Combining Bayesian Networks and Formal Reasoning for Semantic Classification of Student Utterances

Maxim Makatchev [a,1] and Kurt VanLehn [b]
[a] *The Robotics Institute, Carnegie Mellon University*
[b] *Learning Research and Development Center, University of Pittsburgh*

Abstract. We describe a combination of a statistical and symbolic approaches for automated scoring of student utterances according to their semantic content. The proposed semantic classifier overcomes the limitations of *bag-of-words* methods by mapping natural language sentences into predicate representations and matching them against the automatically generated deductive closure of the domain givens, buggy assumptions and domain rules. With the goal to account for uncertainties in both symbolic representations of natural language sentences and logical relations between domain statements, this work extends the deterministic symbolic approach by augmenting the deductive closure graph structure with conditional probabilities, thus creating a Bayesian network. By deriving the structure of the network formally, instead of estimating it from data, we alleviate the problem of sparseness of training data. We compare the performance of the Bayesian network classifier with the deterministic graph matching-based classifiers and baselines.

Keywords. Dialogue-based intelligent tutoring systems, Bayesian networks, formal methods, semantic classification

1. Introduction

Modern intelligent tutoring systems attempt to explore relatively unconstrained interactions with students, for example via a natural language (NL) dialogue. The rationale behind this is that allowing students to provide unrestricted input to a system would trigger meta-cognitive processes that support learning (i.e. self-explaining) [1] and help expose misconceptions. WHY2-ATLAS tutoring system is designed to elicit NL explanations in the domain of qualitative physics [7]. The system presents a student a qualitative physics problem and asks the student to type an essay with an answer and an explanation. A typical problem and the corresponding essay are shown in Figure 1. After the student submits the first draft of an essay, the system analyzes it for errors and missing statements and starts a dialogue that attempts to remediate misconceptions and elicit missing facts.

Although there are a limited number of classes of possible student beliefs that are of interest to the system (e. g., for the Pumpkin problem, about 20 correct and 4 incorrect

[1]Correspondence to: Maxim Makatchev, The Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA, 15213, USA. Tel.: +1 412 268 3474; Fax: +1 412 624 7904; E-mail: maxim.makatchev@cs.cmu.edu.

Question: Suppose you are running in a straight line at constant speed. You throw a pumpkin straight up. Where will it land? Explain.

Explanation: Once the pumpkin leaves my hand, the horizontal force that I am exerting on it no longer exists, only a vertical force (caused by my throwing it). As it reaches it's maximum height, gravity (exerted vertically downward) will cause the pumpkin to fall. Since no horizontal force acted on the pumpkin from the time it left my hand, it will fall at the same place where it left my hands.

Figure 1. The statement of the problem and a verbatim explanation from a student who received no follow-up discussions on any problems.

beliefs), for each class there are multiple examples of NL utterances that are semantically close enough to be classified as representatives of one of these classes by an expert. Typically the expert will classify a statement belonging to a certain class of student beliefs if either (1) the statement is a re-phrasal of the canonical textual description of the belief class, or (2) the statement is a consequence (or, more rarely, a condition) of an inference rule involving the belief. An example of the first case is the sentence "pumpkin has no horizontal acceleration" as a representative of the belief class "the horizontal acceleration of the pumpkin is zero." An example of the second case is the sentence "The horizontal component of the pumpkin's velocity will remain identical to that of the man's throughout" as a representative of the belief class "The horizontal average velocities of the pumpkin and man are equal": the letter can be derived in one step from the former via a domain rule. The second case occurs due to the coarseness of the classes: the expert would like to credit the student's answer despite the fact that it doesn't match any of the pre-specified semantic classes, based on its semantic proximity to the nearby semantic classes. To summarize, utterances assigned to a particular semantic class by an expert can have different syntactic and semantic features.

While *bag-of-words* methods have seen some successful applications to the problems of semantic text classification [4], their straightforward implementations are known to perform weakly when the training data is sparse, the number of classes is large, or classes do not have clear syntactic boundaries[1] [6]. This suggests using syntactic and semantic parsers and other NLP methods to convert the NL sentences into symbolic representations in a first-order predicate language [7]. However, uncertainty inherent in the various NLP methods that generate those representations [5] means that the same utterances can produce representations of different quality and structure. Figure 2, for example, shows two representations for the same utterance, produced by different NLP methods.

