Automatic Extraction of Facts from Press Releases to Generate News Stories

Peggy M. Andersen, Philip J. Hayes
Alison K. Huettner, Linda M. Schmandt
Irene B. Nirenburg
Carnegie Group, Inc.
5 PPG Place
Pittsburgh, PA 15222, USA

Steven P. Weinstein
Reuters Ltd
85 Fleet Street
London, EC4P 4AJ England

Abstract

While complete understanding of arbitrary input text remains in the future, it is currently possible to construct natural language processing systems that provide a partial understanding of text with limited accuracy. Moreover, such systems can provide cost-effective solutions to commercially-significant business problems. This paper describes one such system: JASPER. JASPER is a fact extraction system recently developed and deployed by Carnegie Group for Reuters Ltd. JASPER uses a template-driven approach, partial understanding techniques, and heuristic procedures to extract certain key pieces of information from a limited range of text.

We believe that many significant business problems can be solved by fact extraction applications which involve locating and extracting specific, predefined types of information from a limited range of text. The information extracted by such systems can be used in a variety of ways, such as filling in values in a database, generating summaries of the input text, serving as a part of the knowledge in an expert system, or feeding into another program which bases decisions on it. We expect to develop many such applications in the future using similar techniques.

1. Introduction

While a computer program that can provide complete understanding of arbitrary input text remains a distant dream, it is currently possible to construct natural language processing systems that provide a partial understanding of certain types of text with limited accuracy. Moreover, such systems can provide cost-effective solutions to commercially-significant business problems. This paper describes one such system: JASPER. JASPER (Journalist's Assistant for Preparing Earnings Reports) is a fact extraction system recently developed and deployed by Carnegie Group for Reuters Ltd. JASPER uses a template-driven approach and partial understanding techniques to extract certain key pieces of information from a limited range of text. Specifically, JASPER takes as input a live feed of company press releases from PR Newswire. It identifies which of those releases contain information on company earnings and dividends, and for those releases, it extracts a predetermined set of information. It then reformats that information into a candidate Reuters news story and ships it off to a financial journalist for validation or editing. JASPER improves both the speed and accuracy of producing Reuters stories and hence provides a significant competitive advantage in the fast-paced world of financial journalism.

JASPER gets excellent results in terms of both accuracy and speed. It does this by combining frame-based knowledge representation, object-oriented processing, powerful pattern matching, and heuristics which take advantage of stylistic conventions, including lexical, syntactic, semantic, and pragmatic regularities observed in the text corpus. The shallow, localized processing approach that we have adopted focusses on the information to be extracted and ignores irrelevant text. The first phase of JASPER has been deployed at Reuters for use and testing. It provides a low-risk and high-value solution to a real-world business problem.

JASPER's architecture facilitates transfer to other fact extraction applications; the domain-independent core which controls processing is separate from the application-specific knowledge base which makes decisions about extracting information, so only the latter needs to be rewritten for other applications. Still, the knowledge engineering required to build an application is significant. We estimate that the JASPER application involved approximately eight person months in knowledge engineering, apart from basic system development.

Many significant business problems can be solved by similarly focussed fact extraction applications. The information extracted can used in a variety of ways, such as

filling in values in a database, generating summaries of the input text, serving as a part of the knowledge in an expert system, or feeding into another program which bases decisions on it. We expect to develop many such applications in the future using similar techniques.

2. Related Work

Most text understanding systems have generally fallen into two categories:

- systems which attempt to perform a complete linguistic analysis of the text
- systems which perform partial understanding to accomplish certain specific understanding tasks

Most of the linguistically-based systems perform a more or less pure syntactic analysis and a semantic and/or pragmatic analysis to arrive at a representation of the meaning of the text. TACITUS [3], PROTEUS [5], PUNDIT [2], CAUCUS [9], and the News Analysis System (NAS) [6] all fall into this category. The systems differ in the specifics of the syntactic, semantic and pragmatic analysis used and in the degree to which the different levels of processing are integrated. For example, TACITUS' syntactic step does enough semantic processing to produce a logical form; a second step performs pragmatic tasks such as reference resolution. Some systems base their processing on a particular linguistic theory; for example, CAUCUS uses Lexical Functional Grammar and NAS uses a Government-Binding approach to syntax and semantics. Other systems use more idiosyncratic approaches to the analysis.

