Creating a PurposeNet Ontology: An insight into the issues encountered during ontology creation

Rishabh Srivastava
LTRC
IIIT Hyderabad
(rishabh.srivastava@research.,

Soma Paul
LTRC
IIIT Hyderabad
soma@)iiit.ac.in

Abstract

PurposeNet is an ontology based on the principle that all artifacts (man-made objects) exist for a purpose and all its features and relations with other entities are giverened by its purpose. We provide instances of ontology creation for two varied domains from scratch in the PurposeNet architecture. These domains include MMTS domain and recipe domain. The methodology of creation was totally different for the two domains. MMTS domain was more computaionally oriented ontology while recipe domain required a post-processing after manually entering the data. The post-processing step uses hierarchical clustering to cluster very close actions. MMTS ontology is further used to create a simple template based QA system and the results are compared with a database system for the same domain.

1 Introduction

This paper shows the procedure, advantages and disadvantages of enriching the datasets. The raw data we observed was different for the 2 domains. Once we decide the structure, the MMTS domain data was collected automatically. Although the integral actions had to be entered manually. On the other hand, for the recipe domain, after the initial manual population of data, post-processing for clubbing the nearest actions had to be carried out.

A question answering application in the domain of MMTS train timing has also been talked about in brief in this paper. The results are very satisfactory as they show a significant improvement over a similar RDBMS system. The prepared ontology can be put to use in several applications. Question answering, dialog systems, machine translation systems, etc.

1.1 Literature Review

Multiple attempts have been made in the past for populating data in PurposeNet. Some have tried automatic means while the others do it manually. (Singh and Srivastava, 2011a) have used surface text patterns (STPs) to extract information directly from Wikipedia text. They had tried populating only some of the descriptor features of the ontology. Other major works in automatic population for PurposeNet Ontology (Mayee et al., 2010a; Mayee et al., 2010b) have also tried to populate purpose of the artifact directly into PurposeNet using surface text pattern, typed dependency parser and neural networks. Although, the results of the dependency parser are quite convincing, the methodology by both the aforementioned techniques cannot be used to populate the complete ontology. (Singh and Srivastava, 2011b) give a deep insight into the various kinds of artifacts one can find in the real world. They also give a brief introduction to extract these into an ontology. None of the ontologies created in the PurposeNet architecture have been created completely using automatic methods.

One key thing to notice here is that both the domains chosen deal with abstract artifacts. When we talk of the MMTS timing domain, we talk about the artifact *journey* and MMTS train is just a medium to complete that journey, whereas when considering the recipe domain, we talk about an abstract artifact (a *knowledge* which can aid us in preparation of a physical artifact (Recipe is comparable to knowledge as mentioned by Singh and Srivastava (2011b)) . PurposeNet till now has only been populated for physical artifacts (vehicle and hotel (Rallapalli and Paul, 2012) artifacts).

Recently, a very active interest can be seen in the community for complex domains(Kambhatla et al., 2012). A very early work dealing with train-scheduling expert system has been reported in (Chang, 1988). They have made the system for engineers to construct and schedule an AC electrified railways. Chinese train ontology has been created using UML has been presented in (Hu et al., 2006).

Works of Kalem and Turhan (2005) try to explain how to create a recipe ontology in detail. The difference however is that it captures only the description part of the recipe and for the actual recipe, redirects user to the url. It doesn't capture the series of steps and intermediates required for a recipe. Our work is much influenced by ColibriCook (DeMiguel et al., 2008). Ontology defined in TAAABLE is also nearly the same as PurposeNet recipe ontology created by us (Badra et al., 2008).

