COMPARING HUMAN AND MACHINE PERFORMANCE FOR NATURAL LANGUAGE INFORMATION EXTRACTION: Results from the Tipster Text Evaluation

Craig A. Will

Institute for Defense Analyses
Computer and Software Engineering Division
1801 N. Beauregard Street
Alexandria, VA 22311

ABSTRACT

This paper presents results from a study comparing human performance on the text of natural language information extraction with that of machine extraction systems that were developed as part of the ARPA Tipster program. Information extraction is shown to be a difficult task for both humans and machines. Evidence for one set of text material, English Microelectronics, indicated that a human analyst produces about half the errors as does machine systems.

INTRODUCTION

In evaluating the state of technology for extracting information from natural language text by machine, it is valuable to compare the performance of machine extraction systems with that achieved by humans performing the same task. The purpose of this paper is to present some results from a comparative study of human and machine performance for one of the information extraction tasks used in the Tipster/MUC-5 evaluation that can help assess the maturity and applicability of the technology.

The Tipster program, through the Institute for Defense Analyses (IDA) and several collaborating U.S. government agencies, produced a corpus of filled "templates" —structured information extracted from text. This corpus was used both in the development of machine extraction systems by contractors and in the evaluation of the developed systems. Production of templates was performed by human analysts extracting the data from the text and structuring it, using a set of structuring rules for "filling" the templates and computer software that made it easier for analysts to organize information. Because of this rather extensive effort by analysts to create these templates, it was possible to study the performance of humans for this task in some detail and to develop methods for comparing this performance with that of machines participating in the Tipster/MUC-5 evaluation.

The texts that the templates were filled from were newspaper and technical magazine articles concerned either with joint business ventures or microelectronics fabrication technology. Each topic domain used text in two languages, English and Japanese. This paper discusses preparation of templates and presents detailed results for human and machine performance; a shorter paper [1] discusses preparation of templates and basic results.

The primary motivation for this study was to provide reliable data that would allow machine extraction performance to be compared with that of humans. The MUC and Tipster programs have included extensive efforts to develop measurements that can objectively evaluate the performance of the different machine systems. However, although these measures are capable of reliably discriminating between the performance of different machine systems, they are not very useful by themselves in evaluating how near the technology is to providing reliable performance and the extent to which it is ready to be used in applications. Sundheim [2] initiated human performance study for extraction by providing estimates of human performance for the task used in the MUC-4 evaluation; the present study provides human data for the Tipster/MUC-5 evaluation that was produced under relatively controlled conditions and with methods and statistical measures that assess the reliability of the data.

A second motivation for the study was for its value in helping produce better quality templates so as to allow high-quality system development and reliable evaluation. The quality and consistency of the templates being produced were monitored as analysts were trained and gained experience, and particular efforts were made to identify the causes of errors and inconsistency so as to develop strategies for reducing error and increasing consistency.

A third motivation for studying human performance was to better understand the nature of the extraction task and the relative performance of humans compared with machines on different aspects of the task. Such an understanding can particularly help in the construction of human-machine integrated systems that are designed to make the best use of

what are at the present time rather different abilities of humans and machines [3].

This paper is organized as follows:

The paper begins with a discussion of how the templates were prepared, with particular emphasis on the strategies that were used that served to minimize errors and maximize consistency, including detailed fill rules, having more than one analyst code a given template, and the use of software tools with error detection capabilities.

The paper next describes the results of an investigation into the extent to which template codings made by analysts that are playing different roles in the production of a particular template influence the resulting key, which provides clues to the effectiveness of the quality control strategies used in the template preparation process.

The results of an experimental test of different methods of scoring human performance are then presented, with the goal of selecting a method that is statistically reliable, minimizes bias, and has other desirable characteristics. Data that indicates overall levels of human performance on the task, variability among analysts, and reliability of the data are then presented.

The results of an investigation into the development of analyst skill are then presented, with the significant question being the need to understand whether the performance levels being measured truly reflect analysts who have a high level of skill.

The performance of humans for information extraction is then compared with that of machine systems, in terms of both errors and metrics that attempt to separate out two different aspects of performance, *recall* and *precision*.

The results of a study comparing the effect of key preparation on the evaluation of machine performance are then presented. This is particularly relevant to the question of how keys should be future MUC and Tipster evaluations.

A study is then presented of the extent to which machines and humans agree on the relative difficulty of particular templates.

The results of a pilot study in which the performance of humans and machines is compared for particular kinds of information, to see what information machines are comparatively worse or better than humans in extracting, is then presented.

A final section of the paper makes some general conclusions about the results and their implications for assessing the maturity and applicability of extraction technology.

THE PREPARATION OF TEMPLATES

The development of templates for the English Microelectronics corpus began in the fall of 1992. It began with an interagency committee that developed the basic template structure and, when it had evolved to the point that it was relatively stable, two experienced analysts were added to the project so that they could begin training toward eventual production work. About two months after that, two more production analysts joined the project.

The template structure was object-oriented, with a template consisting of a number of *objects*, or template building blocks with related information. Each object consisted of *slots*, which could be either fields containing specific information or *pointers*, that is, references, to other objects. Slot fields can either be *set fills*, which are filled by one or more categorical choices defined for that field, or by *string fills*, in which text is copied verbatim from the original article. In cases of ambiguity, analysts could provide *alternative fills* for a given slot. In addition, analysts could include *comments* when desired to note unusual problems or explain why a particular coding was selected. Comments were not scored and were primarily used when analysts compared two or more codings of a given article to determine which was the more correct. For more information on the template design see [4]. Also see [5] for a discussion of selection of the articles in the corpora and preparation of the data, and [6] for a discussion of the different extraction tasks, domains, and languages.

Previous experience with the English and Japanese Joint Ventures corpus had made it clear that producing templates with a high degree of quality and consistency is a difficult and time-consuming task, and we attempted to make the best use of what had been learned in that effort in producing templates for English Microelectronics with quality and consistency appropriate to both the needs of the project and the resources we had available.

"Quality" refers to minimizing the level of actual error by each analyst. "Error" includes the following: (1) Analysts missing information contained in or erroneously interpreting the meaning of an article; (2) Analysts forgetting or misapplying a fill rule; (3) Analysts misspelling a word or making a keyboarding (typographical) error or the analogous error with a mouse; and (4) Analysts making an error in constructing the object-oriented structure, such as failing to create an

object, failing to reference an object, providing an incorrect reference to an object, or creating an extraneous object.

"Consistency" refers to minimizing the level of legitimate analytical differences among different analysts. "Legitimate analytical differences" include the following: (1) Different interpretations of ambiguous language in an article; (2) Differences in the extent to which analysts were able or willing to infer information from the article that is not directly stated; and (3) Different interpretations of a fill rule and how it should be applied (or the ability or willingness to infer a rule if no rule obviously applies).

To improve quality and consistency, three steps were taken:

Development of Fill Rules

First, a set of relatively detailed rules for extracting information from articles and structuring it as an object-oriented template was developed (with the rules for English Microelectronics a 40-page, single-spaced document). These rules were created by a group of analysts who met periodically to discuss problems and to agree on how to handle particular cases via a general rule. One person (who was not one of the production analysts) served as the primary person maintaining the rules. Because of the highly technical nature of the topic domain, an expert in microelectronics fabrication also attended the meetings and resolved many problems that required technical knowledge.

Coding by Multiple Analysts

The second step was the development of a procedure in which two analysts participated in coding nearly half of the articles, and the reconciliation of different codings to produce final versions. For 300 articles in a "high quality" development set and for the 300 articles in the test set, the following procedure was followed: Two analysts first independently coded each article, with the resulting codings provided to one of these same analysts, who produced a final version, or "key". The remaining 700 development templates were coded by only one analyst, with each of four analysts coding some portion of the 700 articles. The purpose of the two-analyst procedure was to correct inadvertent errors in the initial coding and to promote consistency, by allowing the final analyst to change his or her coding after seeing an independent coding by a different analyst. The procedure also promoted consistency in the long run by providing analysts with examples of codings made by other analysts so that they could see how other analysts handled a given problem. It also helped improve the fill rules by allowing analysts to detect recurring problems that could be discussed at a meeting and lead to a change in the fill rules.