These linguistically-based systems have a tremendous potential for complete understanding of a wide range of text, because, in theory, they do a complete analysis of the text. However, the processing of such systems tends to be relatively slow; in addition, these systems have tended to be used in research contexts in part because the range of coverage that they can provide is necessarily limited. A full analysis of text that covers diverse topics or that must be processed at a high rate of throughput is not feasible given the current state of the art.

Systems which do not attempt a complete understanding of the text, but rather focus on specific understanding tasks are more likely to result in deployable applications. ATRANS [7], the only major deployed fact extraction system before JASPER, is the most notable example. ATRANS operates in the domain of international banking telexes, dealing with one major subclass of such telexes -- money transfer telexes. ATRANS automatically extracts the information required to complete the transfer (the various banks mentioned in the telex, their roles in the money transfer, payment amounts, dates, security keys, etc.) and formats it for entry into the bank's automated transaction processing system. The understanding techniques used in ATRANS are based on caseframe analysis using the Conceptual Dependency formalism [8] which relies on semantics over syntax, and does not require a complete analysis of the text.

General Electric's SCISOR system [4] uses a hybrid approach, combining syntactic and caseframe parsing. This allows it to exploit the strong top-down domain expectations provided by caseframes to deal with relevant fragments from text that it cannot fully analyze, while at the same time generating complete linguistic analyses when possible. SCISOR is also designed so that general grammatical knowledge and domain-specific knowledge are kept separate. This will greatly facilitate its transfer to other domains.

3. Business Problem

A major component of Reuters business is to provide real-time financial news to financial traders. Corporate earnings and dividend reports are two routine, but extremely important, types of financial news that Reuters handles. Publicly-traded companies must, by law, provide this information periodically, and equities traders rely on news services like Reuters to distill the companies' reports and make the information available within minutes or even seconds so that they can use it to make decisions about which stocks to buy and sell. It is imperative that the reports be generated very quickly and very accurately; if Reuters can produce important earnings and dividend stories first, they will have the edge in the very competitive real-time financial news market.

One important electronic sources of earnings information is PR Newswire, which provides a wide range of press releases on many topics to subscribers. Figure 1 is a typical earnings press release received through the PR Newswire service. Figure 2 shows the corresponding Reuters news story which a reporter would generate from this release.

While the production of these reports is crucial to Reuters business, it is a routine, tedious task which requires just enough domain knowledge and human intelligence to require trained reporters. JASPER helps Reuters produce earnings and dividend news stories substantially faster, with fewer errors, and with less tedium. JASPER automatically generates draft earnings and dividend stories from the press releases carried on PR Newswire and makes them available

/FROM PR NEWSWIRE MINNEAPOLIS 612-871-7200/
TO BUSINESS EDITOR:

GREEN TREE ANNOUNCES THIRD QUARTER RESULTS

ST. PAUL, Minn., Oct. 17 /PRNewswire/ -- Green Tree Acceptance, Inc. (NYSE, PSE: GNT) today reported net earnings for the third quarter ended Sept. 30 of $10,395,000, or 70 cents per share, compared with net earnings of $10,320,000, or 70 cents per share, in the same quarter of 1989.

For the nine months, net earnings were $26,671,000, or $1.70 per share, compared with the first nine months of 1989, which had net earnings of $20,800,000, or $1.21 per share.

GREEN TREE ACCEPTANCE, INC. STATEMENT OF EARNINGS
(in thousands)

	Three Months		Nine Months	
	9/30/90	9/30/89	9/30/90	9/30/89
Earnings before income taxes	16,903	16,785	43,368	33,825
Net earnings	10,395	10,320	26,671	20,800
Earnings per share:	.70	.70	1.70	1.21
Weighted average common shares outstanding	11,599	11,494	11,597	11,450

-0- 10/17/90

Figure 1: An example PR Newswire release

GREEN TREE ACCEPTANCE, INC <GNT.N> Q3 NET
ST. PAUL, Minn, Oct 17
Shr 70 cts vs 70 cts
Net 10.4 mln vs 10.3 mln
Avg shrs 11.6 mln vs 11.5 mln
Nine Months
Shr 1.70 dlrs vs 1.21 dlrs
Net 26.7 mln vs 20.8 mln
Avg shrs 11.6 mln vs 11.5 mln

Figure 2: An example Reuters news story

to reporters for editing. Reporters need only check the information and make any necessary changes.

