Integrating declarative knowledge : Issues, Algorithms and Future Work

Doo Soon Kim and Bruce Porter
The University of Texas at Austin
Computer Science Department
{onue5, porter}@cs.utexas.edu

Abstract

As information technology progresses, computers of these days can access more information than in the past, and thereby the need to integrate many pieces of knowledge increases. This paper discusses the integration of declarative knowledge. More specifically, it discusses the construction of a coherent knowledge representation out of small pieces of knowledge using a background KB. The paper will cover several challenging issues arising in knowledge integration, our approach and future work.

One of the goals of AI is to build a comprehensive knowledge base incrementally by acquiring fragments of knowledge from multiple sources, probably by reading texts(fig. 1). The key facility in a system - apart from natural language processing, which is beyond the scope of this paper - is knowledge integration: the ability to combine fragments of knowledge into a coherent whole. Although some recent systems (Harrington & Clark 2007) (Yatskevich, Welty, & Murdock 2006) (Noy 2004) provided aspects of this facility, their main focus was on identifying co-references among the snippets of knowledge. In contrast, in this paper, KI entails considerably more than coreference identification. A KI facility, such as the one we are developing, would have application for a variety of AI tasks, such as NLP, Knowledge Actuisition, and computational models of human learning.

Challenges in KI

Fig. 2 shows an example of KI which combines two pieces of information about *Blood Circulation*. Background knowledge, in this case, about the concept, PUMPING, is used to improve the coherence of the combination. With the example, this section presents challenging issues of KI which received little attention in AI.

- **Maximizing coherence** Typically, multiple snippets of knowledge can be combined in many ways. A challenge in knowledge integration is finding the combination that maximizes coherence. While researchers have identified some of the factors that influence coherence, a complete theory has not emerged.

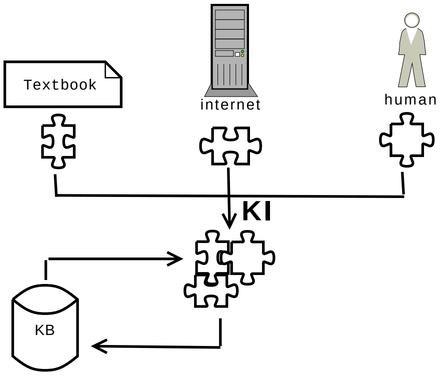


Figure 1: Integration of small pieces of knowledge from different sources. The figure illustrates a symbiotic relationship between a knowledge base (KB) and knowledge integration (KI) in which KB assists KI and KI enriches KB.

- **Semantic matching** Integrating multiple knowledge representations requires aligning them. The alignement process involves identifying semantically matched parts among the representations. This can be challenging because same information can be represented in different ways. For example, “Blood moves from the heart to the lung” and “A lung receives the blood from the heart” express the same meaning, though their formal representations would be different.
- **Implicit information** Humans are so good at filling in unspecified gaps in several pieces of knowledge that are related. One of the challenges to KI is to identify such unspecified information. For example, fig. 2 shows a causal relation between MOVement of BLOOD in source1 and PUMPING of HEART in source2 (dashed line).
- **Granularity difference** A special case of previous two challenges is that representations could be different in the level of detail. For example, source2 uses HEART and LUNG as an origin and a destination of MOVEment of BLOOD, whereas background knowledge, PUMPING, encodes that FLUID moves from CHAMBER to CONDUIT (shaded parts of fig. 2). KI has to identify such granularity differences and combine the representations based on

