Automatic coding of learners' self-explanation when learning from diagrams

Shaaron Ainsworth, Richard Forsyth, David Clarke, Laura Robertson and Claire O'Malley

School of Psychology & Learning Sciences Research Institute
University of Nottingham

Nature of the Problem

- To understand the process by which student learn, protocol analysis is often required.
- But protocol analysis is extremely time and labour intensive (and I hate doing it).
  - Developing, applying and checking the reliability of coding schemes takes between 10 and 50 hours per hour of protocols
- Wouldn't it be lovely if I had a research assistant, who was reliable, accurate, fast, free and always available
  - I could work with learners outside the laboratory contexts, with many more participants, over much longer periods of time
- Maybe I should build one….

Structure of the Talk

- What features would we want in our hypothetical system?
- CODELEARNER
  - Initial explorations in a system that analyses protocols (non-technical explanation)
- Case Study
  - Learning by self-explanation from concrete and abstract diagrams of the cardiovascular system
- Conclusions

The Process of Protocol Analysis

- Researchers must record, transcribe, segment and code natural language. Each segment is typically a few words in length.

Text segment	Code
The septum runs down the middle of the heart	Paraphrase
It must be there to stop the different types of blood from mixing	Self-explanation
I don't know what the septum is for	Monitoring

Our Imagined Scenario

- A researcher is faced with a corpus which contain hundreds to thousands of text segments.
- This researcher may have a novel or non-standard coding scheme.
- She is willing to apply this scheme by hand to some segments (and check them) but could not possibly code many thousands.
- We assume she wants the system coding to be accurate and that she should code as few as she can.

The Decision Cycle

- Code a few text segments
- Test the system with the cases coded so far to see how accurate the system is
- decide whether to continue coding more examples
  - stop, since the desired accuracy level has been already attained;
  - continue to code more training cases to approach the desired accuracy level, because it can be reached but only with more data;
  - abandon, since the necessary level of accuracy will never be reached or not reached without excessive effort.

Key Functionality

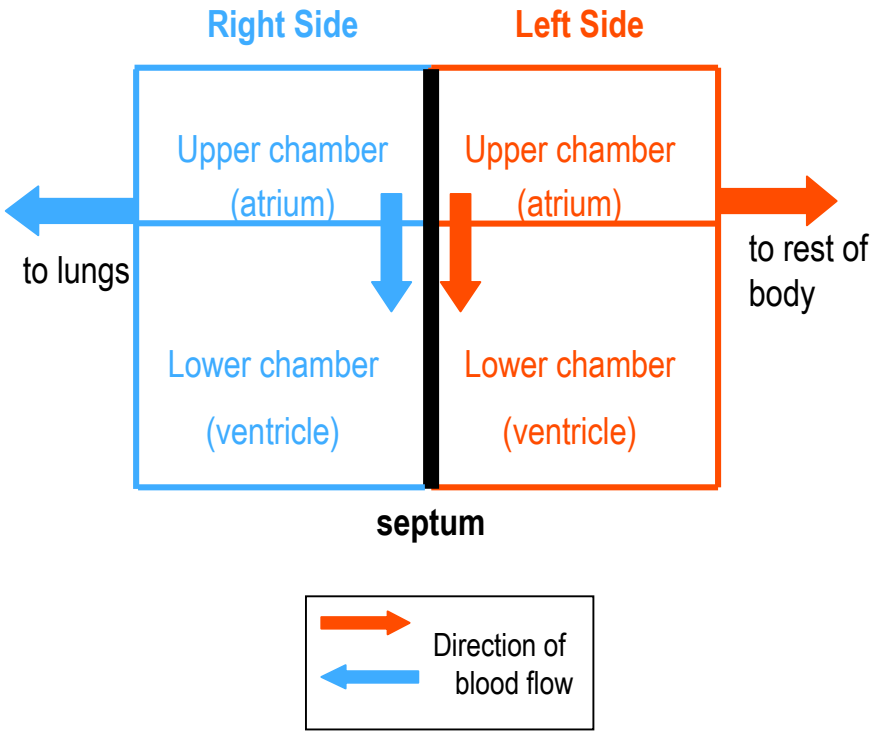
- Accuracy: the ability of the system to assign the same code to text segment that an expert human coder would.
  - Aim for 100% agreement, but in reality this will never happen
  - The degree of accuracy tolerated will depend upon the research question
- Economy: the ability of the system to get close to its final accuracy level with modest amounts of training data.
  - E.g. 80% human coding much less useful than 20%
  - Again the degree of economy depends upon the situation
- Predictability: the system must be able to estimate its own performance on a full data set from a relatively small subset.
  - e.g. we would not want to code 50% of the data and then find the required level of accuracy could not be reached. Nor would we want to code 50% if only 10% was required level of accuracy.