In all, JASPER attempts to extract 56 different values from an earnings release, though not all of these will ever be present in any given release. Most of the values that JASPER extracts are numbers -- net income, per share income, revenues, sales, average number of shares outstanding, etc. -- and most information types are reported for four time periods: the quarter just ended, the corresponding quarter of the prior year, the fiscal year to date just ended, and the corresponding year to date period of the prior year. Other information types have only one value; these include the quarter being reported (Q1, Q2, Q3, or Q4), the end date of the quarter being reported, the place of origin of the release, the dividend, the date on which the dividend will be paid, etc.

The JASPER system was developed between December, 1990 and August, 1991. The software was installed in early August, 1991, and reporters in New York and other Reuters offices in the United States began experimental use of the system immediately.

Results of this use have shown that JASPER does its job quickly and accurately.

- JASPER processes the average earnings or dividend release in approximately 25 seconds.
- By the standard measures of recall and precision, the system is over 96% accurate overall in selecting relevant releases for processing.
- By corresponding measures for fact extraction, the system is over 84% accurate overall in extracting the desired information from the selected releases. Over 90% of the values that JASPER places in the stories it generates are correct.
- JASPER handles 33% of targeted releases perfectly. It handles 21% of all earnings stories with no errors or omissions whatever; and handles 82% of all dividend releases with no errors or omissions.

4. Technical Approach

Upon receiving a press release from PR Newswire, JASPER first determines whether it is "relevant" -- that is, whether it is one of the earning or dividend releases from which we wish to extract information. Carnegie Group's Text Categorization Shell (TCS) [1] is used to do this selection. Only about 20% of the information on the wire is relevant.

JASPER has a frame representation which defines the specific information types to be extracted from relevant texts. These frames guide the remainder of the processing. The slots of the frame define what information is to be extracted and also hold information about how the processing for each slot is to be performed.

For each slot in the frame, the system tries to match against each sentence an associated set of patterns of words; if any of the patterns match, a procedure, or extraction method, also associated with the particular slot, is called to decide whether the patterns which matched can be used to assign a value to the slot. The extraction method may decide that no slot value should be assigned, or it may translate the information that matched into a canonical form and store it in the frame. Once all available information has been extracted and stored in the frame, JASPER generates a news story from the information and makes the story available to reporters for editing.

Together, the patterns and the extraction methods make up the application-specific *rulebase*. The rulebase is tailored to the syntactic structures and vocabulary that we have observed in our analysis of the corpus. JASPER does not do complete syntactic parsing or complete semantic analysis of the text. Instead, it matches "sketchy" patterns, looking only for relevant words or phrases within sentences. The extraction methods too were written expressly to handle the forms that we have observed in PR Newswire texts. The rulebase makes certain assumptions about the language it expects to find in a text; while these assumptions are not always borne out, they are in most cases, and JASPER reaches a very high level of accuracy because of them.

The input press releases often have a table along with the textual part, as in the example in Figure 1. The information contained in the two parts often overlaps, but in most cases neither the textual nor the tabular part gives all the required information. We therefore extract the information from both the text and the table and then merge the two sets of values. In this paper we do not discuss the techniques used to extract information from tables.

JASPER runs under Ultrix on a DECstation 3100. The dedicated standalone DECstation has loose system interfaces to the PR Newswire feed and to a Tandem computer on which the reporters edit stories. The core extraction system runs in Lucid Common Lisp and uses the Common Lisp Object System (CLOS) to represent its frames.

4.1. Text Understanding Control

The control of the text understanding component of JASPER follows a simple algorithm. For each sentence in the release, JASPER checks every item on an ordered list of targeted information types, or slots, to determine whether a value has already been assigned to the corresponding slot. If no value has yet been stored, JASPER tries to match the current sentence against a set of patterns associated with that slot. If any pattern matches, tentatively identified values from the sentence are bound to pattern matcher variables, and the extraction method associated with that information type is called to interpret the results of the pattern matching.

