Towards a Domain-independent Method for Evaluating and Scoring Analogical Inferences

Joseph A Blass, Irina Rabkina, & Kenneth D. Forbus

Northwestern University, Evanston, IL, USA
{joeblass, irabkina}@u.northwestern.edu
forbus@northwestern.edu

Abstract. This paper proposes a domain-independent method to evaluate inferences for analogical reasoning, via a prototype system. The system assigns analogical inferences confidences based on the quality of the mapping and the system's confidence in the facts used to generate the inference. An initial implementation is applied to two domains.

Keywords: Analogical Reasoning, Inference Evaluation, Confidence.

1 Introduction

Any reasoning system which asserts facts through the processing and manipulation of previously known information ought to have a measure of confidence in the accuracy of those newly asserted facts. Even if a given reasoning technique is sound, inferred facts are only as accurate as the assumptions upon which they are based. For example, systems that reason via formal logic produce reliable inferences, but if the reasoning environment is complex enough, or a particular axiom is missing, contradictions may pass undetected. Furthermore, forward chaining systems overgenerate inferences, while backchaining systems are directed, but require a known goal for reasoning. On the other hand, probabilistic systems such as Bayes Nets [1] are good at determining how likely a particular inference is, but require a lot of training data or carefully hand-tuned priors. Analogy is a case-based reasoning technique that constructs an alignment between two cases, with a preference for shared structure, and uses that structure to make inferences from one case to another [2]. Inspired by human cognition, analogical reasoning does not require a fully articulated domain theory and can work from single examples and partial information. However, the inferences made by an analogical reasoning system may not be correct, and while there are evaluation measures based on the structure of the mapping and candidate inferences, all of the methods used in previous systems have been domain and/or task specific.

This paper proposes a unified approach to evaluating and scoring analogical inferences. It integrates logical reasoning, analogical reasoning, and probabilistic reasoning to provide confidence estimates for analogical inferences. We present an initial implementation and some experimental results as a proof of concept of these ideas.

1.1 SME, SAGE, and Cyc

The principles underlying our system are domain general. Our implementation uses the Structure-mapping Engine (SME, [3]) and a supplemented Cyc knowledge base [4]. What is important about the Cyc ontology for the present paper is that it provides *microtheories*. Microtheories serve as contexts, e.g. one microtheory might describe modern-day Chicago, while another describes Chicago as it was during the Fire. Microtheories can inherit from each other, e.g. when performing social reasoning, a common microtheory to include is HumanActivitiesMt, which as its name suggests, describes things people commonly do. Microtheories enable locally consistent reasoning, even though the knowledge base (KB) taken as a whole is inconsistent, e.g. there are microtheories describing different, incompatible fictional worlds. For analogical reasoning, we implement cases as microtheories, which enables reasoning to be done with respect to different cases locally. All reasoning is done with respect to a *context*, that is, a microtheory and all of the microtheories it inherits from.

SME [3] is a computational model of analogy that computes mappings between two structured cases, a base and a target. Each mapping includes correspondences between elements in the two cases, candidate inferences based on those correspondences, and a structural evaluation score calculated based on the structural similarity between the two cases. The higher the score, the more similar the cases and the more trusted the mapping. The Sequential Analogical Generalization Engine (SAGE [5]) uses SME mappings to create generalizations between cases. These generalizations can then be used as cases for further SME comparisons. Rather than keep only facts common to all generalized cases, SAGE generalizations are a joint distribution over the facts in all constituent cases. Each fact is stored in the generalization together with its probability, that is, the proportion of cases in that generalization that contains it. Only facts whose probability falls below a preset threshold are not included in the generalization. This scheme allows the generalization to maintain information about which facts are likely, not only which are universal. For example, consider a generalization composed of three cases that describe dogs: a Golden Retriever, a yellow Labrador, and a Dalmatian. The generalization will have the fact that a dog has 4 legs with probability 1.0 and the fact that it has yellow fur with a probability of 0.67. The inference evaluation system makes use of these probabilities, along with the structural evaluation score.

2 Inference Evaluation

When the system reasons its way to a new fact in a context, it can either be certain it is true, certain it is false, or somewhere in between. The system uses disjointness reasoning, logical contradiction and implication, and the parameters of SME mappings to determine the system's confidence that an inference is true. All reasoning is done with respect to the context in which the inference is to be asserted.