Software Support Tools

The third step was the development of software tools that helped analysts to minimize errors, detect certain kinds of errors, and support the process of comparing initial codings. One such tool was the template-filling tool developed by Bill Ogden and Jim Cowie at New Mexico State University (known as Locke in the version designed for English Microelectronics). This tool, which runs on a Sun workstation and uses the Xwindows graphical user interface, provided an interface that allowed analysts to easily visualize the relationships among objects and thus avoid errors in linking objects together. The tool also allowed analysts to copy text from the original article by selecting it with a mouse, entering it verbatim into a template slot, thus eliminating keystroke errors. In addition, the Locke tool has checking facilities that allowed analysts to detect such problems as unreferenced or missing objects. A second tool was the Tipster scoring program (developed by Nancy Chinchor and Gary Dunca at SAIC [8]) which provided analysts making keys with a printout of possible errors and differences between the initial codings. Another program, written by Gerry Reno at the Department of Defense at Fort Meade, did final checking of the syntax of completed keys.

The four analysts who coded templates all had substantial experience as analysts for U.S. government agencies. In all cases analysts making the keys were unaware of the identity of the analyst producing a particular coding. Analysts did often claim that they could often identify the analyst coding a particular article by the comments included in the template coding or the number of alternative fills added, although when this was investigated further it appeared that they were not necessarily correct in their identification.

In addition to the templates and keys created for the development and test sets described above, a small number of codings and keys were made for the purpose of studying human performance on the extraction task. In February, 1993, at about the time of the 18-month Tipster evaluation, a set of 40 templates in the development set were coded by all analysts for this purpose. Similarly, for 120 templates of the 300-template test set that was coded in June and July, 1993 extra codings were made by the two analysts that would have not normally participated in coding those articles, resulting in codings by all 4 analysts for 120 articles.

INFLUENCE OF ANALYSTS PLAYING DIFFERENT ROLES ON KEY

The two-analyst procedure for making keys used for English Microelectronics was intended as an efficient compromise between the extremes of using a single analyst and the procedure that had been used for English Joint Ventures in which two analysts would independently make codings and provide the codings to a third analyst who would make a key.

It is of interest to know whether this form of checking is achieving its intended result—that of improving the quality and consistency of templates. We can investigate this indirectly by measuring the level of influence analysts playing different roles (and producing different codings) have on the key that is produced. The question of influence is also of interest—as will be seen in the next section—for its implications in understanding the characteristics of different methods for measuring human performance.

To investigate this influence, data from the set of 120 templates that all analysts coded were analyzed separately based on the role in the production of the key played by the particular coding. Figure 1 shows the relationship between different codings and the analysts producing them. Analyst 1 produces, based on the original article, what will be called here the *primary* coding of the article. Analyst 2 produces independently the *secondary* coding of the article. The primary and secondary codings are then provided to Analyst 1, who produces the key. Analysts 3 and 4 also produce other codings of the article that have no effect on the key. Each analyst plays a particular role (Analyst 1, 2, 3, or 4) for 30 of the 120 templates.

Note that when Analyst 1 uses the primary and secondary codings in making the final coding, or key, there is a natural bias toward the primary coding. This is primarily because Analyst 1 created that coding, but also because the analyst typically does not create the key from scratch with the Locke tool, but modifies the primary coding (probably reinforcing the tendency to use the primary coding unless there is a significant reason for changing it).

Figure 2 shows the results of this analysis, with performance expressed in terms of error per response fill, as calculated with the methodology described by Nancy Chinchor and Beth Sundheim [7] and implemented by the SAIC scoring program [8]. All objects in the template were scored, and scoring was done in "key to key" mode, meaning that both codings were allowed to contain alternatives for each slot.

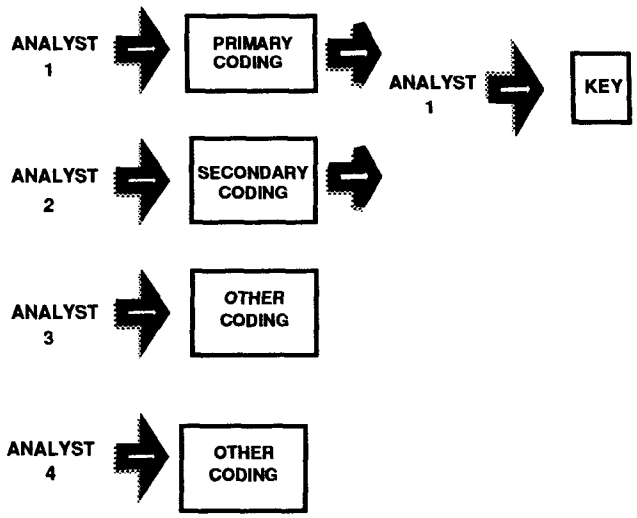

Figure 1: Procedure for Coding Templates and Making Keys for the 120 Articles Coded by All Analysts

(See the Appendix for details of producing the error scores and calculation of statistical parameters and tests.)

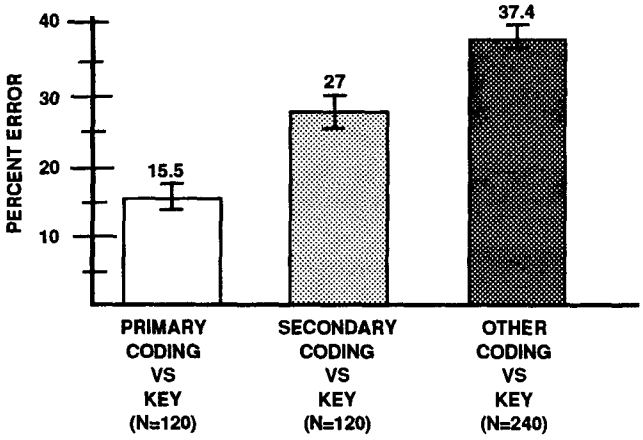

Figure 2: Error for Coding Compared with Key Depending upon Role of Coding in Making Key

The data is shown for three conditions, with each condition reflecting the accuracy, in terms of percent error, of a coding playing a particular role (or no role) in producing a key. The conditions are: (1) error for the primary coding when measured against the key (shown as an unfilled vertical bar); (2) error for the secondary coding when measured against the key (shown as a light gray vertical bar); and (3) error for other codings when measured against the key (shown as a dark gray vertical bar).

Also shown for all conditions in the form of error bars is the standard error of the mean. Because the mean shown is cal-

culated from a small sample it can be different from the desired true population mean, with the sample mean only the most likely value of the population mean. The standard error bars show the range expected for the true population mean. That mean can be expected to be within the error bars shown 68% of the time. See the appendix for details about how the standard error of the mean was calculated.

Figure 3 makes clear the role of different codings: the primary coding is made by the analyst who also later made the key, while the secondary and other codings were made by other analysts. The primary and secondary codings are both used in making the key, while the other codings are not used.

	PRIMARY	SECONDARY	OTHER
MADE BY	Analyst who Also Later Made Key	Other Analyst	Other Analyst
USED IN	Used in Making Key	Used in Making Key	Not Used in Making Key

Figure 3: Characteristics of Different Coding Roles

The result (in Figure 2) that the primary coding when compared to the key shows a mean error considerably above zero indicates that analysts quite substantially change their coding from their initial version in producing the key. Presumably, this results from the analyst finding errors or more-desirable ways of coding, and means that quality and consistency is improved in the final version. (All differences claimed here are statistically significant—see Appendix for details).