The extraction methods are application-specific procedures associated with individual slots which use the results of pattern matching to determine whether any slots should be filled and what value(s) should be used. If an extraction method assigns a value to a slot, the slot is marked as "done" and is removed from the list of slots to try on subsequent sentences.

4.2. The Pattern Matcher

One important component of Carnegie Group's Text Categorization Shell is a powerful pattern matcher which matches complex patterns of words written in a specialized pattern language against text. This technology is also central to JASPER's fact extraction technology. The network-based left-to-right pattern matcher includes disjunction, negation, optionality, and skipping operators, and performs regular and irregular English morphology transformations when words are specified as nouns or verbs. The following pattern illustrates the pattern matching operators:

```
((profit +N ! earnings)
 (&skip 8 ($n ?million dollar +N))
 (&n (per share)))
```

This pattern says to match either the word *profit* or *profits* (`+N` indicates that it is a noun) or *earnings*, followed within eight words by any number (`$n`), followed optionally by *million*, followed by *dollar* or *dollars*; and a match will fail if the phrase *per share* follows *dollar*. This pattern would match in sentences like the following:

- *ABC Company announced profits of more than 50 million dollars last year.*

The pattern will not match in the following sentences, however:

- *XYZ Company's profits will be 2.25 dollars per share.*
- *XYZ Company announced that its earnings for the third quarter of 1990 will exceed expectations at 45.6 million dollars.*

The former sentence will fail because *per share* follows *dollars*. The latter will fail because more than eight words intervene between *earnings* and the number.

JASPER uses an extended version of the TCS pattern matcher for extracting information. It not only provides a boolean indication of whether a pattern matched, but also saves the information that we want to extract from the matches as special variables. A variable binding operator was added which can transform words matched in the text into a canonical form or simply save the words that matched. For example, this pattern

```
(&if ($n) %number)
```

will match any number and bind the number that matched to the pattern matcher variable `%number`.

This variable binding operator can also canonicalize values, as shown in the following pattern:

```
(&if
 (((fourth ! 4th) (quarter ! qtr)) !
   (Q4)) (%quarter = 4))
```

Patterns like this one can match a variety of expressions with the same meaning, binding a pattern matcher variable to a single form representing this meaning. This pattern matches all of the following phrases and binds the variable `%quarter` to *4* in every case: *fourth quarter*, *4th quarter*, *4th qtr*, *fourth qtr*, *Q4*. Once the crucial information is saved as pattern matcher variables, it can be used by the extraction methods to fill in values in a frame representation of the text.

4.3. Knowledge Representation

JASPER uses CLOS to control the extraction processing and to store the extracted information. Each type of release from which we extract information -- earnings and dividends -- has a frame, or CLOS class, associated with it,

with a slot for each information type that JASPER extracts. Figure 3 shows a portion of the earnings frame.

```
{{EARNINGS
  net-income-group:    <net-income-group-object>
  current-quarter-net: <net-income-object>
  prior-quarter-net    <net-income-object>
  currrent-ytd-net:    <net-income-object>
  prior-ytd-net:       <net-income-object>
  . . .
  period-reported:     <period-reported-object>
  . . .
}}
```

Figure 3: Earnings Extraction Frame

As mentioned above, we are interested in extracting numbers for four different time periods for many information types. The slots **current-quarter-net**, **prior-quarter-net**, **current-ytd-net**, and **prior-ytd-net** in Figure 3 represent the four slots for net income. All four slots are processed together using the same patterns and extraction methods; in order to accomplish this, a group slot, **net-income-group** in the example, is defined to hold the information required for processing these slots. The individual slots corresponding to each time period then hold the specific values extracted from the text.

Other information types have just one slot; for example **period-reported** in the example represents the period for which earnings are being reported (Q1, Q2, Q3, or Q4). This slot contains the information about how to extract the information -- the patterns and extraction methods -- and also holds the value once it is extracted.