1.2 PuporseNet Architecture

The architecture of PurposeNet is described in details in (Sangal et al., 2013). There a few major points we would like to emphasize here:

- All artifacts exist for a purpose, and all its components aid the artifact to satisfy its purpose.
- PurposeNet defines 3 kinds of relations, namely descriptor features, action features and action frames and relationships with other artifacts. They have listed a series of relations below these 3 categories. With a change of domain, one can change these listed relations, but one cannot change the 3 basic relations. PurposeNet by default allows:
  - 8 **artifact-artifact relation**
  - 25 **descriptor properties**
  - 3 **action features** (4 including maintenance)

  which totals to 37 relations.
- Similar is the claim for action frame. Precondition, subaction, outcome and semantic roles are enough to complete an action frame. Although one can increase the number of semantic roles according to a domain. PurposeNet by default allows:
  - 4 **action frames** (7 in nested)
  - 21 **semantic relations**

  which totals to 28 relations. Total default PurposeNet relations equals to 65 relations.
- Descriptor features can also be changed according to the domain.

Sanctity of these rules must be maintained, if one wants to create a PurposeNet ontology. For various domains, one needs to have a substantial amount of data to understand all the possible variations.

1.3 Ontology reasoning

We emphasize on using OWL because firstly, it is a W3C standard for ontologies and secondly, ontologies can help in inferencing of facts. Even a proposition expressed in FOL (first order logic) is more expressible than its sibling in relational database. OWL supports description logic while is a higher logic than FOL.

2 MMTS domain

Figure 1 shows the structure of the MMTS ontology.

2.1 Entities, Actions and Relations

In this section we explain the additional features provided to the Ontology which contribute to the real richness of the Ontology.

2.1.1 Entities

The prime entities featuring in the ontology include:

1. Day
2. Money
3. Coach
4. Place
   - Station
   - Non-Station
5. Route
6. Ticket
7. Train
   - AllDayTrain
   - AllPublicTrain
   - LadiesTrain
   - SundayTrain

2.1.2 Actions

1. Buy
2. Travel
3. StopAtStation

2.1.3 Relations

We had to include certain relations in the MMTS ontology as PurposeNet architecture doesn't completely clarify its support on timing, duration, routes or stoppages. Apart from the relations declared by PurposeNet, we had to include:

1. atStation: to mark the stop of a train at a station.
2. comprisesOf (inverse belongsTo): to mark what a route comprises of.
3. forRoute: to mark the stop is for what route.
4. hasStopNo (inverse ofTrain): to mark the stops of a train.

[1]sr means semantic roles

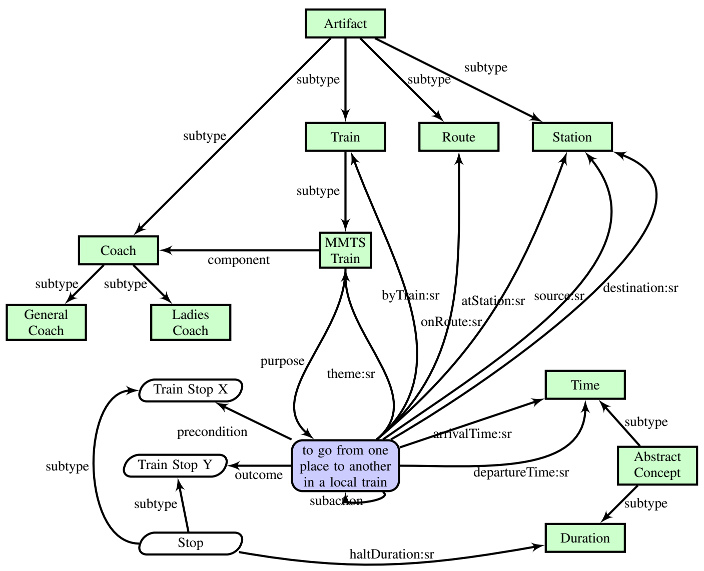


Figure 1: Figure shows the ontological structure of the MMTS train domain. Solid green boxes represent entities, blue rounded boxes represent actions and white trapeziums represent states. The upper ontology has not been shown for diagram clarity. [1]

5. isTravelledBy (inverse travelsThrough): to mark the trains which travel through a route.
6. reverseRouteOf (symmetric): to mark the route which is reverse of a route.
7. runs: to mark the days on which a train runs.
8. arrivesOn: to mark the arrival time of a train at a station.
9. departsAt: to mark the departure time of a train from a station.
10. stationHaltTime: to mark the halt time of a train at a station.

