COREFERENCE RESOLUTION STRATEGIES FROM AN APPLICATION PERSPECTIVE

Lois C. Childs, David Dadd, Norris Heintzelman

Lockheed Martin Corporation

P.O. Box 8048
Philadelphia, PA 19101

lois.childs@lmco.com

(610) 354-5816

1. INTRODUCTION

As part of our TIPSTER III research program, we have continued our research into strategies to resolve coreferences within a free text document; this research was begun during our TIPSTER II research program. In the TIPSTER II Proceedings paper, "An Evaluation of Coreference Resolution Strategies for Acquiring Associated Information," the goal was to evaluate the contributions of various techniques for associating an entity with three types of information: 1) name variations, 2) descriptive phrases, and 3) location information. This paper discusses the evolution of the coreference resolution techniques of the NLToolset[1], as they have been applied to an information extraction application, similar to the MUC Scenario Template task. Development of this application motivated new coreference resolution algorithms which were specific to the type of entity being handled. It also has raised the importance of understanding the structure of a document in order to guide the coreference resolution process.

In the following paper, Section 2 discusses entity related coreference resolution techniques and Section 3, the relevance of document zoning. Section 4 concludes with a discussion of future work, which will include location merging, event coreference resolution, and event merging.

The NLToolset

The NLToolset is a framework of tools, techniques, and resources designed for building text processing applications. It is a pattern based system which uses world knowledge resident in a lexicon, a location gazetteer, and lists of universal terms, such as first names and Fortune 500 companies. This knowledge base is extensible with generic, as well as domain-specific, information. The NLToolset applies lexico-semantic pattern matching in the form of basic structural patterns (*possible-title firstname middle-initial lastname*), as well as contextual knowledge (*possible-person-name, who is X years old*). The NLToolset currently contains generic packages of rules to extract dates/times, percentages, money, phone/fax numbers, passports, identification numbers, social security numbers, person names, organization names, locations, vehicles, and drugs. It can extract from upper case and mixed case text.

The NLToolset has been applied to routing, indexing, name spotting, information extraction, and document management. It is an object-oriented system, implemented in C++ and ODBC to make it portable to both Unix and NT platforms, as well as multiple databases.

An Application

One application, developed with the NLToolset, extracts a complex template which describes cocaine seizures by law enforcement personnel. The application fills a template which holds information about the type of document (document identification,

[1] The NLToolset is a proprietary text processing product, owned by Lockheed Martin Corporation.

classification, and source), the entities involved (seizing and trafficking organizations, arrested persons), the drug information (amount, type, and method of concealment), the platform information (name and type of vehicle involved), the relevant locations (origin and destination of drugs, and place of seizure), and the date of the event.

An important step in merging all of the relevant information together into one template is the resolution of all coreferences among the relevant entities. In the following example, knowing what *the vessel* refers to would help in linking that information to the seizure event, without having to merge the finding and seizing events.

Drugs were found on board a cruise ship in Miami's harbor. Local law enforcement seized 17 kg. of cocaine on the vessel.

The entities involved in this application range over several types: persons, organizations, vehicles, drugs, locations, and the events themselves. In our work, we find it helpful to exploit the nature of the entity in finding its coreferences.

2. ENTITY COREFERENCE

Some coreference resolution techniques can be applied with only slight modifications across entity types. Identifying names and their variations is the first step in sorting out the person and organization entities. The NLToolset stores each newly recognized named entity, along with its computed variations and acronyms. The variations and acronyms are algorithmically generated, based on entity type, without reference to the text. For example, in general, person names can have nicknames, but organization names can have acronyms. (Persons sometimes also have acronyms, e.g *JFK*, but these are exceptions which must be stored as world knowledge.) Generated variations are stored in a temporary lexicon so that naturally occurring variations in the text can be recognized and linked to the original occurrence.

In linking noun phrases with named entities, the NLToolset has rule packages which find noun phrases of specific types: organization, person, vehicle, drugs. This allows the NLToolset to limit the search space for referents.

During context-based name recognition, entities are directly linked, via variable bindings within the patterns, with descriptive phrases that make up their context. These will be found in a set of four syntactic forms which are universal across entity types: appositives, predicate nominatives, prenominals, and name-modified head nouns.

APPOSITIVE: *Lockheed Martin, the aerospace giant,*

PREDICATE NOMINATIVE: *Lockheed Martin is a leader in information technology.*

PRENOMINAL: *the defense contractor, Lockheed Martin Corporation,*

NAME-MODIFIED HEAD NOUN: *the Lockheed Martin conglomerate*

These descriptive phrases can make up a document-specific ontology, or semantic filter, for the named entity which can be used to link isolated noun phrase references. This semantic filter had its origins in our TIPSTER II research on linking organization names with descriptive noun phrases.