Each of the slots in the earnings frame in turn has a class as its value; these classes store information about how to do the extraction, and once information has been extracted from the press release, they store the value extracted. Each of these classes has the following slots associated with it for extracting from text:

- a set of patterns to be used for extracting information from text
- a procedure, or method, for extracting information from text
- the value extracted

4.4. The JASPER Extraction Rulebase

This section describes application-specific patterns and procedures used for fact extraction. In analyzing the relevant texts, we found tremendous regularity in the language and syntactic structures used due to stylistic conventions followed by U.S. companies in reporting earnings and dividends. The patterns and extraction methods take advantage of these regularities, handling the forms that are most likely to occur in the text with a high level of accuracy, and the forms that occur less frequently or not at all less accurately.

The patterns used for extraction tend to match "sketchy" phrases, with skipping between the relevant elements of the pattern. For example, in order to find the net income we need to know that earnings are under discussion and we need to know what the amount of the earnings was; we can skip over other irrelevant information. A pattern like the following was used for net income:

```
((profit +N ! earnings)
 (&skip 8
  ((&if ($n) %number)
   ?(&if (million) %mult) dollar +N))
 (&n (per share)))
```

The patterns and extraction methods follow a few main strategies, depending on the kind of information to be extracted. Each of the strategies is described below.

4.4.1. Extracting Information for Simple Slots

Several slots for earnings and dividends required a very simple strategy. The reporting period for earnings is an example of this type of slot. The patterns match simple phrases and bind a variable to the value to be extracted; the extraction method then takes the value bound to the variable, canonicalizes it if necessary, and fills in the appropriate value in the frame.

Below is the pattern for the fourth quarter reporting period:

```
(&if
 (((fourth ! 4th) (quarter ! qtr)) !
  (Q4))
 (%quarter = 4))
```

If this pattern is matched, the pattern matcher variable **%quarter** is bound to the value *4*. The extraction method for the **reporting-period** slot is then called to fill in the value for the slot in the frame.

4.4.2. Understanding Time Context in Text

Earnings figures are generally given for four periods. In order to interpret the numbers in an earnings release, the system must not only find the figures reported and determine which information type they refer to (e.g. net income), but must also know the time period they apply to -- the current or prior year, and the quarter or the year to date.

For efficiency and for accuracy in handling elliptical time expressions, we handled time phrases separately, maintaining a time context which is then used to determine which of the four group slots to fill with the figures extracted. This time context makes it possible to process pairs of sentences like the following:

- *Earnings during the fourth quarter of 1990 were 50.5 million dollars. Sales were 74.3 million dollars.*

When JASPER processes the first sentence it stores as the

time context in working memory the fact that the last period mentioned was a quarter and the last year mentioned was the current one. After the time context is set up in this way, the earnings information is processed. The following sentence gives sales information, but does not provide any information about time. Despite this, the persistent time context in working memory allows us to determine that the slot to fill is the sales slot for the current quarter rather than for the prior quarter or for one of the year-to-date slots.

The extraction procedures for time handling use heuristics based on our analysis of the particular texts to be handled and on our knowledge of English syntax, semantics, and pragmatics. While JASPER does not handle all time contexts correctly, it performs very well on the types that occur in the corpus of PR Newswire earnings reports.

4.4.3. Extracting Numbers for Group Slots

JASPER uses the same strategy for filling in all slots in earnings releases that require number values. We will use net income as an example. Net income has four specific slots to fill, one for each of the reported time periods; all are handled together by the `net-income-group` slot, which has a single set of patterns to match and a single extraction method to sort out which of the specific slot(s) to fill when relevant patterns match.

The `net-income-group` slot has two sets of patterns, informally called *current patterns* and *prior patterns*:

- *current patterns* match a word or phrase like *earnings* followed at some distance by a number; the number is bound to a pattern matcher variable. For example,

```
((profit +N ! earnings)
 (&skip 8
  ((&if ($n) %number)
   ?(&if (million) %mult) dollar +N))
  (&n (per share)))
```

- *prior patterns* match, in different orders, a word like *earnings* and a comparison word (e.g., *compared, increase ... from, rise ... from, versus*, etc.) followed at some distance by a number, which is bound to a pattern matcher variable. The following is an example of one such pattern:

```
((profit +N ! earnings)
 (&skip 8 (increase +V ! decrease +V)
 (&skip 8 (from))
 (&skip 8 ((&if ($n) %number)
           ?(&if (million) %mult)
           dollar +N))
 (&n (per share)))
```

These two patterns match the net income from the current and prior period in sentences like the following:

- *XYZ Company's profits for the current year increased from 45.5 million dollars last year to 50 million dollars.*

The time context described above is used to help determine which time period the extracted numbers refer to.