The uncertainty in parsing and in other NLP components adds to the syntactic and semantic variability among representatives of a semantic class. Encoding these sources of variabilities explicitly appears to be infeasible, especially since the properties of the NLP components may be difficult to describe, and the reasoning behind an expert assigning a semantic class label to an input may be hard to elicit. A method to combine different sources of uncertainty in an NLP pipeline using a Bayesian network has been proposed in [3]. Our objective is to adapt this idea to the scenario when semantic classes themselves, as well as relationships between them are uncertain. In particular, we would

[1] Syntactic features alone become insufficient for classification when classes depend on the semantic structure of the domain (as described in the previous paragraph), including the sensitivity to conditionals and negation.

Representation I:

```
(position pumpkin ...)
(rel-position ...pumpkin at)
(quantity2b ...0 ...)
(acceleration pumpkin ...)
(rel-coordinate ...)
```

Representation II:

```
(acceleration man horizontal ...0 ...)
```

Figure 2. Representations for the sentence "There is no horizontal acceleration in either the pumpkin or in the man, and therefore is inconsequential," produced by two different methods. For the sake of simplicity we omitted uninstantiated variables.

like to learn the relationships between the various elements of symbolic representations and semantic classes directly from the data.

In this paper we propose to combine the expert's knowledge about the structure of the domain with the structure's parameter estimation from the data. In particular, we utilize a Bayesian network with nodes representing the facts and domain rule applications, and directed edges representing either an antecedent relationship between a fact and a rule application, or a consequent relationship between a rule application and a fact. In addition, subsets of nodes are grouped as antecedents to the nodes representing semantic class labels. The design of the classifiers is described in Section 2. The details on our dataset are given in Section 3.

The structure of the Bayesian network is derived as a subset[2] of the deductive closure C via forward chaining on the givens of the physics problem using a problem solver, similarly to the approach taken in [2]. However, unlike [2], we estimate the conditional probabilities from the data. We will compare the Bayesian network classifier with (a) a direct matching classifier that assumes syntactic similarity and NLP consistency in generating symbolic representations; (b) a classifier based on the match of the input with a particular subgraph of the deductive closure C and checking the labels in the neighborhood of this subgraph; (c) a majority baseline and (d) a Bayesian network with untrained parameters. The evaluation results are presented in Section 4. In Section 5 we summarize the results and outline the ways to improve the system.

2. Classifiers

In general, our classification task is: given a symbolic representation of a student's sentence, infer the probability distribution on a set of student beliefs representing knowledge of certain statements about domain. By the domain statements here, we mean physics principles, instantiated principles (facts) and misconceptions. In this paper we will consider an evaluation of the classifying of facts, which is a part of the measure of *completeness*, as opposed to classifying of misconceptions, which we referred to as *correctness* in [8]. We treat these problems differently: analyzing a utterance for misconceptions is viewed as a diagnosis problem (a student's utterance can be arbitrary far in the chain of

[2]For the sake of simplicity we will refer to the subset of the deductive closure that is fixed throughout the study as the deductive closure.

reasoning from the application of an erroneous rule or fact), while coverage of a fact or a rule by an utterance is viewed as a problem of semantic proximity of the utterance to the fact or the rule. We will compare the performances of the classifiers described below.

2.1. Direct matching

Each of the semantic classes of interest is equipped with a manually constructed symbolic representation of the "canonical" utterance. The direct matching classifier uses a metric of graph similarity based on largest common subgraph [10] and a manually selected threshold to decide whether the symbolic representation of a student's utterance is similar enough to the representation of the canonical member of the semantic class. The method is fast and does not require running any logical inference procedures. However, obviously it can only account for limited variations in the semantic representations.