Organizations

During our TIPSTER II research, it was found that organization names sometimes contain embedded semantic information which can be useful in resolving noun phrase coreferences. An experiment with the NLToolset's MUC6 performance, as reported in the TIPSTER II Proceedings, showed that using this information contributed five points of recall and seven of precision to the organization descriptor score. The technique used was to devise a semantic filter for an organization noun phrase and compare it to previous organization names to see if they can be linked. In the following example, the noun phrase and named organization have *jewel* references in common, which would be enough to link them.

Semantic Filters:

the jewelry chain	=>(jewelry jewel chain)
Smith Jewelers	=>(smith jewelers jeweler jewel)

If there is more than one candidate named entity, file position is considered as a factor, the closest name being the most likely referent.

Persons

As the NLToolset's coreference resolution techniques were expanded to other types of entities, it was found that previous methods would not always be applicable. Person names do not generally contain semantic information. For example, *John Smith* would not automatically be recognized as *a toilet manufacturer*. For this reason, the semantic filter must rely solely on syntactically linked semantic information. For persons, however, the standard set of four forms (appositive, prenominal, predicate nominative, and name-modified head noun) can be expanded to include person-specific information, such as titles, as in the following example.

The Judiciary Committee voted today on the impeachment of President Nixon. The president has announced that he will resign.

Vehicles

The vehicle category is problematic because entities are often referred to by the type of vehicle, rather than by a specific name. For example, an airplane name might be *Boeing 747* or *F-14*. Since it is possible to have several vehicles of the same type discussed in a document, all with the same "name," the NLToolset's standard name linking algorithm does not apply. The decision to link names must come later, at the event level, when more information is known.

Once the air vehicle names have been identified, airplane noun phrases are found and coreference resolution is performed, using the following algorithm. Assume that a noun phrase match belongs with the most recently seen entity, unless there is some contradictory information. If there is, then the current match is compared to the next most recently seen entity. If a match contradicts all previously seen entities, then it represents a new entity. The possible types of contradictory information currently are model information, manufacturer, military branch, airline, and flight number. The variable binding feature of the NLToolset pattern language allows the developer to extract type information during the name recognition process. For example, when the pattern for F-14 is constructed, the developer can inject the knowledge that plane types beginning with the letter F are considered fighter planes. This knowledge will allow the NLToolset to link the phrase "the fighter" to the named plane; moreover, it will prevent the phrase "the helicopter" from being linked. The algorithm for person and organization coreference resolution assumes that a noun phrase is not related unless there is some evidence to prove it, in direct contrast to that for vehicles.

Quantified Artifacts

Quantified artifacts, such as drug amounts, are handled with a straightforward algorithm that is usually successful, having achieved accuracy above 90% in the prototype application.

All measured amounts of drugs are identified as unique entities. Generic noun phrases then can refer to the last mention of a drug, based on the specificity of the drug type. For example, *the drugs* would refer to the last drug entity regardless of type, while *the cocaine* would refer to the last cocaine entity. An exception to the rule is the case where the noun phrase is actually referring to a group of drug amounts. In that case, context clues would need to be considered in order to handle that occurrence. This is an area that has been identified for improvement.

Measurement terms alone can indicate a drug amount within an ellipsis, as in the following example.

17 kg. of cocaine was found in the trunk of the car, while 2 kg. were found in the glove compartment.

To resolve this coreference to a common drug type, *cocaine*, the algorithm picks up the last mention of the drug from a drug stack, which keeps track of which drug was mentioned last.

A problematic case is that in which a drug seizure is referred to in general terms, giving the total amount of drugs seized, and then gives a breakdown of the amounts. The NLToolset will identify all measured drug amounts as unique. Currently, there are no heuristics to check on redundancy of seizure information, based on quantity captured. This will be an area to explore in future work, as the prototype is brought to an operational level.

3. DOCUMENT ZONING

During the development of the drug seizure application, it became apparent that knowledge of the structure of the document would be of help in limiting the coreference resolution to semantically related zones. Often, a document is sent to convey information on multiple topics and/or locations. If the text processing system does not recognize a topic shift, it may incorrectly relate unrelated information. The challenge is to zone the document before information extraction begins. As with most text-processing problems, zoning must be determined via both structure and meaning, i.e. the syntax and semantics of the document.

Authors often use visual cues, such as skipped lines and indentation to alert the reader to shifts into new topics. Since text processing techniques have been developed for character streams, it is difficult for them to interpret the visual cues that are two dimensional in nature, rather than linear. Our current work is seeking to apply image understanding techniques to this problem by constructing an auxiliary grid representation of the text and applying two dimensional pattern matching, in order to extract the nature of the document's structure.