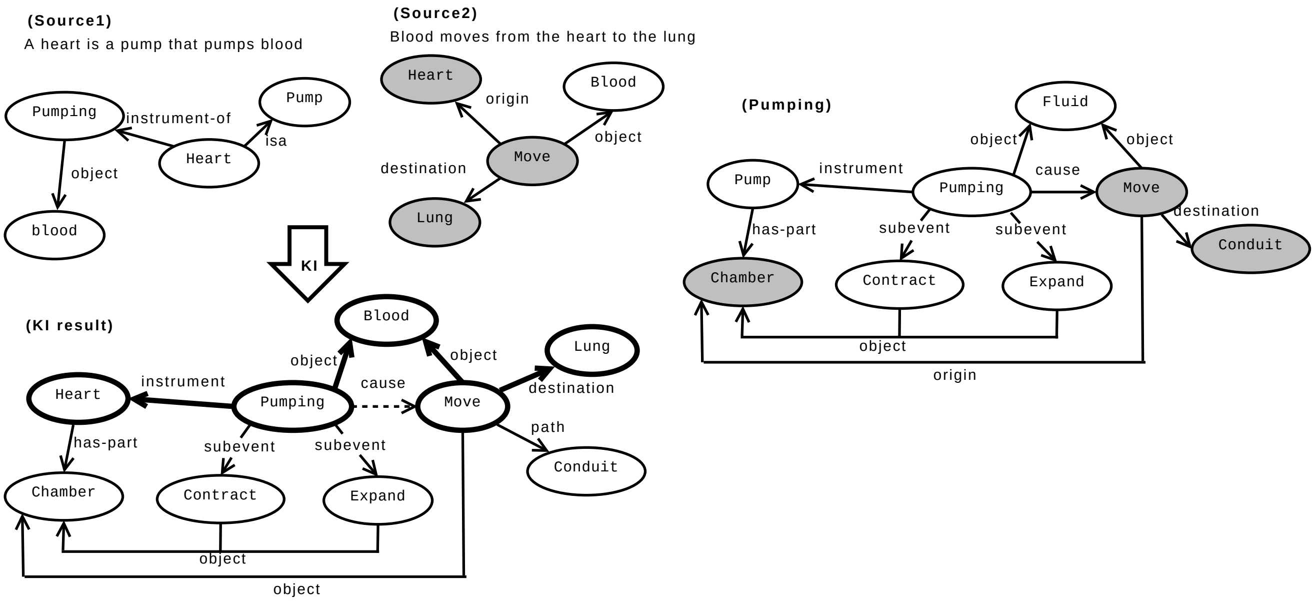


Figure 2: Integration of two logical forms about *Blood circulation*. Background knowledge, PUMPING, improves the coherence of the integration by revealing a causal relation between the input representations (dashed line). The bold lines in the resulting knowledge structure signify the representations in source1 and source2.

the identification. (Kim & Porter 2007) identified several types of granularity mismatches commonly arising in knowledge integration.

Because of these challenges, simple graph-matching (Sowa 1984) is insufficient.

Our approach and algorithm

This section introduces an algorithm, KI (Algorithm1), to handle the challenges in Section 2. The algorithm consists of two steps: *stitch* and *elaborate*. In the stitch phase, two representations are combined using GRAPH-MATCH (Algorithm3), and in the elaboration phase, the combined result is further elaborated by a knowledge base (KB). The elaboration phase also uses GRAPH-MATCH. GRAPH-MATCH extends simple graph-matching (Algorithm2) in the following ways:

- **Demons** In simple graph-matching (Algorithm2), two triples are aligned if one subsumes the other. For example, (Animal has-part Limb) can be aligned with (Human has-part Leg) in that way. GRAPH-MATCH extends this alignment by considering not only subsumption relations but other types of alignment, such as resolution of granularity differences. More specifically, GRAPH-MATCH runs multiple demons, each of which looks for a specific pattern and combines the two representations if the pattern is found. If more than one demon is applicable, the algorithm branches to create all possible alignments. The next section describes the use of demons in more detail.
- **Use of beam** Reasoning in KI is uncertain, so the graph match algorithm maintains a beam of multiple states, each of which corresponds to a different way of integrating the representations. Because the beam size is limited, the algorithm requires an intelligent way of selecting beam elements which will be retained. Devising such a selection method is one of our future goals.

