Dimensions of Difficulty in Translating Natural Language into First Order Logic

Dave Barker-Plummer,[1] Richard Cox[2] and Robert Dale[3]
dbp@csli.stanford.edu, richc@sussex.ac.uk, rdale@science.mq.edu.au
[1] CSLI, Stanford University, CA, USA
[2] University of Sussex, UK
[3] Macquarie University, Australia

Abstract. In this paper, we present a study of a large corpus of student logic exercises in which we explore the relationship between two distinct measures of difficulty: the proportion of students whose initial attempt at a given natural language to first-order logic translation is incorrect, and the average number of attempts that are required in order to resolve the error once it has been made. We demonstrate that these measures are broadly correlated, but that certain circumstances can make a hard problem easy to fix, or an easy problem hard to fix. The analysis also reveals some unexpected results in terms of what students find difficult. This has consequences for the delivery of feedback in the Grade Grinder, our automated logic assessment tool; in particular, it suggests we should provide different kinds of assistance depending upon the specific 'difficulty profile' of the exercise.

1 Introduction

The translation of sentences in natural language (NL) into first-order logic (FOL) is a key part of the logic curriculum; indeed, it can hardly be said that a student understands formal logic if they are not able to carry out this translation task competently. For many students, however, it is a difficult task. The difficulties students face are, at least in part, due to characteristics of the natural language sentences themselves. For example, we would expect it to be relatively easy to translate a natural language sentence when the mapping from natural language into logical connectives is transparent, as in the case of the mapping from *and* to $\wedge$, but more difficult when the natural language surface form is markedly different from the corresponding logical form, as in sentences of the form *A provided that B*.

In this study, we aim to characterize translation tasks based on the student's responses to them. Rather than constructing a model of the student (as for example in [5]), we seek to model the task, and to do so in an empirically grounded way, rather than based on intuitions of the author of the exercise. This is in contrast with our past work, in which we focussed on the nature of the errors that students make when performing translation tasks [1].

1. *If **a** is a tetrahedron then it is in front of **d**.*
2. ***a** is to the left of or right of **d** only if it's a cube.*
3. ***c** is between either **a** and **e** or **a** and **d**.*
4. ***c** is to the right of **a**, provided it (i.e. **c**) is small.*
5. ***c** is to the right of **d** only if **b** is to the right of **c** and left of **e**.*
6. *If **e** is a tetrahedron, then it's to the right of **b** if and only if it is also in front of **b**.*
7. *If **b** is a dodecahedron, then it's to the right of **d** if and only if it is also in front of **d**.*
8. ***c** is in back of **a** but in front of **e**.*
9. ***e** is in front of **d** unless it (i.e., **e**) is a large tetrahedron.*
10. *At least one of **a**, **c**, and **e** is a cube.*
11. ***a** is a tetrahedron only if it is in front of **b**.*
12. ***b** is larger than both **a** and **e**.*
13. ***a** and **e** are both larger than **c**, but neither is large.*
14. ***d** is the same shape as **b** only if they are the same size.*
15. ***a** is large if and only if it's a cube.*
16. ***b** is a cube unless **c** is a tetrahedron.*
17. *If **e** isn't a cube, either **b** or **d** is large.*
18. ***b** or **d** is a cube if either **a** or **c** is a tetrahedron.*
19. ***a** is large just in case **d** is small.*
20. ***a** is large just in case **e** is.*

Figure 1. The 20 sentences in Exercise 7.12.

We are interested in developing a richer understanding of what it is that makes a translation exercise difficult. We identify two different measures of difficulty, and explore the relationship between them; this allows us to characterize each translation problem in terms of its **difficulty profile**, which in turn can be used to vary the kind of feedback provided.

Section 2 provides some background to the present study and introduces the corpus we use. Section 3 introduces our two measures of difficulty; Section 4 then discusses the relationship between these two measures, and what they reveal about the nature of the problems students face. In Sections 5 and 6, we focus on two particular difficulty profiles, exploring what they mean for our assessment tool. Finally, Section 7 provides some conclusions and points to future work.

