In Proceedings of the 5th Pacific Rim Conference on Artificial
Intelligence (PRICAI 1998), November 1998, Singapore

Blitz:
A Preprocessor for Detecting Context-Independent Linguistic Structures[1]

Boris Katz Deniz Yuret Jimmy Lin Sue Felshin Rebecca Schulman Adnan Ilik

MIT Artificial Intelligence Laboratory
545 Technology Square,
Cambridge, MA 02139
USA
E-mail: {boris, deniz, jimmylin, sfelshin, rebecka, adnan}@ai.mit.edu

Abstract

The flow of natural language is often broken by constructions which are difficult to analyze with conventional linguistic parsers. To handle these constructions, which include numbers, dates, addresses, etc., and, to a lesser extent, proper nouns, NL systems typically implement specialized new rules. This leads to a level of complexity which renders maintenance or improvement difficult. Analyzing and tokenizing these constructions with an independent preprocessor can alleviate the burden on already taxed systems. Because these constructions have highly regular forms, strict structure, and can be largely understood in the absence of context, it is possible to shift the burden of processing away from the primary parser, and onto a simpler, faster, non-linguistic preprocessor.

This paper describes Blitz, a hybrid database- and heuristic-based natural language preprocessor, which has been integrated into the START Natural Language System in order to demonstrate how non-linguistic preprocessing can improve parsing. As a result, START's ability to analyze real-world sentences has improved considerably. Advantages of Blitz over existing systems are also discussed.

Introduction

Linguistically motivated NL parsers have been plagued by the vast complexity of language. Certain parsers which attempt to handle the richness of unrestricted language often grow to contain unmanageably large grammars. Other parsers which reduce language to a simple and tightly constrained linguistic model either cannot analyze syntactically odd structures or cannot decide between multiple interpretations. Certain natural language constructions contribute significantly to the complexity of grammars by introducing ambiguity, but are actually quite simple in form. These constructions, which include numbers, dates, times, addresses, emails, URLs, and proper nouns, do not exhibit the richly hierarchical structure of typical language constructions, and therefore lend themselves well to heuristic-based non-linguistic analysis in the absence of surrounding context. These structures can be analyzed by a preprocessing module and converted to single tokens, vastly speeding and simplifying parsing.

Other natural language constructions, particularly names, present a problem to linguistically motivated parsers because their structures are extraordinarily ambiguous and because they involve large numbers of ever-changing vocabulary items. For example, while a parser could contain rules such as NP → VP in order to parse sentences such as "Who directed Gone with the Wind," such rules will bog the parser down in generating endless absurd interpretations of input. A parsing system could store multi-word tokens like "Gone with the Wind" in its lexicon, but this would require storing thousands or even millions of lexical entries, and would also require constantly updating the parser-specific lexicon as new tokens are coined.

Blitz is a preprocessing system used by the START NL System developed at the MIT AI Laboratory. It is a hybrid preprocessor which uses both very simple heuristic rules and preconstructed symbol databases, or "symbol tables," to extract the above-mentioned constructions from free text and return results in a uniform frame structure. Blitz components are compartmentalized in separate layers, yielding a high customizable modular system. Ultimately, all frames are passed back to START (or any NL system), endowing it with the ability to understand sentences that it otherwise would not be able to understand.

Methodology

Blitz was developed with the following premises and philosophy:

[1] This research was supported in part by the Defense Advanced Research Projects Agency under Air Force Research Laboratory grant F30602-98-1-0036. The views expressed in this paper are those of the authors and do not reflect the views of the sponsoring agencies.

Minimal linguistic and lexical knowledge. The heuristic component of Blitz recognizes typographical properties (character position and case), certain closed classes of words, e.g., the names of the twelve months, cardinal and ordinal digits, etc., and employs very simple rules for generating constructions, e.g., a month name and an ordinal represent a date (“June 3rd”). These simple rules do not result in much overgeneration because most special constructions take highly defined forms. Other components which recognize fixed tokens access lists of symbols compiled from databases without reference to significant linguistic knowledge. For example, a database of famous people can be compiled into a list of proper name tokens.