2.2 Algorithm for ontology population

The ontology population was primarily divided into 2 sections. One of them dealt with the train and stop information (Section 2.2.1), while ther other with the actions part of the ontology (Section 2.2.2).

2.2.1 Train Ontology

We start with the data from the MMTS website[2] and then apply the algorithm 1 for data population.

2.2.2 Action Ontology

The major actions in the MMTS train domain include:

```
1. Pick a route.
2. For all the trains that run in the route start.
3. Start from the train with the earliest starting time.
4. Start from the first stop and proceed towards the last
   and for each stop enter the following information.
     • route            • station
     • train number     • departing time
```

Algorithm 1: Algorithm shows the steps to be followed to populate MMTS ontology using MMTS online data.

1. buying a ticket
2. boarding a train at a stop
3. travelleing
4. alighting a train at another stop

They are basic actions and the information about them cannot be found in texts. We have tried to break these actions into more basic ones. eg. for buying a ticket, one has to pay money and receive ticket.

3 Recipe domain

We started populating this ontology from scratch, manually. All the information was filled in the PurposeNet framework. One of the interesting observations was that the recipe domain contained a lot of states which cannot be named. We called

[2]http://www.mmtstraintimings.in/

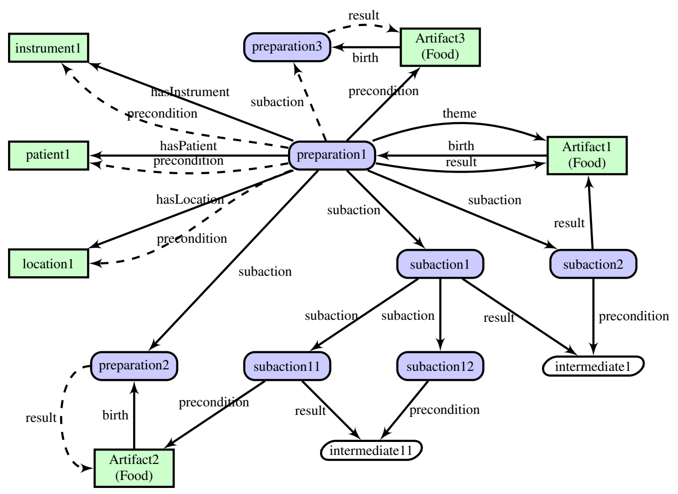


Figure 2: Figure shows the ontological structure of the recipe domain. Solid green boxes represent entities, blue rounded boxes represent actions and white trapeziums represent states. Dashed arrows represent the inferred relations.

them intermediates in our ontology. All the four action features including preconditions, subactions, semantic roles and outcomes, played a major roles in describing the ontological details. Location and themes were the two most prominent semantic roles in the action ontology. To describe an action or state, theme stated the artifacts. Figure 2 shows the structure of the recipe ontology.

3.1 Entities, Actions and Relations

3.1.1 Entities

A well structured list of Entities including the major classes of food so that hierarchy can later be explited has been included in the ontology. The upper hierarchy is enlisted below.

1. Natural artifacts
   - Spices
   - Vegetables
2. Physical artifacts
   - Food Artifact
     - Item Food Artifact
     - Prepared Food Artifact
     - Processed Food Artifact
   - Intermediate Artifact
   - Kitchen Artifact

3.1.2 Actions

This is a long list of basic actions which are used in the cooking domain. We have tried to build a hierarchy of actions here. Some part of this work is onfluenced by the work of Rajan (2013). The upper action hierarchy include:

1. Add
   - Apply
   - Fill
   - Sprinkle
2. Cool
3. Heat
   - Boil
   - Fry
   - Roast
4. Make
   - Prepare
5. Mix
   - Knead
6. Place
   - Cover
7. Press
8. Roll
9. Separate
   - Break
   - Chop
     - Grate
   - Cut
   - Mash
   - Peel
   - Scoop

3.1.3 Relations

No special relation has been added as the core-PurposeNet architecture is sufficient for recipe ontology.

