

Finding Instances of Deduction and Abduction in Clinical Experimental Transcripts

M. Amalfi, K. Lo Presti, A. Proveti¹, and F. Salvetti^{2,3}

Dip. di Fisica, Università degli Studi di Messina.
Sal. Sperone 31. S. Agata di Messina, I-98166 Italy
<http://informatica.unime.it/>
ale@unime.it

Dept. of Computer Science, University of Colorado at Boulder.
430 UCB, Boulder, CO 80309, USA
Umbria Inc.
1655 Walnut St., Boulder CO 80302, USA
<http://umbrialistens.com>
franco.salvetti@umbrialistens.com

Abstract. This article describes the design and implementation of a prototype that analyzes and classifies transcripts of interviews collected during an experiment that involved lateral hemisphere-brain-damage patients. The patients' utterances are classified as instances of categorization, prediction and explanation (abduction) based on surface linguistic cues. The agreement between our automatic classifier and human annotators is measured. The agreement is statistically significant, thus showing that the classification can be performed in an automatic fashion.

1 Introduction

This article describes a software, with a Prolog-based automated reasoner at its core, that we have designed and implemented in the context of our work on measuring utterances that are evidence of inferential ability, viz., prediction and explanation, trying to match a human annotator. The starting point has been a clinical experiment that involved three groups: patients who have been diagnosed brain damage in the left hemisphere (henceforth left-patients), patients who have been diagnosed brain damage in the right hemisphere (henceforth right-patients) and a control group. The experiment was carried out at the Boston VA hospital by professional clinicians; the results were available in writing. The experiment consisted in showing to the subject the picture in Figure 1 (without the A..D frames on) and asking them

“I have a picture here. As you see, there's a lot going on. Look it over and tell me about it.”

Patients' answers were transcribed and analyzed to find evidence that the subject had verbalized some form of reasoning that allowed them to either *explain* certain details of the pictured situation or to *predict* what would happen immediately next. Finding evidence of mental inference in these documents

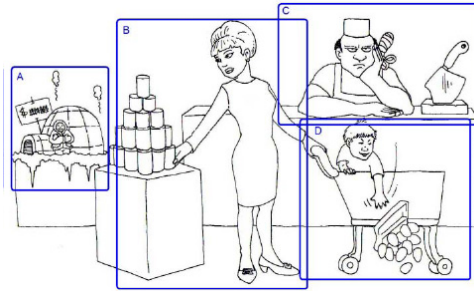


Fig. 1. Picture shown to the patients (with frames added)

is sometimes a challenging task. Also, opinions on what counts as a prediction/explanation may differ widely. However, the experiment was conceived in a way to circumscribe the domain of discourse to the few characters and objects present in the picture. Thus, automated text analysis could be carried out by relatively standard Artificial Intelligence techniques such as pattern matching and backward-chaining reasoning with a Prolog interpreter.

We have noticed that the sub-scenes, framed and labeled A to D in Figure 1 but not in the experiment, suggest more-or-less straightforward *predictions*, e.g., the pile of cans, (frame B) will fall down, and *explanations*, e.g., the boy (frame D) threw off the eggs. This experiment lends itself to automated annotation by text analysis.

To the best of our knowledge, this is the first implemented system that supports automated annotation over this type of clinical test.

A cognitive science interpretation of this experiment is found, among others, in [1]. Their interpretation can be summarized as follows. Left hemisphere-brain-damage patients exhibit verbal evidence of reasoning which, compared to those of a control group, suggests that while capable of performing reasoning, they have less problems in producing sentences with evidence of prediction than sentences with evidence of explanation. Vice versa, right-hemisphere brain damage patients exhibit verbal evidence of reasoning which, compared to those of a control group, suggests that even though abstract reasoning and the ability to make inferences may be impaired, they have fewer problems producing sentences with evidence of explanation than sentences with evidence of prediction.