Conflicts between multiple matches are resolved by a heuristic procedure which allows JASPER to handle very complex sentences like the following with perfect accuracy:

- *ABC Company reported net earnings of 50 million dollars or 45 cents per share on revenues of 62 million dollars this year compared to earnings of 55 million dollars or 51 cents per share on revenues of 71.1 million dollars last year.*

5. Status and Results

JASPER was deployed for testing and use by reporters in early August 1991. Reporters in New York and other Reuters offices in the United States are currently using the system as an aid in producing earnings reports from PR Newswire announcements.

Accuracy tests run at Carnegie Group on a set of press releases that the system developers had never seen showed that JASPER's accuracy compares favorably with the results seen at the Second Message Understanding Conference (MUCK-II) [9].

JASPER also runs quickly enough to be used in this real-time application at an average of about 25 seconds per relevant press release. Reuters required processing to be less than 30 seconds in order for the journalists to get the stories out in the very tight timeframes they have to work with.

5.1. Accuracy

Before delivering JASPER we ran an accuracy test on 100 earnings releases and 50 dividend releases that the system developers had not seen or analyzed prior to the test. Accuracy scores were calculated by manually comparing the values extracted by JASPER with the correct values specified by a Reuters journalist. We measured accuracy separately for selection of relevant releases and fact extraction. Results are reported below.

5.1.1. Selection

Selection refers to the identification of relevant earnings and dividend reports in the stream of press releases from PR Newswire. Selection is measured with the standard measures of *recall* and *precision*. Recall is the percentage of actual earnings and dividend announcements that the selection process succeeds in finding. If recall is high, the system is not missing many items that it should select. Precision is the percentage of announcements that JASPER selects that are actually relevant, i.e. relate to earnings or dividends. If precision is high, the system is not wrongly selecting many items that should not be selected. These measures correlate closely with the recall and precision

measures used for MUCK-II, with only minor differences.

The figures in Figure 4 are based on 1047 PR Newswire releases, representing four days transmissions. The "Expected" figures represent the number of relevant releases actually present in the sample; the "Assigned" figures represent the number of releases selected by JASPER. We calculate overall accuracy as the average of the recall and precision scores.

	Earnings	Dividend	Combined
Expected	115	25	140
Assigned	117	25	142
Correct	112	24	136
Recall	97.4	96.0	97.1
Precision	95.7	96.0	95.8
Overall	96.6	96.0	96.5

Figure 4: JASPER Selection Accuracy

To compare our results with those of MUCK-II, we have chosen the highest score for recall and precision for each of four tests: two tests each with two different data sets. The first test on each data set was run "cold" -- the system developers had not seen the data in advance. The second test in each case was run after the system developers had made some changes to accommodate the test data. The best recall and precision scores for each test do not necessarily come from the same system.

	TST1 Cold	TST1 With Changes	TST2 Cold	TST2 With Changes
Recall	65%	85%	100%	100%
Precision	100%	90%	100%	100%
Overall	83%	88%	100%	100%

Figure 5: Best MUCK-II Selection Accuracy

5.1.2. Extraction

We use two measures of accuracy for fact extraction: *completeness* and *correctness*. Completeness corresponds roughly to the recall measure used in MUCK-II, and to the recall measure used for selection; it measures the percentage of targeted values available in the PR announcements that are actually extracted correctly by the system. A *targeted* value is one that should, according to Reuters practice and style guidelines, appear in the Reuters news story.

$$\text{completeness} = \frac{\text{correct values extracted}}{\text{total targeted values}}$$

Correctness corresponds roughly to the precision measure used in MUCK-II and the precision measure used for selection; it measures the percentage of times that a value extracted by the system is correct.

$$\text{correctness} = \frac{\text{correct values extracted}}{\text{total values extracted}}$$

JASPER was designed with an emphasis on correctness rather than on completeness on the assumption that reporters are less likely to overlook gaps than wrong values in the story. To compensate for this built-in bias, we also calculate an overall accuracy figure for extraction by averaging the percentages obtained for completeness and correctness.

