

# Reliability of human annotation of semantic roles in noisy text

Derrick Higgins  
Educational Testing Service  
Mail Stop 12-R  
Princeton, NJ 08541  
dhiggins@ets.org

## Abstract

*This paper addresses the question of how to obtain consistent semantic annotation on the basis of a set of noisy texts. Many potential real-world applications of semantic computing are faced with the need to handle texts which are not well-edited, and for which a resource-intensive treebanking effort is not feasible.*

*Student-produced short answers contain many grammatical and lexical errors, making consistent annotation a challenge. Nevertheless, this paper demonstrates that semantic role annotation can be done in a consistent and useful manner even under these constraints.*

## 1 Introduction

*c-rater*<sup>®</sup> is an automated scoring program for content-based short-answer questions. Given a set of training responses to a short-answer question and a “model” built by a content specialist on the basis of those responses, *c-rater* can automatically assign scores to new responses, based on its estimate of which important points are addressed in each. A fuller description of the method by which *c-rater* models are built and applied can be found in [12].

The important aspect of *c-rater* for this paper is that it operates by identifying a set of propositions which ought to be represented in an ideal response, and scoring responses according to the strength of the evidence that these propositions

are indeed present. Identifying particular propositions in a response involves looking for sentences which match a template consisting of a predicate associated with one or more arguments.

*c-rater* uses heuristic rules to extract potential predicate-argument pairs from student responses, but another approach to this problem could also be considered—namely *semantic role labeling* (SRL). Semantic role labeling comprises the identification of the noun phrases associated with particular argument slots of a predicate, typically by means of a machine learning model. If such a model could be successfully applied to *c-rater*, it could allow for both greater scoring accuracy and easier system maintenance.

Evaluation of the performance of semantic role labeling in *c-rater* scoring is not entirely straightforward. Because scoring models for a particular test item are authored by human content experts, who specify a set of paraphrase rules on top of the argument identification component, it is hard to evaluate the independent impact of the argument identification component on the accuracy of the scores. Instead of evaluating scoring accuracy directly, we have human annotators create a gold-standard set of semantic predicate-argument relations in a set of student responses, so that we can compare the performance of an SRL model with *c-rater*'s current argument identification component on this narrower task.

This sort of evaluation presupposes that annotators can consistently identify predicate-argument relations in noisy texts such as those scored

by *c-rater*. Electronically captured student responses are characterized by spelling errors, misuse of whitespace, capitalization, and punctuation, grammatical mistakes, and extreme variation in writing style. Whether annotators can in fact identify semantic relationships in such texts is the question addressed in this paper.

## 2 Semantic Role Labeling

Semantic role labeling, as conceived of in current research, involves the automatic identification of the meaning structure of a sentence. Specifically, this meaning structure is represented by a set of participant roles such as AGENT, PATIENT, and GOAL, which are assigned to the entities involved in the action described by a sentence. These roles are embedded in a frame structure, which expresses the propositional content of the text. Previous research has produced promising results on this task using machine learning tools to derive generalizations from the Berkeley FrameNet database ([1]), and the PropBank annotations of the Penn Treebank ([10]).

The first work to address the problem of semantic parsing from a statistical point of view is that of [9]. Gildea & Jurafsky showed that a statistical system trained on FrameNet could achieve fairly good results in assigning participant role labels within a sentence, with much better robustness than hand-built systems. Since 2002, many other researchers have taken up the issue of automatic labeling of participant roles, and this continues to be a topic of great interest to many computational linguists. It was one of the shared tasks of both CoNLL ([5]) and Senseval ([13]) in 2004.

The best performing semantic role labeling systems, exemplified by [16], use an architecture in which a syntactic parser is applied to each sentence in a first pass, and the task of the semantic role labeling system is to identify certain constituents identified by the parser as semantic arguments of a predicate. This classification of constituents is typically done with complex machine learning systems such as support vector machines, which are capable of handling the large feature

space necessary for this application.

Because the simpler set of roles in PropBank allows machine learning models to generalize more successfully, and because of the more convenient sampling method used for annotating predicates in this corpus, we consider models built on PropBank, rather than FrameNet, in the following. (See [14] for more information about the differences between these resources.)