In this article we are not concerned with the cognitive interpretation of the data, nor with the validation of a particular hypothesis relating lateral brain damage to specific types of reasoning impairment. Rather, we would like to validate our approach to automated text classification by showing, via statistical analysis, that the results are comparable with those of human annotators. Our architecture is designed to avoid commitment to any particular model of rationality but could serve as a tool for validating Cognitive Science theories.

Indeed, one could say that the relatively simple software architecture described here is effective for textual analysis only when simple sentences having

a limited lexicon are considered. However, the advantage of using an automated tool will be can evident when similar experiments will be administered to large populations and human annotation will become uneven or even impossible. Although our software, described in Section 3 is not suitable for large-scale activities *as is*, standard computational complexity analysis yields that our approach can indeed scale up to several hundreds transcripts.

1.1 Relating reasoning to speech

Abductive reasoning is a well-studied topic in knowledge representation and automated reasoning, e.g., in automated diagnosis. So do predictive reasoning and categorization. However, the characterization of abductive phrases as opposed to deduction or categorization phrases requires some preliminary agreement on what constitutes evidence of abduction. As will be explained in the next sections, all human annotators taking part in the validation of our system were given a standard set of rules on what should count as evidence of abduction. The definition of the *is_explan* and *is_predict* predicates in Section 3.3 can be taken as a case-by-case definition of the guidelines¹. It is interesting to assess how the *abductive* reasoning considered here relates to the general use of term.

In systematic treatment of abductive reasoning, Magnani [2] introduces the term *manipulative abduction*. Manipulative abduction happens when we are thinking *through doing* and not only, in a pragmatic sense, *about doing*. In Magnani's classification an abduction is always related to one of these *conjectural template*:

1. curious and anomalous phenomena;
2. dynamical aspects;
3. artificial apparatus or
4. epistemic acting.

The latter conjectural template is the one where our examples fit more easily. Epistemic acting involves interesting features:

- simplification of the reasoning task;
- treatment of incomplete and inconsistent information;
- control of sense data;
- external artifactual models and
- natural objects and phenomena.

We appreciate that this experiment circumscribes the abduction activity in a simple and directed way. That is, abduction is reduced to *simplification of the reasoning task*, since other forms would not apply here. Nonetheless, we found that to have a clear distinction between abduction and the other two forms of reasoning a temporal dimension is still necessary, namely to separate

¹ The guidelines for panels of human annotators, however, are available from <http://mag.dsi.unimi.it/inference-finder/>

phrases about the past from phrases about the present or the immediate future. For instance, consider the phrase *butcher is angry*, which is found in several transcripts. Prima facie, it fits both the *curious and anomalous phenomena* and the *natural objects and phenomena* features. Yet, we can only suppose that the butcher is angry because he cut his finger.

1.2 The underlying Cognitive Model

It is fair to notice that from the point of view of cognitive science the results given by our automated annotator are qualified by a set of framework assumptions about what constitutes evidence of reasoning (deductive or not). The framework assumptions can be summarized as follows:

- it is possible to find evidence of cognitive process by analyzing utterances;
- in particular, it is possible to evaluate an individual’s inferential ability through the analysis of his transcript, and
- the experiment has been effective in circumscribing the domain of discourse to a fixed set of reasoning instances that can be looked upon in the transcripts.

2 Related Research

Deductive reasoning is at the heart of logic-based knowledge representation and reasoning. Recently, it is becoming a topic of interest in Cognitive Science and even in Neuroscience. The reader may refer, for instance, to the survey in [3]. However, the most research efforts seem to use different techniques than the traditional logic-based AI methods. To the best of our knowledge, the work which is closest in spirit to our line of research is the set of experiments that Bucciarelli and her co-authors have designed and performed to verify the predictions of the well-known Johnson-Lairds Mental Model Theory (MMT) by [4].

According to MMT, humans make inferences by constructing mental models that are internal (mental) representations of some external state of affairs [5]. This theory postulates that reasoning depends on understanding the meaning of premises, and then using this meaning and general knowledge to construct mental models of the possibilities under description.