The result that the secondary coding when compared to the key shows a mean error that is substantially above that of the primary coding condition indicates that the analyst's original (primary) coding does in fact influence the key more strongly than does the secondary coding (produced by another analyst). At the same time, it is clear that the secondary coding does itself substantially influence the key, since the mean error for the secondary coding is substantially less than that for the "other" codings, which are not provided to the analyst making the key and thus have no influence on it. This provides good evidence that analysts are indeed making use of the information in the secondary coding to a substantial extent. This probably resulted in an improvement in both quality and consistency of the templates above what would be the case if only a single coder (even with repeated checking) was used, although we do not have direct evidence of such improvement and the extent of its magnitude is not clear.

METHODS FOR SCORING HUMAN PERFORMANCE

Before human performance for information extraction can be effectively compared with machine performance, it is necessary to develop a method for scoring responses by human analysts.

The problem of measuring machine performance has been solved in the case of the MUC-5 and Tipster evaluations by providing (1) high-quality answer keys produced in the manner described in the previous section; and (2) a scoring methodology and associated computer program.

The primary additional problem posed when attempting to measure the performance of humans performing extraction

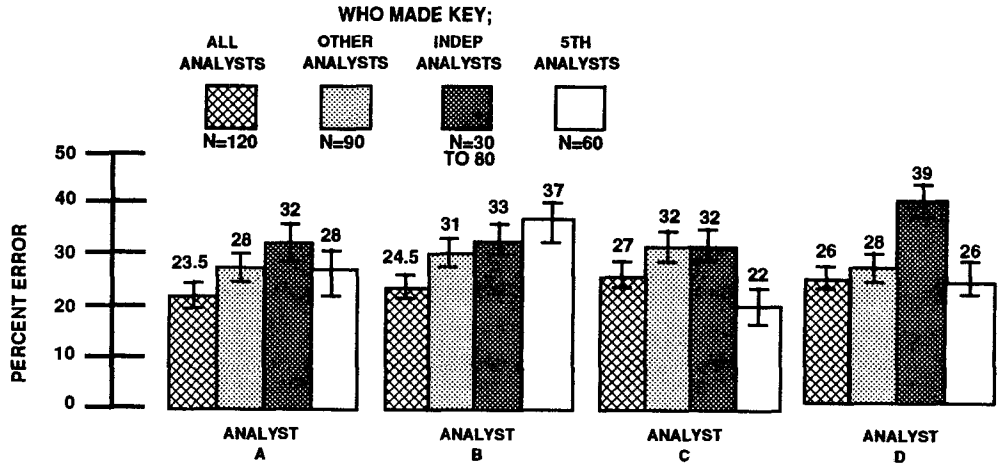

Figure 4: Comparison of Methods for Scoring Human Performance

is that of "who decides what the correct answer is?" In the case of the English Microelectronics analysts, the four analysts whose performance we are attempting to measure became—once they had substantial training and practice—the primary source of expertise about the task, with their knowledge and skill often outstripping that of others who were supervising and advising them. This made it especially difficult to measure the performance of particular analysts.

We approached the problem of determining the best method for scoring humans empirically: We compared experimentally four different methods for scoring codings by human analysts. In general, the criteria is objectivity, statistical reliability, and a perhaps difficult-to-define "fairness" or plausibility of making appropriate comparisons, both between different human analysts and between humans and machines.

In evaluating different scoring methods, the 120 templates in the Tipster/MUC-5 test set that had been coded by all four analysts were used. As was described in the previous section, keys for each template in this set were made by one of the analysts, using as inputs codings done independently by the analyst making the key and one other analyst. Each of the 4 analysts made keys for 30 of the 120 templates and also served as the independent analyst providing a coding to the analyst making the keys for a different set of 30 templates. In addition, for 60 of the 120 templates, a fifth analyst made a second key from codings made by the four analysts.

Figure 4 shows data comparing the four scoring methods for each of the four analysts. The data is shown in terms of percent error, with all objects scored, and with "key to response" scoring being used. In key to response scoring, alternative fills for slots are allowed only in the key, but not in the coding being scored. Since in the data collected here, analysts did normally add alternative fills (since their goal was to make keys), these alternatives were removed before scoring, with the first alternative listed assumed to be the most likely one and thus kept, and others deleted. The purpose of using key-to-response scoring was so that the resulting data could be directly compared with data from machine systems, which produced only one fill for each slot. Scoring was done in batch mode, meaning that human analysts were not used (as they are in interactive mode) to judge cases in which strings did not completely match.

In the "All Analysts" condition, all 120 templates made by each analyst were scored, using as keys those made by all 4 analysts (including the analyst being scored). In the "Other Analysts" condition, only templates that have keys made by analysts other than the analyst being scored were used in scoring each analyst (with each analyst having 90 templates of the 120 templates coded by that analyst scored). In the "Independent Analysts" condition, only templates for which the analyst being scored neither made the key nor produced a coding that was used as an input for making the key were used in scoring each analyst. (This resulted in from 30 to 80 templates being scored for each analyst, depending upon the analyst.) In the "5th Analyst" condition, a 5th analyst made the answer keys (with 60 templates scored in this condition). This 5th analyst did not code production templates but was in charge of maintaining the fill rules and the overall management of the English Microelectronics template coding effort.

The "All Analysts" condition showed the most consistent performance across analysts, with a variance calculated from the means for each analyst of 1.82 (N=4). The "Other Analysts" condition was nearly as consistent, with a variance of 3.16. The "Independent Analysts" and "5th Analyst" conditions were much less consistent, with variances of 9.08 and 30.19, respectively. The high variance of the "Independent Analysts" condition, however, resulted only from the performance of analyst D, who had a very small sample size, only 30 templates. If analyst D is left out, the variance becomes only 0.32 for this condition. The high variability across analysts for the 5th analyst could be a result either of the small sample size or, more likely, a tendency for the 5th analyst to code articles in a manner more similar to some analysts (especially analyst C) than others (especially analyst B).

The subjective opinions of the analysts and their co-workers suggested that all English Microelectronics analysts here were generally at the same level of skill, which is consistent with the above data. (This was not true of the English Joint Venture analysts, for example, where both the data and the opinions of analysts and others suggested considerable variability of skill among analysts.) However, it should be noted that all of the conditions in which analysts are being scored by other analysts run the risk of making the differences among analysts artificially low. Consider, for example, the case of a very skilled analyst being scored against a key made by an analyst who is poorly skilled. The more skilled analyst is likely to have some correct responses scored incorrectly, while a less-skilled analyst may have his or her incorrect responses scored as correct. However, the patterns of analyst skill elicited by the different scoring methods do not show any reliable evidence of such differences in skill, and it appears that analysts have similar levels of skill and that any effect of a "regression toward the mean" of mean analyst scores is minimal.

Figure 5 shows the same data comparing scoring methods that was shown in the previous figure, but in this figure data has been combined to show means for all analysts in each of the scoring conditions. This combining allows the overall differences among the different scoring methods to be seen

more clearly. In addition, combining the data in this way increases reliability of the overall mean.

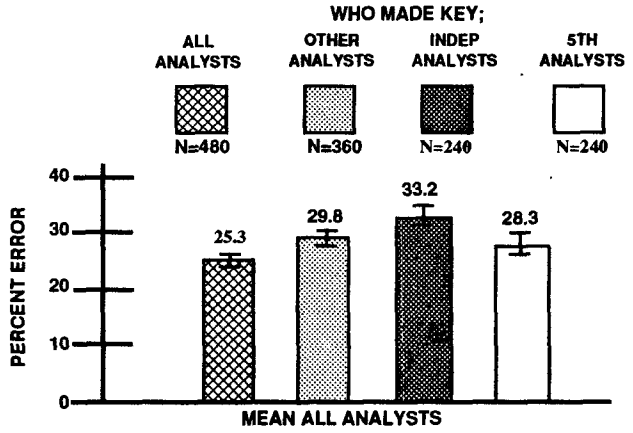

Figure 5: Comparison of Scoring Methods Using Mean of All Analysts

Figure 6 shows a summary of the characteristics of different scoring methods as discussed above. The "All Analysts", "Other Analysts", and "Independent Analysts" methods all use the expertise of the most practiced (production) analysts.