S**upplementation, not Replacement.** Blitz was not designed as a standalone product, but rather as a component of a comprehensive NL system which assists in parsing and understanding. The natural language parser, equipped with greater syntactic and semantic knowledge, will consider each suggested token and attempt to incorporate it into the sentence.

Speed and robustness. The Blitz system was designed with speed and robustness in mind. Its structure as an integrated series of layers renders development of comprehensive and efficient systems easier.

Compartmentalization. Embodying the concept of compartmentalization in its system architecture, Blitz isolates each component from another, creating independent sections that can easily be interchanged and switched on or off. This architectural design allows Blitz to be specifically adapted to any application, e.g., when processing sports pages, the module which recognizes companies might be switched off. This compartmentalization strategy leads to a system that is easily fine-tuned, maintained, and improved.

Comprehensiveness and accuracy. Blitz recognizes a wide range of constructions which are syntactically impoverished and limited to a relatively small number of forms; not only are they composed from closed category lexical items, but in addition, it is possible to enumerate the rules for forming them. Blitz can very accurately extract the information within each recognized token, such as the value of a written number.

Recall is more important than precision. All suspected special constructions are detected by Blitz, even under the threat of overgeneration. However, this is acceptable because a true natural language parser will decide the final treatment of all tokenized constructions, employing semantic and linguistic knowledge.

Frames

Blitz communicates extracted tokens in the format of a “frame” which encodes the lexical information for the token, following this template: (type "*string*" :span (begin end) :attribute *value* ...)

The frame consists of *type* (the type of construction) and a quoted string of the construction, followed by *span* information, which maps the string to the character position within the input sentence, and then an arbitrary number of attribute and value tags pairs containing specific extracted information.

Heuristic Layer

The heuristic layer of the Blitz system consists of several independent modules. This design facilitates the removal, addition, or improvement of any module without drastic changes to the system architecture.

Email and URL

These email module recognizes the ubiquitous ‘@’ sign and checks for domain endings, and the URL module recognizes the limited set of URL prefixes such as “http://,” “mailto://,” “www.,” etc.

Numbers

The number module extracts a number and calculates its value, accomplished with very limited lexical knowledge. The following is a sample output of frames as constructed by Blitz.

Three hundred sixty-fourth

```
(number "three hundred sixty-fourth" :span (0 25) :value 364 :notation ordinal)
```

42.2 million

```
(number "42.2 million" :span (0 11) value 42.2+e7 :notation natural)
```

Because it is possible to write a single number with the conjunction “and,” it is difficult to separate cases of two actual numbers from one single number constructed with “and.” True disambiguation is often impossible, requiring additional insight offered by context and grammar, provided by the main parser.

Proper Nouns

In truth, the extraction and disambiguation of proper nouns is extremely difficult to accomplish in the absence of context. Proper nouns by their very nature are deeply intertwined with the basic lexical and semantic fabric of the sentence; hence it is difficult to understand and extract such information successfully without processing the entire sentence with a full parser. The follow five sentences poignantly demonstrate a small sample of such ambiguities.

(1) The New York Times is a newspaper.
(2) In The New York Times today there was an article about artificial intelligence.
(3) For Better or Worse is a popular comic strip.
(4) The copy of the New York Times John read was missing an entire section.
(5) Is Mary Joe Frank's daughter?

"The New York Times" is the full name of the popular newspaper, but it is impossible to derive such information except with *a priori* knowledge. The beginning of every sentence is capitalized; therefore heuristics cannot determine whether or not that word is part of a proper noun. This is also the problem encountered in sentence (2), where a preposition might be mistaken for part of the actual proper noun. In this case, disambiguation is difficult unless there exists a large list of common words that should be excluded from any proper noun, which might include all prepositions. However, even that scheme is far from foolproof, because prepositions can legitimately begin a proper noun, as in sentence (3). Sentence (4) further demonstrates compounded ambiguity when two proper nouns are adjacent to each other, unbroken by any punctuation. Once again, it is almost impossible to handle such cases without the benefit of a true parser. (Even then, parsers might have difficulty.) Finally, there are truly ambiguous sentences, such as (5), where it may be that Mary is the daughter of Joe Frank, or that Mary Joe is Frank's daughter.

