A News Analysis System

Robert J. KUHNS

Artificial Intelligence Center
Arthur D. Little, Inc.
Cambridge, MA 02140 USA

Abstract

This paper describes a prototype news analysis system which classifies and indexes news stories in real time. The system extracts stories from a newswire, parses the sentences of the story, and then maps the syntactic structures into a concept base. This process results in an index containing both general categories and specific details. Central to this system is a Government-Binding parser which processes each sentence of a news item. The system is completely modular and can be interfaced with different news feeds or concept bases.

1.0 Introduction

This paper reports on a prototype news analysis system (NAS) which classifies and indexes news stories in real time. That is, the system receiving reports from a newswire is capable of classifying the reports and constructing an index of them. Once a news item is classified, it can then be routed to users for whom the story is pertinent. This system, which runs as an independent background process, is automatic and greatly reduces the amount of irrelevant information users must cope with.

From a theoretical view, one significant aspect of the system is that its parser is a deterministic principle-based Government-Binding (GB) system. Basing NAS on such a processor demonstrates capabilities of a syntax-oriented natural language parser and how linguistic and world knowledge (primarily financial) can be interfaced to provide a useful application. While a pre-prototype of NAS was written in ZetaLisp on Symbolics and later ported to the TI Explorer and Explorer II, the current version is implemented in Symbolics Common Lisp.

2.0 Background

Much of the text processing work has focused on methods for obtaining information or retrieving texts from large databases. Approaches are wide and include key wording, statistical analysis, pattern matching, and a method using lexical, syntactic, and semantic filters. However, there are other applications for which these techniques seem inadequate. (/Hayes et al. 1988/ does describe a strictly pattern matching approach to news categorization.) For instance, consider the news and financial industries where those who gather and report news or trade stocks and bonds must read, analyze, and react to current electronic information from many different sources almost instantly. This need to process information within seconds, coupled with the fact that there is already an overwhelming amount of information that individuals must sort through in order to find relevant news, clearly shows the need for rapid and accurate indexing and routing systems.

It was in this context that NAS was developed. The goal was to build a system with the capability of processing news stories received from active newswires, i.e., to be able to categorize each story against a set of general topics and more detailed subtopics in a matter of seconds. These stories and their associated indexes can then be routed to those interested individuals, thereby helping to reduce the load of irrelevant information that they must see in order to find the items which are pertinent to them.

Having identified this need, a pre-prototype was built around an existing parser based closely on the work of Marcus /Marcus 1980/. Stories in the pre-prototype were manually entered and were chosen so that the concepts do not directly appear in the story. The system was not using key words or phrases and was characterizing stories at a certain level of abstraction. While there were only about 12 different concepts from which this system could process, it was able to classify a story as a merger/acquisition, for example, and provide the companies involved, including the buyer and the acquired company, and tell whether the merger was successful or not.

The present version incorporates a more powerful parser than the one in the previous version, and it can identify over 200 concepts ranging from specific (a name of a company or currency) to abstract (marketing reports vs. analysis for some industry). Since NAS processes unrestricted sets of stories, concepts can be either explicit (specific ones) or implicit (abstract ones) and NAS will select those which are appropriate for each story.

The remaining sections will discuss the major components of NAS, the way it handles errors, and future directions. Several example stories and their indexes are also provided.

3.0 The Architecture of NAS

NAS consists of four major subsystems, viz., a stream filter, a lexical scanner, a parser, and a semantic processor or filter working sequentially as listed. The stream filter is the component which scans the news stream received via a satellite dish and selects only the textual news reports. Once a story is identified, it enters a lexical scanner resulting in sequences of words that are distinguished as sentences of the incoming report. These sentences are passed to the parser which constructs syntactic structures which are then used by the semantic processor in mapping these structures onto representations of the concepts. The story and index can then be directed to interested users and/or entered into a database for future consultations.

The underlying design consideration of this architecture was to modularize stream, linguistic, and application-specific semantic processing. In this way, new interfaces to different newswires, enhancements to the parser, or changes of application or a different concept base can be completed without impacting the other components, thereby enabling easy modification to NAS.

3.1 The Stream Filter

NAS is currently interfaced with a news feed which transmits news related to financial and commodity markets. Since the textual reports are interspersed with various quotations which are not input to NAS, the stream filter screens the non-textual items and directs only the stories containing text to NAS. When a quote is prefaced with several lines of text, the stream filter will send the item for processing of the text while the numerical quotes are ignored.

