CS 10701: Final project report.
Textual entailment in the domain of physics

Maxim Makatchev
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
maxim.makatchev@cs.cmu.edu

Abstract

Bag-of-words methods to the problems of semantic text classification and textual entailment have seen some successful applications [3], their straightforward applications are known to break when the training data is sparse, the number of classes is large, or classes do not have clear syntactic boundaries (for example when negational, conditional sentence markers significantly affect classification). These are, however, properties of a typical semantic classification problem in the domain of natural language tutoring systems. Recently formal methods have been evaluated for reasoning about entailment using the logical representations of natural language propositions [5]. This work extends those methods to account for uncertainty in generating logical representations of natural language sentences by using Bayesian networks with observable nodes representing the logical propositions in the domain of the tutorial dialogue corpus, latent nodes corresponding to domain rule applications, and semantic class label nodes. The problem of sparseness of training data is dealt with by using logical inference engine to generate the network structure, and using informative priors for parameter estimation. The results demonstrate improved performance over the formal reasoning approaches and other baselines.

1 Introduction

1.1 Problem

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 metacognitive processes that support learning (i.e. self-explaining) and help expose misconceptions WHY2-ATLAS, is designed to elicit NL explanations in the domain of qualitative physics [6].

The system presents the 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 propositions.

Although there are limited amount of classes of possible student beliefs that are of interest to the system (of 20 statements representing semantic classes for the Pumpkin problem the approach described here will target 16 selected as described in Section 2), there are many possible NL sentences that are semantically close to be classified as representative 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 rephrasal of the 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 velocity of the pumpkin doesn't change" as a representative of the belief class "the horizontal acceleration of the pumpkin is zero": the former can be derived in one step from the letter via a physics domain rule. These examples suggest that a model an expert's classification of student beliefs would have to account not only for syntactic, but also for inferential proximity of the statements. Note that in general, syntactic proximity alone appears to be insufficient to predict of inferential proximity. In this paper we attempt to augment syntactic proximity analysis with a graph of semantic relationships over the set of domain statements. We will compare deterministic and probabilistic inference algorithms that use this graph for a sentence classification.

1.2 Existing system overview

The sequence of natural language processing is as follows:

- A combination of a semantic-syntactic parser, template-filling classifier and a bag of words statistical classifier generates a first-order predicate logic (FOPL) representation of the input sentence [4].
- Based on the semantic representation of the student's input, the completeness and correctness analyzer attempts to classify whether the input sentence corresponds to any of the pre-specified classes of student's beliefs. For example, if the student types "pumpkin has no horizontal acceleration," the analyzer may infer that student believes that the horizontal force of the pumpkin is zero.

In the early versions of WHY2-ATLAS, the reasoning about the student's beliefs was done by generating abductive proofs of the observed student's input on-the-fly. More recently we have used pre-generated deductive closure as a graph of semantic relationships in the space of problem-specific domain statements and deterministic inference mechanism based

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.

on graph matching. We will compare the new approach with these existing deterministic approaches in the experiments described in this report.

1.3 Desired extension

The deterministic mapping from the formal representation of the input to the graph of deductive closure does not account for the uncertainty in generating formal semantic representation. It is desirable to extend the graph of logical relations over the domain statements (a subset of the deductive closure of givens and false assumptions) into a probabilistic graphical model, such as a Bayesian network, and estimate its parameters based on the actual expert labeling of student sentences. In this project, we implement such such extension.

1.4 Related work

Bayesian networks have been gaining popularity as tool of choice for user modeling (e. g. [1]). The have proven particularly suitable for applications that benefit from visualisable user model, such as tutoring systems with inspectable student models.

In the area of NLP, Bayesian networks have been used for reduction of cascading errors in linguistic pipelines [2], among other applications. There, the probability distributions generated by each individual component of NLP system have been accounted for by assigning a the component to a corresponding variable in the network, and performing the approximate inference.

The present work follows the motivation and the general idea of reducing cascading errors with the work by Finkel, Manning and Ng [2]. The major difference is that the Bayesian network in our work is constructed in a semi-automatic fashion, the affordance of the relatively well formalizable domain of qualitative mechanics.