### 2.1 *c-rater* tuples

The alternate method used by *c-rater* to identify the predicate-argument structure in a response is described in [12]. *c-rater* generates a syntactic analysis of each sentence, and then uses a set of heuristic rules to generate candidate arguments for each predicate identified in the sentence. These predicate-argument relations are known as *tuples* in the *c-rater* system. The roles assigned to each argument are very generic ones, such as (deep) “subject” and “object”, which are useful for distinguishing the different argument slots for a single predicate, but are not differentiated according to different predicate types (as FrameNet roles are). Ultimately, *c-rater* tuples target participant roles, just as SRL models do, and while there may be some differences in the way arguments are encoded, it is not unreasonable to compare the two directly in terms of argument identification accuracy.

## 3 Data

A set of student responses was assembled to serve as a test set for the evaluation of the performance of semantic role labeling and *c-rater*’s current techniques in identifying predicate-argument pairs. These are responses to content-based short-answer questions, written by first- and second-year college students as part of a pilot study.

These responses span two different *prompts*, or *topics*, which are the questions to which the responses are directed. One of the prompts targets *reading comprehension*; the student is asked to read a passage, and then answer a question based

on information within the text itself. The other, *critical thinking*, prompt targets analytical skills, requiring students to synthesize multiple pieces of information from the text in order to produce a correct response.

From the available data, we selected a subset of responses which used specific words we believed to be important predicates for scoring the accuracy of the response. This is discussed in more detail in Section 4.3 below, where we introduce the PropBank-style annotation procedure we used for these predicates. In addition to an electronic version of the topics, and the responses on these topics, we also had access to *scoring rubrics* for each task. These rubrics specify exactly which criteria a response must meet in order to receive credit. This allowed us to make an informed decision about which predicates are most critical for each task, by looking at the sorts of statements required by the rubric for a response to count as a correct answer. In some cases, there was only one point to be addressed in the question, and credit is given in an all-or-nothing fashion (i.e., a score of 1 or 0). The rubrics for other topics allow for partial credit if a student covers some important points, but not the full set of points the question is supposed to elicit. These topics have a score scale consisting of 0, 1, or 2.

The average number of sentences per response ranges between one and four across topics, and the number of words per topic from around 20 to around 60. These are truly short responses in which students provide an answer to a narrowly-focused question, and not essays.

## 4 Annotation

This project involved three major stages of annotation, which were designed to mimic the processing steps which *c-rater* would undertake if it were to use semantic role labeling, but with human intervention to mitigate errors caused by deficiencies in the automated processing systems.

The first step of annotation is light copy-editing, in which annotators fix spelling errors and other egregious problems which are likely to ad-

versely affect later processing steps, such as parsing. This corresponds to the spelling-correction and tokenization step in *c-rater*.

The edited student responses are then processed with a statistical parser, to produce a syntactic tree structure for each sentence. This structure is the basis for the second stage of annotation: syntactic parse correction. This involves making minimally sufficient changes to the parse trees to eliminate structural problems and parse structures which seriously violate the Penn Treebank annotation guidelines.

Finally, our annotators performed PropBank style annotation on selected predicates from each response in order to construct a gold-standard test set of important predicate-argument relations to be identified for each response.

### 4.1 Editing

The annotators for this stage were instructed to correct a limited set of error types (listed in Table 1), and were given examples of these errors from student responses, as a model to guide their editing. Annotators were instructed to err on the side of conservatism, correcting errors only when they were clearly mistakes, and not just questionable stylistic choices. A subset of 130 responses were edited by two independent annotators so that we could assess their agreement on this task.

Each annotator was asked to record the number of corrections of each type which were made to each response. A more exact measure of agreement could be made on the basis of exactly which corrections were made by each editor. However, we have observed that annotation of grammatical errors requires evaluation at a more abstract level. There is some ambiguity inherent in the correction process, because a given error can often be remedied in multiple ways. For example, a subject-verb agreement error can be fixed by changing the number of either the subject or the verb, if both interpretations are plausible. Requiring an exact match between corrections made by each annotator might be too stringent a criterion from this perspective.

Table 1 shows a summary of the corrections made by Annotator A. Somewhat surprisingly, the most common error type among all responses was grammatical errors. (The spelling errors in these responses tend to be very salient.) This may be because the class of grammatical errors is relatively broad, so that annotators were using this code as a bit of a catch-all for errors not easily classified into one of the other types.