3.2 Algorithm for ontology population

For every dish, we seached for a legitimate recipe on the web and followed the algorithm 2.

3.3 Post-processing

Although the data populated by following algorithm 2 was quite correct, we identified 2 major areas where post processing was required.

1. We did not extract the components/ingredients required for making a dish. Section 3.3.1 talks about the solution of this problem.
2. There was a need for structuring the ontology. The identification of problem is discussed in section 3.3.2

- Continue from the first sentence. All the sentences will have an action.
- If the action is already existant, link it.
- else, create the action and find the thematic roles.
- Mostly, there will be 4 thematic roles, which include:
  - location
  - instrument
  - patient
  - theme

  There is a posibility of precondition also.
- if these are existing (or are raw material), add them, or create an entry.
- for the outcome, if it is existing add it, else
  - if the outcome is a named entity, i.e. it can be called something, create the named entity,
  - else, create an intermediate.

Algorithm 2: Algorithm shows the steps to populate recipe ontology using web recipes.

while section 3.3.3 talks about the solution of this problem.

3.3.1 Extracting raw materials and intermediates for a recipe

To identify the components of a dish, one information was to identify the preconditions. But as we have already seen, the subactions as well as the conditions can be recursive and so the identification of all the raw materials and intermediates becomes somewhat difficult[3]. To ease it, we have put in 4 set of rules in the ontology itself which are as follows:

1. If the birth of an artifact needs another as precondition, then the former requires latter.
2. Defines requires as a transitive entity.
3. If the requirement is a food artifact, then the food artifact is a component.
4. If the requirement is a natural thing, then the natural thing is a component.

3.3.2 Need for extraction of hierarchy

Although the process was quite straightforward, on studying recipes from various websites, we got a different picture altogether.

- Same subdish is prepared by Multiple dishes

  (aloo gobhi paratha: 1.5 cups mashed potatoes)

  (aloo paratha: 2 1/2 cups boiled , peeled and mashed potatoes)

[3]We have not differenciated core-coponents from other components as of now.

- different recipe documents of the same recipe have different text representation. In our ontology, we try to capture a generic version of those variation of dishes.

  (aloo paratha source 1: hierarchy)

  (aloo paratha source 2: flat)
- all the recipe documents consider different individuals as their starting point (ingredients)

  (3 medium potatoes or aloo)

  (2 1/2 cups boiled , peeled and mashed potatoes)
- some recipe documents give chunked information
  - for clarity
    (Masala dosa: Preparing the Dosa Batter, Preparing the potato filling-sabzi, Preparing the Masala Dosa)
  - sometimes chronology plays a part of subgrouping sub part of the dishes
    (Preparing the Dosa Batter: Cover and let the batter ferment for 8-9 hours.)
- similar/trivial activities are given together in recipe documents

  (chop the aloo and add all the spice powders, green chilies, salt.)

  (add the mashed potatoes. add chopped green chilies. mix well and keep aside.)

After the manual population of the PurposeNet data with the recipe information step by step, we got a rough data prepared. The next task at hand was to cluster the similar elements in an ontology so that the ontology can be properly structured. Section 3.3.3 explains how we post-process the ontology semi-automatically to create a structured ontology.

3.3.3 Extracting hierarchy of action in recipe domain

To begin with the experiments, we populated the ontology with our previous algorithm. We then apply agglomerative hierarchical clustering on this data to identify the sub-clusters/subgraphs which are very close. As the hierarchical clustering requires a metric, we chose the metrics as the frequency of edges required for making all the dishes (equations 1, 2 and 3).

$$FrequencySet(FS) = \nu(e), \quad \forall e \in all\_edges\_in\_the\_graph \tag{1}$$

$$max(FS) = argmax(FS) \tag{2}$$

$$merge(FS(i)), \forall n(i) = k, \quad \text{where } max(FS) \geq k \geq threshold \tag{3}$$

To identify the clusters in the recipes, we identified the edges which were most common in the various recipes. We fixed a certain threshhold and being above that threshhold the edges were selected. A subgraph of the selected edges is created and the outgoing edge of this subgraph is declared as the output of the subgraph. A subaction consisting all the actions and relations of tbis subgraph is entered into the ontology. The actions for which these subactions were a subaction is replaced with a higher subaction.