In a recent work [6], Bucciarelli and her collaborators studied the ability of comprehend logical connectives both as abstract verbal entities and inside a complex pragmatic context. To validate the MMT basic predictions that

- hardness of a certain mental task depends on the number of models that need to be considered, and
- reasoning about falsity is harder than reasoning about truth,

[6] describes an experiment where participants are given questions that involve evaluating the truth-value of non-atomic phrases.

Bucciarelli et al. results strongly suggest that the right hemisphere plays a great role in deductive reasoning. In particular, right-patients seem at loss w.r.t. control patients when it comes to reasoning with problems, reporting them and paraphrasing them. However, such disadvantage is not uniform across the spectrum of tests. Indeed, although syllogisms over a unique model are harder for the right-patients than for the control group, syllogisms over multiple models proved hard for all the participants. That finding confirms the MMT forecast that problems with multiple models are difficult *tout court*.

An alternative explanation of Bucciarelli's results, which was put forward by the same authors, is that while the analogical component plays a great role in the reasoning process, the oral component is equally as important in reasoning with syllogisms and periphrasis; indeed on those aspect the the scores of right patients did not differ much from those of the control group.

3 The software architecture

Our objective is that of implementing a software for the recognition of categorizations, explanations and predictions that gives results as close as possible to those of human annotators. To do so, our first step has been to reduce categorizations, explanations and predictions to the occurrence of distinct <subject/verb/object> triples within a transcript [7]. The three types of inference are distinguished by the particular choice of verb tense. Categorization, explanation and prediction are defined as occurrence of key-words in the document. To this purpose, we have formulated a definition of the three concepts in terms of occurrence of words in the sentences.

Figure 1 shows the areas of interest of the picture. Each frame is associated to one or more possible inferences. The *categorization* has been considered as the presence of a word (or its synonym) that individualizes an object, a situation or a behavior represented in the picture. Thanks to this definition, the program can detect categorizations by searching and counting the co-occurrences of one or more key-words at sentence level.

Verbal acts of *explanation* and of *prediction* have been equated to the occurrence of a triple of words inside a sentence. The triples are intended as subject-verb-object (SVO), in this order, which are considered as sufficient evidence of a explanation/prediction act.

For instance, if < *butcher, cut, finger* > is found in a sentence the program should give a score +1 to the interviewed patient (and to his/her group) for having explained why the *butcher* has his finger bandaged. Similarly, finding *woman, demolish* and *cans* in a sentence will result in one point score given for prediction.

Clearly, what triple should be considered good evidence of prediction and/or explanation reasoning depends on the experiment, i.e., on the picture being shown to the subjects of the experiment. Hence, the triples to be searched for are part of the input to the program. It should be noticed at this point that the program is independent from the experiment: to adapt it to the annotation

of a set of transcripts pertaining a different experiment (same modality, different picture) one needs to change exactly three predicate definitions: *is_term*, *is_explan* and *is_predict*. The first definition can be obtained automatically as the output of a lexical analyzer whereas the latter two need a careful analysis of the experiment.

Let us now describe in detail the architecture and the data representation adopted in this project². The system consists of two main components that have been designed and implemented for the experiment described above.

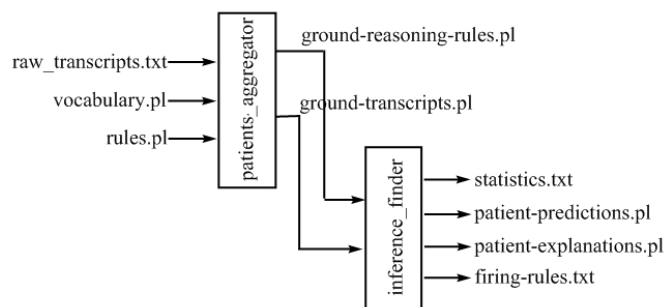


Fig. 2. Overview of the architecture

Their input-output description follows:

1. program *patients-aggregator* takes as input data:
 - the patients' transcript;
 - local vocabulary (since the experiment consider phrases related to the picture in Figure 1, the number of words of interest is finite and rather limited, i.e., 149 words as categorizations. So, it has been possible to represent all tokens of interest by means of Prolog facts) and
 - the general-purpose deduction rules for the Prolog inferential engine.