Algorithm 1 (KI) Top-level algorithm for KI. It first relates two input representations and then elaborates further the related representations using a background KB.

```
Input : $G_1$, $G_2$ : input graphs
        $KB$, a background KB
Output : combined knowledge representation
Algorithm
result1 ← Use GRAPH-MATCH to combine $G_1$ and $G_2$ (stitch phase)
result2 ← Use GRAPH-MATCH to combine result1 and a knowledge structure $G_{KB}$ in $KB$ (elaboration phase)
return result2
```

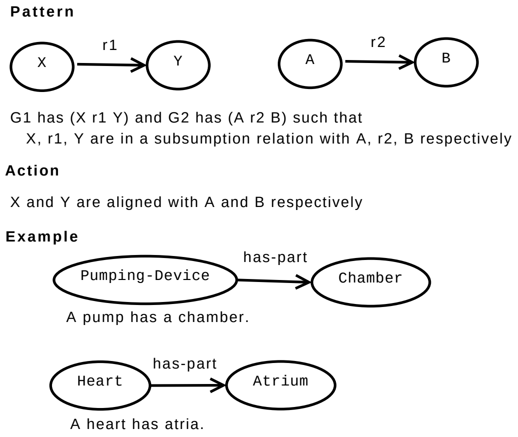
Demons

The demons introduced in this section are mainly aimed at handling granularity differences introduced in (Kim & Porter 2007). But, the use of demons can apply for other types of alignment in a framework of GRAPH-MATCH as well. We will explain each demon with a specific pattern that it looks for and an action that it executes. In this section, the Component Library (Barker, Porter, & Clark 2001) is used as our ontology.

Subsumption-based alignment This demon aligns two triples in a subsumption relation. This is a pattern used in simple graph-matching.

Algorithm 2 simple graph-matching (Variable names are shown in *a slanted style*)

```
Input : $G_1, G_2$ : input graphs
Output : combined knowledge representation
Algorithm
node-pair-queue ← initial starting nodes from G1 and G2 that are aligned with each other
while node-pair-queue is not empty do
  (c1, c2) ← the top element in node-pair-queue
  for all triples, t1, adjacent to c1 in G1 do
    find a triple in G2, t2, that t1 subsumes (or vice verca), and then add (t1, t2) to mappings
    add to node-pair-queue a new node mapping from t1 and t2
  end for
end while
Join $G_1$ and $G_2$ based on mappings
```



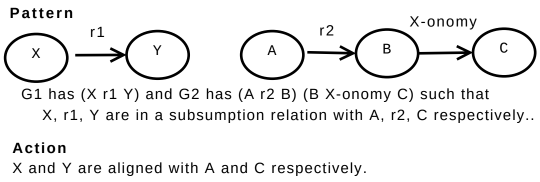
G1 has (X r1 Y) and G2 has (A r2 B) such that X, r1, Y are in a subsumption relation with A, r2, B respectively

X and Y are aligned with A and B respectively

A pump has a chamber.

A heart has atria.

X-onomy based alignment This pattern generalizes the one used in subsumption-based alignment. During graph matching, the algorithm determines co-references among the entities and events mentioned in the graph. This can be challenging because the entities and events can differ in level of granularity. The algorithm uses a demon to handle this case. The pattern of this demon uses a relation, X-onomy. X-onomy is a general relation that includes all relations that involve hierarchy, such as has-part(partonomy), enclose, isa(taxonomy), etc.