SCORING METHOD	EXPERTISE OF ANALYST MAKING KEY	BIAS DUE TO SELF-SCORING	BIAS TOWARD CERTAIN ANALYSTS	STATISTICAL RELIABILITY
ALL ANALYSTS	HIGH	SUBSTANTIAL	NONE	HIGH
OTHER ANALYSTS	HIGH	SOME	NONE	HIGH
INDEP. ANALYSTS	HIGH	NONE	NONE	HIGH
5TH ANALYST	MODERATE TO HIGH	NONE	PROBABLY SOME	HIGH

Figure 6: Characteristics of Scoring Methods

To make the key, while the "5th analyst" method uses an analyst the expertise of which is somewhat more questionable because of putting much less time into actual coding articles. The "All Analysts" method appeared to have substantial bias (producing artificially low error scores) from analysts scoring their own codings, while the "Other Analysts" method appeared to produce some (but less) such bias. Neither the "Independent Analysts" nor the "5th Analyst" method suffered from this kind of bias. The "All Analysts", "Other Analysts", and "Independent Analysts" methods are unbiased with respect to particular analysts (because of counterbalancing to control for this), but the "5th Analyst" method appeared to have some bias, presumably because of a coding style more similar to some analysts than others. Finally, the "All Analysts", "Other Analysts", and "Independent Analysts" methods had relatively high statistical reliability, while the "5th analyst" method had much less reliability.

Figure 7 shows a Recall-Precision scatterplot for the four analysts and for each of the four conditions shown in Figure 4. Analysts scored by the "All Analysts" method are shown as solid circles, while analysts scored by the "Other Analysts" method are shown as solid triangles. Analysts scored by the "Independent Analysts" method are shown as deltas, and analysts scored by the "5th Analyst" method are shown as solid squares. Note that only the upper right-hand quadrant of the usual 0-100% recall-precision graph is shown. Performance is expressed in terms of *recall* and *precision*, which are measures borrowed from information retrieval that allow assessment of two independent aspects of performance. Recall is a measure of the extent to which all relevant information in an article has been extracted, while precision is a measure of the extent to which information that has been entered into a template is correct. Details of the method for calculating recall-precision scores for the Tipster/MUC-5 evaluation can be found in the paper by Chinchor and Sundheim [7].

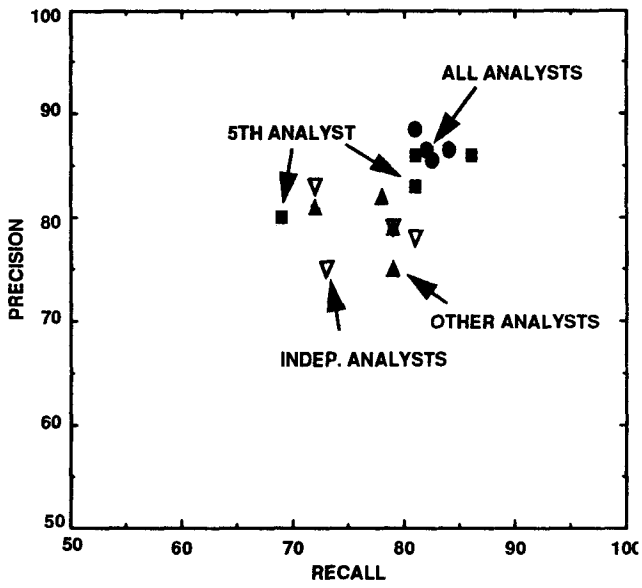

Figure 7: Recall Versus Precision Scores for Human Analysts Scored by Different Methods

THE DEVELOPMENT OF ANALYST SKILL

In interpreting the levels of performance shown by analysts for the extraction task, and, particularly, when comparing human performance with that of machines, it is important to know how skilled the analysts are compared to how they might be with additional training and practice. Comparing machine performance with humans who are less than fully skilled would result in overstating the comparative performance of the machines.

Four analysts were used in production template coding, all having had experience as professional analysts. One analyst had about 6 years of such experience, another 9 years of experience, a third 10 years of experience, and the fourth about 30 years of experience. All were native speakers of English. None of the analysts had any expertise in microelectronics fabrication.

We compared the skill of analysts at two different stages in their development by analyzing two sets of templates, each coded at a different time. The first, or "18 month" set, was coded in early February, 1993, at about the same time as the 18 month Tipster machine evaluation, after analysts had been doing production coding for about 3 months. The second, or "24 month" set was coded in June and July, 1993, somewhat before the 24 month Tipster/MUC-5 machine evaluation, and toward the end of the template coding process, when fill rules were at their most developed stage and analysts at their highest level of skill. There was some difference in expected skill between the two pairs of analysts, since one pair (analysts A and B) had begun work in September, although their initial work primarily involved coding of templates on paper and attending meetings to discuss the template design and fill rules, and during this period they did not code large numbers of templates. The second pair began work in November, and did not begin production coding of templates until a few weeks after the first analysts.

Data for the 18 month condition was produced by first having all analysts code all articles of a 40-article set in the development set. Each analyst was then provided with printouts of all 4 codings for a set of 10 articles, and asked to make keys for those articles. Data for the 24 month condition was produced as described previously for the "All Analysts" condition, using the 120 templates that all analysts coded in the test set, with each of the 4 analysts making keys for 30 of the 120 templates. Note that analysts making the keys in the 18 month condition used as inputs the codings of all 4 analysts, while analysts making the keys in the 24 month condition used as inputs the codings of only 2 analysts. In both conditions and for all analysts, "key to key" scoring was used, in which all alternatives in both codings were used in determining the differences between template codings.

Figure 8 shows data, in terms of percent error, for each of the two pairs of analysts in both the 18 month and 24 month conditions. The pairing of analysts is based on when they started work, with analysts A and B ("Early Starting Analysts") beginning work on the project before analysts C and D ("Later Starting Analysts"). Note that analysts who started early appeared to make slightly fewer errors in the 18 month condition (27%) than in the 24 month condition (28.3%), although the difference is not statistically significant.

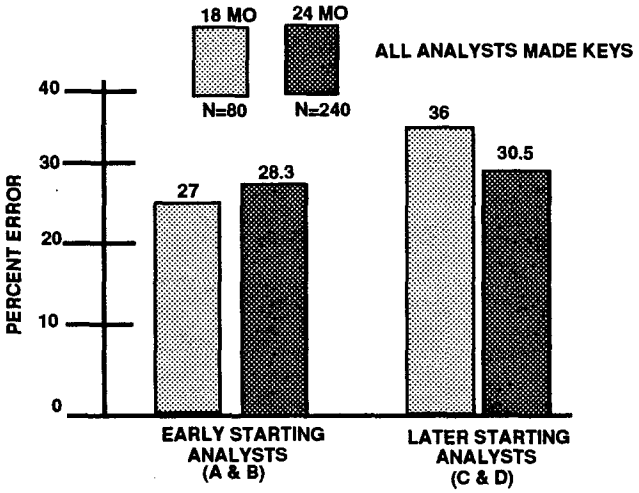

Figure 8: Performance of Analysts Who Stated Early or Later

This difference can be explained at least in part by the difference in the method of making the keys. In the 18 month condition, all 4 analyst codings influenced the key, while in the 24 month condition only 2 of the analyst codings influenced the key. This results in the 18 month condition producing scores that are artificially low in terms of errors, compared to the 24 month condition. The difference, based on the data in Figure 5, can be estimated at from about 4 to 8 percentage points. Thus, it appears that analysts who started early did not improve their skill, or improved it minimally, between the 18 and 24 month tests. However, analysts who started later did appear to learn substantially, with error scores of 36% in the 18 month condition and 30.5% in the 24 month condition, with the amount of learning for the analysts who started later probably somewhat more than shown because of the difference in the method of making keys. Because of the differences in scoring methods between the conditions and the small sample size in the 28 month condition, the above results are only suggestive. An alternative approach to assessing the development of skill of analysts (that does not require keys for scoring) compares the pattern of disagree-