The stream filter can be deactivated and stories that have been stored on text files are then used as the corpus. The effect of changes to the system can now be traced when using static input.

3.2 The Lexical Scanner

The lexical scanner receives the news reports from the stream filter and provides the parser with processible input. It decomposes the incoming stream into words, numbers, distinguished characters, e.g., $ (dollar sign) and % (percent sign), punctuation, and sentences, and it also equates abbreviations with its unabbreviated form. The scanner has access to the lexicon and when it recognizes a word, it associates all the lexical information with that word.

As the scanner is analyzing the stream of characters, it is also determining the presence of sentences, i.e., sentence delimiters. Sentences ending in question or exclamation marks are easily detectable. Although news services differ, algorithms which rely on the formatting scheme of the news source have been developed which find declarative sentences even in the cases where they contain period-final abbreviations. The beginning and end of a news story are characterized by distinguishing features, so story identification is trivial.

3.3 The Parser

Central to NAS is the parser which provides syntactic structures that are eventually mapped onto concepts resulting in an index for a story. The parser is a principle-based GB parser and is a substantially revised version of /Kuhns 1986/. (See /Abney 1986/,/Berwick 1987/, /Kashket 1987/, /Thiersch 1987/, and /Wehrli 1987/ for descriptions of principle-based parsers.) The parsing strategy is deterministic in that no temporary structures are built or information deleted during the course of a parse (/Berwick 1987/ and /Marcus 1980/). It should be noted that in connection with this type of application, speed is crucial and although a deterministic parser is strict in that it cannot backtrack or produce alternate parses in ambiguous sentences, its speed of approximately 100 words/second in linear time is essential.

The parser has two main subsystems, viz., the set of interacting GB-modules and the lexicon. These modules include principles and constraints from Case and bounding theories and, especially, X-bar, thematic or θ, trace, binding, and control theories. These latter subsystems have a particularly prominent role for the parser. Predicate-argument relations or θ-role assignments to arguments of predicates are determined by θ-theory. In the case where movement has occurred, trace theory will relate an argument which now must reside in a position which cannot receive a θ-role with its empty category or trace in a θ-marked position from which the constituent has moved. This enables the θ-role of the argument to be determined. Possible coreferential relations for pronominals and anaphors are identified with principles of binding and control theory. Moreover, the Extended Projection Principle, θ-Criterion, and Case filter are observed by the parser. (For a full discussion of these modules and principles see /Chomsky 1981/.)

The primary output of the parser is a set of licensing relations. "Licensing" is a cover term for any of a number of possible relations between projections. Nonmaximal projections are licensed by maximal projections via X-bar theory and these maximal projections are licensed by an argument or a trace of an argument, a predicate, or an operator. Specifically, a predicate licenses its internal arguments or complements and its external argument or its subject. (Again for a more detailed discussion of these aspects of GB theory see /Chomsky 1986/.)

In that the goal of the parser is to license projections of each element of a sentence, it can perform two basic operations. It can construct a projection of a lexical item in direct use of X-bar theory or it can establish or assert a licensing relation between two maximal projections with respect to other constraints of GB Theory. The parser proceeds by first building a maximal projection and then attempts to license it to another maximal projection or vice versa, i.e., another projection to it.

Upon encountering a lexical item, the parser creates a maximal projection consisting of a set of features. Each node receives a type in terms of X-bar primitives ($\pm$N, $\pm$V), an index, and its lexical item from which it has projected. Relevant GB systems are invoked during the parse to determine binding relations and θ-role assignments. The proper index to encode binding or coreference will be incorporated in the projection and co-indexed projections share all of their features. However, it is not always possible to assign an index or θ-role at the inception of a projection because of inadequate information. The parser will not commit itself and will only include the syntactic structure that it can derive at that stage of the parse. When the relevant information is available, the parser will incorporate it in the incomplete node which preserves the monotonicity of parsing information. This process is constrained to the current cyclic node which is the left bounded context of the parser. (/Kuhns in preparation/ will discuss the specifics of this parser.) The parser produces a list of licensing relations for each sentence of a news story. In turn it outputs an ordered list of the relations corresponding to the sentences of a news report. This set is then passed to the semantic processor.