Table 2 shows the Pearson product-moment correlation between the number of corrections of each type made by annotators A and B on each response. We believe these correlations are reasonable, given the size of the data samples on which they are based. However, the correlation for the number of spelling errors found was somewhat disappointing. (Again, this may be due to uncertainty about the classification of a given error.)

In the further annotation steps described below, the texts edited by Annotator A were used, for the sake of consistency, and because this annotator had more experience with the task.

## 4.2 Treebanking

The second stage of annotation involved the correction of parse trees produced by the OpenNLP statistical parser.<sup>1</sup>

The optimal way of ensuring a clean syntactic representation would be to have trained linguists do full Treebanking of the responses, just as if they were new additions to the Penn Treebank. However, this would be a very time-consuming and resource-intensive process. As a compromise, we use the OpenNLP statistical parser to provide an initial structure, so that the burden of our annotators is reduced to the task of correcting gross errors committed by the parser. The OpenNLP parser is based on Adwait Ratnaparkhi’s maximum-entropy parser ([17]), which still ranks among the best-performing statistical parsers. This parser is used in *c-rater* as well, which makes it a reasonable starting point.

Previous work has demonstrated that correcting the output of an automatic parser can reduce

the amount of human effort needed to develop a full syntactic tree for a sentence ([7, 15]). Chiou et al. cite a 50% reduction in annotation time, although this result is for Chinese. To our knowledge, though, no previous work has evaluated the quality of the resulting parses by assessing the agreement of independent annotators in producing a corrected tree structure. Our evaluation below fills this gap in the literature.

The task of correcting parses requires some elaboration, because it is not immediately obvious which sorts of errors ought to be corrected, and which should be left alone. Furthermore, the output of a typical parser typically contains much less information than a Treebank tree produced from scratch. For example, most parsers do not identify the location of traces and other empty categories. They also do not indicate binding relationships, or grammatical relations, which are provided in parts of the Penn Treebank.

As in the previous annotation task, we instructed our annotators to operate conservatively, refraining from changes which are not clear-cut, or whose effects on further processing steps would be negligible. Empty categories, binding indices, and grammatical relations were not added to the trees. Annotators were instructed to correct tagging errors when they crossed major category boundaries, and thereby engendered parse problems, but distinctions within a major category (such as NN vs. NNP) were not corrected. Annotators were instructed to focus on “global” errors, in which incorrect delimitation of a given constituent affects a large portion of the parse tree, rather than localized errors involving a single constituent label. These global errors comprise the biggest potential problem for applying semantic role labeling to the task of short answer scoring.

We used three annotators for this project, all of whom were trained linguists (either holding a PhD, or currently enrolled in a PhD program). All three of the annotators had previous experience with Penn Treebank trees, and consulted the Penn Treebank bracketing guidelines in the course of their work.

Table 3 provides some summary statistics

<sup>1</sup><http://opennlp.sourceforge.net/>

Topics	Total Num. Responses	Capitalization Errors	Spelling Errors	S-Breaking Errors	Grammar Errors	All Errors
All topics	1260	159 (0.13)	554 (0.44)	265 (0.21)	966 (0.77)	1946 (1.54)

**Table 1. Error types corrected by Annotator A. The number in parentheses is the average number of errors per response.**

Capitalization Errors	Spelling Errors	S-Breaking Errors	Grammar Errors	All Errors
0.909	0.588	0.665	0.878	0.900

**Table 2. Correlation between number of errors reported per response by each annotator.**

about the 964 sentences for which we have corrections by multiple annotators. The corrected parses produced by the two annotators are of approximately the same size as those in the original parses, with just over 36 nodes per sentence.

The most marked difference between the original parses and the annotators’ corrected versions is that the latter seem to contain many fewer NP nodes. This difference in the number of NP nodes suggests that our parser is predisposed to produce more recursive NP structures than is warranted, and is likely a manifestation of the “label bias” problem cited elsewhere in the NLP literature ([3, 11]). The count of other constituent types is more constant across parses, which is to be expected, because the structure of NP constituents is much more variable than that of VPs and PPs.

We performed an evaluation of the agreement between annotators, using two different sets of measures. This evaluation uses the same set of data described in Table 3, for which we have corrections by two annotators.