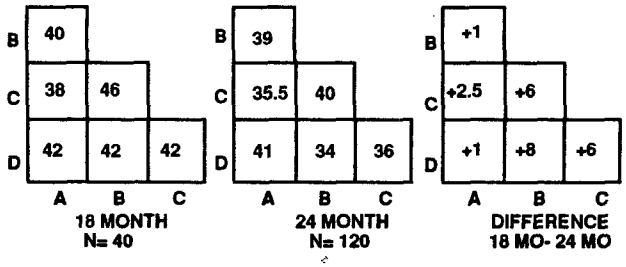

Figure 9: Disagreement Matrices for 18 Month and 24 Month Tipster Tests and Differences Between the Two Matrices

ment among analysts for the 18-month and 24-month tests, and is more convincing. Such a pattern can be constructed by running the scoring program for a given set of templates in "key to key" mode (so that it calculates a measure of disagreement between two codings) for all pairs of codings for the four analysts.

Figure 9 shows such patterns, termed here "Disagreement Matrices", for the 18- and 24-month tests, along with a third "Difference" matrix (shown at the far right) created by subtracting scores in the 24-month matrix from those in the 18-month one, resulting in a measure of the extent to which consistency between particular analysts has improved. Note that all cells of the Difference matrix have positive scores, indicating that consistency between all pairs of analysts has increased.

Figure 10 shows comparisons of the scores from the Difference matrix for three specific cases of analyst pairs. For the Early Starting Analysts pair (A and B), shown at the far left, consistency between the two analysts increased by only one percentage point, suggesting that even at the 18-month test, these analysts were already near their maximum level of skill. For the Later Starting Analysts (C and D), however, shown at the far right, consistency between the two analysts increased by 6 percentage points, indicating that these analysts were still developing their skill. For the case where the mean of all pairs of early-late analysts (AC, AD, BC, and BD) is calculated, shown as the middle vertical bar, consistency increased by an average of 4.375 percentage points, indicating that the less-skilled analysts had increased their consistency with the more-skilled analysts.

The general finding here is that (1) the analysts who started earlier improved their skill minimally from the 18 to 24 month tests; and (2) analysts who started later improved their skill considerably. Because by the time of the 24 month test the later starting analysts had as much or more practice coding templates as did the early starting analysts at the time of the 18 month test, it is reasonable to assume that their

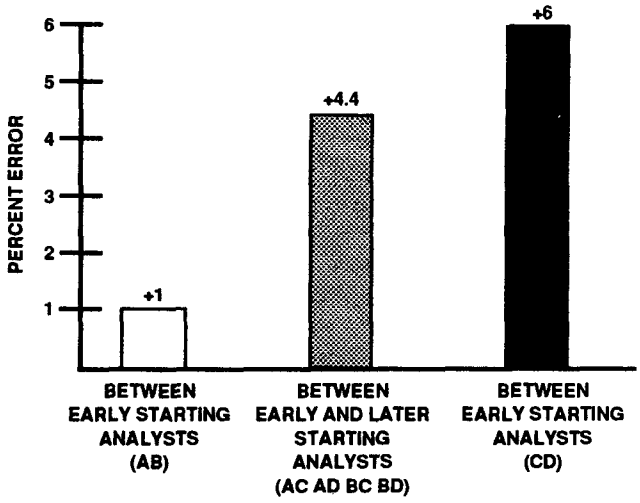

Figure 10: Improvement in Consistency From 18-Month to 24-Month Tests

increase in skill reflects an early part of the learning curve and that by the 24 month test all analysts have started to reach an asymptotic level of skill.

The evidence in the literature for the development of skill in humans suggests that skill continues to develop, if slowly, for years or decades even on simple tasks, and it can be expected that continued practice on information extraction by these analysts would increase their level of skill. However, it does appear that the analysts were very highly skilled by the end of the study and were at an appropriate level of skill for comparison with machine performance.

COMPARISON OF HUMAN AND MACHINE PERFORMANCE

The most critical question in the Tipster/MUC-5 evaluation is that of how performance of the machine extraction systems compares with that of humans performing the same task.

Figure 11 shows mean performance, in percent error, for the 4 human analysts, using the "Independent Analysts" condition discussed in a previous section and shown in Figure 5, for the 120 articles coded by all analysts from the English Microelectronics test set. Also shown is the corresponding machine performance for the same articles for the three machine systems in the Tipster/MUC-5 evaluation that had the best official scores for English Microelectronics.

The differences are very clear, with the mean error for human analysts about half that of the machine scores. Both

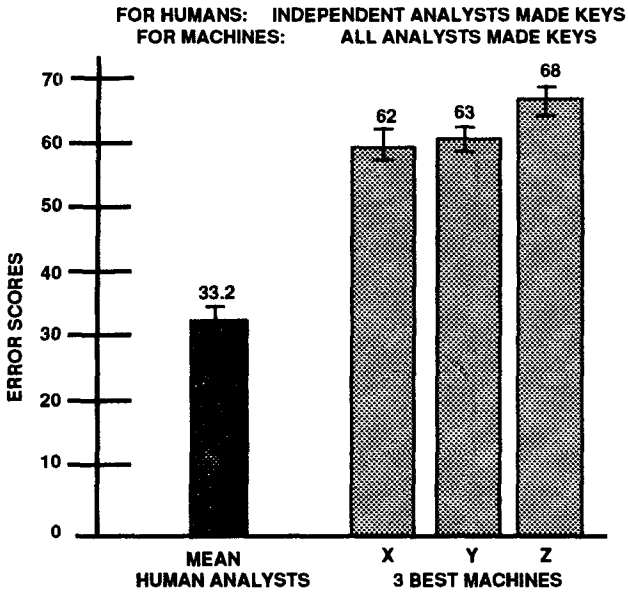

Figure 11: Comparison of Human and Machine Performance

the human and machine scores are highly reliable, as is shown by the standard error bars.

Figure 12 shows essentially the same data expressed in terms of recall and precision.

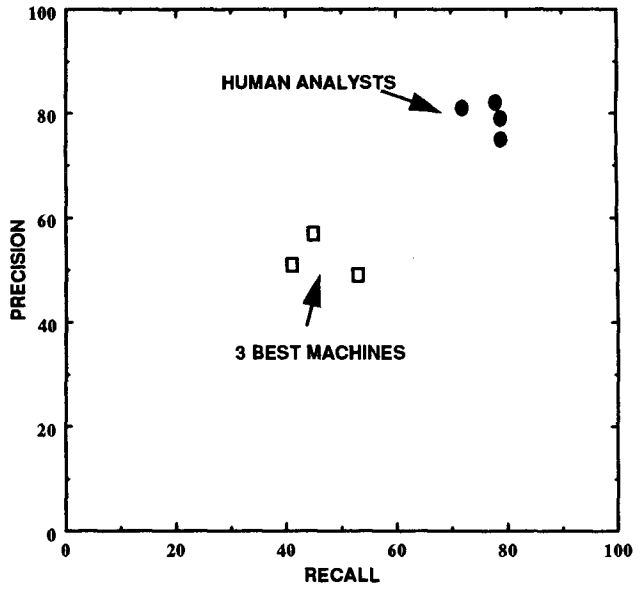

Figure 12: Comparison of Human and Machine Using Recall and Precision Scores

What is surprising about this data is not that the machines have a seemingly rather high error rate, but that the human rate is so high. The recall-precision diagram suggests that machines can have even more similar performance to humans on either recall or precision, if one is willing to trade of the other to achieve it. Machine performance is likely to be at least somewhat better than this in a real system, since resource constraints forced developers to run incomplete systems (that, for example, did not fill in slots for which information was infrequently encountered).