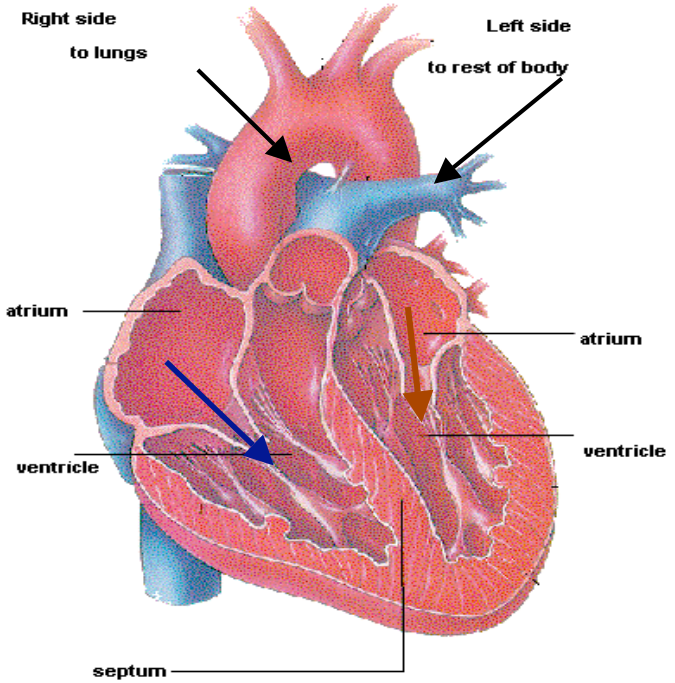
Case Study Details

- Background: Learners who self-explain learn more than those who do not (Chi, Renkl, etc)
- Diagrams can increase the quality and quantity of self-explanations (Ainsworth)
- The coherence of text influences the self-explanation effect with less coherent text increasing the amount of self-explanation (Ainsworth)

Case Study

- Is the self-explanation effect influenced by the how abstract the diagrams are?