It produces:

- rules data about categorizations, explanations and predictions;
- patients' data.

Next, *patients-aggregator* scans the transcripts and creates a suitable Prolog representation of the phrases. Then, it also produces the schematic rules that the subsequent Prolog interpretation will use to discover the instances of prediction and explanation in the transcripts.

2. program *inference-finder* is written in Prolog; it takes as input the data generated by *patients-aggregator* and it produces:
 - statistics;
 - patient predictions and explanations and
 - firing rules³.

² The software (source and binary codes), the results and documentation is available from our group page: <http://mag.dsi.unimi.it/inference-finder/>

³ Prediction and explanation rules verified for each patient.

3.1 Representing interview transcripts

The first program, called *patients-aggregator*, takes as input the 3 files with the interviews and it finds out the individual words and delimits sentences⁴.

Every transcript begins with “Patient #” followed by the patient’s identification, the program will write the relative id, type and transcript into file *ground-transcripts.pl*.

For instance, the following is the transcript phrase *Patient #000001*

The butcher / had a / thumb / the butcher / had a bandaid in his thumb / he cuts / his knife / his thumb / he cut / his thumb / with the knife / while cutting cheese / he looks angry / and upset / The boy / in the carriage / has dropped / the eggs / eggs / will fall / on floor / and breaks / break. / The woman / is pulling out / the can / from the bottom. / The can / will fall / in the floor. Frozen food / is on sale / The picture shows the igloo / and the eskimo. / The igloo / is / on sale / too. That’s about it.

and *patients-aggregator* finds out the following rules:

```
patient(000001).
patient_type(000001, lh).
phrase( [000001, [the, butcher, had, a, thumb, the, ..., and, breaks, break],
        [the, woman, is, pulling, out, the, can, from, the, bottom],
        [the, can, will, fall, in, the, floor],
        [frozen, food, is, on, sale, the, picture, shows, the, igloo, and, the, eskimo],
        [the, igloo, is, on, sale, too],
        [that, s, about, it]]).
```

Patients-aggregator will write also the file *ground-reasoning-rules.pl* (a file witch contains schematic rules with variables witch describe the reasoning rules).

3.2 Format of the output

For each patient, our Prolog analyzer produces as output:

1. Patient’s identifier,
2. Patient’s proposed classification (*lh*, *rh* or *control*)
3. words count,
4. categorizations count,
5. explanations count, and
6. predictions count.

⁴ Special characters e.g., “,” “/,” “-” and “(” mark the end of words while the usual “.” “?” and “!” mark the end of sentences. This simple way for tokenization and segmentation is here considered sufficient for the purposes of this research.

3.3 Representing rules

The following rules were used to describe possible categorizations for the example at hand.

```
is_term(lady).
is_term(butcher).
is_term(can).
...
```

This is an example of an explanation rule we used:

```
is_explan('butcher cuts finger',
          [butcher, he, man, boy, mister, kid, guy],
          [cut, sore, wrapped, sliced, cuts, have, ...],
          [finger, fingers, knife, bandage, ...]).
```

whereas this is an example of a prediction rule:

```
is_predict('woman demolishes cans',
           [she, lady, mother, woman],
           [pull, pulls, get, demolish, pulling, ...],
           [can, bottom, display, cans, stack, ...]).
```

By means of lists, we have extended the triples so as to provide a moderate tolerance to stemming, i.e. in the examples *cuts* is also accepted. These extensions, however, have to be discussed on a one-by-one basis to account for verb tenses, which can be very strong indicators in differentiating abduction from prediction. The following Prolog predicates are for parsing the interviews, applying the recognition rules and counting the *hits*.