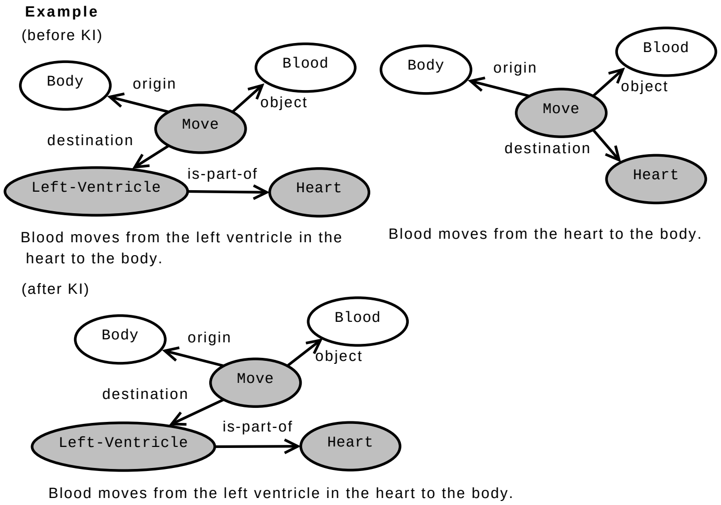


G1 has (X r1 Y) and G2 has (A r2 B) (B X-onomy C) such that X, r1, Y are in a subsumption relation with A, r2, C respectively..

X and Y are aligned with A and C respectively.

Algorithm 3 (GRAPH-MATCH) This algorithm extends Algorithm2. *Italic parts* are new steps added to Algorithm2.

```
Input : $G_1, G_2$ : input graphs
Output : combined knowledge representation
Internal Data Structure
  State ::= {node-pair-queue}
Algorithm
node-pair-queue ← initial starting nodes from G1 and G2 that are aligned with each other
create state $S_1$ with node-pair-queue
add S1 to beam
for all states S in beam whose node-pair-queue is not empty do
  (c1, c2) ← the top element in node-pair-queue of S
  for all triples, t1, adjacent to c1 in G1 do
    Each demon aligns t1 with G2 in its own way and then updates node-pair-queue.
    Create a new state, $S_{new}$, with the updated node-pair-queue and add $S_{new}$ into beam
  end for
  If beam contains more states than the threshold, select the top N states
end for
Join $G_1$ and $G_2$ based on the mappings of the best state in beam
```



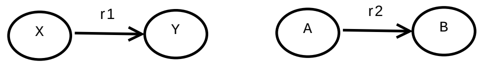
Blood moves from the left ventricle in the heart to the body.

Blood moves from the heart to the body.

Blood moves from the left ventricle in the heart to the body.

The above demon can be transformed to another one that can align two triples whose heads and relations are taxonomically aligned with each other but tails are not aligned. This is a pattern in which (B X-onomy C) is ablated from X-onomy based demon. On detecting such pattern, this new demon aligns the partially-aligned triples by installing a X-onomy relation between the tails.

Pattern



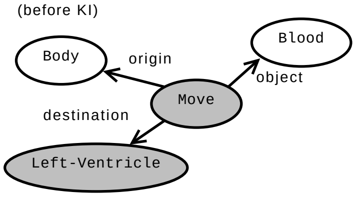
G1 has (X r1 Y) and G2 has (A r2 B) such that
X and r1 are in a subsumption relation with A, r2 respectively
but, Y is not with B

Action

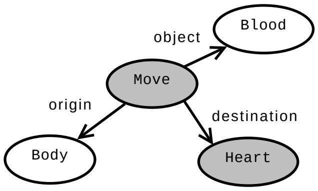
1) X is aligned with A
2) Add a new triple (Y X-onomy B) in the combined representation

Example

(before KI)

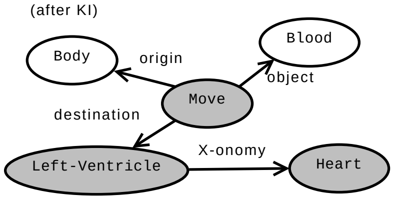


Blood moves from the left ventricle to the body.



Blood moves from the heart to the body.

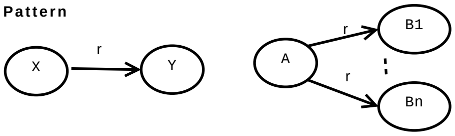
(after KI)



Blood moves from the left ventricle to the body
The left ventricle is in X-onomy relation with the heart.

Replication-based alignment Another common case in granularity mismatch is when one representation contains many repetitions of a triple in the other.