2.2. Matching against a deductive closure subset of radius 0

For each of the physics problems we generate off-line a deductive closure of problem givens, likely student's beliefs and domain rules. In the case of the Pumpkin problem covered in this paper, the deductive closure has been built up to the depth 6, has 159 nodes representing facts, and contains the facts corresponding to the 16 semantic classes of interest. In fact, out of the total of 20 semantic classes relevant to the solution of the problem we have limited our investigation to the 16 classes that we were able to cover by the deductive closure created with a reasonable knowledge engineering effort. This effort included manual encoding of 46 problem-specific givens and 18 domain rules that can be shared across a range of physics problems.

The nodes of the deductive closure are then automatically labeled (off-line) with the semantic class labels via a graph matching algorithm used in the direct matching classifier described in Section 2.1. During the run-time, the symbolic representation of a student's utterance is matched against the deductive closure via the graph matching algorithm and the matching nodes of the closure are then checked for sufficient coverage of any of the semantic classes by counting their semantic labels [8].

2.3. Matching against a deductive closure subset of radius 1

The radius here refers to the size of the neighborhood (in terms of the edge distance) of the subset of the nodes of the deductive closure that match the symbolic representation of the input utterance. In the previous case, we considered only those nodes that matched the utterance. Here, we consider the set of nodes that are reachable within one edge of the nodes matching the input utterance [8]. Thus the neighborhood of radius 1 contains the neighborhood of radius 0. It is natural to expect that this classifier will over-generate semantic labels, improving recall, but likely decreasing precision.

2.4. Bayesian network, untrained

Our intention is to use the structure of the deductive closure graph to construct a Bayesian network classifier. We augment this graph with additional nodes corresponding to the semantic class instances and semantic class labels. Thus, the resultant Bayesian network graph in our proposed method consists of following types of nodes:

- 159 nodes corresponding to the domain facts in the original deductive closure. These nodes are observed in each data entry: a subset of them that is matched to the symbolic representation of the student utterance is considered to be present in the input utterance, the rest of these nodes are considered not present in the utterance;
- 45 nodes corresponding to domain rule applications in the original deductive closure. These nodes are unobserved. Parents of such node are the nodes corresponding to the antecedent facts of the rule application, and children of such node are the nodes corresponding to the facts generated by the rule application (consequences);
- 16 nodes representing the class label variables. They are childless and their parents are nodes corresponding to the instances of the semantic class among the subsets of fact nodes. These nodes are observed for every data entry of the training set according to the human-generated semantic labels of the utterance.
- 62 unobserved nodes corresponding to the instances of the semantic class in the deductive closure.

Each of 282 nodes in the network is boolean valued. We use informative priors for conditional probabilities: boolean OR for the class label nodes, and boolean OR with probability $p = 0.1$ of reversing its values for the other nodes. In this baseline classifier we don't train the parameters, using just the default conditional probabilities.

2.5. Bayesian network, trained

This classifier consists of the Bayesian network built in the same way and as the classifier above, but this time the network parameters are estimated via Expectation-Maximization (EM) with the same informative priors on conditional probabilities as above, using 90% of the dataset at each step of the 10-fold cross-validation.

3. Dataset

The data set consists of 293 labeled NL utterances collected during a Spring and Summer of 2005 study with student participants. The features are:

- a mapping to the values of the observable nodes of the Bayesian network (deductive closure),
- a human-generated score (an integer between 1 and 7) indicating the quality of the symbolic representation,
- zero or more of human-generated class labels from the set of 16 semantic class labels and their respective binary confidence values (high/low).

From the histogram of the semantic labels shown in Figure 3, it is clear that the data is skewed towards the empty label, with 35.49% of the examples not labeled as corresponding to any of the 16 semantic classes. Thus we expect that the majority baseline classifier that always predicts the empty label would perform quite well. However since it is particularly important to give a credit to a student's contribution when there is one, a slight raise of the performance above the empty-label majority baseline can mean a

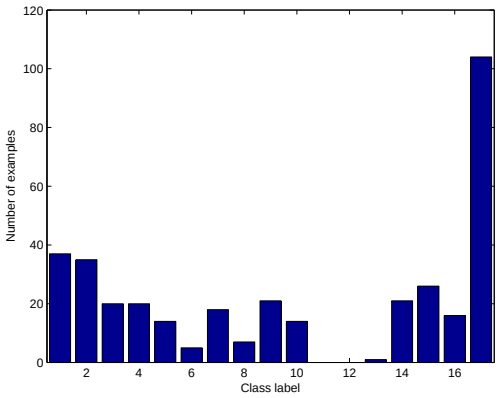

Figure 3. Histogram of the 16 semantic labels and the "empty" label (the rightmost column) in the dataset.

significant change in the application's responsiveness to the student's non-empty contributions.

