of most of the ML algorithms would help us to increase the accuracy meanwhile parallel running decrease the run time. This is done by first clustering and then classifying the data. Clustering is categorized under unsupervised learning and popular clustering methods are Principal Component Analysis (PCA), K-means clustering, Hierarchical clustering and so on. The suggested algorithm uses two types of clustering, first PCA is used which is great at reducing noise [5]. Noise in data specially in text data means the meaningless and useless texts or data. By reducing those not needed data computation would be much faster. Next, K-means clustering is added to those filtered data as it is shown in figure 2, level two is where k-means is applied. Clustering these data is the same idea as making them domain-specific since the clusters are non-overlapping [5]. By having them

into different clusters, data is more sensible and their similarities and dissimilarities of data would be more obvious. Moreover, it is breaking them down to independent data as much as possible for parallel running purposes. Then, classifying each cluster separately in parallel and finding them hierarchical structures, this step happens at level three. At the end, we run the classifier algorithm on the data points which were closed to the line separator in the level two. The closer the data points are to the line the closer the meaning and relation would be thus by analyzing and computing only those data points we are able to finalise our hierarchical structure.

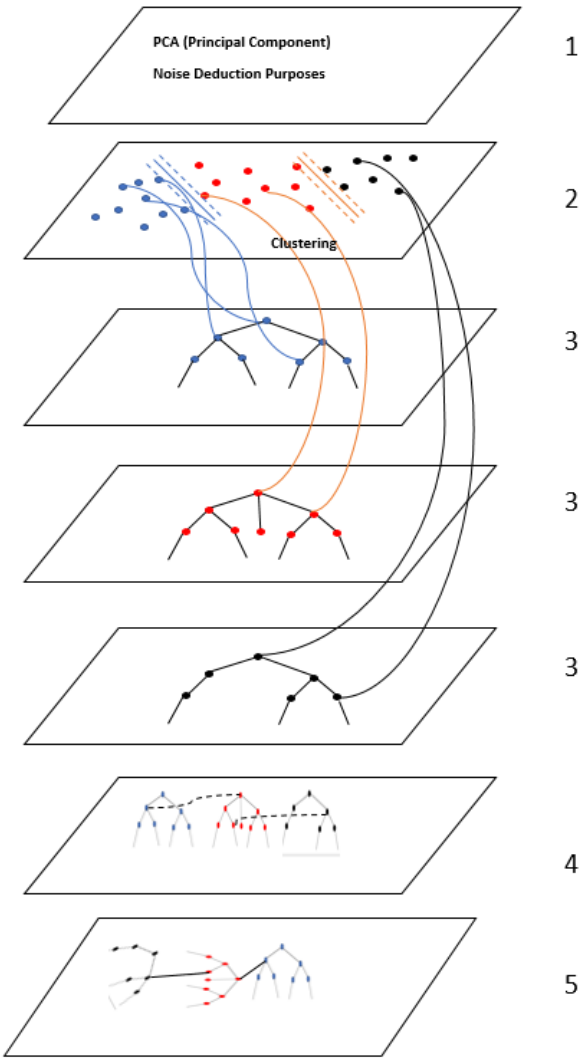

Figure 2. Structure of running different ML algorithms. Level or stage number is on the right-hand side.

To find which algorithm to be used for data, main algorithm can decide which algorithm or clusters to choose from depending on data set type. It can ask for multiple data set types and by just having the rule-based or probabilistic methods, the most applicable algorithm would be chosen. These methods are very simple since many ML algorithms have been ran on many different data types and by comparing its result it's possible to choose the most effective algorithms.

4. RELATED WORK

4.1. SVM

Overfitting happens in a framework where algorithm gets to choose the better matcher for the dataset. In [6], instead of using least square regression a machine learning boosting algorithm called Adaboost, has been used for selection of those matchers when doing meta-matching.

4.2 GLUE

GLUE's learners are general-purpose classifier, [3] suggests that selection of the features to be done in a which might improve its performance yet have it as "global-perspective". Meaning taking advantage of the available ontologies which domain experts put them together for that specific field. Moreover, it has been mentioned by adding a thesaurus like WordNet the performances may increase [3]. Another suggestion for lack of their performance was given, to have an extension to GLUE but the way that author explains it means additional instances. Additional instances are not a great idea since the learning algorithm is not learning but having some additional cases added to it. However, it might work well if the classifier was domain-specific.

4.3 Parallelization

In [8], other techniques of machine learning which haven't been really focused on ML field it has been used. The method is *rewarding* the system when it does a task correctly. [8] only emphasized on parallelizing by using a microtask platform. The platform, divides the task into multiple independent tasks thereby running them in parallel. However, their algorithm has 50% of accuracy only.