Categorizations:

```
category1( _, [], 0).
category1( IdPatient, [Term|TermList], N):-
    phrase([IdPatient|Interview]),
    count_occurrences(Term, Interview, N2),
    category1(IdPatient, TermList, N1),
    sum(N1, N2, N).
```

Explanations:

Ph = phrase where we search in the order subject-verb-complement

NameExp = name of explanation

SubL = subjects list

VerbL = verbs list

CompL = complements list

explain(Ph, SubL, VerbL, CompL) = search in the order subject-verb-complement

```
explain( Ph, NameExp):-
    is_explan(NameExp, SubL, VerbL, CompL),
    search_sub(Ph, SubL, VerbL, CompL);
fail.
```


Predictions:

Ph = phrase where we search in the order subject-verb-complement

NamePre = name of prediction

SubL = subjects list

VerbL = verbs list

CompL = complements list

predict(Ph, SubL, VerbL, CompL) = search in the order subject-verb-complement

```
predict( Ph, NamePre):-  
    is_predict(NamePre, SubL, VerbL, CompL),  
    search_sub(Ph, SubL, VerbL, CompL);  
    fail.
```

Ph = one or more phrases over which subject is searched

[SubLH|SubLT] = subjects list

VerbL = verbs list

CompL = complements list

phrase_remainder = search subject into phrase

search_verb(Ph_Remain, VerbL, CompL) = search verb into ph. remainder after subject is found

search_sub(Ph, SubLT, VerbL, CompL) = calls itself with SubLT

```
search_sub( _, [ ], _, _):-  
    fail.
```

```
search_sub( Ph, [SubLH|SubLT], VerbL, CompL):-  
    phrase_remainder(SubLH, Ph, Ph_Remain),  
    search_verb(Ph_Remain, VerbL, CompL);  
    search_sub(Ph, SubLT, VerbL, CompL).
```

List_H = element that we search (Subject/Verb/Complement)

[Ph_H|Ph_T] = phrase over which we search element List_H

Ph_Tail = phrase remainder after element List_H

```
phrase_remainder( _, [ ], _):-  
    fail.  
  
phrase_remainder( List_H, [Ph_H|Ph_T], Ph_Tail):-  
    List_H==Ph_H, Ph_Tail=Ph_T;  
    phrase_remainder(List_H, Ph_T, Ph_Tail).
```

Let us now see in detail how the classifier works vis-a-vis the results of a human panel of evaluators.

4 Results and comparisons

All the interviews considered in this work were annotated by two independent panels of human annotators, here called B (for Boulder) and M (for Messina). Each panel was made of two graduate students of Computer Science, who received similar instructions and very precise instructions on how to annotate interviews.

4.1 Annotating individual phrases

Table 1 summarizes the number of instances of inferential reasoning that were found for this interview.

Results		
-	expl.	pred.
B panel	3	4
M panel	3	1
Program	1	2

Table 1. Annotations on patient #000001 transcript

Let us now see some actual instance. To wit, the M panel found these instances of explanation:

1. *he cuts / his knife / his thumb,*
2. *he looks angry and*
3. *the boy / in the carriage / has dropped / the eggs / eggs / will fall / on floor*

The classifier has been able to find only the first instance of explanation with the *'butcher cuts finger'* rule, which was matched against the following phrase:

[the, butcher, had, a, thumb, the, butcher, had, a, bandaid, in, his, thumb, he, cuts, his, knife, his, thumb, ...]

Considering prediction, The M panel found only this instance:

1. *the woman / is pulling out / the can / from the bottom. / The can / will fall / in the floor.*

whereas the classifier found two instances of explanations:

1. *woman demolish cans, and*
2. *pile crashing down*

In other words, the classifier split the phrase above into two.

4.2 Annotations by the B panel

In the same interview, the B panel found the following instances of explanation:

1. *he cut / his thumb / with the knife*
2. *while cutting cheese*
3. *has dropped / the eggs*

The B panel and the M found the same number of explanations and they are in agreement on only two explanations, the 1st and the 3rd.