Due to the relatively small amount of labeled data, for the current experiment we decided to discard the human-generated confidence values of class labels and quality scores of symbolic representation to reduce the dimensionality of the data. However, we account for the degree of confidence in matching the symbolic representations of utterances with the nodes of the deductive closure by generating two instances of the dataset that differ only in the threshold for the graph matching algorithm [10] that decides what nodes of the deductive closure graph correspond to the graph of symbolic representation of the utterance. Namely, for the dataset *data07* the predicate representation of NL utterances is mapped to the nodes of the deductive closure with the more permissive threshold of 0.7 (less overlap of the structure and labels of the two graphs is required), while for the dataset *data09* the threshold is set to the more restrictive value of 0.9 (more overlap of the structure and labels of the two graphs is required). The data with lower confidence are, somewhat counter-intuitively, harder to obtain due to increased search space of the graph matcher. The more permissive similarity matching results in more matched nodes of the deductive closure and therefore of the Bayesian network, potentially making the dataset more informative for training and prediction. This is one of the hypothesis that we test in our experiment in Section 4.

4. Evaluation

The evaluation consists of 10-fold cross-validation on *data07* and *data09* datasets (Tables 1 and 2). The performance measures are average recall and average precision values for each of the entries. The following classifiers have been compared:

- *direct*: Deterministic matching directly to the class representations (no deductive closure structure).
- *radius0*: Deterministic matching to the deductive closure and then checking the labels of the matched closure nodes (uses deductive closure structure).

- *radius1*: Deterministic matching to the deductive closure and then checking the labels of the closure nodes within inference distance 1 from the matched closure nodes (uses deductive closure structure).
- *BNun*: Probabilistic inference using untrained Bayesian Network with informative priors (uses deductive closure structure).
- *BN*: Probabilistic inference using EM parameter estimation on a Bayesian Network with informative priors (uses deductive closure structure).
- *base*: Baseline: a single class label that is most popular in the training set.

Classifier	Recall	Precision	F-measure
direct	0.4845	0.4534	0.4684
radius0	0.5034	0.4543	0.4776
radius1	0.5632	0.4120	0.4759
BNun	0.2517	0.0860	0.1282
BN	0.4948	0.5000	0.4974
Base	0.3897	0.3897	0.3897

Table 1. Performance of 6 classifiers on *data07*.

Classifier	Recall	Precision	F-measure
direct	0.5138	0.5103	0.5120
radius0	0.4690	0.4690	0.4690
radius1	0.4713	0.3957	0.4297
BNun	0.2069	0.0787	0.1140
BN	0.4701	0.4707	0.4704
Base	0.3931	0.3931	0.3931

Table 2. Performance of 6 classifiers on *data09*.

The first observation is that the higher confidence dataset data09 did not result in better performance of the Bayesian network classifier. We attribute this to the fact that high confidence data contained very sparse observations that were insufficient to predict the class label and to train the parameters of the network.

Second, the deterministic methods that take advantage of the deductive closure outperform the deterministic direct matching that does not use the deductive closure on both the recall and precision (*radius0*), or just the recall while sacrificing the precision (*radius1*) (Table 1). Moreover the method that uses a neighborhood of the matching subset of the deductive closure, *radius1*, has better recall (and a worse precision) than the method that uses a just the matching subset of the deductive closure, i. e. *radius0*.

Third, the structure alone is insufficient to improve the precision, since the Bayesian network that doesn't learn parameters (using the default values), *BNun*, performs poorly (worse than the majority baseline).