2.1 Disjointness Reasoning, Contradiction, and Implication

If the system has inferred that an entity is of a certain type, and there is already a contextualized assertion that it is of another type that is by definition disjoint from the first, the system simply rejects that inference. For example, if Fluffy is a dog, it cannot assert that it is a cat unless it first retracts that it is a dog. In the Cyc knowledge base, certain collections are marked as disjoint collection types, such that if an entity is an instance of one of those types, it cannot be an instance of another. When our system detects that an inference is of the form `(in-Context ?context (isa ?entity ?newType))`, it gathers all the other declarations of that entity's type in context `?context`. If any of those other types are disjoint with `?newType`, then the system rejects the inference. Inferences can also be rejected if they are contradicted by known implication rules. If there is a rule of the form `A -> ~I`, where `I` is the analogical inference, and `A` is known to be true, the inference can be rejected. Similarly, if there is a rule of the form `I -> A`, and `A` is explicitly known to be false, the inference can be rejected.

Implication is similar: If there is a rule of the form `A -> I`, and `A` is true, the inference has been confirmed. Similarly, if there is the rule `~A -> I`, and `A` is known to be false. The confidence in the implied fact is a function of the confidence assigned to the facts used to imply it. Contrapositives of the rules for implication and contradiction are generated on-the-fly. We do not assume rules are sufficiently complete to generate all inferences generated by analogy. Even if they were, analogy would be useful for focusing logical reasoning. The system makes use of forward chaining in a targeted fashion, only for verification, which is more efficient than simply forward-chaining.

2.2 Inferences from Analogical Reasoning

When the system derives an inference using an analogical mapping, it may be able to directly prove or disprove it. Failing that, it is desirable to have a measure of the extent to which the inference is trusted. The normalized SME match score is one such signal. Another is the degree to which the facts the inference is based on (in the base and target cases) are trusted. If the base case is a SAGE generalization, then the fact probability in the generalization tells us how likely that fact is within that generalization. For a non-generalized case, the system does not know the extent to which the case itself is an outlier or whether any one fact in the case is core to the overall concept that the case encodes. Inferences from individual cases should be trusted less than high-probability generalization facts, since there is evidence from the generalization that the high-probability facts are more common.

Putting it all together, analogical inferences are assigned confidence scores thus:

$$\mathrm{P(Inference)} = MatchScore * BaseTrust * \prod_{t\,\epsilon\,target\ support} P(t)$$

The BaseTrust is as described above: If the base case is a SAGE generalization then:

$$\mathrm{BaseTrust} = \prod_{b\,\epsilon\,base\ support} P(b)$$

And otherwise it is set to the default normalizing value (currently, 0.7).

Given this formula, confidence scores are always on the interval (0, 1): normalized match scores are on that interval, and fact confidence scores are on the interval (0, 1]. Confidence cannot be zero since zero-confidence facts are simply suppressed, rather than asserted with confidence zero. Normalized match scores are a measure of the *degree* of overlap between cases, rather than the total amount of information being mapped from one case to another. These can be low but are never zero: for a mapping to be generated at all, there must be some degree of overlap.

We use a product, rather than, say, a sum, of these components to keep confidence on the (0,1) interval. Intuitively it makes sense that a fact inferred from many facts should be trusted less than one inferred from only a few (if we are equally uncertain of the supporting facts). The more facts used to support an inference, the greater the chance that one of them is false and that the inference is therefore invalid. If the confidence scores were allowed to be greater than one, then the confidence of inferences might become greater as we moved further out along inference chains.

The system uses a Truth Maintenance System, which has a single argument to mark a belief as in or out. This renders combining evidence from multiple arguments moot.

2.3 Implementation

In the current implementation, facts that are assumed (for example, the details of the case that is to be reasoned about) have a confidence of 1. Our inference evaluator first tries to determine whether an inference is contradicted or implied; if it fails, it checks whether that inference is from analogy and scores it appropriately, and otherwise, assigns it the default normalizing score. Contradiction and implication are handled using backward chaining from axioms in the knowledge base, using resource bounds.