In refer to the predictions, the B panel found four instances of them:

1. *eggs / will fall / on floor*
2. *and breaks / break*
3. *The woman / is pulling out / the can / from the bottom*
4. *The can / will fall / in the floor*

The first two were not found by the M panel, while the last two are indicated by the M panel as the same prediction.

5 Validation of the results

The overall number of annotations obtained during our experimental annotation is illustrated in Table 2.

Instances Found		
-	expl.	pred.
B panel	99	71
M panel	87	35
Program	71	32

Table 2. Overall no. of instances found

In the following we describe a statistical analysis of the annotations that supports a more sophisticated understanding of the results.

5.1 The Kappa index

The Kappa index [8], introduced by Cohen [9], has been proposed as a measure of the specific agreement for category among two observers. Kappa measures the accord among the answers of two observers (or the same observer in different moments), that appraises couples of objects or diagnostic categories.

This index captures and corrects the so-called *accidentals agreements*. An agreement is called accidental when two observers reach the same conclusion even though they did so by employing completely different sets of criteria to distinguish between the presence/absence of relevant conditions. In such cases the raw agreement index would not reflect a real agreement. The idea underlying the Kappa index is that the actual accord between two observers is as the difference between the raw agreement and the agreement we would have under the hypothesis that between the two there is no accord and thus their answers may coincide only by chance.

To define K formally, first we introduce the following notation. For any two possible classifications i j , let p_{ij} be the proportion of cases that were classified as i by the first observer and j by the second. Clearly, p_{ii} is the proportions of cases that have been agreed upon to be of type i . Also, let p_i . (resp. $p_{.i}$) be the proportion of cases that the first (the second) observer classified as i (sometimes called marginal frequencies). Now, let

P_o be the proportion of frequencies observed of accords among the two evaluators, and

P_e it is the proportion of accords expected under the *void hypothesis*: accord is determined by the product of the marginal frequencies p_i . and $p_{.i}$.

The value of K is given by the ratio between the excess agreement ($P_o - P_e$) and the maximum obtainable agreement ($1 - P_e$) :

$$K = \frac{P_o - P_e}{1 - P_e} \quad (1)$$

For ordinal variables, which are the case with our scores, the weighted-kappa index, is defined. Weighted-Kappa assigns less weight to agreements as categories are further apart.

Let m be the overall number of categories and let w_{ij} be the weight assigned to the [dis]agreement of the i th and j th categories, with $w_{ii} = 1$, $w_{ij} \geq 0$ ($\forall i \neq j$) and $w_{ij} = w_{ji}$. Then

$$w_{ij} = 1 - \frac{(i - j)^2}{(m - 1)^2} \quad (2)$$

$$P_o^w = \sum_{i=1}^m \sum_{j=1}^m w_{ij} p_{ij} \quad (3)$$

$$P_e^w = \sum_{i=1}^m \sum_{j=1}^m w_{ij} p_i \cdot p_j. \quad (4)$$

$$K^w = \frac{P_o^w - P_e^w}{1 - P_e^w}. \quad (5)$$

If there is a complete agreement, then K (resp. K^w) will be equal to 1. If the observed agreement is greater than or equal to the agreement attended only by chance obtained then the K index will result near zero or even slightly negative. Values of K above 0.6 suggest that there is a substantial agreement; values below 0.21 indicate a weak agreement. Table 3 from [10] shows an interpretation of the values.

To sum it up, K^w is the right type of index to assess the quality of our program vis-à-vis human analysis of some experimental results. Measuring the degree of agreement among two rules and the program, we have the following results:

Between the M panel and the system a substantial agreement is found; whereas group B has only fair agreement both with group M and with the program.