The other component of the linguistic processor of NAS is the lexicon which contains words and distinguished strings, together with their syntactic and subcategorization features including X-bar primitives ($\pm$N, $\pm$V), number, name or referential expressions, complement types, control features (for interpreting empty subjects (PRO) of infinitival complements), and θ-grids or θ-role assignments for predicates. An ambiguous lexical entry has features for all of its potential types associated with that item and lexical ambiguity resolution procedures choose the appropriate features during the parse (/Milne 1983/ and /Milne 1986/).

Morphology is minimal, reflecting only relations between roots and their derivational

forms and associations between words and affixes. Lexical redundancy rules for specifying correspondences between sets of features have been implemented. Since news reports frequently have abbreviations, lexical entries which have an abbreviated form will be marked as such, and when the abbreviation appears in a story, the lexical scanner retrieves the lexical information of the unabbreviated form. Relationships between lexical items and their extragrammatical features will be discussed below (Section 3.4).

The lexicon consists of less than 15,000 members and in building the lexicon the emphasis has been on the inclusion of verbs, adjectives, and prepositions. Names, especially of individuals, corporations, and geographical locations, not present in the lexicon are found in news reports regularly. While many familiar names are in the lexicon, unfamiliar nouns are handled by the error handling routines (Section 5.0).

While the lexicon is updated as needed, the way it was originally constructed was to collect distinct "words" from stories received from a satellite feed. Numbers were disregarded but names and abbreviations were included. During several non-continuous weeks of scanning the stream for new words, the task of assigning syntactic features to each valid item began. While this is a laborious and time-consuming process, it was aided by a menu-driven facility for feature assignment where typing was minimized and much time saved.

Also, during the time that previously unknown words were being "collected," a counter was indicating the number of current words in increments of 100. When the list was slightly over 7,000, the number of new words being added to it slowed. Furthermore, a point of convergence seemed to occur under 9,500 items. At this stage of lexicon development, a comparison of the existing words against a sample of over 50 words (mainly verbs and adjectives) taken from another news service suggested that the present list was sufficient in that it contained every word taken from the news stories. This is significant because a system which is to parse sentences within a story must have the capability of recognizing each word. Since it appears that the vocabulary of reports is bounded (with the exception of names), rapid linguistic processing of news is realizable with respect to lexical recognition.

3.4 The Semantic Processor

The semantic processor is an automatic pattern matcher which incorporates world knowledge that is used to determine the "meaning" of its linguistic input with respect to a set of topics and designators in its concept base. The term concept refers to a general notion such as merger/acquisition, terrorism, currency report, or strikes and lockouts. Designators are subtopics which provide detail to an index. A story categorized as a merger/acquisition could be further characterized by designators indicating specific companies involved or by the industries impacted. The existing system has the capability of processing over 200 concepts and designators. The output of this processor (and NAS) is a classification or index of a story consisting of one or more general concepts and their designators. If no general concept is found, the system may still assign designators. In other words, a story may be about Air France while the general classification is unknown.

Structurally, the processor can be viewed as having a concept base and a θ-relation interpreter which takes as input the predicate-argument structures denoted by θ-relations and attempts to find matches with elements in the concept base. The concept base itself possesses an internal structure consisting of several levels of abstraction. The most concrete level consists of names which enter into an index whenever present in a story. This level primarily contains names of corporations, industries, corporate executives, government officials, and geographical locations. In order to keep linguistic and the application dependent concepts independent, pointers between the lowest level of the concept base and the lexicon are used. A change to the concept base or substitution of a new one will not affect the linguistic component.

Representations at the next level reflect commonality which the elements at the first level share and together they provide designators for a story. The objects at this more abstract level are called entity types and they further characterize the members of the first level. Two common entity types are industry type and company. The semantic processor can assign an industry designator to a story if either the industry is explicitly mentioned in the story or if companies or individuals mentioned in the story are related to a particular industry. So a news item about Swiss Air will have both the name Swiss Air and its associated industry, viz., Airline Industry, assigned to its index.

The last and most abstract level is that of a general concept such as merger/acquisition, currency report, strikes and lockouts, and terrorism. These are represented by frames where there is one action slot and at least one entity type slot (determined from the previous level). Moreover, one concept may have several different representations. The action slot is a list of one or more synonomous words or phrases that denote an action or the "doing" component of a concept. The members of the action slot are not semantic primitives but are actual words. Furthermore, they are word stems and not all of their morphological variants. The entity type slots contain types of entities which are found in the previously discussed level of the concept base. For example, a partial representation for merger/acquisition is:

(1) Merger/Acquisition
Action: buy, take over
Agent: company
Object: company

where buy or take over is the action and the entity type slots are labeled agent and object and their members must be of the type company. Details of this formalism are discussed below in connection with the θ-relation interpreter.