2 Method

2.1 Classifier

The observed data in our problem is the mapping of the formal representations of NL text to the corresponding nodes in the graph of deductive closure. We augment this graph with additional nodes corresponding to class instances and class labels. Thus, the resulting Bayesian network graph in our proposed method (Figure 3, a fragment) consists of following types of nodes:

- 159 nodes representing domain statements in the original deductive closure (ovals in Figure 3). These nodes are observations generated by the semantic parser/matcher;
- 45 nodes corresponding to domain rule applications in the original deductive closure (diamonds in Figure 3). These nodes are unobserved. Parents of these nodes are nodes that are in the body of the rule application, and its children are the nodes that are in the head of the rule application;
- 16 nodes representing the class label variables (triangles in Figure 3). They don't have children and their parents are nodes corresponding to the representatives of the class among the subsets of domain statement nodes;
- 62 nodes corresponding to the representatives of the class (rectangles in Figure 3).

Each of 282 nodes in the network is boolean valued.

2.2 Parameter learning

The parameter learning is done via EM algorithm with both informative and uninformative priors using Bayes Net Toolbox for Matlab. Since the uninformative priors generated significantly worse results than informative ones we report only results with informative priors in this paper. The informative priors that worked best were boolean OR for class label nodes, and boolean OR with probability p=0.1 of reversing its values for the other nodes. Other priors that were discarded included boolean ANDs for all the nodes except for the class labels.

2.3 Datasets

The data set is a set of labeled natural language sentences collected during a study with real student subjects during Spring and Summer of 2005. The features are automatically generated map to the statement nodes of the deductive closure (observable (oval) nodes of the Bayesian network), human generated degree of quality of the semantic representation $(1, 2, \ldots, 7)$; the labels are human generated class labels (multiple labels from the set of 16 can be assigned to each sentence) with binary confidence grades (high/low). A typical entry is shown in Figure 2.

```
<entry>
<sentence>
Since f_equals_m_times_a, if the acceleration is zero then so is
the net force'the first part justifies the second.
</sentence>
<dprop>
((FORCE ID146441 ?VAR606326 ?VAR606327 *X3720018 *X3720017
*X3720016
X3720015 *X3720014 *X3720013 *X3720012 *X3720011 *X3720010
X3720009)
(REL-COORDINATE *X3720013 *X3720008)
(DEPENDENCY ID146442 ID146441 ?VAR606334 INDEPENDENT *X3720004
X3720003))
</dprop>
<typing>
NIL
</typing>
<quality>
2
</quality>
<tags>
p30a high
</tags>
</entry>
```

Figure 2: A typical data entry consisting of the natural language string, the formal representation, quality of representation (2 — human tagged), and the class label with confidence level, tagged based on the input text string only (p30a, high — human tagged).