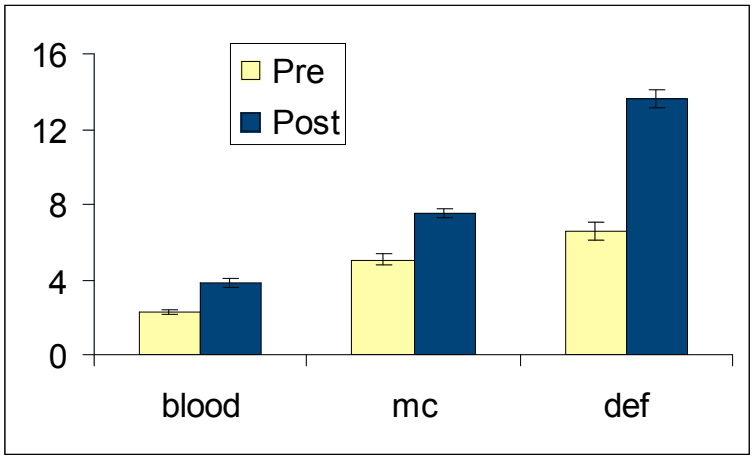


Study Method

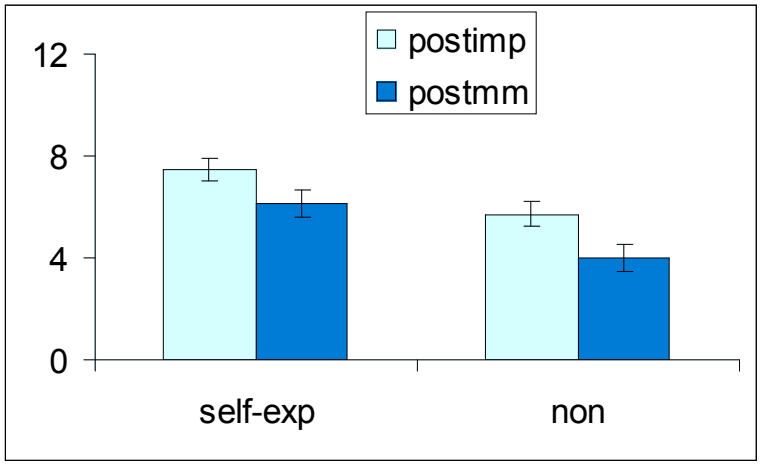
- 2 by 2 design (trained explainers, non-explainers)/ (abstract, concrete diagrams)
- 48 undergraduates without biology past 16
- Pre-tests (blood path diagrams, multiple choice, definitions)
- Un-timed intervention with 10 diagrams with minimal prompting in the self-explanations conditions
- Post-tests (pre-tests + implicit questions and knowledge inference questions)

Results 1

- All participants improved pre to post-test (F (42,3) = 111, p < 0.0001)

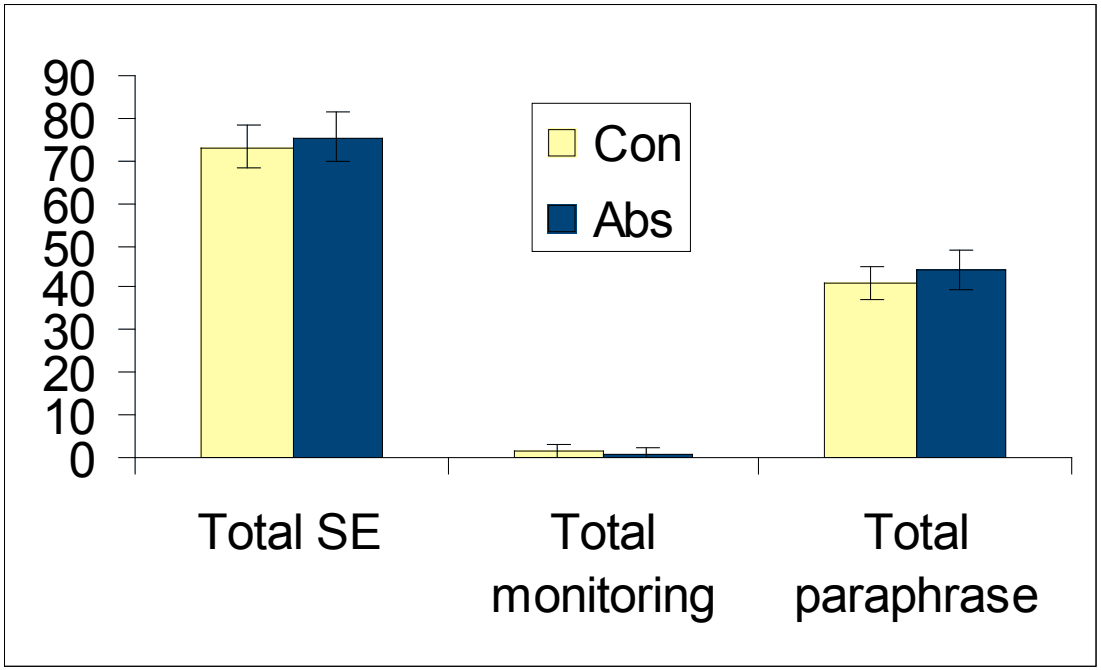


- Self-explainers outperformed non-self explainers on deeper knowledge questions (F (43,2)= 4.58, p < 0.016).