In our implementation, all inferences are given a confidence score and a reason for that score. The reason is the facts and axioms that were used to generate the score. For implied facts, the score is the product of the confidences of the facts that imply it (because perhaps those antecedents are not trustworthy). Contradicted facts are currently simply rejected, although in future implementations they will be scored based on the likelihood of the facts used to reject them. Confidence scoring for analogical inferences is described above. SME mapping scores can be normalized in three different ways, all of which are on in the interval [0:1]. The base normalized score is a measure of how much of the base case is mapped in the mapping, that is, how much of the base case overlaps with the target case. If the target is much larger than the base but the base is highly alignable with a sub-set of the target, the base normalized score will be quite high even if the match score is low. The target normalized score is the corresponding measure for how much of the target case is mapped, and the normalized score is the average of the base and target normalization scores. The default is the average normal score. Base normalization tends to be used in recognition tasks, where covering the entire base is the criteria, whereas target normalization tends to be used in reasoning tasks, where finding precedents that can lead to inferences within a more complex situation is important.

3 Evaluation

We tested the confidence scoring and contradiction components of this initial implementation on two tasks: Analogical Chaining and Moral Reasoning. Analogical Chaining is a commonsense reasoning technique that elaborates a case description by repeated analogy to small cases called Common Sense Units (CSUs) [6]. These CSUs can be extracted automatically from natural language, and are thus easy to provide to the reasoning system. As analogical chaining uses analogical reasoning, it does not require a fully articulated domain theory or rules constructed by experts, can reason with partial knowledge, and can use the same case for prediction or explanation. Analogical chaining has been tested on questions from the Choice of Plausible Alternatives commonsense reasoning test (COPA, [7]). As analogical chaining asserts inferences by analogy, then asserts new inferences building on those previous inferences, it is very valuable to give it a measure of confidence in those inferences.

We examined the performance of the inference evaluation system on 11 COPA questions, whose internal representations were automatically extracted from the English text of the question using EA-NLU [8]. These questions were selected because they require repeated analogical inference (i.e., chaining) to solve. The system had a case library of around 50 cases it could retrieve and reason with. For every question tested, the confidence scores assigned to inferences were lower the further down they were along the inference chain; this means the inferences that enabled the system to answer the questions had lower confidence scores than the intermediate inferences used to infer them, reflecting the system's lower confidence the further out it went from established facts. Inference scores ranged from 0.02 (for an inference made only using facts from the COPA question itself) down to 1×10^{-6} (for an inference several steps removed from the question facts). All but three questions did not involve any dead-end reasoning: analogical chaining found the correct answer for those questions without exploring any fruitless inference chains. We will examine two cases that involved dead-end reasoning in detail.

One question asks: "The egg splattered. What was the cause of this?" The answers are "I dropped it" and "I boiled it." The system first hypothesized that the egg splattering was caused by some unknown violent impact, and assigned that inference a confidence score of 0.01 (low inference scores are discussed below). It then hypothesized, as an alternative explanation for the egg splattering, that the egg hit the floor. This did not involve the abstract impact from the first inference, but was based only on the question facts. However, the mapping had a lower match score than the first, so it was given a confidence of 0.0004. The system then pursued, in separate reasoning contexts, explanations for the first two inferences. In the system's case library was a case describing how an object was violently impacted when it was hit with a rock, so it hypothesized that perhaps the unknown impact on the egg was caused by a rock. Despite being based on a higher confidence inference (the first inference asserted, where p=0.01), this inference had a low match score and therefore resulted in a score of 2×10^{-6}. Finally, the system used a fourth case to explain the inference that the egg hit the floor by hypothesizing that it was dropped. Despite being based on a lower-confidence fact than the inference about the rock, this inference had a higher match score and thus received a confidence of 2×10^{-5}. While low, this score is still an order of magnitude higher than the dead-end hypothesis about the rock based on more highly-trusted initial inference.

Another question asks: “The truck crashed into the motorcycle on the bridge. What happens as a result?” The answers are “The motorcyclist died” and “The bridge collapsed”. The automatically constructed question representations involve only one statement about a motorcycle (and no motorcyclist) but several statements about the crashing event (who was involved, where it happened, etc.). The system retrieves cases based on what is present in the case, so it began by reasoning about a familiar case involving a vehicle crashing. In that case the vehicle was an airplane, so the system first posited that perhaps the crash in question involved an airplane malfunctioning ($p= 8\times10^{-5}$). The system then retrieved a story about a child falling out of bed and crashing onto the floor. It used this case to posit that the crash was caused when the truck fell out of bed ($p = 9\times10^{-5}$). Building on the airplane inference, it hypothesized that the airplane lost power ($p=1\times10^{-6}$), then that the motorcycle lost power, (0.012), and finally, having exhausted its knowledge of crashes, that the motorcyclist dies ($p=2\times10^{-5}$). In this case the correct inference has a lower confidence score than all but one of the dead-end inferences.