2 The Data

The corpus consists of student-generated solutions to exercises in *Language, Proof and Logic* (LPL) [3], a courseware package consisting of a textbook together with desktop applications which students use to complete exercises.[1] Students may submit answers to 489 of LPL's 748 exercises online; the other exercises require that students submit their an-

[1] See http://lpl.stanford.edu.

swers on paper to their instructors. The electronic submissions are processed by the Grade Grinder (GG), a robust automated assessment system that has assessed approximately 1.8 million submissions of work by more than 46,000 individual students over the past eight years; this population is drawn from approximately a hundred institutions in more than a dozen countries. These submissions form an extremely large corpus of high ecological validity.

For the work reported here, we focus on a specific exercise we selected from LPL; this is a natural language (NL) to first-order logic (FOL) translation exercise of moderate difficulty, i.e. one that psychometrically discriminates between students. Exercise 7.12 from Chapter 7 (which introduces conditionals) was selected because it has the highest proportion of erroneous submissions of all translation exercises.

This exercise involves translating each of twenty English sentences into FOL. Our desktop applications offer relatively little useful feedback for exercises of this type, so compared to other exercise types, a higher proportion of submissions to the Grade Grinder contain errors. A **submission** for Exercise 7.12 consists of a solution for all twenty sentences, and is considered erroneous if the student makes an error on at least one of the solutions. For this study we focus on the calendar year 2007, during which period a total of 2558 students attempted this exercise. 14% got the exercise right without making a single error. The high proportion of submissions containing errors is in part a reflection of the fact that the exercises contains twenty translation tasks, all of which the student would have to have correct first time to avoid being found erroneous. For this study, we examined the corpus of erroneous submissions of Exercise 7.12, representing the work of a set of 2221 students.

A translation for a sentence (which we refer to here as a **solution**) is considered correct if it is equivalent to a **reference solution**; there are infinitely many correct answers for any sentence, so a theorem prover is employed to determine equivalence. The sentences from Exercise 7.12 are presented in Figure 1. The reference solution for Sentence 1 in Figure 1 is Tet(a) $\rightarrow$ FrontOf(a,d).

The emphasis in the Grade Grinder is on self-remediation of errors. The Grade Grinder's response to an erroneous submission of the form FrontOf(a, d) $\rightarrow$ Tet(a), a common error, takes the form:

> *** Your first sentence, "FrontOf(a, d) -> Tet(a)", is not equivalent to any of the expected translations.

This very uninformative, 'canned' feedback is typical of the Grade Grinder's responses. Students using the Grade Grinder may make as many submissions of a given exercise as they need to obtain the correct answer. This means that the corpus contains repeated submissions by the same student of the same task, and enables us to track their path to a solution. A student's work is reported to their instructor only when the student chooses, typically because it is reported as correct or because their deadline has arrived. Further