Pattern



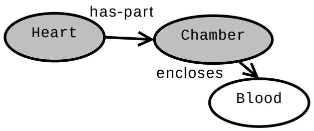
G1 has (X r1 Y) and G2 has n repetition of (A r Bx) such that
X and Y are in a subsumption relation with A and Bx respectively.

Action

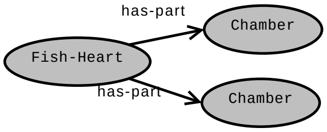
Replicate the relevant parts in G1 n times, and X, Y1, ..., Yn are mapped with A, B1, ..., Bn.

Example

(before KI)

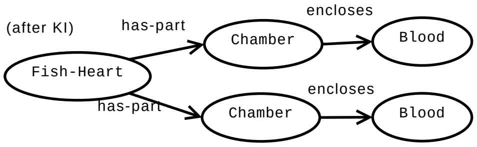


A heart has chambers filled with blood.



A fish heart has two chambers.

(after KI)

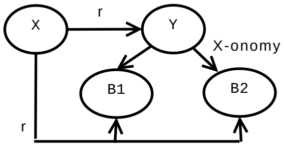


A fish heart has two chambers filled with blood

Generalization-based alignment In this case, several pieces of similar information can be generalized. For example, "A heart has a left atrium" and "A heart has a right atrium" can be generalized into "A heart has atria".

Pattern

(Template)



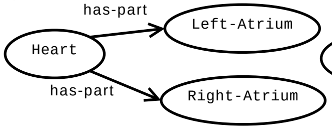
G1 and G2 has the subgraphs that can be aligned with the above template

Action

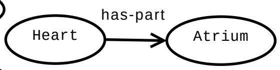
Align G1 with G2 through the template

Example

(before KI)

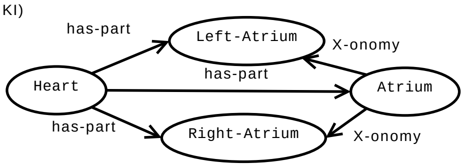


A heart has a left atrium and a right atrium.



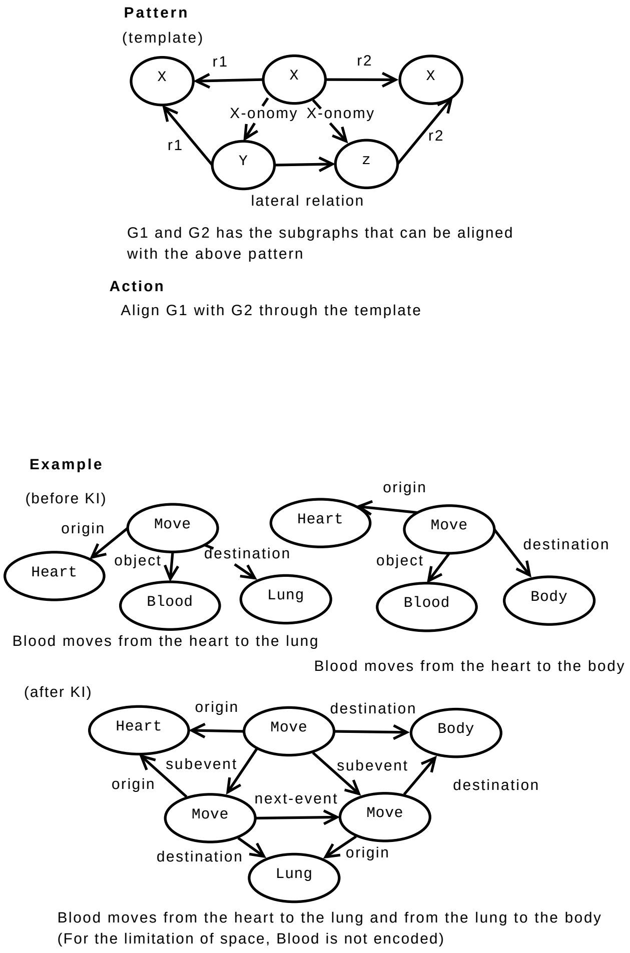
A heart has atria.