This example illustrates the pitfalls both of Analogical Chaining and of the inference evaluation system. After the system had posited the airplane, it was all too happy to continue reasoning about it, and the match scores were high enough along that reasoning chain (and low enough for the case that gave it the answer) that those erroneous inferences were scored much higher than the one that seems obvious to humans (humans of course have much prior knowledge the system lacks). The system can be led astray and mask the utility of useful inferences if it marks even one incorrect inference as highly probable. Furthermore, it seems wrong to give the system a hard-and-fast rule stating that airplanes cannot be involved in car crashes. Such a situation may be extremely unlikely, which could be recognized by accumulating cases in a generalization about car and motorcycle crashes, but, as Hollywood has shown us, it’s not impossible. This raises an important point about the interplay between analogical reasoning and first-principles reasoning. Analogical learning can provide explicit evidence of what can happen, because analogical generalizations provide structured, relational probabilistic representations of what has happened. But analogical learning only implicitly gathers evidence about what cannot happen. First principles reasoning is better at ruling out the kinds of things that are impossible (e.g. vehicles cannot fall out of beds because they cannot fit in them).

MoralDM is a computational moral reasoning system that makes decisions by analogy to moral generalizations [9,10]. In one experiment, generalizations are formed from cases that either involve the principle of double effect [11], or do not. This principle states that harm caused as a side effect of preventing a greater harm is morally acceptable, but not harm caused in order to prevent that greater harm. The canonical example illustrating this principle is that most people say it is morally acceptable to switch a trolley that will hit five people onto a side track where it will hit one person (double effect), but not to instead push someone in front of the trolley to save those same five people (not double effect). In these moral generalizations, the facts indicating whether the case involves double effect and which case-specific action should be taken have probability 1, whereas the facts specific to the case (whether it is a trolley or torpedo doing the harm, for example, how many people are hurt, or what the mechanisms are to save those people) have lower probabilities. We took the inferences made in reasoning about moral cases by analogy to these generalizations and checked them with the inference evaluator. This was both to test the generalization normalizing component and to get a sense of whether even highly trusted inferences have low scores (the mappings that generate these inferences have high unnormalized scores). While the scores for the high-confidence facts are still quite low (in all cases approximately 0.02), the scores for the low-confidence facts are much lower, corresponding to the lower proportion of constituent cases in which they appear. In the same mapping where the decision fact was scored at 0.02, for example, the fact about the form the harm took was scored at 0.005. This example demonstrates the utility of taking generalization fact probability into account: had these inferences been made by analogy to an ungeneralized case, the inference evaluator would have given them both scores of 0.014. Using generalization probabilities gives the system a means to assess different inferences from the same mapping.

4 Related Work

Most previous work on analogical inference validation has been domain specific. For example, Ouyang and Forbus used first principles reasoning within the physics problem solving domain to validate candidate inferences produced by SME [12]. While the validation improved their system's performance, a complete domain model had to be assumed. Similarly, Klenk and Forbus used a small set of hand-encoded heuristics to verify candidate inferences during transfer learning [13]. While these were not a complete domain model, the heuristics were specific to inferences that could be made in the test domain. While the system described in this paper allows for domain-specific verification (i.e. through implies statements), it is domain-general. Furthermore, unlike previous systems which rated an inference as true or false, the current system allows for intermediate rankings.

Similar intermediate rankings have been used to evaluate inferences derived by non-analogical reasoning systems. Examples include fuzzy logic networks [14], Bayesian Logic models (BLOG, [15]), and Markov Logic Networks (MLNs, [16]). By assigning a fuzzy truth space to antecedents, fuzzy logic networks are able to derive fuzzy truth values for inferred consequents. They allow for incomplete domain knowledge, but do require a handwritten set of rules. Fuzzy rules can be used in combination with data sampled in a particular space to rule in or out inferences made within that space ([17]). Fuzzy logic networks assign qualitative truth values (e.g. "mostly true") to inferences, rather than calculating a quantitative confidence measure.