Kappa	Strength of Agreement
< 0.00	Poor
0.00–0.20	Slight
0.21–0.40	Fair
0.41–0.60	Moderate
0.61–0.80	Substantial
0.81–1.00	Almost Perfect

Table 3. The K benchmarks

Explanations			
-	B	M	Program
B panel	1	0,497	0,386
M panel	-	1	0,635
Program	-	-	1

Table 4. The K -weighted degree of agreement on explanations

Predictions			
-	B	M	Program
B panel	1	0,336	0,363
M panel	-	1	0,531
Program	-	-	1

Table 5. The K -weighted degree of agreement on predictions

5.2 Interpretation of the results

The statistics described above show a good agreement between the program scores and those given by the M panel. Vice versa, the B panel results have a relatively low agreement K^w with both the program and the M panel. The B panel consistently finds more instances of reasoning (of any type) than the M panel and the system. These differences can be explained by the fact that the mental model of the M panel annotators is reflected in the program. These results are very satisfying from an Artificial Intelligence perspective: they show that, e.g., from the point of view of the B panel, the classification given by the M panel and that given by the system are hardly distinguishable.

6 Conclusions

We have described the design and implementation of a prototype that analyzes and classifies transcripts of interviews collected during a cognitive science experiment that concerned assessing reasoning bias in lateral-brain damage patients. Our Prolog-based software takes a static description of reasoning rules and matches them on patients' transcripts. Hence, patients' utterances were classified as instances of categorization, prediction and explanation (abduction) based on surface linguistic cues. The agreement between our automatic classifier and human annotators is measured. The agreement is statistically significant, w.r.t. the inherent limitations of the experiment thus showing that the classification can be performed in an automatic fashion. The statistical results support our claim that our software can be safely applied to automate the analysis of experimental results of the type described earlier. Our program can be useful as a provider of second opinions to reveal possible overlooks or mistakes in the diagnostic analysis.

From a Cognitive science point of view, our project may be considered limited by the fact that it can analyze only verbal (transcribed) responses to experiments. Vice versa, from an A.I. point of view the pattern matching mechanism, though rather basic vis-à-vis current natural language processing techniques is implemented fairly elegantly and efficiently in Prolog.

We are currently working to incorporate such techniques, (e.g., regular expressions) into our program. It would be interesting to apply our classifier to the transcripts of the experiment in [6] since their experiment seems within the reach of the techniques we have employed. Another promising direction of research consist in attaching to the token words some *semantics* obtained by automated reference to Wordnet⁵.

acknowledgments

Thanks to Enea Zaffanella for carefully reviewing our submitted version and suggesting several improvements. Thanks to Monica Bucciarelli for suggestions on how to extend this work.

⁵ <http://wordnet.princeton.edu>

References

1. Salvetti, F.: Reasoning and brain damage: prediction and explanation in left- and right-hemisphere patients. Proc. of the XXVth Conf. of the Cognitive Science Society (CogSci) (2006)
2. Magnani, L.: An abductive theory of scientific reasoning. Proc. of the int. workshop on computational models of scientific reasoning and applications (2002)
3. Goel, V.: Cognitive Neuroscience of Deductive Reasoning. Cambridge Univ. Press (2004)
4. Johnson-Laird, P.N.: Mental models. Cambridge University Press, Cambridge, UK (1983)
5. Johnson-Laird, P.N.: Human reasoning and rationality. Int'l Sym. on Foundation and the Ontological Quest: Prospects for the New Millennium (2001)
6. Sacco, K., Bucciarelli, M., Adenzato, M.: Mental models and the meaning of connectives: A study on children, adolescents and adults. Proc. of the XXIIIth Conf. of the Cognitive Science Society (2001) 875–880
7. LoPresti, K.: An automated Classifier for recognizing inferential capacity biases. Graduation Project, Univ. of Messina (2005)
8. Carletta, J.: Assessing agreement on classification tasks: the kappa statistic. Computational Linguistics **22(2)** (1996) 249–254
9. Cohen, J.: A coefficient of agreement for nominal scales. Educational and Psychological Measurement **20(1)** (1960) 37–46
10. Soliani, L.: Statistics Manual for Research and Profession (In Italian). (2005)