The Blitz proper noun module looks for sequences of adjacent capitalized words that may potentially be separated by a very small list of connecting words such as "and," "the," and "of." All combinations of the entire token are then enumerated, in anticipation of the ambiguities mentioned above, e.g., New York Times would lead to "New York Times," "New York," "York Times," "New," "York," and "Times." Since the proper noun demon detects all combinations of capitalized words, it may return a large number of frames. "Confidence values," discussed in detail below, are used to choose among frames.

Time, Date, Address, and Quantity

These modules recognize clock time, calendar dates, street addresses, and quantities of measure. For example:

7:12 pm

```
(time "7:12 pm" :span (0 6) :hour 7 :minute 12 :time pm)
```

Friday, February 13, 1998

```
(date "Friday, February 13, 1998" :span (0 24) :day Friday :month February :date 13 :year 1998)
```

77 Massachusetts Avenue

```
(address "77 Massachusetts Avenue" :span (0 22) :number 77 :location "Massachusetts Avenue")
```

$23.5 billion

```
(quantity "$23.5 billion" :span (0 12) :value 2.35e+10 :unit $)
```

two feet

```
(quantity "two feet" :span (0 8) :value 2 :unit feet)
```

Symbol Table Layers

Because heuristics are only effective in extracting closed constructions delimited by strict forms, it is necessary for Blitz to incorporate other knowledge sources for the recognition of proper nouns which have no set form. The easiest way to accomplish this is through lists of symbols for individual categories; e.g., a list of all famous people, Fortune 500 Companies, movie titles, professional sports players, etc. The rich resources available on the World Wide Web make it possible to create such long symbol lists with relative ease.

In addition to simple heuristics, Blitz is currently coupled with multiple common proper nouns databases. When a sentences is preprocessed, it is checked against the database for matches, which are also packaged in frames and ultimately returned to the natural language system.

Compartmentalization is also relevant in the context of symbol tables. Due to the potentially huge number of symbols in each database and the number of databases, it is imperative to isolate knowledge sources from each other to ensure scalability and flexibility. For example, the movies database should be separate from the database of Fortune 500 Companies. This modularization of data assists in the management of complexity, making the addition and updating of individual databases easier.

There are several advantages to storing symbols in database format. The first is that large amounts of data can easily be added or changed, allowing great flexibility in preprocessing applications. More importantly, however, storing additional information about symbols is possible with this scheme. For instance, the symbol "Gone with the Wind," stored as a movie title, could also contain information about the director, date and cast of the movie. Such information, in addition to the fact that "Gone with the Wind" is a movie title, can be passed on to a natural language engine, and thus provide extra information about text. And for natural language systems on the World Wide Web, URL information can be included, so that such symbols may be hyperlinked.

Confidence and Conflict Resolution

Blitz overgenerates symbols because it works without regard to context, because some input is inherently ambiguous, and because identical symbols can be detected by more than one means (e.g., by both the proper noun demon and a symbol table of proper names). Blitz errs on the side of false positives when detecting symbols, leaving the natural language parser ultimate responsibility for ruling out unwanted symbols. Nevertheless, Blitz evaluates the likelihood and accuracy of frames, insofar as it can, to assist the parser, returning its calculations as “confidence” values within frames, expressed as decimals between zero and one.

Confidence in Isolation

In some cases, a heuristic module or symbol table can decide confidence without reference to other modules. For example, the proper noun demon always assign lower confidence to a proper noun at the beginning of the input, since the first word might be capitalized purely because it starts the sentence. Also, confidence values are adjusted accordingly if the input is in uniform uppercase or lowercase.

Relational Confidence

In some cases, Blitz can adjust the confidence of a frame based on the presence of another frame. For example, given "We went to a concert on May 1st," Blitz can lower the confidence on "May" as a month because a month name is highly unlikely to occur next to a possible date ordinal without being part of the larger date. On the other hand, given "Profits were high this year for Dewey, Cheatham, and Howe," Blitz has no way to assign higher probability to either the three-frame or one-frame interpretation (unless the symbol table contains the symbol).