The first measure of agreement between annotators uses the PARSEVAL metrics ([2]), which are typically used to evaluate the accuracy of parsers (e.g., [6, 8, 17]), but may also be used to evaluate the agreement between annotators in producing a syntactic tree structure ([7, 4]). These metrics include labeled and unlabeled *precision*, *recall*, and *f-measure*, in addition to the number of cross-brackets per sentence. In Table 4, the first tree structure is arbitrarily treated as the “guessed” structure, while the second structure is taken as the “gold” structure, for the purposes of

computing the PARSEVAL metrics.

Table 4 shows that the agreement between annotators is much higher than that between the original parse and either annotator individually, indicating that the annotators are indeed converging on a common gross syntactic structure. The unlabeled f-measure between the two annotators, which is a measure of how well the overall structures of two parse trees correspond, is 0.968, representing a significant ( $p < .001$ ) reduction in the disagreement rate of more than 65% over the unlabeled f-measure between either annotator and the original parse. The number of crossing brackets per sentence also sees a dramatic reduction, from 2.63 to 0.50.

The other metric by means of which we can measure the agreement between annotators in correcting the output of our parser is the leaf-ancestor metric ([18, 19]). This metric was developed in order to address some perceived shortcomings of the PARSEVAL metric in capturing human intuitions about the relative seriousness of differences between parse trees. In a nutshell, the leaf-ancestor metric evaluates the agreement between parse trees on a word-by-word basis, instead of a node-by-node basis. This causes the agreement calculation to be more sensitive to the relative length of constituents.

Table 5 shows the leaf-ancestor agreement figures for the same pairings of parses considered in Table 4, computed as an average value over sentences, tokens, and words (excluding punctuation). The agreement between our annotators is in all cases significantly ( $p < .001$ ) higher than the

Source of Parses	Responses	Sentences	Words	Nodes	NP Nodes	VP Nodes	S Nodes	SBAR Nodes	PP Nodes
parser output	334	964	18646	35386	6155	5755	3218	1517	1470
Annotator B	334	964	18646	35364	5890	5800	3215	1522	1448
Annotator A	334	964	18646	35146	5616	5881	3155	1480	1443

**Table 3. Summary statistics for responses in data used to establish human agreement in parse correction**

agreement between either annotator individually and the original parse, which is consistent with the results derived using the PARSEVAL metrics.

### 4.3 Semantic Role Labeling

The final stage of annotation was the assignment of PropBank semantic role labels to the arguments of selected predicates in each response.

As mentioned above, the responses used for these experiments had been selected to contain specific words we believed to be important predicates for scoring the accuracy of the response, in order to ensure that the evaluation in terms of argument identification is actually relevant to the larger question of scoring accuracy. These important predicates were the ones selected for PropBank annotation.

Constraints on annotator availability limited the total number of responses which could be annotated for syntactic roles, but we were able to collect PropBank annotations for two complete prompts (**CP-2** and **CP-4**). For prompt **CP-2**, we identified three important verbs: *reduce*, *decrease*, and *lose*.<sup>2</sup> For prompt **CP-4**, we identified only one verb: *introduce*.<sup>3</sup> All of the responses from prompts **CP-2** and **CP-4** were double-annotated for PropBank arguments.

For the semantic role annotation undertaken here, there is little difference between requiring matching arguments to have the same constituent

span, and requiring them to have the same head only. Therefore, we report only the head-based measure. However, we do calculate both labeled and unlabeled agreement statistics. Unlabeled agreement may be more meaningful in considering the usefulness of semantic parsing for *c-rater*, since mislabeled arguments might still be usable in scoring, given appropriately defined scoring rules.

Tables 6–7 show the agreement between annotators in labeling PropBank arguments, according to these measures. The rows  $N_A$  and  $N_B$  indicate the number of arguments identified by each annotator.