Knowledge of the structure must then be supplemented with knowledge of the semantics of the structure. This will make it possible for the text processing system to go beyond the structural components of paragraph and table to find the semantic zones of the document which tie structural components together. For example, a single word at the beginning of a paragraph may have significance only because of the fact that it is a country name.

SPAIN

LOCAL POLICE HAVE SEIZED ...

Depending on the source of the document, the author may insert outline characters to help the reader interpret the structure of the story.

A. PUERTO RICO: ON JULY 5, MARITIME OFFICERS ...

The outline styles are often standard forms which have been tailored by the author, so an automatic system must be able to interpret varying styles. In the following, the location will span several blocks of text, marked alphabetically.

1. CALIFORNIA

A. JULY 5, OFFICERS SUSPECTED ...

B. JULY 7, LOCAL LAW ENFORCEMENT SEIZED ...

Pattern matching techniques are being applied at the tokenizer level to identify structure markers and look for the outline patterns.

Zoning is a topic of ongoing research which is also being applied to the problem of tabular information.

BlockFinder

Image understanding techniques have been applied to the problem of recognizing the structure of text. BlockFinder, a prototype of a new NLToolset tool, uses two-dimensional patterns to find the edges in a grid of text. It converts an input text file into a list of blocks separated by white space.

The BlockFinder is the component of NLToolset that looks at a text file from a two dimensional perspective. Characters from the file are arranged in a two dimensional grid where the rows of the grid are separated by newline characters. By treating this character grid as an image, it is possible to find sections of text which are isolated by white space. Now that the computer has a representation of a file that reflects how the characters would appear on a page, it is feasible to look "above" and "below" characters in a file to find boundaries where text meets white space. In this way the BlockFinder can pick up on zoning cues which are obvious to a human reader but which have proved elusive for computers.

Here is an overview of the BlockFinder algorithm.

1. Characters from the file stream are inserted into a 2-D array. A newline character starts a new line of the array. Tab characters insert white space up to the next eight character tab stop.
2. Each character is classified as text, punctuation, or white space.
3. White space consistent with normal word spacing within a block of text is filtered out. These space characters are reclassified as text.
4. Punctuation consistent with standard English is filtered out. These punctuation characters are reclassified as text.
5. The boundaries between text and non-text characters are marked as edges.
6. Adjacent edges are linked together to form longer straight edges.
7. The long straight edges are grouped to trace the boundaries of text blocks.
8. These text blocks are flagged as document zones.

Most of the blocks detected by the BlockFinder are sections and paragraphs within a document. These blocks are continuous; they can be represented by a start and end position in the file stream. The BlockFinder finds other (non-continuous) blocks as well. A primary example of a non-continuous block is a column of a table. Work is underway to have the BlockFinder isolate and organize these column blocks into a table structure that would allow the NLToolset to interpret tabular data.

Whether the identified blocks represent tabular columns, paragraphs, or sections of a document, they contain important clues to the document's organization. These clues help the human reader to understand the document. The BlockFinder allows the NLToolset to use these same clues to break the document into logical zones which should, in turn, improve the quality of the coreferences generated.

Outline Matching

The author of a document has an almost infinite variety of conventions from which to choose to indicate text grouping. Sections, sub-sections, or paragraphs can be separated by blank lines, or by outline symbols, or by some arbitrary indentation with no blank lines. A system that can also use outline characters and indentation as well as blocks will be more successful than one that works with blocks alone.

The outline hierarchy of a document is indicated by the order in which the outline symbols appear.

In our prototype, during tokenization, an outline label (a letter or number) is recognized as one or two digits or letters followed by a "." at the beginning of a line. The tokenizer then inserts the title "outline-letter," "outline-number," or "outline-roman."[2]

Pattern matching is used to determine hierarchy by position in the file. If the first occurrence of an outline title is a Roman numeral, then we know that Roman numerals are being used as the top outline level. Similarly, if the second type of outline title to appear (that is not Roman) is an outline letter and lastly an outline number, then we have identified the style. This pattern matching is used to create new labels that indicate hierarchy: outline1, outline2, outline3, etc. Next, we simply find and group each outline label and the text associated with it into component objects.

Internal structures are used to group the outline components into parent-child relationships that represent zone structure.

Indentation

Next, we intend to look at indentation to indicate the breaking of a block of text into smaller units (i.e. paragraph). In our prototype, if the indentation is greater at the break point than the indentation at the start of the containing block, the new units will be grouped as children of the containing block. Otherwise, the new units are siblings to the containing block.

Semantic Zones

The idea behind using the blocks, outlines, and indentation, is to create the basic document structure first, refining at each step, using the new information. Once the structure has been built, we will use semantic pattern matching to determine the meaning of the structure based on prior information concerning the document style. For example, in the case of the application under discussion, document sections are sometimes marked with location names that have either no outline labels or an outline1 label. So, the collection of components that start with a location name is a document section, and it and all its children can be treated as a single zone.

