Topic Modeling to Detect Student Expressions of Understanding in Collaborative Problem-Solving Dialogues

Angelica Willis, Ashana Evans, Jung Hee Kim, Kelvin Bryant, Michael Glass
Dept. of Computer Science, North Carolina A&T State University
Dept. Computing and Information Sciences, Valparaiso University

Introduction

When students are working together solving a problem, can a computer gauge how often they show understanding? In the COMPS (COMputer-supported collaborative Problem-Solving) project, students in small groups engage in typed-chat problem-solving dialogues. The instructors can oversee and join the conversations. This project applies topic modeling toward real-time computer assessment of the degree of discussion of the problem, with the aim of posting an assessment of the state of the conversation to an instructor dashboard.

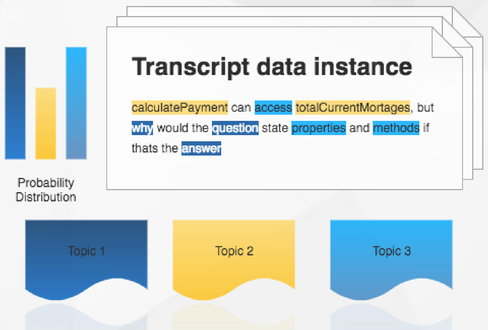
Working from transcripts, we manually annotated dialogue turns where students exhibit understanding or reasoning in the domain of the exercise. In these experiments the students were solving Java programming problems. Then we applied topic modeling to automatically annotate the same turns with automatically-derived features. Finally we trained machine classifiers to recognize the understanding-turns, using the automatically-derived features.

Reasoning in the Domain

The target behavior is reasoning in the domain that exhibits some understanding. Even incorrect understanding counts. The manual annotators identified dialogue where students:

a) Utilized Java knowledge: Java concepts, programming language constructs, or tokens from the program.
b) Showed some thinking about the Java. Simply uttering a variable name, e.g., didn't count.
c) Didn't say they were confused or lacked understanding.

Topic Modeling



Topic modeling is a clustering algorithm commonly used to analyze the thematic meaning of large volumes of unlabeled text.

A COMPS dialogue transcript can be thought of as a collection of mixed topics, each dialogue turn contains some of these topics. The topic modeler uses Latent Dirichlet Allocation (a probabilistic model) to discover collections of words that tend to occur together within the same dialogue turns. One topic is a probability distribution over words.

Each dialogue turn is modeled as containing a mixture of the topics.

The intuition behind this work is that the words used together during reasoning in the problem domain may be discovered as topics.

Example Annotated Dialogue

Who	Turn Text	DU	U0	A	DA	U1	Duration	Overlap	DeltaT	Pauset	Part.	?	Disc. M	You	U1 || U2
Student A	so since this is the only static method , calculatePayment is the only method it has access to	0	0	0	0	1	19122	0	9062	1579	0.733074	0	0	0	1
Student A	and the static variable	0	0	0	0	0	35317	29	738	14104	0.491555	0	1	0	0
Student B	calculatePayment can only access totalCurrentMortgages since they are both static and static methods can only access other static variables	0	1	0	0	0	8793	41	5116	1104	0.504167	0	0	0	1
Student C	can't it return a value into principalAmount and interestRate from the reference variable passed to the method	0	1	0	1	0	43854	107	-4939	1906	0.728464	0	0	0	0
Student B	it can return a value, but the method itself cant acccess anything that isnt static	0	1	0	0	0	24418	102	-18983	1592	0.447435	0	0	0	1
Student C	thats sort of accessing it or is it not?	1	0	0	0	0	13563	64	-4932	801	0.729796	0	0	0	0
Student B	and since the other variables arent static you cant reference them in the method	0	1	0	0	0	10809	46	-4312	1480	0.456597	1	0	0	1
Student A	it doenst ask about returning anyhting	0	0	0	0	0	13147	38	-6192	1261	0.731059	0	0	1	0
Student C	i know i was confues about accessing and returning values	0	1	0	0	0	8585	0	3860	616	0.496142	0	0	0	1
Student B	accessing a variable means using it in the method. the method can only use the sttaic variable of totalCurrentMortgages	0	1	0	0	0	10529	57	3288	1032	0.461964	0	0	0	1
Student C	so thats our answer then	1	0	0	0	0	19382	79	2615	1228	0.726893	0	0	0	0
Student B	yes	0	0	1	0	0	5225	26	10292	528	0.470273	0	1	0	0
Student B	calculatePayment can only access totalCurrentMortgages since they are both static and static methods can only access other static variables	0	1	0	0	0	1015	3	-728	187	0.728157	0	0	0	1
Student A	agreed	0	0	1	0	0	263	2	19201	42	0.734474	0	0	0	0
Student C	I agree with it	0	0	1	0	0	37353	11	8683	18496	0.489226	0	0	0	0

[Legend for transcript header] (Do we want to keep U0 and U1? or just U?)