Results 2

- No difference in the amount of self-explanation by condition



CODELEARNER

- *Naive Markov Classifier* (NMC):
  - it finds simple features that are typical of each category of text
  - and then combines them in a mathematical recipe that says how likely each (familiar or new) item of text is to belong to the various categories.
- Cross Validation:
  - selects random samples, of increasing size, from the full data set as training sets, and then tests its 'classification recipes' on new samples to ensure the principles generalise.

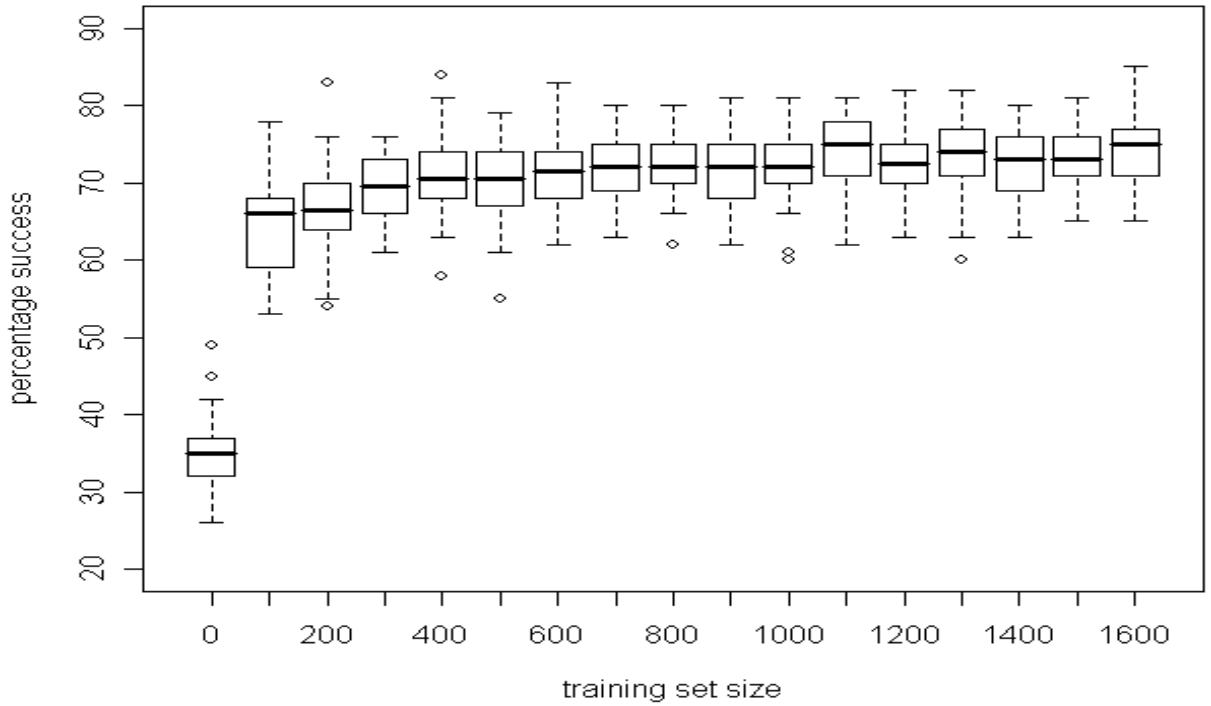
RESULTS: Accuracy

	SE	Monitoring	Paraphrasing
Program-SE	1454	51	719
Program-M	5	103	7
Program-P	467	22	2172

- 74.6% agreement (k=0.51) ☹ human-system
- But only 79.0% agreement (k=0.61) human-human
- Self-explanation coding from diagrams is difficult for humans too

RESULTS: Economy

- The system exhibits very steep learning curves (on this date): only 300 coded segments allows it to achieve an accuracy rate of 70%.



RESULTS: Self Prediction

Conclusions

- Relatively few systems out there (e.g. TagHelper)
- Accuracy not good enough on this problem
  - (although much better on others tasks e.g. Poirot/Marple, Dating Adverts).
- Fares much better on economy and self-predictability
- Is not language specific, adapts easily to others tasks, adapts to multiple classes (with skewed distributions)
- More researched needed, as I (and I suspect you) would really like an automated protocol analyst.