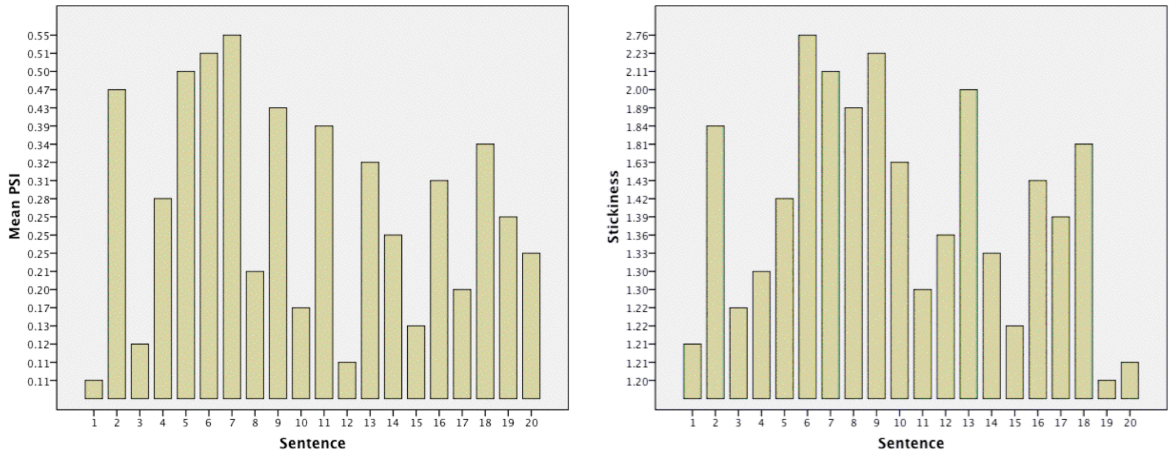

Figure 2. Proportion of students who get each sentence wrong (left), and average stickiness values for the 20 sentences (right).

information on the Grade Grinder, and samples of feedback reports, can be found on the GG website.[2]

3 Measuring Difficulty

How do we measure the difficulty of a translation exercise? Of course, the author of a textbook uses intuitions about difficulty when preparing the exercises used in that textbook. However, even when based on extensive experience, this invariably involves some degree of subjectivity, and it fails to acknowledge that different students may find different problems difficult to different degrees. Ideally we would like to determine the difficulty of an exercise on the basis of empirical data, and further, be able to take account of the fact that different students may face different problems.

We consider here two possible measures of difficulty based on the data we have available.

- First, we can look at the proportion of students who get a particular exercise wrong; the assumption here is that the more students who get an exercise wrong, the more difficult that exercise must be. We refer to this value as the sentence's **PSI** (for 'the *P*roportion of *S*tudents who provide an *I*ncorrect answer'.) Figure 2 (left) shows, for each of the 20 sentences, the proportion of this sample whose initial attempt at that sentence resulted in an incorrect answer. On this basis, we can observe, for example, that Sentences 1 and 12 are relatively easy, whereas Sentences 5, 6 and 7 are relatively difficult.

[2]See http://ggww2.stanford.edu/GUS/gradegrinder.

- A second measure can be obtained by considering how many attempts it takes for a student to determine the correct answer once they have made their initial mistake. We call this the **stickiness** of the error. Thus, every student has a stickiness value for every sentence they get wrong; for any sentence they correct on their second attempt, the stickiness value of that sentence for that student is 1. If we average this value over all students, we obtain a stickiness measure for the sentence. Figure 2 (right) shows the average stickiness values for the 20 sentences in the exercise based, for each sentence, on the subset of the 2221 students in the sample that made an error on it. This illustrates that Sentence 6 is much stickier—which is to say that it is 'harder to fix', even given the Grade Grinder's feedback—than Sentence 5 which with it shares a high PSI.

Since we have both values for each sentence in the data, we can combine these to produce a tuple we refer to as the **difficulty profile** of the sentence; this captures both the likelihood of a student getting the sentence wrong, and the average number of attempts it takes to fix the error.

4 Correlating the Dimensions

As noted above, 2221 students who made submissions to Exercise 7.12 made one or more errors. For each of these students, a co-occurrence matrix was constructed for the 20 sentences. Each cell of an individual student's matrix coded the relationship between one distinct pair of sentences. For example, if the student made an error on Sentence 3 and an error on Sentence 12, then the cell at [3,12] would be coded 1 (co-occurrence of error), otherwise zero. The individual subject matrices were summed to produce a combined matrix for all 2221 subjects.

The summed matrix was input to the SPSS Proximities procedure to produce a similarity matrix. The similarity matrix formed the input to the SPSS Cluster procedure, which was used to compute a multilevel, agglomerative, hierarchical cluster analysis using Ward's method (see, e.g., [4]). The item clusters are arranged hierarchically with individual items at the leaves and a single cluster at the root. Dendrograms provide a graphical display of cluster analysis output. The dendrogram for all subjects' data (Figure 3) shows the sentence clusters. Bifurcations that are more distant from the leaf nodes mean that the clusters are more dissimilar. For example, in Figure 3, the cluster of Sentences 1, 12, 3, and 15 indicates that many students who made errors on, say, Sentence 3 also tended to make errors on Sentences 1, 12 and 15. The primary branch at the root level indicates two quite distinct major clusters and three somewhat less dissimilar subclusters within the upper main branch (labelled 1–4 in Figure 3).