The summary statistics in the lower right-hand corner of Table 6 show that overall, agreement between annotators is quite high, with an f-measure of 0.95 on prompt **CP-4**, and 0.93 on prompt **CP-2**. However, their agreement is much lower on the infrequent argument classes **ARG2** and **ARG3**, and in labeling **ARG0s** for the verbs *reduce* and *decrease*. It is somewhat understandable that the less frequent argument classes would display less consistency, since the annotators have less experience in dealing with them, and also because they are found only in unusual uses of these verbs. The lower agreement in labeling **ARG0s** for *reduce* and *decrease* may reflect the fact that *decrease* displays an ergative alternation in which surface syntactic relationships do not map simply to semantic roles (*She decreased her selection*  $\longleftrightarrow$  *Her selection decreased*). Students have a tendency to use *reduce* in the same way, although this violates the prescriptive rules of its use.

We consider an overall f-measure in the mid-nineties to be satisfactory for this annotation task, given the level of performance by automatic sys-

<sup>2</sup>Prompt **CP-2** concerns a bookseller who is faced with competition from a chain store. She has decided to remove some bookshelves and install a café. However, this alienated some of her loyal customers. The three verbs are relevant to this prompt because they are used frequently both in descriptions of the store’s inventory (*reduced* selection), and in discussing the results of the remodeling (*losing* customers).

<sup>3</sup>Prompt **CP-4** asks about the discourse function of a particular (introductory) passage in the readings.

tems reported in the literature, and the error analysis we have performed thus far.

## 5 Future work

The current paper has demonstrated that reliable annotation of semantic argument structure is feasible even for noisy texts. Human agreement on editing for grammatical mistakes, correcting automatic syntactic parses, and assigning Prop-Bank role labels is high enough to allow the construction of a useful testbed for further analysis.

Now that such a testbed has been developed the obvious next step in our work is to evaluate machine learning models for semantic role labeling against *c-rater*'s current method of argument identification. The same agreement measures used to compare human annotators in this paper can be used to evaluate the success of each model in identifying the semantic arguments relevant to a student response.

Another important task is the establishment of a set of best practices for the correction of automatic parses. Previous work has established the efficiency gains which this methodology can achieve, and this paper has demonstrated that it can result in high agreement between human annotators. In future, it would be very valuable to investigate factors contributing to annotator agreement in parse correction.

## References

- [1] C. F. Baker, C. J. Fillmore, and J. B. Lowe. The Berkeley FrameNet project. In *COLING/ACL-98*, pages 86–90, 1998.
- [2] E. Black, S. Abney, S. Flickenger, C. Gdaniec, C. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. Procedure for quantitatively comparing the syntactic coverage of English grammars. In *HLT '91: Proceedings of the Workshop on Speech and Natural Language*, pages 306–311, Morristown, NJ, USA, 1991. Association for Computational Linguistics.
- [3] L. Bottou. *Une approche théorique de l'apprentissage connexionniste; applications à la reconnaissance de la parole*. PhD thesis, Université Paris XI, Orsay, France, 1991.
- [4] T. Brants. Inter-annotator agreement for a German newspaper corpus. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC-2000)*, Athens, Greece, 2000.
- [5] X. Carreras and L. Màrques. Introduction to the CoNLL-2004 shared task: Semantic role labeling. In *Proceedings of CoNLL-2004*, pages 89–97. Boston, MA, USA, 2004.
- [6] E. Charniak. A maximum-entropy-inspired parser. In *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL 2000)*, pages 132–139, 2000.
- [7] F.-D. Chiou, D. Chiang, and M. Palmer. Facilitating treebank annotation using a statistical parser. In *HLT '01: Proceedings of the First International Conference on Human Language Technology Research*, pages 1–4, Morristown, NJ, USA, 2001. Association for Computational Linguistics.
- [8] M. Collins. *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania, Philadelphia, PA, 1999.
- [9] D. Gildea and D. Jurafsky. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288, 2002.
- [10] P. Kingsbury and M. Palmer. From Treebank to PropBank. In *Proceedings of LREC-2002*, Las Palmas, Canary Islands, Spain, 2002.
- [11] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.
- [12] C. Leacock and M. Chodorow. C-rater: Scoring of short-answer questions. *Computers and the Humanities*, 37(4):389–405, 2003.
- [13] K. Litkowski. Senseval-3 task: Automatic labeling of semantic roles. In R. Mihalcea and P. Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 9–12, Barcelona, Spain, July 2004. Association for Computational Linguistics.
- [14] M. Palmer, D. Gildea, and P. Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Comput. Linguist.*, 31(1):71–106, 2005.
- [15] S.-Y. Park, Y. Cho, S. Son, U.-S. Song, and H.-C. Rim. Tree annotation tool using two-phase parsing to reduce manual effort for building a treebank. In *Proceedings of The Second International Joint Conference on Natural Language Processing (IJCNLP-05)*, Jeju Island, Korea, 2005.
- [16] S. Pradhan, W. Ward, K. Hacioglu, J. Martin, and D. Jurafsky. Shallow semantic parsing using support vector machines. In *Proceedings of the Human Language Technology Conference/North American Chapter of the Association of Computational Linguistics (HLT/NAACL)*, Boston, MA, 2004.
- [17] A. Ratnaparkhi. Learning to parse natural language with maximum entropy models. *Machine Learning*, 34:151–175, 1999.
- [18] G. Sampson. A proposal for improving the measurement of parse accuracy. *International Journal of Corpus Linguistics*, 5:53–68, 2000.
- [19] G. Sampson and A. Babarczy. A test of the leaf-ancestor metric for parse accuracy. *Natural Language Engineering*, 9(4):365–380, 2003.