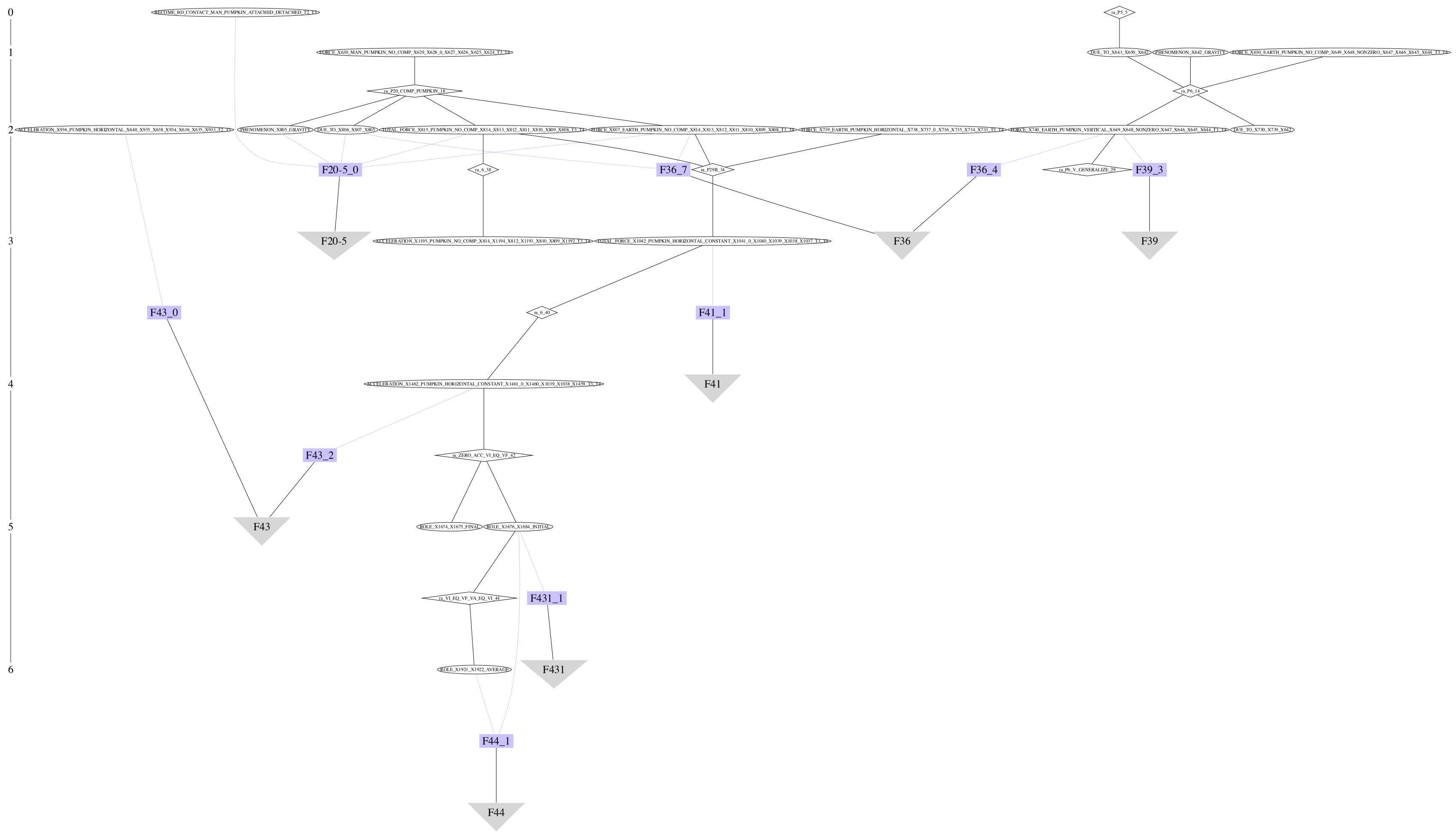
0
1
2
3
4
5
6
BECOME_BD_CONTACT_MAN_PUMPKIN_ATTACHED_DETACHED_T2_T3
FORCE_X630_MAN_PUMPKIN_NO_COMP_X629_X628_0_X627_X626_X625_X624_T3_T4
ra_P20_COMP_PUMPKIN_18
ACCELERATION_X936_PUMPKIN_HORIZONTAL_X640_X935_X638_X934_X636_X635_X933_T2_T3
PHENOMENON_X805_GRAVITY
DUE_TO_X806_X807_X805
TOTAL_FORCE_X815_PUMPKIN_NO_COMP_X814_X813_X812_X811_X810_X809_X808_T3_T4
FORCE_X807_EARTH_PUMPKIN_NO_COMP_X814_X813_X812_X811_X810_X809_X808_T3_T4
FORCE_X739_EARTH_PUMPKIN_HORIZONTAL_X738_X737_0_X736_X735_X734_X733_T3_T4
FORCE_X740_EARTH_PUMPKIN_VERTICAL_X649_X648_NONZERO_X647_X646_X645_X644_T3_T4
DUE_TO_X730_X739_X642
ra_P5_5
DUE_TO_X643_X650_X642
PHENOMENON_X642_GRAVITY
FORCE_X650_EARTH_PUMPKIN_NO_COMP_X649_X648_NONZERO_X647_X646_X645_X644_T3_T4
ra_P6_14
F20-5_0
ra_6_38
F36_7
ra_P29B_34
F36_4
ra_P6_V_GENERALIZE_29
F39_3
F20-5
ACCELERATION_X1195_PUMPKIN_NO_COMP_X814_X1194_X812_X1193_X810_X809_X1192_T3_T4
TOTAL_FORCE_X1042_PUMPKIN_HORIZONTAL_CONSTANT_X1041_0_X1040_X1039_X1038_X1037_T3_T4
F36
F39
F43_0
ra_6_40
F41_1
ACCELERATION_X1462_PUMPKIN_HORIZONTAL_CONSTANT_X1461_0_X1460_X1039_X1038_X1459_T3_T4
F41
F43_2
ra_ZERO_ACC_VI_EQ_VF_42
F43
ROLE_X1674_X1675_FINAL
ROLE_X1676_X1684_INITIAL
ra_VI_EQ_VF_VA_EQ_VI_44
F431_1
ROLE_X1921_X1922_AVERAGE
F431
F44_1
F44