We then looked at where the individual sentences lay on a scatter plot of PSI against Stickiness; we noted that the dendogram clusters correspond to four distinct bands in Figure 4. The figure makes it clear that there is not a direct correspondence between our two mea-

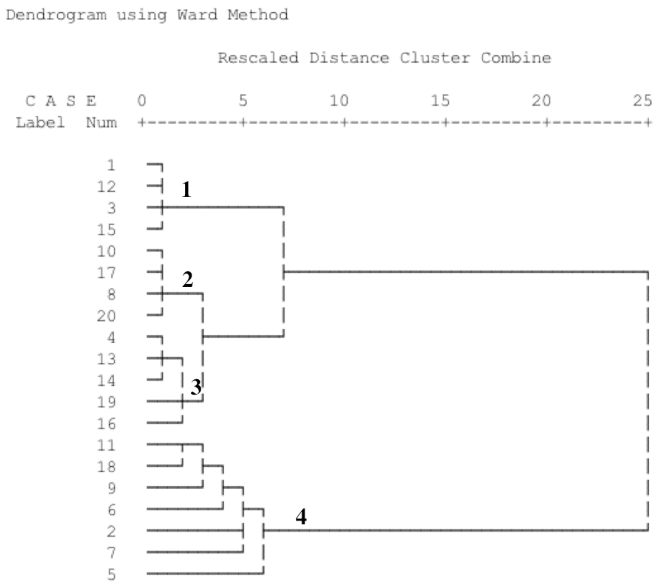

Figure 3. Dendrogram representation of cluster analysis outputs. Leaf node numbers correspond to Exercise 7.12 Sentences 1–20. Clusters labelled 1, 2, 3 and 4 correspond to bands in Figure 4.

sures of difficulty, although there is reasonable correlation in the case of many sentences (Spearman's ρ = .61, p = .004). We single out for particular attention two situations involving outliers (and thus where our two measures of difficulty in some sense 'conflict'): sentences with a high PSI but a low stickiness, and sentences with a low PSI but a high stickiness.

5 Hard to Get Right, But Easy to Fix

We can characterise sentences with a high PSI and a low average stickiness as being hard to get right—many students get them wrong first time—but easy to fix: once you know you've got it wrong, it's easy to work out what the correct answer is.

The most salient examples of this category in our data are Sentences 19 and 20, repeated here for convenience:

19. **a** is large just in case **d** is small.
20. **a** is large just in case **e** is.

Both of these use the natural language expression *just in case*, which translates into FOL as the biconditional, '↔'. As we noted in [1], students find this expression particularly

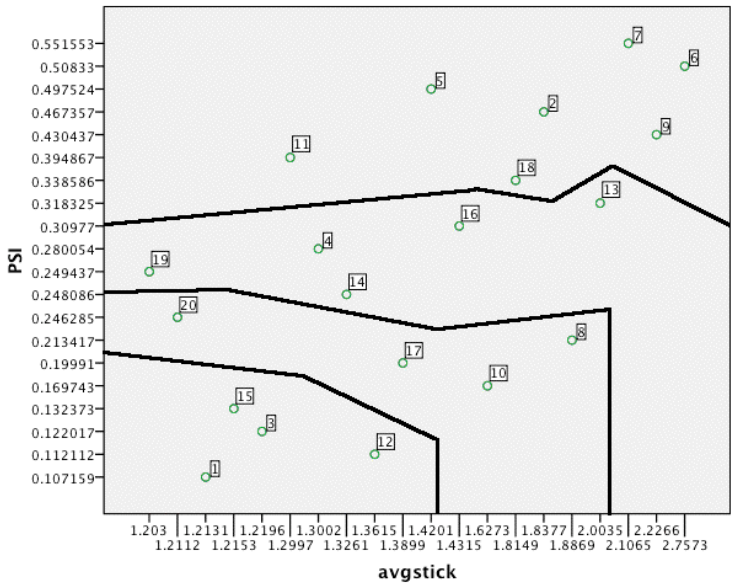

Figure 4. Scatterplot of the difficulty measures PSI and Stickiness. Data point labels indicate Sentences 1–20 of Exercise 7.12 and the 'bands' correspond to the four clusters identified in Figure 3.