Lastly, the trained Bayesian network *BN* seems to be best overall, according to the F-measure, at least when the symbolic representation was generated with a more permissive threshold, as in *data07*. However the improvement is modest, and more investigation is required to determine the behavior of the classifier on larger datasets.

5. Conclusion

From a pragmatic stand point, this work has shown that each increment in technology has increased semantic classification accuracy. According to the F-measures, taking an advantage of the graph of semantic relationships (deductive closure) improved the the performance of the deterministic classifiers from 0.4684 (*direct*) to 0.4776 (*radius0*). A Bayesian network that has been built on top of the deductive closure with its parameters estimated via EM-based training outperformed the deterministic methods (with the score 0.4974) in the case when the mapping of the data onto the network is more permissive. Incidentally, this disproved our hypothesis that the training data with higher confidence

in the labels (i. e. less permissive mapping onto the network) must necessarily result in better performance. Finally, we demonstrated a feasibility of deriving of the Bayesian network structure via deterministic formal methods, when the amount of training data is insufficient for structure learning from the data. Although these results are encouraging, they also indicate just how hard this particular classification problem is.

An interesting direction for future work would be to extend the Bayesian network with the nodes representing uncertainty in observations, namely in semantic labels and in symbolic representation. Eventually, we would like to incorporate the Bayesian network in a framework that would guide the tutorial interaction, for example by generating a tutorial action to maximize an information gain or a certain utility function, as in [9].

Acknowledgements

This research has been supported under NSF grant 0325054 and ONR grant N00014-00-1-0600. The authors would like to thank all members of the Natural Language Tutoring group, in particular Pamela Jordan, Brian 'Moses' Hall, and Umarani Pappuswamy. The evaluation was done using Kevin Murphy's Bayes Net Toolbox for Matlab.

References

[1] Michelene T. H. Chi, Nicholas de Leeuw, Mei-Hung Chiu, and Christian LaVancher. Eliciting self-explanations improves understanding. *Cognitive Science*, 18:439–477, 1994.

[2] C. Conati, A. Gertner, and K. VanLehn. Using bayesian networks to manage uncertainty in student modeling. *Journal of User Modeling and User-Adapted Interaction*, 12:371–417, 2002.

[3] Jenny Rose Finkel, Christopher D. Manning, and Andrew Y. Ng. Solving the problem of cascading errors: Approximate bayesian inference for linguistic annotation pipelines. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 618–626, 2006.

[4] Arthur C. Graesser, Peter Wiemer-Hastings, Katja Wiemer-Hastings, Derek Harter, Natalie Person, and the TRG. Using latent semantic analysis to evaluate the contributions of students in autotutor. *Interactive Learning Environments*, 8:129–148, 2000.

[5] Pamela W. Jordan, Maxim Makatchev, and Kurt VanLehn. Combining competing language understanding approaches in an intelligent tutoring system. In *Proceedings of Intelligent Tutoring Systems Conference*, volume 3220 of *LNCS*, pages 346–357, Maceió, Alagoas, Brazil, 2004. Springer.

[6] Claudia Leacock and Martin Chodorow. C-rater: Automated scoring of short-answer questions. *Computers and the Humanities*, 37(4):389–405, 2003.

[7] Maxim Makatchev, Pamela W. Jordan, and Kurt VanLehn. Abductive theorem proving for analyzing student explanations to guide feedback in intelligent tutoring systems. *Journal of Automated Reasoning*, 32:187–226, 2004.

[8] Maxim Makatchev and Kurt VanLehn. Analyzing completeness and correctness of utterances using an ATMS. In *Proceedings of Int. Conference on Artificial Intelligence in Education, AIED2005*. IOS Press, July 2005.

[9] R. Charles Murray, Kurt VanLehn, and Jack Mostow. Looking ahead to select tutorial actions: A decision-theoretic approach. *J. of Artificial Intelligence in Education*, 14:235–278, 2004.

[10] Kim Shearer, Horst Bunke, and Svetha Venkatesh. Video indexing and similarity retrieval by largest common subgraph detection using decision trees. *Pattern Recognition*, 34(5):1075–1091, 2001.