An example: Suppose we enter 5 dishes, 3 of which need a peeled boiled potato. So when we traverse the graph once to enquire the edges being traversed, we find the edges of boiling a potato, getting boiled potato as an outcome, peeling a potato and the outcome boiled peeled potato thrice. Now we create a subgraph of these 4 edges, with boiled peeled potato as outcome. We club these 4 into a single cluster.

Figure 3 represents the process of clubbing subactions in which Figure 3a shows 3 dishes all of which contain boiled and peeled potato either as a precondition or as a result of some subactions. Figure 3b shows the result of running the algorithm after which a new action *prepare peeled boiled potato* is introduced which has the subaction as boil potato and peel potato.

3.3.4 Algorithm

Algorithm 3 is used for extracting the subgraphs.

4 QA Application

To test the effectiveness of the MMTS ontology, we created a dummy QA application which answers questions related to the MMTS train timings. The input was html templates, which contained 20 different types of questions and these were internally mapped to sparql (PrudHommeaux et al., 2008) queries. The pellet reasoner was running over the MMTS ontology. The system is compared with a similar system created in mysql (RDBMS). Table 1 shows the results of the two system for same questions.

The results clearly show that by using an ontology one can answer a lot more questions by just inferring than answering by the information already provided. Traversal is not a very positive step, but at least one does not need to provide an extra information for the same. 14 questions out of 20 are inference based and 4 are traversal based for an ontology. Only information for 2 questions needed to be specifically provided.

- For all the food artifacts, choose their preparation actions and for all such preparations, do the following,
    - Create a graph around the preparations.
    - create an indices of edges with the preparation.
    - While creating the dag keep an account of all the edges traversed.
    - Keep on incrimenting the edges as and when they are traversed.
- Sort the edges according to the frequency of their occurence.
- Set a threshhold below which the edges will not be considered.
- for all the frequqncies try to form a subgraph.
- The final state(s) will be taken as the outlet.
- Check in the index and in all the dishes where any one of the considered edges occurs, replace the subactions enlisted which occur in subgraph with a precondition of the final state in the obtained in the above step.

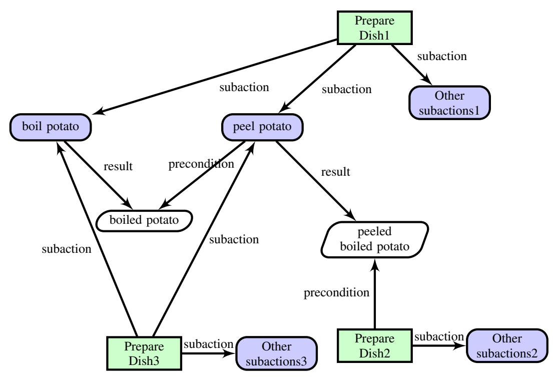
Algorithm 3: Algorithm shows the steps to post process the recipe domain ontology.

Table 1: Table shows the number of questions answered by both the system categorized by the type of answer required. The numbers in red indicate the strenghth of an ontology

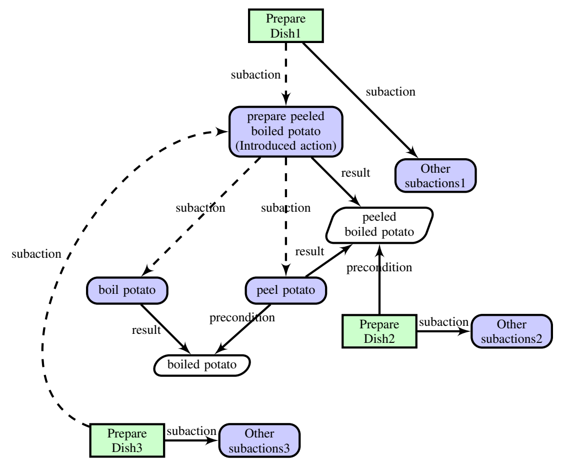
Category	**DB**	**OWL**
Inference based answering	8	14
Information based answering	6	2
Traversal based answering	5	4
N.A.	1	0
Total	20	20