Comparison	Labeled			Unlabeled			X-Brackets Per Sentence
	P	R	F	P	R	F	
Annotator A vs. parser	0.842	0.835	0.838	0.905	0.896	0.900	2.62
Annotator B vs. parser	0.857	0.855	0.856	0.908	0.906	0.907	2.63
Annotator A vs. Annotator B	0.916	0.911	0.913	0.972	0.965	0.968	0.50
Annotator B vs. Annotator A	0.911	0.916	0.913	0.965	0.972	0.968	0.50

**Table 4. Agreement between annotators in correcting parses, measured according to PARSEVAL metrics**

Comparison	Average Sentence Score	Average Token Score	Average Word Score
parser vs. Annotator A	0.914	0.898	0.896
parser vs. Annotator B	0.920	0.903	0.901
Annotator A vs. Annotator B	0.959	0.957	0.955

**Table 5. Agreement between annotators in correcting parses, measured according to leaf-ancestor metric**

		CP-2 <i>reduce</i>	CP-2 <i>decrease</i>	CP-2 <i>lose</i>	CP-2 ALL	CP-4 <i>introduce</i>
<b>ARG0</b>	$N_A$	28	26	178	233	121
	$N_B$	25	24	178	228	101
	Precision	0.86	0.73	0.94	0.91	0.83
	Recall	0.96	0.79	0.97	0.95	1.00
	F-measure	0.91	0.76	0.95	0.93	0.91
<b>ARG1</b>	$N_A$	45	37	195	278	130
	$N_B$	46	39	194	280	130
	Precision	0.98	1.00	0.96	0.97	0.97
	Recall	0.96	0.95	0.97	0.96	0.97
	F-measure	0.97	0.97	0.97	0.97	0.97
<b>ARG2</b>	$N_A$	13	10	0	23	15
	$N_B$	5	1	1	7	15
	Precision	0.38	0.00	0.00	0.22	1.00
	Recall	1.00	0.00	0.00	0.71	1.00
	F-measure	0.56	0.00	0.00	0.33	1.00
<b>ARG3</b>	$N_A$	–	1	–	1	–
	$N_B$	–	0	–	0	–
	Precision	–	0.00	–	0.00	–
	Recall	–	0.00	–	0.00	–
	F-measure	–	0.00	–	0.00	–
<b>All ARGs</b>	$N_A$	86	74	373	535	266
	$N_B$	76	64	373	515	246
	Precision	0.85	0.76	0.96	0.91	0.91
	Recall	0.96	0.88	0.96	0.95	0.98
	F-measure	0.90	0.81	0.96	0.93	0.95

**Table 6. Agreement between annotators in identifying labeled PropBank arguments (common head)**

		CP-2 <i>reduce</i>	CP-2 <i>decrease</i>	CP-2 <i>lose</i>	CP-2 ALL	CP-4 <i>introduce</i>
<b>All ARGs</b>	$N_A$	86	74	373	535	266
	$N_B$	76	64	373	515	246
	Precision	0.85	0.77	0.95	0.91	0.91
	Recall	0.96	0.89	0.96	0.96	0.98
	F-measure	0.90	0.83	0.96	0.93	0.95

**Table 7. Agreement between annotators in identifying unlabeled PropBank arguments (common head)**