difficult, perhaps because it is so rarely used (at least with this interpretation) in natural language.[3]

The vast majority of students who get Sentence 19 wrong offer either of the following two translations, reflecting the common misunderstanding of *just in case* as a bare conditional:

- Small(d) → Large(a)
- Large(a) → Small(d)

The analogous errors also occur very frequently for Sentence 20. In each case, we hypothesise that the students realise that the translation involves some kind of conditional; if they don't get it right the first time, and incorrectly offer a conditional as in the cases above, the most obvious next alternative is the biconditional, which is the correct answer. In effect, there is a very small space of plausible alternative answers given the belief that some flavor of conditional is required, and therefore a relatively low likelihood of making a second error. This space of potential answers is further constrained by the simple nature of the sentences: they are amongst the shortest sentences in the exercise. Sentence 19 mentions only two objects (**a** and **b**) and two unary predicates (Large and Small).

[3] In everyday language, the expression *just in case* is most often used as an approximate synonym for 'as a precaution'. This is not what is meant when it is used by logicians or mathematicians, as is explicitly taught in the LPL textbook.

This suggests that stickiness is related to the extent to which each submitted (or re-submitted) sentence translation by the student reduces the space of plausible solutions.

6 Easy to Get Right, But Hard to Fix

We can characterise sentences with a low PSI and a high average stickiness as being easy to get right—most students get them right first time—but hard to fix: if you get it wrong, it's hard to work out what the correct answer is.

Sentence 8 represents a reasonable example of this phenomenon:

8. **c** is in back of **a** but in front of **e**.

The implication of the difficulty profile of this example is that most students know how to translate *but* into FOL, but if a student doesn't understand this, being told they are wrong (the current feedback provided by the Grade Grinder) is not of any particular help. The high average stickiness of this sentence suggests that students are at a loss as to what the correct answer might be, perhaps even trying random variations on their initial solution to see what works.

In contrast to Sentence 19, discussed in the previous section, students are less likely to make errors on Sentence 8 and it is much stickier (Figure 2). That the PSI is low is probably explained by the fact that the logical connective that translates *but* is the relatively simple logical *and* ($\wedge$). This translation is introduced in an earlier chapter in LPL and may have been internalized by many students by the time they attempt this exercise.

The surface structure of the NL sentence suggests many possible reasons to suppose that students who err in their first attempt find this sentence sticky. Sentence 8 mentions three objects (**a**, **c**, and **e**) and involves two binary relations (BackOf, FrontOf), and the NL sentence contains an elided reference to **c** which has to be made explicit in the FOL translation. Given the uninformative feedback from the Grade Grinder, we conjecture that students do not know which of these features they have misunderstood. The situation is probably compounded by the fact that this exercise appears in the chapter of the book devoted to conditionals, but does not require a conditional in its translation.

7 Conclusions

What should the implications of the analysis presented here be? We have endeavoured to demonstrate that a unidimensional estimation of difficulty based simply on how many students get an exercise right or wrong masks important variations in the kinds of problems students face. In the analysis presented here, we distinguish the proportion of students who

get an exercise wrong as a measure from a measure of how easy it is for a student to *correct* a wrong answer.

More broadly, we can identify four extremes that characterise individual problems:

Hard to Get Wrong, Easy to Resolve: Such problems may be of limited pedagogical value, although they might serve to build a learner's confidence.

Easy to Get Wrong, Easy to Resolve: These might play a role in encouraging care or vigilance, and so might be appropriate for delivery to a careless student.