Software Processing Overview

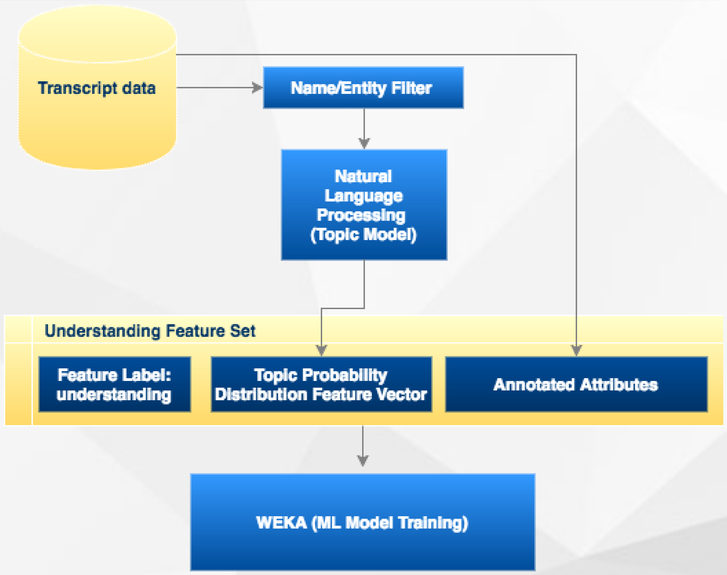
Transcript data includes the text from which topics are extracted, as well as the additional **annotated attributes** (some by hand and by calculation) from the transcript as seen above.

Name/Entity recognition is applied to the transcript text to remove unhelpful data (i.e. the names of students, as they address each other). This issues that topics do not form around a subject based on who is being addressed.

The Topic Model intakes a string of cleaned text, and outputs a distributed probability of how likely the given text contains each of the predefined themes, which becomes our primary **feature vector.**

The **Feature Label,** also called a class variable, this is what we are trying to classify with our model, and is the key output component in prediction. For training, it is already defined so that the system may learn. This system aims to predict cognitive reasoning in the learning domain.

WEKA - A Machine Learning library that can be implemented in Java, and contains the building blocks needed to develop a ML system, including some popular ML algorithms. WEKA is used to train the model, and can be saved using java serialization for repeated use in testing, classification and retraining



Experiment

Our experiments clustered transcript data into 20, 70 and 100 topics. Our best results we discovered using J48 Decision Trees algorithm, closely followed by the LogitBoost Additive Logistic Regression algorithm within WEKA.

Example Extracted Topics & most popular words found

Topic 1
answer (51) correct (38) yeah (30) yea (26) part (25) agree (23) move (18) makes (14) wrong (14) problem (12) cool (12) explanation (11) sense (11)

Topic 2
private (119) double (86) data (81) type (74) int (40) modifier (38)

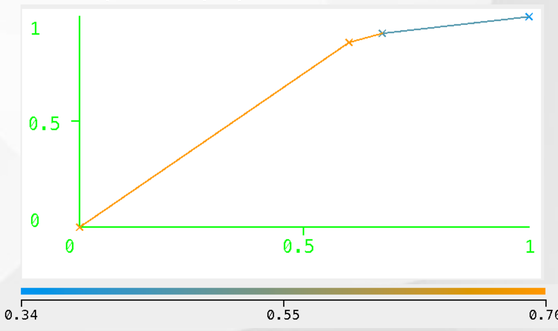
Topic 2
rate (31) amount (27) interest (25) principal (23) variable (16) total (16) number (16) current (15) class (15) double (15) parameters (14)

Results

	J48 Decision Trees			LogitBoost		
Num Topics	Precision	Recall	Percent Correct	Precision	Recall	Percent Correct
20	73.0%	33.2%	71.4%	67.5%	45.1%	72.3%
70	66.66%	35.4%	70.2%	70.4%	43.0%	72.8%
100	68.5%	35.4%	70.7%	59.2%	52.3%	69.7%

It can be noticed that, as the number of clusters (topics) used increases, the Precision seems to decrease while the Recall increases, and the overall percentage of correctly classified instances remains roughly constant.

Receiver Operator Curve
(J48: 20 Topics) Area under curve =.64



Acknowledgment

Partial support for this work was provided by the National Science Foundation's Improving Undergraduate STEM Education (IUSE) program under Award No. 1504917. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.