The performance data shown in the figure, other data, and the subjective accounts of individual analysts and their co-workers support the general conclusion that for this group of analysts the level of skill for information extraction was very similar for each analyst. This uses the "Other Analysts" scoring method, with recall and precision scores for individual analysts not particularly meaningful for the otherwise more reliable "Independent Analysts" condition. (See Figure 7 for recall and precision scores for all scoring conditions).

EFFECT OF METHOD OF KEY PREPARATION ON MACHINE PERFORMANCE

A practical consideration in evaluating machine performance of importance for future evaluations (such as MUC-6) is the extent to which it is necessary or desirable to use elaborate checking schemes to prepare test templates, or

whether templates prepared by a single analyst will serve as well.

In an attempt to provide some data relevant to this issue, the performance of the three best machine systems was measured using two different sets of keys. In one condition ("Normal key") the keys used for evaluating the machines were those normally used in the 24 month evaluation, for the 120-article set for which templates were coded by all analysts and a checked version produced by a particular analyst using codings of multiple analysts. In the other condition ("Orig. Coding"), the keys used for evaluating the machines were the original unchecked templates coded by all 4 analysts.

Figure 13 shows the resulting data for both conditions for each of the three machines. For all machines, there is little difference (and none that is significant) between performance between the two conditions.

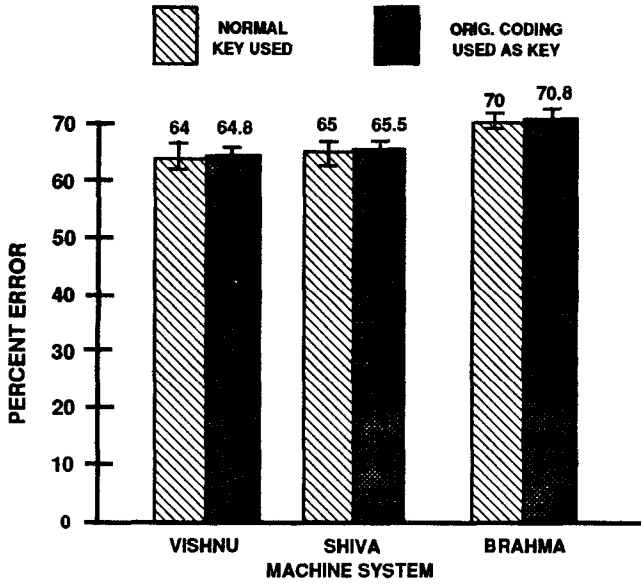

Figure 13: Performance of 3 Best Machines Measured Individually with Key or Original Coding

Figure 14 shows the same data, but combined for all three machines. Again, there is no significance difference, and because of the large sample size and resulting small standard error, the result is highly reliable. This finding may seem surprising given the results presented earlier that show substantial differences between original and (checked) final codings. The difference can be explained by the relative precision involved. Comparisons between original and final codings by analysts might be seen as analogous to different shades of colors: if an original

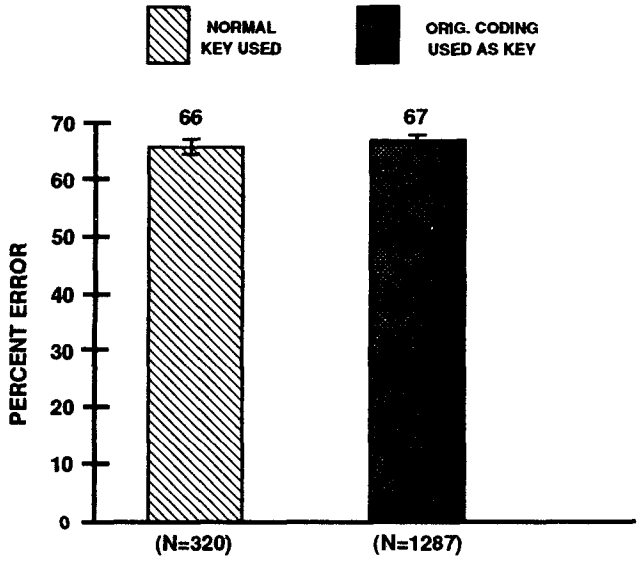

Figure 14: Performance of 3 Best Machines Measured Together with Key or Original Coding

analyst codes light green, while a second analyst produces a checked version of dark green, a measure of differences may show a substantial magnitude. At the same time, the machines may be producing codings ranging from blue to orange. While comparing light green with orange may yield considerable differences, it is plausible that there may be little or no difference between the magnitudes resulting when orange is compared first with light green and then with dark green. It can be expected that as machine performance improves, there will be an increasing difference between evaluations using original and checked keys.

AGREEMENT ON DIFFICULTY OF PARTICULAR TEMPLATES

The extent to which different analysts (and machines) agree on which templates are difficult and which are easy is of interest in understanding the task and human and machine performance for the task.

This was measured first by obtaining scores for different analysts for particular templates, and calculating Pearson product-moment correlation coefficients for corresponding templates between pairs of analysts.

Figure 15 shows these correlations, with correlations between the 4 analysts shown at the far left, correlations between the 3 best machines shown correlation between ran-

domly in the center, and selected pairs of humans and machines shown at the far right.

Correlations among humans were relatively low, with R^2 from .04 to .20 (median at 0.13). Correlations among machines were moderate, with R^2 from .21 to .44. Correlations between a particular human and a particular machine were low to moderate, with R^2 from .07 to .21.

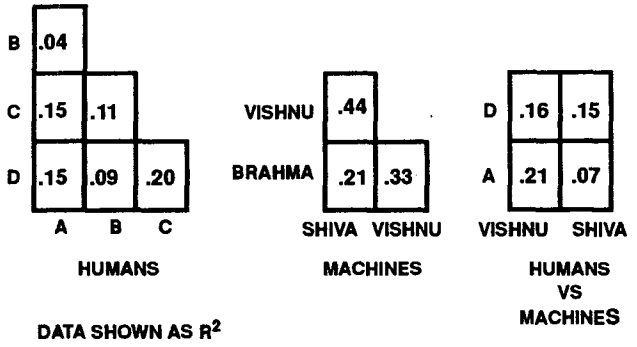

Figure 15: Correlation of Scores on Particular Templates Between Different Analysts

A second approach to studying the same issue was taken by

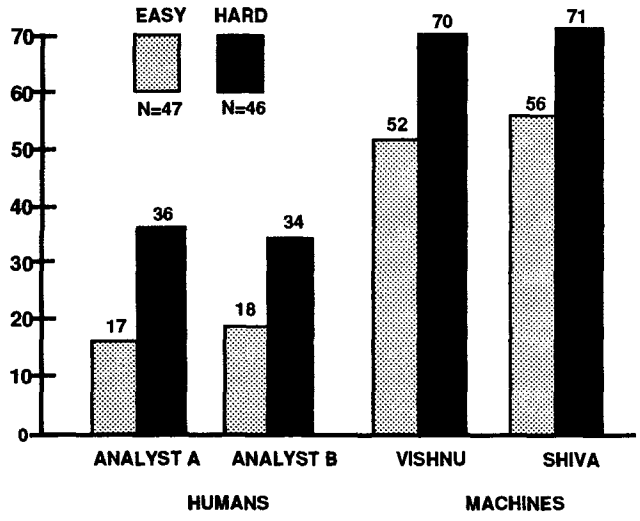

Figure 16: Performance for Easy and Hard Templates by Individual Analysts and Machines

dividing a set of 93 temples into two groups, either "easy" or "hard". The division was made by first calculating an error score for each template when one analyst is measured against another as a key. This was done for all 93 temples for 2 pairs of analysts, with the mean difference calculated for both pairs for each template. The templates were then divided into "easy" and "hard" groups, with the "easy" group consisting of those templates with the lower mean difference scores and the "hard" group consisting of those templates with the higher scores.