The other module of the semantic processor is a θ-relation interpreter which maps θ-relations of each sentence of a news story into the concept base, or, in other words, onto specific concepts and designators. This mapping is executed as follows. First, recall that the parser returns a set of licensing relations including θ-relations for each story. Each member of this set is a list of the relations for a sentence of the story. In examining the θ-relations for a sentence, the interpreter attempts to establish general concepts by pairing the predicate and arguments of a

θ-relation with the action and entity type slots of a concept, respectively. For example, consider a merger/acquisition frame (1) and a θ-relation which has bought as a predicate with its agent being Acme Corp. and its object as Software Inc.
The θ-relation interpreter first determines that bought is related to buy and that buy is a member of the action slot. Since this comparison is successful, the interpreter then derives the entity types of Acme Corp. and Software Inc. from the abbreviations Corp. and Inc. Both have an entity type of company, and the interpreter can match the argument structure of the θ-relation with the entity type slots of (1), resulting in a merger/acquisition classification being assigned to the story.

In attempting to determine a general categorization, the interpreter is encountering specific company names and, perhaps, their associated industry names. If these are contained in the concept base, they are also entered into the index. In this hypothetical example, if Software Inc. was listed in the concept base and related to the computer industry, then independent of the general classification, the final index would contain both the name of the company and its industry. In this way, a user can specify a particular company and receive all stories mentioning it, although there may not be any further index.

Since the mapping of the interpreter between the θ-relations of the parser and the concepts in the concept base is structure preserving, the items within indexes can also exhibit certain relationships. Arguments which are either an agent or object in a θ-relation will correspond to entity slots marked agent and object in a concept, respectively. Thus, the index will reflect the roles in which the participants are engaged, e.g., in a merger/acquisition the buyer and the acquired could be distinguished.

The next section provides several examples.

4.0 Examples

This section illustrates the type of indexes which NAS produces. The stories are from Reuters and the results are actual outputs from NAS.

Story 1

Montreal, Nov 3 - Air Canada's 8,500 groundworkers plan rotating strikes in the next few days following a collapse in contract talks with the government-owned airline earlier today, a union spokesman said.

Chief union negotiator Ron Fontaine said the workers will give 24 hours notice of a walkout but only two hours notice of which airports or maintenance centres they will strike.

The airline has warned that it will lock out any workers participating in rotating strikes until a new contract agreement is reached. The union last went on strike in 1978, shutting down the airline for two weeks.

Indexes:

Strikes and Lockouts
Industry - Airlines

The system has the concepts of strikes and lockouts and airlines industry in its concept base. The designator Airlines Industry is arrived at by a relation between Air Canada and its industry. The more general notion of Strikes and Lockouts appears as a frame in the concept base of the form:

(2) Strikes and Lockouts
Action: plan, participate
Agent: employee
Object: strike

where the action slot consists of plan and participate and the agent slot is of type employee of which groundworkers is so marked. The word strike is simply marked as strike. The parser returns a θ-relation for the first sentence with plan as a predicate, groundworkers as the agent, and strikes as the object. The θ-interpreter operates as described in the previous section and the Strikes and Lockouts frame is satisfied. Other typical results of processing by NAS are stories 2 and 3. Only the first sentence of each are provided since the remaining sentences of these news reports did not add any new information to the index.

Story 2

Valley Forge, Pa, November 3 - Alco Standard Corp. said it sold two of its gift and glassware companies for an undisclosed amount of cash to management groups in leveraged buyouts.

.
.
.

Indexes:

Divestment
Industry - Giftware
Trade - Glass

Story 3

Kuwait, November 3 - A booby-trapped car bomb exploded in Kuwait City on Tuesday morning, the official Kuwait news agency Kuna report.

.
.
.
.

Indexes:

Terrorism
Location - Kuwait
Instrument - Bombings

Since the details of indexing are identical to those above, they will be omitted here. However, it is noteworthy to indicate that the word divestment does not appear anywhere in Story 2. (Clearly, the verb sold alone could not trigger a divestment.) Similarly, in Story 3 terrorism is never used, yet NAS correctly indexes the story and also identifies the location and the weapon or instrument used.