4.4 Layering

In [6], *AgreementMaker* uses layer formatting to combine matchers, but none of them has used ML techniques.

- 1st layer: features of different concepts such as their class and instances are determined with use of a cosine similarity method.
- 2nd layer: two matchers have been used: descendant's similarity inheritance and sibling's similarity contribution [7].
- 3rd layer: By combining those two layers and computing a linear weighted combination over them, outputs are more desirable.

5. EVALUATION

There is no doubt that adding the already mapped data helps us train our training set much better. Using this method itself increases the accuracy drastically. By adding it to GLUE which has accuracy 66 - 97% [3], it might be possible to reach 100% depending on data type of course. In addition, having them run in parallel for classification purposes decreases the run time. In parallelizing the trade off is the critical path which may cause a bottleneck in the system. Bottleneck may occur when the entire process depends on one task which hasn't finished processing and the rest of the tasks are in the middle of their process. However, with use of multi-threading and *Semaphore* which is part of a real-time programming it is possible to overcome this issue. As in [8], have done their processes in parallel it is time to emphasis on running algorithms in parallel and break the tasks into small and independent ones. Unfortunately, in [8] their accuracy is not as high as the top ranked algorithms are. This could be due to the weak algorithms or could be due to parallelism since the microtask platform is breaking down the tasks it might be eliminating some important information thereby causing low accuracy. Applying PCA or Principal component analysis can help to extract the useless data and then be sent to a breaking down platform such as microtask platform. This could improve the accuracy while decreasing the run time.

In [3], author proposed couple of solutions and one of them seem to be including adding more instances. The learner is supposed to learn and not given additional instances therefore, this is not an optimal solution. If all the mapped data get gathered and fed into training sets, the training dataset increases thereby having a better performance rather than before. In above proposed method where the possible errors are taken into the account and running them using probabilistic reasoning over them it is possible to avoid the local maxima. In [3], "global-perspective" have been suggested for local maxima problem where it hasn't been talked about in details. However, it's been concluded that has the same meaning as global optima. Global optimization consists of very complex formulation and algorithms where it has shown a way to avoid local maxima and minima.

Layers of algorithm helps to take advantage of most of ML algorithms and their combination improves accuracy. It's been proven that PCA reduce noise and applying it before clustering improves the learning data. This method might face difficulty when K-means algorithm is in higher dimension, therefore the data points close to margin can be many which cause increase in execution time. At the end by parallel running the classifiers, decreased run time might get cancelled out when having higher dimensions in clustering. Another, potential downside is that number of clusters should be given to the machine by the user. So, if giving a very different value e.g. having two domains but four or six clusters the data is not well organized and it may decrease the performance.

6. CONCLUSION

In conclusion, the proposed methods can, theoretically, revolutionize semantic web through highly efficient and intelligent ontology mapping. By lowering run-time there will be more resources to be allocated towards accuracy of performance. These solutions require the user to have the ability to run real-time parallel processes. Further advancements can take place by investigating global optimization algorithms since GLUE has a considerable potential to reach %100 accuracy.

7. REFERENCES

[1] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. Ventura, CA: Academic internet Publishers, 2007**.**

[2] Kalfoglou, Y. and Schorlemmer, M. 'Ontology mapping: the state of the art", *The Knowledge Engineering Review*, 18(1), 2003: 1–31.

[3] A. Doan, J. Madhavan, P. Domingos, and A. Halevy, "Ontology Matching: A Machine Learning Approach," *Handbook on Ontologies*, pp. 385–403, 2004.

[4] Ming Mao, Yefei Peng, Michael Spring, "An adaptive ontology mapping approach with neural network based constraint satisfaction", *Web Semantics: Science, Services and Agents on the World Wide Web*, 8(1), pp: 14-25, 2010.

[5] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning: with applications in R*. b: Springer, 2013.

[6] Pavel Shvaiko, J´erˆome Euzenat. "Ontology matching: state of the art and future challenges.", *IEEE Transactions on Knowledge and Data Engineering,* 25 (1), pp. 158-176, 2003.

[7] I. F. Cruz and W. Sunna, "Structural alignment methods with applications to geospatial ontologies," *Transactions in Geographic Information Science*, vol. 12, no. 6, pp. 683–711, 2008

[8] C. Sarasua, E. Simperl, and N. F. Noy, "CrowdMap: Crowdsourcing Ontology Alignment with Microtasks," *The Semantic Web – ISWC 2012 Lecture Notes in Computer Science*, pp. 525–541, 2012.