This was intended a a way of constructing a simulation of a corpus and task that was easier (and harder) than the Tipster task, which was viewed by many in the Tipster project as excessively demanding. The hypothesis was that the machines might do comparatively better than humans on the "easy" set than on the "hard" set.

The results are shown in Figure 16 for two analysts and two machines. The opposite of the expected (and hoped-for) hypothesis appeared to be the case. The human analysts produced roughly twice as many errors on the "hard" set of templates as on the "easy" set, while the machines were only somewhat better on the "easy" versus "hard" set.

Figure 17 shows the same data, but in terms of the means for each pair of humans and machines. In addition, data (at the far right) is presented in which the mean machine error is divided by the mean human error for each set, thus normalizing the difference. Here the comparative difference between machine and human was much larger for the "easy" set compared with the "hard" set.

Whether this method allows a realistic simulation of the effects of difficulty of the text and task is unclear and the meaning of this data is hard to interpret. It would be valuable to develop tests sets for future MUC and Tipster evaluations that could effectively assess the effect and nature of text and task difficulty.

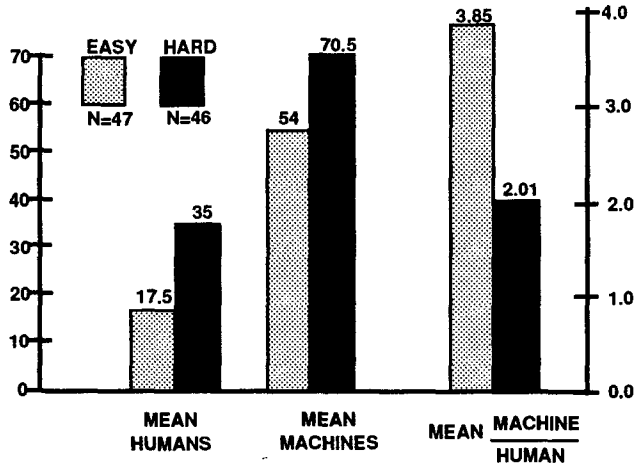

Figure 17: Performance for Easy and Hard Templates by Means for Humans and Machines

COMPARING HUMAN AND MACHINE PERFORMANCE FOR SPECIFIC INFORMATION

A particularly interesting question about human and machine performance is that of how the two compare for different aspects of the extraction task. Such differences are most easily seen by comparing human and machine performance on different slots in a template object or on an entire object.

This issue was investigated by making use of a 60-template subset of the MUC-5/Tipster test set that was coded by all analysts and for which a key was made by the 5th analyst.

The scoring program, in addition to calculating overall scores for templates or sets of templates, also provides scores for each individual slot and object in the template.

Scores for each of the four human analysts and the three best machine systems were obtained for each object and slot using the scoring program. Only those objects and slots with at least 10 examples of nonblank responses in the keys were further scored.

Because of the wide disparity between scores for humans and for machines, the data representing performance on each slot and object were normalized in the following manner: First, performance for each slot and object for the 4 human analysts was averaged by calculating a mean error for each slot and object. A rank order score was then assigned to each slot and object, reflecting the lowest to highest comparative performance for humans for that slot or object.

Finally, a calculation was made of the comparative difference in performance in scores for particular slots and objects between humans and machines, by subtracting the rank order score for machines from the rank order score from humans for the corresponding slot or object.

Figure 18 shows the results, listing the 5 comparatively "worst" slots from the point of view of the machines and then the 5 comparatively "best" slots from the point of the machines. For further comparison, one slot in which the performance of humans and machines are comparatively equal is listed.

The column at the left shows the rank order difference score, with +30 indicating <equip>name, the slot for which machines did the worst compared with humans. The two columns at the right list the error scores for humans and machines, with the <equip>name slot resulting in a machine score of 84.3% error, but a human score of 18.0% error. Note that this extreme comparative difference results

MACHINE COMPARATIVELY WORSE

DIFF	SLOT	MACHINE	HUMAN
+30	<equip> Name	84.3	18.0
+12.5	<litho> equip	68	20
+11	<layering> equip	67	19.3
+10	<pkg> device	76.3	22.8
+10	<pkg> pl-count	60	17.5

MACHINE COMPARATIVELY BETTER

-20	<layering> type	49.0	29.5
-17	<layering> OBJ	45	25.5
-13	<layering> film	66	37.8
-13	<pkg> unit	68	46
-8	<litho> type	57	24.25

COMPARATIVELY EQUAL

0	<device> function	68.7	29.3

Figure 18: Comparisons of Human and Machine Performance on Specific Slots

from two factors: the machines did particularly bad on this slot (84.3%) compared to their overall performance (68.7%), and humans did particularly well on the slot (18.0%) compared to their overall performance (29.3%).

This data is essentially a pilot experiment towards investigating the question of how human and machine performance might compare on specific tasks, with slot and object fills the best available way of obtaining this information with the given data. Without detailed investigation, we can only speculate on the reasons for the results. It appears, however, that many or all machine developers, pressed for time particularly in the case of microelectronics, simply did not bother to code specific slots, viewing them as unimportant to the final score. Those slots would likely appear in a list of slots that machines did comparatively bad on (though it may also be necessary for humans to do particularly badly on the slots as well). It appears that, in the case of the slots that machines did comparatively well on, that these were slots with large sets of categorical fills, with the set sufficiently large and the items sufficiently obscure that humans had a difficult time remembering them well enough to effectively detect them when they appeared in text. Because these words (or acronyms) tended to be context-free, relatively simple strategies for detecting these keywords and matching them to slots could be used. This does suggest that the abilities of humans and machines are quite different, and that an approach in which an integrated human-machine system is used rather

than a machine-only system, as is described in [3], might be appropriate.

CONCLUSIONS

The present study has shown that on the English Microelectronics extraction task, the best machine system performs with an error rate of about 62%, a little less than twice that of the 33% error produced by highly skilled and experienced human analysts.

This level of performance suggests that machine extraction systems are still far away from achieving high-quality extraction with the more difficult texts and extraction problems characterized by the Tipster corpus. However, machine performance is close enough to the human level to suggest that practical extraction systems could be built today by careful selection of both the text and the extraction task, and perhaps making use of integrated human-machine systems that can harness the abilities of both humans and machines for extraction rather than depending upon a machine-only system.

ACKNOWLEDGEMENTS

The following persons contributed to the effort resulting in the human performance measurements reported here: Deborah Johnson, Catherine Steiner, Diane Heavener, and Mario Severino served as analysts for the English Microelectronics material, and Mary Ellen Okurowski made keys to allow comparison with all analysts. Susanne Smith served as a technical consultant on microelectronics fabrication. Beth Sundheim, Nancy Chinchor, and Kathy Daley helped in various ways, particularly with respect to the scoring program used. Nancy Chinchor also provided some statistical advice. Boyan Onyshkevych also helped in defining the problem and approaches to attacking it, and was a coauthor on some early presentations of pilot work on human analyst performance at the Tipster 12-month meeting in September, 1992 and the National Science Foundation Workshop on Machine Translation Evaluation on November 2-3, 1992, both in San Diego. Mary Ellen Okurowski provided valuable discussions about the human performance work and comments on this paper. Larry Reeker helped with project management of the overall template collection effort. and provided comments on this paper.

REFERENCES

1. Will, Craig A., "Comparing Human and Machine Performance for Natural Language Information Extraction: Results for English Microelectronics from the MUC-5 Evaluation." *Proceedings of the Fifth Message Understanding Conference (MUC-5).* Baltimore, MD, August 25-27, 1993. San Mateo, CA: Morgan Kaufmann, Inc., 1994.