Blitz also compares adjacent and overlapping frames in order to normalize their confidence values. For example, by default Blitz is more confident in longer frames, but it does not assign confidence by absolute length, but by length in relation to adjacent frames:

I read an article in the Boston Globe.

```
(propernoun "Boston" :confidence .5)
(propernoun "Globe" :confidence .5)
(propernoun "Boston Globe" :confidence 1)
```

I read an article in the Herald.

```
(propernoun "Herald" :confidence 1)
```

Combining Frames

Blitz can combine frames and reduce overgeneration, and in some cases can assign higher confidence when combining frames. For example, given "who wrote Gone With The Wind," Blitz can combine the heuristically derived proper name frame with the symbol table frames, reducing the total number of frames. As another example, Blitz will assign higher confidence to

[Michael Jordan] of the [Chicago Bulls]

in which both symbols (or at least one) can be found in symbol tables than to

[Michael Jordan of the Chicago Bulls]

in which the symbol is a legitimate proper noun, but does not exist in any symbol table. (In fact, if one includes enough databases, nearly everything becomes a symbol, which can also be weighted with confidence values.)

Using Confidence Values

A natural language parser will need some minor interface code in order to integrate information supplied by Blitz. The parser will likely want to combine its own lexical and syntactic knowledge with Blitz's confidence values in order to decide on the proper interpretation of the input. Thus it may prefer Blitz's interpretations in some cases:

* I saw [$_{det}$ The] [$_{pronoun}$ Who] in concert.
I saw [$_{NP}$ (propernoun "The Who")] in concert.

but not in others:

[$_{aux}$ May] I go now?
* [$_{NP}$ (date "May")] I go now?

START and Blitz

Integrating Blitz with the START Natural Language System has improved START's ability to handle real-world sentences dramatically. The Globe and AI Lab Servers, developed with the START Natural Language System, have been serving thousands of users and answering hundreds of thousands of questions on the World Wide Web since 1993.[2] Previously, unknown words and constructions would trigger an interaction such as the one below:

```
> Englebert Humperdinck wrote Mary Had A Little Lamb
          Could you phrase that a little differently, I didn't understand.
```

However, an integrated START system taking advantage Blitz's pre-processing ability does understand such sentences. In this case, the sentence is passed to the preprocessor, which tokenizes "Mary Had a Little Lamb" and "Englebert Humperdinck" as proper nouns. When this information is returned to START, the sentence is transformed into the equivalent of "*A wrote B*," which is then easily parsed.

```
> Englebert Humperdinck wrote Mary Had A Little Lamb
> Who wrote Mary Had A Little Lamb?
          Englebert Humperdinck wrote Mary Had A Little Lamb.
```

Furthermore, the addition of Blitz allows START to gracefully handle cases in which it does not have a specific answer to the query. Instead of returning a canned response like "*I don't understand. Please try another sentence.*" START will use the parse tree of the question to generate a response which conveys a sense of understanding, and when available will return Web links to symbols found in the input.

```
> Were there any unaccredited directors involved in Gone with the Wind?
```

I am not sure whether there were any unaccredited directors involved in Gone with the Wind, but I think you can find the relevant information here:

- Gone with the Wind

Discussion

There already exist several products which specialize in the extraction of proper nouns and names [1,2,4,5,7,8]; NetOwl by IsoQuest [8], and Nominator by IBM and the University of Pennsylvania [5] being the most notable. Both systems are continual entrants in ARPA's Named Entity Test at the Message Understanding Conferences, performing consistently well. However, the focus of the Blitz system differs somewhat from these other products. Although the relatively old concept of extracting names and proper nouns is incorporated into these previous systems, little work has been done in the understanding of other special constructions, which Blitz handles with remarkable precision. Furthermore, neither NetOwl nor Nominator aims towards natural language understanding, but towards somewhat less ambitious goals of automatic indexing, keyword extraction, and summary generation.