When the document sections and subsections have been identified, the code can verify that the reference token retrieved is in the same document section as the current token. If it is not, than it is not an accurate reference. Also, since the location in the section header has been identified, it is clearly the default location for any event found in that section.

4. CONCLUSIONS AND FUTURE WORK

Research is ongoing to expand the capability of the NLToolset's coreference resolution module.

Location Merging

The location template for the drug seizure application contains slots to hold information about the locale (a descriptive, such as *Highway 40*), the city, state, country, latitude/longitude, region or body of water. To extract a complete representation of the location for an event, the NLToolset must collect all location references and merge them into a complete description of the location. In the following example, a pattern to extract seizure information may pick up both the city, *San Felipe*, and the locale, *Highway 32*. These must then be merged into one template, based on the knowledge that they are related via the seizure extraction pattern. Additionally, the country information must then be added. While it is possible to use the gazetteer to look up city names in order to find the associated country, sometimes a city name has been used in more than one country, and other information, such as zoning information, must be used to disambiguate. Another problem is that not all events occur in large cities; small towns are not usually listed in the gazetteer.

1. BOLIVIA

A. LA PAZ: ON JULY 8 COAST GUARD PATROLS SIGHTED ...

B. SAN FELIPE: JULY 7, LOCAL LAW ENFORCEMENT OFFICERS SEIZED 2 TONS OF COCAINE ON HIGHWAY 32.

[2] Exceptions are made for single digit Roman numerals, I, V, X, etc., which can either represent Roman numerals or letters, depending on the context.

Location merging capability, based on event and zoning information, will be added to the NLToolset in the near future.

Event Coreference Resolution

During development of the application prototype, event coreference resolution was identified as a necessary technique to better the accuracy of the system. The following example illustrates the problem.

17 KG. OF COCAINE WERE SEIZED IN MIAMI. THE OPERATION WAS CONDUCTED BY A TEAM CONSISTING OF THE FBI, THE COAST GUARD, AND LOCAL AUTHORITIES.

In order to tie in the seizing organizations to the seizure event, the system must be able to identify the referent of *operation* as the entire seizure event. This is coreference resolution at a later stage of processing than that for entities; it must occur after the main events have been identified. The plan is to apply patterns which match nominalized event forms, and to link them to the known events, based on zoning information.

Event Merging

Event merging is a challenging part of extracting complex scenario templates. Authors usually spread information across several sentences, depending on the understanding of the reader to link the related information. The following example illustrates this point.

17 KG OF COCAINE WAS SEIZED ON THE HMS PINAFORE. THE VESSEL HAD EMBARKED FROM CALI AND WAS HEADED FOR MIAMI.

Thoroughly understanding the text is not something that automatic text processing systems currently do successfully. In fact, the most successful information extraction systems long ago gave up the goal of completely understanding free text. Targeted extraction of relevant information has been the most fruitful strategy, thus far.

To continue in this tradition, our TIPSTER research has identified two techniques to investigate as solutions to the event merging problem. The first is entity-based event merging. This technique is based on the observation that the entity coreference resolution can act as a vehicle for linking secondary information. In the previous example, having linked *the vessel* with the platform *HMS Pinafore* would allow the origin and destination of the vessel to migrate back to the extracted seizure event via the coreference chain.

The second technique to be developed is based on the idea that a particular event is usually composed of a finite set of predictable activities. For example, a successful Coast Guard seizure operation may be composed of patrolling, boarding, arresting, and seizing activities. This is not a new idea in the field of Artificial Intelligence.

Since extracting isolated event information is something that the NLToolset does very well, it is thought that a profile of an event can be modeled. The profile would consist of a main event and its associated events. The NLToolset could then merge the extracted information based on the compatibility of its participating entities and zoning information. Something like this was developed on a limited basis for the joint venture scenario template of the original TIPSTER program. In that case, the LISP version of the NLToolset allowed ownership information to be merged into the main event of joint venture based on entity compatibility.

The differences between the two techniques, entity-based and profile-based event merging, are subtle. Both require the construction of patterns for extracting associated event information. The main difference is that, in the former, the associated information, e.g. vessel destination, is tied to the entities involved. This method does not preclude the possibility that an entity may be involved in more than one event; however, event merging, as a step after event extraction, is not required.

With profile-based event merging, the entity information is kept associated with the extracted event and merging takes place after all events have been extracted. As the application is expanded to handle more than one type of main event, there may be overlaps among the profiled subevents.

Both techniques will be investigated under the remainder of the current TIPSTER research effort.

Summary

This paper has discussed the evolution of the coreference resolution techniques of the NLToolset, as they have been applied to an information extraction application, similar to the MUC Scenario Template tasks. It has also discussed current work on understanding document structure, as well as future work on improving information merging techniques.