2. Sundheim, Beth. "Overview of the Fourth Message Understanding Evaluation and Conference." *Proceedings of the Fourth Message Understanding Conference (MUC-4)* (p. 18 and 20). McLean, VA, June 16-18, 1992. San Mateo, CA: Morgan Kaufmann, Inc., 1992.

3. Will, Craig A., and Reeker, Larry H. "Issues in the Design of Human-Machine Systems for Natural Language Information Extraction." Presented at the 18-month Tipster meeting, February 22-24, 1993, Williamsburg, VA. Paper available from authors.

4. Onyshkevych, Boyan A. "Template Design for Information Extraction." *Proceedings of the TIPSTER Text Program, Phase One.* San Mateo, CA: Morgan Kaufmann, Inc., 1994.

5. Carlson, Lynn, Onyshkevych, Boyan A., and Okurowski, Mary Ellen. "Corpora and Data Preparation for Information Extraction." *Proceedings of the TIPSTER Text Program, Phase One.* San Mateo, CA: Morgan Kaufmann, Inc., 1994.

6. Onyshkevych, Boyan, Okurowski, Mary Ellen, and Carlson, Lynn. "Tasks, Domains, and Languages for Information Extraction." *Proceedings of the TIPSTER Text Program, Phase One.* San Mateo, CA: Morgan Kaufmann, Inc., 1994.

7. Chinchor, Nancy, and Sundheim, Beth. "MUC-5 Evaluation Metrics." *Proceedings of the Fifth Message Understanding Conference (MUC-5).* Baltimore, MD, August 25-27, 1993. San Mateo, CA: Morgan Kaufmann, Inc., 1994.

8. SAIC. "Tipster/MUC-5 Scoring System User's Manual." Version 4.3, August 16, 1993. San Diego, CA: Science Applications International Corporation.

APPENDIX:

Details of Statistical Measurements and Tests

Performance is expressed in terms of error per response fill, using the methodology described by Nancy Chinchor and Beth Sundheim [7] and implemented by the SAIC scoring program [8].

Error is defined in this methodology by the following formula:

$$\text{Error} = \frac{\text{incorrect} + (\text{partial} \times 0.5) + \text{missing} + \text{spurious}}{\text{correct} + \text{partial} + \text{incorrect} + \text{missing} + \text{spurious}}$$

where each variable represents a count of the number of responses falling into each category. A *correct* response occurs when the response for a particular slot matches exactly the key for that slot. A *partial* response occurs when the response is similar to the key, according to certain rules used by the scorer. An *incorrect* response does not match the key. A *spurious* response occurs when a response is nonblank, but the key is blank, while a *missing* response occurs when a response is blank but the key is nonblank.

The scoring program is typically given a set of templates and provides an error score, based on all slots in all of the templates. In this paper data is usually reported as means in terms of this error score. However, statistical parameters describing variability are estimated by having the scoring program generate scores for each template, even though the means of the data reported here are calculated across a set of templates. Only about 80% of templates produce an independent score, and only those templates are used in estimating statistical parameters. Thus, in many cases two Ns are given, with the larger number the number of templates scored and the smaller number the number of individual template scores used in estimating the variance, calculating the standard error and confidence intervals, and performing statistical tests.

In the remainder of this Appendix, details are provided for data presented in each Figure, as indicated:

Figure 2: In the "primary" and "secondary" conditions, 120 templates were scored, 30 for each analyst. In the "other" condition, 240 templates were scored, 60 for each analyst. The mean for the primary condition was statistically different from zero at a level of $p < .0001$ ($z=6.74$). The standard error of the mean for the primary condition was 2.30 and the 95% confidence interval (indicating that 95% of the time the true population mean. can be found within this interval) was from 10.9 to 20.7. Of the 120 templates (each of which contributed to the score shown as the mean), 92 templates could be scored independently, and thus N=92 was used for statistical tests. The standard error of the mean for the secondary condition was 2.76 and the 95% confidence interval from 21.5 to 32.5. N was 95. The means for the primary and secondary conditions are statistically different at a level of $p < .01$ ($t=3.19$). The standard error of the mean for the "other" condition was 2.13, and the 95% confidence interval was from 33.2 to 41.6, with an N of 193. The means for the secondary and other conditions were significantly different ($p < .01$, $t=2.88$).

Figure 4: The standard error of the mean for the "All analysts" condition for the 4 analysts (A,B,C, and D, respectively) was as follows: 2.8. 2.6, 3.1, and 2.9. For the "Other analysts" condition: 3.4, 3.2, 3.7, and 3.3. For "Independent analysts": 4.1, 3.7, 3.9, and 5.7. For "5th analyst": 4.7, 4.5, 4.1, and 4.4.

Figure 5: The mean across analysts in the "All Analysts" condition is 25.3, with a standard error of the mean of 1.45 and a 95% confidence interval from 22.4 to 28.2 (N=374). The mean across analysts in the "Other Analysts" condition is 29.8, with a standard error of the mean of 1.74 and a 95% confidence interval from 26.39 to 33.21 (N=283). The mean across analysts in the "Independent Analysts" condition is 33.2, with a standard error of the mean of 2.14 and a 95% confidence interval from 29.0 to 37.4 (N=190). The mean across analysts in the "5th Analyst" condition is 28.3, with a standard error of the mean of 2.24 and a 95% confidence interval from 24.0 to 32.6 (N=187). The mean of the "All analysts" condition is significantly different from that of the "Other analysts" condition ($t=2.00$), while the mean of the "Other analysts" condition is not significantly different from that of the "Independent analysts" condition.

Figure 7: In the "All Analysts" condition, analyst A had recall and precision scores of 84 and 86.5, respectively, analyst B 81 and 88.5, analyst C 82.5 and 85.5, and analyst D 82 and 86.5. In the "Other Analysts" condition, analyst A had recall and precision scores of 79 and 79, analyst B 72 and 81, analyst C 79 and 75, and analyst D 78 and 82. In the "Independent Analysts" condition, Analyst A had recall and precision scores of 81 and 78, Analyst B 72 and 83, Analyst C 79 and 79, and Analyst D 73 and 75, respectively. In the "5th analyst" condition, analyst A had recall and precision scores of 81 and 83, analyst B 69 and 80, analyst C 86 and 86, and analyst D 81 and 86, respectively.

Figure 11: The standard error of the mean for the human analysts was 2.14, and the 95% confidence interval was from 29.0 to 37.4 (N=190). The mean error for system X (Vishnu) was 62%, with a standard error of 2.41 and a 95% confidence interval from 47.28 to 66.72. The mean error for system Y (Shiva) was 63%, with a standard error of 2.19 and a 95% confidence interval from 58.71 to 67.29, while the mean error for system Z (Brahma)was 68%, with a standard error of 2.27and a 95% confidence interval from 63.55 to 72.45. The difference between the mean human scores and the mean for the best machine was statistically significant ($p<.001$, $t=8.58$).

Figure 12: The human analysts had recall and precision scores of 79 and 79%, 72 and 81%, 78 and 82%, and 79 and 75%, respectively. The three best machines, in contrast, had recall, and precision scores of 45 and 57%, 53 and 49%, and 41 and 51%, respectively. These data differ slightly from the official scoring for machine performance because they use only the 120 article subset, not the full 300 article test set. In addition, the official scoring of the machines used interactive scoring, in which human scorers were allowed to give partial credit for some answers, while this scoring was done noninteractively. Note, however, that non-interactive scoring was used for all data in this paper, so comparisons between humans and machines are possible. The use of noninteractive scoring for both machine and human data could bias the result slightly, because of the possibility that humans might be better at providing partially or fully correct answers that don't obviously match the key, but again the difference is likely to be slight.

Figure 13: 120 templates were used in the "Normal key" condition, while 480 templates were used in the "Orig. Coding" condition in the calculation of the mean.

Figure 14: 360 templates were used in the "Normal key" condition, and 1440 used in the "Orig. Coding" in calculating the mean. 320 and 1287 were used, respectively, in calculating the error.