5.0 Error Handling

There are several ways in which NAS can fail to perform an analysis. If the scanner reads an unknown word, it will trigger procedures in an attempt to infer its category. For instance, it will look ahead for abbreviations such as inc, corp, or co and if any of the strings are present, the scanner will assign name features with the immediately preceding unidentified words. (Ideally, in a fully deployed application, NAS would have interfaces to specialized databases of names, say, of companies.) Also, the lexical scanner, in failing to find a word in the lexicon and in the absence of certain triggers (e.g., inc), will label the unknown word a noun and pass the word to the parser in the sentence. This method for handling unknown words works well only if verbs, adjectives, and prepositions used in news reports are nearly exhaustively contained in the lexicon, and NAS has been extremely successful by using this technique.

Another potential problem for NAS is an incomplete or incorrect parse. Both cases often indicate insufficient information of a lexical item. However, during execution of NAS, if the parser cannot find a licensing relationship for a projection of an item in its input stream, it will move to the next word. This projection will remain unlicensed or uninterpreted. If the word has a semantic mark that may trigger a designator, the semantic processor will use it for constructing an index. For example, Yen is a low-level designator word and it is also semantically marked as currency. If the parser cannot license a projection containing this word to a verb or a preposition or, perhaps, misassigns a relation, the index will still contain Yen and currency report. What may be missing is a general categorization.

6.0 Future Directions

In addition to extending and enhancing the components of the semantic processor and parser, the near term efforts will focus on establishing quantitative benchmarks for both speed and accuracy using stories from an active newswire. While a pre-prototype of NAS with a different and less-sophisticated scanner and less-developed parser and semantic processor relied on stories from floppy disks or manual entry, the current version is linked to a live feed. A rough performance measure of the pre-prototype on a very small sample of less than 50 stories showed that it was completely correct for over 70% of the stories.

The present semantic processor operates on a much larger conceptual base and while it is premature to make assessments, the system has indexed one day of news stories from Reuters and the results were independently examined by a group of professional indexers. The indexers who had manually indexed the stories supplied over 400 topics for inclusion in the concept base of NAS, some of which were not relevant to any of the stories. There was no communication with these indexers before or during the process and while there were no formal criteria previously specified, the indexers found the results very promising. Currently, a precise evaluation metric for NAS is being formulated with these indexers.

Long-term work will include enhancement to the semantic processor and a refinement of its classification scheme. Inferencing across classified stories is also an option as well as the capability of allowing the user to query those processed stories (using the same parser). Automatic summarization of stories is also a future possibility.

7.0 Acknowledgements

Steve Cushing made valuable comments on an earlier draft of this paper. Dan Sullivan was a co-developer and implementor of the pre-prototype. On the present version of NAS, Steve Gander has made significant contributions to its design and implementation.

8.0 References

Abney, S., (1986), "Licensing and Parsing," (personal communication).

Berwick, R.C., (1987), Principle-Based Parsing, Technical Report 972, MIT Artificial Intelligence Laboratory, Cambridge, MA.

Chomsky, N., (1981), Lectures on Government and Binding, Foris Publications, Dordrecht, Holland.

Chomsky, N., (1986), Knowledge of Language, Praeger, New York, N.Y.

Hayes, P.J., L.E. Knecht, and M.J. Cellio, (1988), "A News Story Categorization System," Proceedings of the Second Conference on Applied Natural Language Processing, Austin, Texas.

Kashket, M.B., (1987), A Government-Binding Based Parser for Warlpiri, a Free-Word Order Language, Technical Report 993, MIT Artificial Intelligence Laboratory, Cambridge, MA.

Kuhns, R.J., (1986), "A Prolog Implementation of Government-Binding Theory," Proceedings of COLING'86, Bonn, West Germany.

Kuhns, R.J., (in preparation), "A Tree-less GB Parser," (tentative title).

Marcus, M.P., (1980), A Theory of Syntactic Recognition for Natural Language, The MIT Press, Cambridge, MA.

Milne, R., (1983), Resolving Lexical Ambiguity in a Deterministic Parser, D. Phil. Dissertation, University of Edinburgh.

Milne, R., (1986), "Resolving Lexical Ambiguity in a Deterministic Parser," Computational Linguistics, Vol. 12, No. 1.

Thiersch, C., and H.P. Kolb, (1987), "Parsing with Principles & Parameters: Prolegomena to a Universal Parser," (personal communication).

Wehrli, E., (1987), Parsing with a GB grammar, (personal communication).