5 Summary

In this paper, we have explained the various data preparation techniques used in this work. A detailed explaination of the MMTS and recipe domain that follow with the structural changes and entity, action and relations is introduced. The methodology of creation was totally different for the two domains emphasizing on the point that a generic framework working for all domains and all resource types cannot be prepared. A small application for QA is also developed and comparison gives Ontology a clear upper hand over RDBMS system.



(a) Figure shows the ontological structure of the recipe domain with example of 3 dishes.



(b) Figure shows the change in the ontological structure after the grouping of identical actions. *boil potato* and *peel potato* form a cluster and *prepare boiled peeled potato* takes its place wherever possible. They become a subaction of the newly introduced action.

Figure 3: Figure shows the how the most frequently occuring action is combined to give a new information keeping the original information intact. Solid green boxes represent entities, rounded blue boxes represent actions and white trapeziums represent states. Dashed lines represent inferd infoormation.

References

Fadi Badra, Rokia Bendaoud, Rim Bentebibel, Pierre-Antoine Champin, Julien Cojan, Amélie Cordier, Sylvie Després, Stéphanie Jean-Daubias, Jean Lieber, Thomas Meilender, et al. 2008. Taaable: Text mining, ontology engineering, and hierarchical classification for textual case-based cooking. In *9th European Conference on Case-Based Reasoning-ECCBR 2008, Workshop Proceedings*.

C Chang. 1988. Development of a train-scheduling expert system for ac railway electrification. In *Control, 1988. CONTROL 88., International Conference on*, pages 259–264. IET.

Juan DeMiguel, Laura Plaza, and Belén Díaz-Agudo. 2008. Colibricook: A cbr system for ontology-based recipe retrieval and adaptation. In *ECCBR Workshops*, pages 199–208.

Xiao-Hui Hu, Xing-she Zhou, and Jian-Wu Dang. 2006. A designing method of simulation software for chinese train control system based on hybrid software agent model. In *Machine Learning and Cybernetics, 2006 International Conference on*, pages 148–153. IEEE.

Güler Kalem and Çiğdem Turhan. 2005. *Semantic web application: Ontology-driven recipe querying*. Atılım University.

Nanda Kambhatla, Sachindra Joshi, Ganesh Ramakrishnan, Kiran Kate, and Priyanka Agrawal, editors. 2012. *Proceedings of the Workshop on Question Answering for Complex Domains*. The COLING 2012 Organizing Committee, Mumbai, India, December.

P. Kiran Mayee, R Sangal, and S Paul. 2010a. Automatic extraction and incorporation of purpose data into purposenet. In *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on*, volume 6, pages V6–154. IEEE.

P. Kiran Mayee, Rajeev Sangal, and Soma Paul. 2010b. Neural networks based detection of purpose data in text. In *ICT*, pages 606–609.

Eric PrudHommeaux, Andy Seaborne, et al. 2008. Sparql query language for rdf. *W3C recommendation*, 15.

Kavitha Rajan. 2013. Understanding verbs based on overlapping verbs senses. *ACL 2013*, page 59.

Sruti Rallapalli and Soma Paul. 2012. Evaluating scope for interpreting nominal compounds using ontology.

Rajeev Sangal, Soma Paul, and P Kiran Mayee. 2013. Purposenet: A knowledge base organized around purpose. In *Conceptual Structures for STEM Research and Education*, pages 29–30. Springer.

Chandni Singh and Rishabh Srivastava. 2011a. *PurposeNet: Knowledge Representation and Extraction*. PDPM-Indian Institute of Information Technology Design and Manufacturing Jabalpur.

Chandni Singh and Rishabh Srivastava. 2011b. Study and population of artifacts. *International Journal of Computer Technology and Electronics Engineering*, (NCETCSIT-Dec'2011):1–5.