(after KI)



A heart has a left atrium and a right atrium. These two atria have X-onomy relation with Atirum

Abstraction-based alignment In this case, two things that are consecutive temporally or spatially can be viewed as one thing. The pattern of this demon uses a relation, "lateral relation", which connects concepts in the same level of detail (e.g. next-event, beside, etc.).



G1 and G2 has the subgraphs that can be aligned with the above pattern

Align G1 with G2 through the template

Blood moves from the heart to the lung

Blood moves from the heart to the body

Blood moves from the heart to the lung and from the lung to the body
(For the limitation of space, Blood is not encoded)

Note that if only **Subsumption-based alignment** is used, the algorithm3 becomes simple graph-matching (algorithm2).

Application to Learning-by-Reading

The goal of Learning-by-Reading project (Barker *et al.* 2007) is to develop a system that reads a scientific text and generates a formal representation of it. The role of KI in the system is to produce a text-level representation by combining together logical forms of individual sentences generated by natural language software. We plan an experiment to measure the contribution of KI to improving the coherence and cohesiveness of the final representation. In particular, connectivity will be measured among the logical forms, since the much connected representations are typically coherent and cohesive (Zadrozny & Jensen 1991). The connectivity can be calculated by the ratio of the associated parts of the logical forms to the sum of the sizes of them. Since connectivity alone is not enough, the correctness of integration made by the algorithms will be evaluated by human judges.

Related Work

KI has been a key component in learning, knowledge acquisition and natural language systems, although it has not been named as KI in many of these systems (Harrington & Clark 2007) (Yatskevich, Welty, & Murdock 2006) (Noy 2004) (Cullingford 1978).

One closely related area attempts to build a large-scale common-sense KB by reading texts in the Internet (Harrington & Clark 2007) (Yatskevich, Welty, & Murdock 2006). These systems typically focus on identifying corefered entities. Ontology merging (Noy 2004) is similar to KI in that both aim at aligning two different representations. However, ontology merging often takes input representations written in different KR languages, whereas KI in this paper assumes the same underlying language.

References

Barker, K.; Agashe, B.; Chaw, S. Y.; Fan, J.; Glass, M.; Hobbs, J.; Hovy, E.; Israel, D.; Kim, D. S.; Mulkar, R.; Patwardhan, S.; Porter, B.; Tecuci, D.; and Yeh, P. 2007. Learning by reading: A prototype system, performance baseline and lessons learned. In *Proceedings of Twenty-Second National Conference on Artificial Intelligence*.

Barker, K.; Porter, B.; and Clark, P. 2001. A library of generic concepts for composing knowledge bases. In *Proceedings of the international conference on Knowledge capture*, 14–21. ACM Press.

Cullingford, R. 1978. Script application: Computer understanding of newspaper stories. Technical Report 116, Yale University Department of Computer Science, New Haven, CT.

Harrington, B., and Clark, S. 2007. Asknet: Automated semantic knowledge network. In *Proceedings of Twenty-Second National Conference on Artificial Intelligence*.

Kim, D. S., and Porter, B. 2007. Handling granularity differences in knowledge integration. In *Proceedings of the Fall AAAI Symposium Series*.

Noy, N. F. 2004. Semantic integration: A survey of ontology-based approaches. *SIGMOD Record* 33(4):65–70.

Sowa, J. 1984. *Conceptual Structures*. Addison-Wesley.

Yatskevich, M.; Welty, C.; and Murdock, J. W. 2006. Coreference resolution on rdf graphs generated from information extraction: first results. In *Proceedings of the ISWC 06 Workshop on Web Content Mining with Human Language Technologies*.

Zadrozny, W., and Jensen, K. 1991. Semantics of paragraphs. *Computational Linguistics* 17(2):171–210.