In the accuracy results given in Figure 6, the "Unadjusted" figures are the raw results of the test. The "Adjusted" figures take into account typographical errors in the PR Newswire input (treating them in our favor), as well as the judgments of the same Reuters reporter regarding permissible deviations from his output. The figures "With Changes" are based on a second test on the same input after some changes had been made to the extraction rulebase.

	Unadjusted	Adjusted	With Changes
Targeted	1549	1542	1542
Extracted	1274	1275	1293
Correct	1153	1170	1190
Completeness	74.4	75.9	77.2
Correctnes	90.5	91.8	92.0
Overall	82.5	83.9	84.6

Figure 6: JASPER Extraction Accuracy

Figure 7 shows results from MUCK-II. The best test scores from each of four test for their correlates of completeness and correctness are given. The completeness and correctness scores do not necessarily come from the same system for any given test. The four tests involved two tests each of two different data sets. The first test on each data set was run "cold" -- the system developers had not seen the data in advance. The second test in each case was run after the system developers had made some changes to accommodate the test data, and so should correspond roughly to our test "with changes". While the MUCK-II accuracy measures differ somewhat from JASPER's, we believe that they are similar enough to show that JASPER compares favorably with the results of the systems which competed in MUCK-II.

	TST1 Cold	TST1 With Changes	TST2 Cold	TST2 With Changes
Completeness	44%	67%	68%	94%
Correctness	93%	95%	93%	98%
Overall	69%	81%	81%	96%

Figure 7: Best MUCK-II Extraction Accuracy

6. Conclusions

JASPER shows that text understanding technology has progressed to the point that it can be applied profitably to real commercial applications. However, the state of the art will not allow the technology to be applied to unconstrained applications. Instead, applications must be selected carefully in order to yield positive results. Certain

characteristics of applications will make them better candidates for fact extraction using Carnegie Group's technology and, we believe, other technologies as well:

- The events or reports have predictable components, or information types to be extracted.
- The information to be extracted tends to be expressed through an unambiguous and predictable, though possibly wide-ranging, set of linguistic forms.

Technologies like JASPER, which extract information from text using shallow, focussed processing techniques based on complex pattern matching and heuristic decision-making, can be profitably applied to applications having these characteristics. JASPER's accuracy compares favorably with other text understanding systems, and its processing speed allows for real-time use of the extracted information in a time-critical application. JASPER is a deployed system which solves a real business problem. We believe there any many other such business problems that could be solved with similar techniques. We expect to see the deployment of many more such applications in the future.

References

1. Hayes, P. J., Andersen, P. M., Nirenburg, I. B., and Schmandt, L. M. TCS: A Shell for Content-Based Text Categorization. Sixth IEEE AI Applications Conference, Santa Monica, March, 1990.

2. Hirschman, L., et. al. The PUNDIT Natural-Language Processing System. Proceedings of the Annual AI Systems in Government Conference, Washington, D.C., March, 1989, pp. 234-243.

3. Hobbs, J., Stickel, M., Martin, P., Edwards, D. Interpretation as Abduction. Proceedings of the 26th Annual Meeting of the Association of Computational Linguistics, Association of Computional Linguistics, June, 1988, pp. 95-103.

4. Jacobs, P. S. and Rau, L. F. "SCISOR: Extracting Information from Online News". *Comm. ACM 33*, 11 (November 1990), 88-97.

5. Ksiezyk, T., and Grishman, R. An Equipment Model and its role in the Interpretation of Noun Compounds. In *DARPA's 1986 Strategic Computing Natural Language Processing Workshop*, Information Sciences Institute, Marina del Rey, CA, 1986, pp. 81-95.

6. Kuhns, R. J. A News Analysis System. COLING88, Budapest, August, 1988.

7. Lytinen, S. and Gershman, A. ATRANS: Automatic Processing of Money Transfer Messages. Proceedings of the Fifth National Conference of the American Association for Artificial Intelligence, Philadelphia, August, 1986, pp. 1089-1093.

8. Schank, R. C. *Conceptual Information Processing.* North Holland, Amsterdam, 1975.

9. Sundheim, B. Second Message Understanding Conference (MUCK-II) Report. NavalOceanSystemsCenter, September, 1989.