Figure 3: A fragment of the Bayesian network structure. Oval nodes correspond to the domain propositions, diamond nodes are rule applications, rectangular nodes are instances of a class, and triangular nodes are class labels. The depth scale is the number of forward changing inference steps to derive an oval node from given nodes (depth 0). All edge arrows point down.

Due to the small amount of labeled data (173 entries in each of two datasets) we decided to discard the confidence values of class labels and semantic mapping for the current experiment to reduce the complexity of the classifier. The two datasets correspond to the same sentences and human labels but have different confidence threshold in the mapping to the deductive closure nodes (in other words, different confidence in the observed values). We will refer to them as data07 for the dataset with confidence level 0.7, and data09, with confidence level 0.9. The data with lower confidence, counter-intuitively is harder to get (due to increased search space of the graph matcher) but results in general in more matches with statement nodes of the deductive closure/Bayesian network, potentially making it a better training set. This is one of the hypothesis that we will test in our experiment.

3 Experiment

The experiment consisted of 10-fold cross-validation on data07 and data09 datasets. 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 direct matching of the corresponding closure nodes to class representations (uses deductive closure structure).
- *radius1*: Deterministic matching to the deductive closure and then direct matching of the closure nodes within inference distance 1 from the corresponding closure nodes to class representations (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
direct	0.5529	0.50
radius0	0.5706	0.4935
radius1	0.6294	0.4414
BNun	0.2559	0.0619
BN	0.5647	0.5618
Base	0.4353	0.4353

Table 1: Performance of 6 classifiers on data07.

Classifier	Recall	Precision
direct	0.5706	0.5618
radius0	0.5235	0.5235
radius1	0.5176	0.4209
BNun	0.2118	0.0614
BN	0.5176	0.5176
Base	0.4353	0.4353

Table 2: Performance of 6 classifiers on data09.

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 structure (*radius0*, *radius1*) outperform deterministic direct matching that does not use the structure on recall rate, although sacrificing the precision (Table 1). Moreover the method that uses larger subset of the closure structure, *radius1* has better recall (and the worse is the precision) than the method that uses smaller subset of the closure structure *radius0*.

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

4 Conclusion

In this study of textual semantic classification (entailment) we demonstrated that knowing the structure of semantic relationships can improve recall of deterministic classifiers, and learning the parameters of the Bayesian network utilizing this structure improves both recall and precision over deterministic methods. We also disproved the hypothesis that the training data with higher confidence in the labels must necessarily result in better performance. However this effect may be mostly due to the small amount of training data. Finally, we demonstrated that a structure of Bayesian network can me derived via deterministic formal methods when the amount of training data is not sufficient for statistical structure learning.

References

[1] 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.

[2] 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.

[3] 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.

[4] Pamela W. Jordan, Maxim Makatchev, and Kurt VanLehn. Combining competing language understanding approaches in an intelligent tutoring system. In *Proceedings of the Intelligent Tutoring Systems Conference*, 2004.

[5] 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.

[6] Kurt VanLehn, Pamela Jordan, Carolyn Rosé, Dumisizwe Bhembe, Michael Böttner, Andy Gaydos, Maxim Makatchev, Umarani Pappuswamy, Michael Ringenberg, Antonio Roque, Stephanie Siler, and Ramesh Srivastava. The architecture of Why2-Atlas: A coach for qualitative physics essay writing. In *Proceedings of Intelligent Tutoring Systems Conference*, volume 2363 of *LNCS*, pages 158–167. Springer, 2002.