BLOG models and MLNs calculate numerical probabilities for inferences. BLOG models do so by defining a probability distribution over a set of possible worlds determined by prewritten axioms. A Metropolis-Hastings Markov chain Monte Carlo approach can then be used to make inferences from the distribution [18]. Using MCMC increases the time and computational cost of inference scoring in these models. MLNs take a different approach: they define a Markov Network over a set of first-order logic sentences and constants, such that a node exists for each grounding of each predicate and a connecting feature exists for every possible grounding of each sentence [15]. Weights are assigned to these features based on the likelihoods of the sentences they describe. A probability distribution is then specified over the ground network. The structure of the network can be learned, given the sentences their possible groundings [19]. The disadvantage of MLNs is scaling: the network grows with additional predicates, as well as additional potential groundings. This also means that every potential grounding of every potential predicate must be present in the training set.

The presented inference evaluation technique could be used in other analogical reasoning systems that score (or could score) the quality of their matches (that is, which have a measure similar to SME's match score). For example, inferences in AMBR ([20]) have evidence accrued in favor and against them, based on semantic and structural similarity. Top scoring hypotheses are asserted into the reasoning environment, but the amount of evidence in favor of them is not. If this evidence were stored as a confidence measure of facts as they are asserted, future inferences could be made not only on the basis of evidence in favor or against them, but the degree to which that evidence is itself believed.

In HDTP (e.g. [21]), analogical mappings are constructed via a process of anti-unification. For example, a formula $p_1(a)$ in a base and $p_2(a)$ in a target is replaced in the mapping by a general predicate P(a), where P is a generalization of both predicates p_1 and p_2. A measure of similarity of P to p_1 and p_2 could be used to score inferences made using formula P(a); the scores of those inferences could then be used to score future inferences made using those inferred facts. In HDTP, inferences are checked for logical consistency; expanding logical consistency checks for inferences is the next extension to be performed on our system.

5 Future Work

Even mappings with high unnormalized match scores, indicating a high quality match, may have low normalized match scores, depending on the relative size of the cases and how much information is left out of the mapping. In the current implementation, low confidence scores assigned to analogical inferences were driven largely by low normalized mapping scores. Small cases with little structural overlap should yield low-confidence scores, since the mappings used to generate the inferences are not seen by the system as being particularly reliable, informative mappings (as indicated by the low score). However, while analogical inferences should have lower scores than logically implied inferences, they should not be vanishingly low. One possibility is to use the highest normalized score as a multiplier in calculating inference confidence scores, rather than always using the same mapping score normalizing function. Each function provides different information, but a high score in either indicates that the mapping includes a high degree of overlap from one case to another. Scoring inferences using the highest normalization score will still involve incorporating the score of the match, the score of the justifying target facts, and the probability of the generalization facts.

Given the ubiquity of certain role predicates (`objectActedOn`, `performedBy`, etc.) analogical chaining can make some inferences that, to a human, seem quite silly. Having the ability to rule out those silly inferences using logical forms of commonsense is desirable but is not being done in the current implementation. The Cyc knowledge base contains millions of axioms, but we are currently only using a small subset (the disjointWith axioms). We plan to explore reasoning techniques that enable us to exploit more of this knowledge, especially horn clauses and implication statements, for constraint-checking (e.g. [22]).

Contradictions should perhaps be asserted with a confidence proportional to the scores of the facts contradicting them, rather than suppressed entirely. If facts are seen as relatively likely, then the contradiction is also likely. If contradictions are asserted, they will must signal which facts contradict them, to keep reasoning consistent.

Many analogy inferences involve positing skolem entities. These are entities present in the base and participating in the candidate inference but which are not present in the target. For example, the event in which the egg was impacted in the above example was posited as a skolem variable. Fundamentally, however, these are open variables, and implication can help resolve them. Contradiction works in a similar way, but instead can only rule out resolutions: just because a rule says that a particular individual cannot fill a role does not mean that it says that no one can.

Finally, further testing is needed on a wider range of domains, as well as further empirical testing of the analogical inference confidence scoring. While we have verified the implication and contradiction through disjointness components of the inference evaluation system are functioning properly, these need to be empirically tested. We can thereafter examine accruing and weighting evidence for and against facts.

6 Conclusion

We presented an initial implementation of a system to evaluate analogical inferences, which have no guarantee of being correct. The system can identify certain inferences as being more likely than others, but further evaluation and extension of the system is needed. Nonetheless, this seems to be a promising direction for inference validation and assessment, and points towards a method for resolving skolem variables in analogical inferences.