Hard to Get Wrong, Hard to Resolve: These probably don't belong in the curriculum, since they are likely to engender frustration in the student.

Easy to Get Wrong, Hard to Resolve: These are the most challenging problems, perhaps best reserved only for those students who are on top of the curriculum.

None of these extremes are ideal; what we really want to do is use exercises that are more or less balanced, perhaps with a bias towards difficulty in one or other dimension for particular pedagogical purposes. As a student works through the LPL exercise set, we may be able to incorporate a measure of how they respond to exercises with different difficulty profiles into a student model. Ideally, appropriate examples should be dynamically generated automatically in response to this measure as it is revealed.

We would like to exploit our richer conception of logic exercise difficulty to enhance and enrich Grade Grinder's feedback and to individualise LPL's curriculum. To achieve this the Grade Grinder requires representations of the characteristics of sentences (such as number of predicates, number of constants, and arity). To this end we are currently developing knowledge representations of **stimulus features**: this task involves characterizing each LPL exercise in terms of its **resources**, represented as matrices of properties (for example, number of constants in NL or FOL sentences, number of predicates, types of predicates, their arity, number and types of connectives, and so on). Then we aim to derive, for each sentence, the space of plausible alternative answers based on the number of terms in a sentence, the number of predicates and their arities, *etc*. A large space of plausible alternative answers suggests *prima facie* a more difficult exercise. We will be able to validate the predictive difficulty measures by correlating them with PSI and stickiness, to determine whether these are all of the (and the only) factors playing into students' experiences with the exercise, and which of the surface features are predictive of PSI and which of stickiness.

The sentence characteristics (including the two difficulty measures and the stimulus features) could be encoded in a manner akin to q-matrices and used to represent students' **concept states** at different stages of learning [2]. Stimulus feature analysis of LPL's resources will also inform their decomposition into constituent sub-concepts and skills. Armed with this knowledge we should then be able to track how many times a student has encountered each sub-concept to-date. We may also employ search algorithms and statistical techniques (for example, multivariate logistic regression) to build models of each student's learning [5]. When the Grade Grinder detects that a student is manifesting a more-than-

average number of errors, it can generate bespoke exercises for individual students using methods akin to those used in AI-based adaptive psychometric item generation [6]. Such approaches require, *inter alia*, rich representations of stimulus features, empirical data on item difficulty, and the application of item response theory (IRT) [7]. The ultimate aim is to generate exercises, judiciously adjusting the difficulty parameters and concepts they contain for each individual student.

References

[1] D. Barker-Plummer, R. Cox, R. Dale, and J. Etchemendy. An empirical study of errors in translating natural language into logic. In V. Sloutsky, B. Love, and K. McRae, editors, *Proceedings of the 30th Annual Cognitive Science Society Conference*. Lawrence Erlbaum Associates, 2008.

[2] T. Barnes. The q-matrix method: Mining student response data for knowledge. In J. Beck, editor, *Proceedings of the AAAI Workshop on Educational Data Mining*. AAAI Press, Menlo Park, CA., 2005.

[3] J. Barwise, J. Etchemendy, G. Allwein, D. Barker-Plummer, and A. Liu. *Language, Proof and Logic*. CSLI Publications and University of Chicago Press, September 1999.

[4] B.S.Everitt. *Cluster Analysis*. Edward Arnold, third edition, 1993.

[5] Hao Cen, K. Koedinger, and B. Junker. Automating cognitive model improvement by A* search and logistic regression. In J. Beck, editor, *Proceedings of the AAAI Workshop on Educational Data Mining*. AAAI Press, Menlo Park, CA., 2005.

[6] S. E. Embretson. Measuring human intelligence with artficial intelligence. In R. J. Sternberg and J. E. Pretz, editors, *Cognition and Intelligence: Identifying the Mechanisms of the Mind*, chapter 13. Cambridge University Press, Cambridge, UK, 2005.

[7] S. E. Embretson and S. P. Reise. *Item Response Theory for Psychologists*. Lawrence Erlbaum Associates, 2000.