The Blitz system differs from other similar systems in three major ways:

1. **Diversity of Heuristics.** Instead of merely concentrating on proper nouns, Blitz has a wide range of heuristic rules to process a spectrum of constructions: numbers, dates, times, addresses, emails, URLs, and proper nouns.
2. **Integration With Natural Language Systems.** Although fully operational as a standalone product, the full potential of Blitz can only be tapped through integration with a natural language system. Combining Blitz with START, for example, creates a system vastly superior in functionality to the individual components. Blitz was created with the ambitious goal of facilitating natural language processing.
3. **System Architecture.** The compartmentalization strategy creates independent layers and modules which have very little dependence on each other, yielding a flexible, highly adaptable system. Isolating knowledge source with an abstraction barrier makes it possible to fine-tune Blitz to suit a wide range of purposes. This architecture is not present in similar systems.

Systems such as NetOwl achieve respectable levels of recall and precision at the cost of complexity and speed. NetOwl implements a large internal lexicon and over two hundred heuristic rules such as:

[2] http://www.ai.mit.edu/projects/infolab

Capitalized First Name + Capitalized Word = Person
Capitalized Word Sequence + Corporate Indicator = Company
…

Although there exist heuristic rules within Blitz as well, the strategy of compartmentalization isolates sets of similar heuristics into separate modules, providing a very effective control on the growth of complexity. Due to the wide variety of forms that special constructions can take, the most obvious solution is to implement new rules to handle the entire spectrum of such constructions. However, this technique can not effectively deal with the explosion of complexity, as any single construction can be interpreted differently by an increasing number of different rules. Blitz alleviates this problem through compartmentalization of knowledge sources and heuristic modules. Much of the conflict is resolved internally within each module, and furthermore through the application of confidence discussed earlier, only the most likely interpretations will be presented.

The architecture of the Blitz system and the "plug-and-play" nature of its individual parts allow fine tuning of the system for a variety of corpuses, increasing both performance and efficiency. The compartmentalization concept implies that each module or layer can be switched on or off trivially, allowing irrelevant parts to be discarded. For example, when parsing the Wall Street Journal, focus should be placed on the Fortune 500 Companies symbol table and the number demon. The symbol table of professional sports players and teams is probably irrelevant in the context of business articles, and therefore should be ignored. This ability for small adjustment demonstrates the versatility of Blitz.

Conclusion

The Blitz system not only demonstrates the validity of the preprocessing concept for the recognition of special constructions, but also the concrete advantages it offers in the ability to handle sentences that would otherwise fail using previous natural language parsers alone. Both the power and limitations of a hybrid heuristic-based system are evident in the performance of Blitz. More research is still required to study the need for further preprocessing and the integration of such systems with existing natural language parsers.

References

[1] Appelt, D., Hobbs, J., Bear, J., Israel, D., Kameyama, M., Tyson, M. "The SRI MUC-5 JV FASTUS Information Extraction System," *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, August 1993.

[2] Hayes, P. "NameFinder: Software the Finds Names in Text," *Proceedings of the 4th RIAO Conference of Computer Assisted Information Searching on the Internet (RIAO-94)*, October 1994.

[3] Katz, B. "Annotating the World Wide Web Using Natural Language," *Proceedings of the 5th RIAO Conference of Computer Assisted Information Searching on the Internet (RIAO-97)*, June 1997

[4] Lehnert, W., McCarthy J., Soderland, S., Riloff, E., Cardie, C., Peterson, J., Feng F. "UMASS/HUGHES: Description of the CIRCUS System Used for MUC-5," *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, August 1993

[5] Ravin, Y., and Wacholder, N. "Extracting Names From Natural-Language Text," *IBM Research Report RC 20338*, April 1997.

[6] Wacholder, N., Ravin, Y., Choi, M. "Disambiguation of Proper Names in Text," *IBM Research Report RC 20735*, February 1997.

[7] *Managing Text with Oracle8 ConText Cartridge*, Oracle Corporation, June 1997.

[8] *NetOwl Extractor Technical Overview*, IsoQuest, Inc., March 1997.