7 Acknowledgements

This research was supported by the Socio-Cognitive Architectures for Adaptable Autonomous Systems Program of the Office of Naval Research, N00014-13-1-0470.

References

1. Pearl, J., Russell, S.: Bayesian Networks. UCLA Cognitive Systems Laboratory, Technical Report R-277, November 2000. In M.A. Arbib (Ed.), Handbook of Brain Theory and Neural Networks, Cambridge, MA: MIT Press, 157-160 (2003).
2. Gentner, D.: Structure-Mapping: A Theoretical Framework for Analogy. Cognitive Science, 7(2), 155-170. (1983).
3. Forbus, K. D., Ferguson, R. W., Lovett, A., Gentner, D: Extending SME to Handle Large-Scale Cognitive Modeling. Cognitive Science. (2016).
4. Lenat, D.: CYC: A large-scale investment in knowledge infrastructure. Comm. of ACM, 38(11), 33-38 (1995).
5. McLure, M. D., Friedman, S. E., Forbus, K. D.: Extending Analogical Generalization with Near-Misses. Procs. of the 29th AAAI Conf. on Artificial Intelligence, Austin, TX (2015).
6. Blass, J. A., Forbus K. D.: Analogical Chaining with Natural Language Instruction for Commonsense Reasoning. Procs. Of the 31st AAAI Conference on Artificial Intelligence. San Francisco, CA. pp. 4357-4363 (February, 2017).
7. Roemmele, M., Bejan, C. A., Gordon, A. S.: Choice of Plausible Alternatives: An Evaluation of Commonsense Causal Reasoning. AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning (2011, March).
8. Tomai, E., & Forbus, K. D.: EA NLU: Practical Language Understanding for Cognitive Modeling. In FLAIRS Conference. (2009, March)
9. Dehghani, M., Tomai, E., Forbus, K. D., Klenk, M.: An Integrated Reasoning Approach to Moral Decision-Making. Procs. Of the 31st AAAI Conference on Artificial Intelligence. pp. 1280-1286 (2008, July).
10. Blass, J. A., Forbus, K. D: Moral Decision-Making by Analogy: Generalizations vs. Exemplars. Procs. of the 29th AAAI Conference on Artificial Intelligence Austin, TX. (2015).
11. Foot, P.: The Problem of Abortion and the Doctrine of Double Effect. Oxford Review 5, pp. 5-15. (1967).
12. Ouyang, T. Y., & Forbus, K. D.: Strategy variations in analogical problem solving. Procs. of the 21st AAAI Conference on Artificial Intelligence. pp. 446-451. (2006, July).
13. Klenk, M., & Forbus, K.: Analogical model formulation for transfer learning in AP Physics. Artificial intelligence, 173(18), pp. 1615-1638. (2009).
14. Zadeh, L. A.: Fuzzy logic. Computer, 21(4), pp. 83-93. (1988).
15. Milch B, Marthi B, Russell S.: BLOG: Relational modeling with unknown objects. ICML Workshop on Stat. Rel. Learning and its Connections to Other Fields. pp. 67-73. (2004).
16. Richardson, M., Domingos, P.: Markov Logic Networks. Mac. Learn, 62, 107-136. (2006).
17. Ughetto, L., Dubois, D., Prade, H.: Implicative and conjunctive fuzzy rules-A tool for reasoning from knowledge and examples. In AAAI/IAAI, pp. 214-219. (1999, July).
18. Milch, B., Russell, S.: General-Purpose MCMC inference over relational structures. Procs. of the 22nd Conference on Uncertainty in Artificial Intelligence. pp. 349-358. (2006, July).
19. Kok, S., Domingos, P.: Learning the structure of Markov logic networks. In Proceedings of the 22nd International Conference on Machine Learning. pp. 441-448. (2005, August).
20. Kokinov, B., Petrov, A: Integrating memory and reasoning in analogy-making: The AMBR model. The analogical mind: Perspectives from cognitive science, pp. 59-124. (2001).
21. Schwering, A., Krumnack, U., Kühnberger, K., Gust, H.: Analogical reasoning with SMT and HDTP. 2nd European Cog Sci Conference, Delphi, Greece. pp. 652-657. (2007).
22. Sharma, A. B., Forbus, K. D.: Automatic Extraction of Efficient Axiom Sets from Large Knowledge Bases. Procs. of the 27th AAAI Conf. on Artificial Intelligence. (2013, June).