Integration of Document Detection and Information Extraction

Louise Guthrie
Lockheed Martin Corporation

Tomek Strzalkowski, Wang Jin and Fang Lin
GE Corporate Research and Development
Schenectady, NY 12301

ABSTRACT

We have conducted a number of experiments to evaluate various modes of building an integrated detection/extraction system. The experiments were performed using SMART system as baseline. The goal was to determine if advanced information extraction methods can improve recall and precision of document detection. We identified the following two modes of integration:

I. Extraction to Detection: broad-coverage extraction

1. Extraction step: identify concepts for indexing
2. Detection step 1: low recall, high initial precision
3. Detection step 2: automatic relevance feedback using top N retrieved documents to regain recall.

II. Detection to Extraction: query-specific extraction

1. Detection step 1: high recall, low precision run
2. Extraction step: learn concept(s) from query and retrieved subcollection
3. Detection step 2: re-rank the subcollection to increase precision

Our integration effort concentrated on mode I, and the following issues:

1. use of shallow but fast NLP for phrase extractions and disambiguation in place of a full syntactic parser
2. use existing MUC-6 extraction capabilities to index a retrieval collection
3. mixed Boolean/soft match retrieval model
4. create a Universal Spotter algorithm for learning arbitrary concepts

LEXICO-SEMANTIC PATTERN MATCHING FOR SHALLOW NLP

The lexico-semantic pattern matching method allows for capturing of word sequences in text using a simple pattern language that can be compiled into a set of non-deterministic finite automata. Each automoton represents a single rule within the language, with several related rules forming a package. As a result of matching a rule against the input, a series of variables within the rule are bound to lexical elements in text. These bindings are subsequently used to generate single-word and/or multiple-word terms for indexing.

Long phrasal terms are decomposed into pairs in two phases as follows. In the first phase, only unambiguous pairs are collected, while all longer and potentially structurally ambiguous noun phrases are passed to the second phase. In the second phase, the distributional statistics gathered in the first phase are used to predict the strength of alternative two-word sub-components within long phrases. For example, we may have multiple unambiguous occurrences of "insider trading", while very few of "trading case". At the same time, there are numerous phrases such as "insider trading case", "insider trading legislation", etc., where the pair "insider trading" remains stable while the other elements get changed, and significantly fewer cases where, say, "trading case" is constant and the other words change.

The experiments performed on a subset of U.S. PTO's patent database show healthy 10%+ increase in average precision over baseline SMART system. The average precision (11-point) has increased from 49% SMART baseline on the test sample to 56%. Precision at 5 top retrieved documents jumped from 48% to 52%. We also noticed that phrase disambiguation step was critical for improved precision.

INDEXING WITH MUC-6 CONCEPTS

In these experiments we used actual MUC organization and people name spotter (from Lockheed Martin) to annotate and index a subset of TREC-4 collection. We selected 17 queries out of 250 TREC topics which explicitely mentioned some organizations by names. The following observations were made:

1. Different queries require different concepts to be spotted: concepts that are universal enough to be important in most domains are hard to find, or not discriminating enough.
2. These differences are frequently query-specific, not just domain-specific, which makes MUC-style extraction impractical
3. The role that a concept plays in a query can affect its usefullness in retrieval: concepts found in focus appear to be radically more discriminating than those found in background roles.

Initial results show that targeted concept indexing can be extremely effective, however, random annotation may in fact cause loss of performance. Overall, the average precision improved by only 3%; however, some queries, namely those where the indexed concepts were in focus roles, benefited dramatically. For example, the query about Mitsubishi has gained about 25% in precision over SMART baseline (from 42% to 52%).

Typical results are summarized in the table below:

	words	annotations	both	merge
Av.PREC	34.1%	18.3%	28.1%	35.5%
REC@50	67%	31%	66%	67%

MIXED BOOLEAN/SOFT RETRIEVAL MODEL

We allow strict-match terms to be included in the search queries in a specially designated field. The hard/soft query mechanism allows a user to specify either in interactive or batch mode a boolean type query which will restrict documents returned by a vector space model match. Documents not satisfying the query will be deemed to be non-relevant for the query.

A two-pass retrieval has been implemented in SMART to allow proper interpretations of such queries. In interactive mode a normal vector query can be entered using 'run' command. When the first results are returned using 'boolean' will place you in editor mode (similar to run). Construct the query and terminate the query with a period on a line by itself. The documents returned by the latest 'run' command are filtered and only those satisfying the query are redisplayed. Using 'more' will always retrieve 'num_wanted' unless there are insufficient documents remaining that are relevant to the initial vector query.

RECOMMENDATIONS FOR AN INTEGRATED SYSTEM

The following were determined to be crucial in building an integrated extraction/detection system:

1. A large variety of extraction capabilities, best if could be generated rapidly on an ad-hoc basis.

2. Rapid discource analysis for role determination of semantically significant terms

3. The need for well-defined equivalence relation on annotations produced by an extraction system.

4. Use of mixed Boolean/soft retrieval model

UNIVERSAL SPOTTER

Identifying concepts in natural language text is an important information extraction task. Depending upon the current information needs one may be interested in finding all references to people, locations, dates, organizations, companies, products, equipment, and so on. These concepts, along with their classification, can be used to index any given text for search or categorization purposes, to generate summaries, or to populate database records. However, automating the process of concept identification in unformatted text has not been an easy task. Various single-purpose spotters have been developed for specific types of concepts, including people names, company names, location names, dates, etc. but those were usually either hand crafted for particular applications or domains, or were heavily relying on apriori lexical clues, such as keywords (e.g., `Co.'), case (e.g., `John K. Big'), predicatable format (e.g., 123 Maple Street), or a combination of thereof. This makes creation and

extension of such spotters an arduous manual job. Other, less salient entities, such as products, equipment, foodstuff, or generic references of any kind (e.g., `a Japanese automaker') could only be identified if a sufficiently detailed domain model was available.

We take a somewhat different approach to identify various types of text entities, both generic and specific, without a detailed understanding of the text domain, and relying instead on a combination of shallow linguistic processing (to identify candidate lexical entities), statistical knowledge acquisition, unsupervised learning techniques, and possibly broad (universal but often shallow) knowledge sources, such as on-line dictionaries (e.g., WordNet, Comlex, OALD, etc.). Our method moves beyond the traditional name spotters and towards a universal spotter where the requirements on what to spot can be specified as input parameters, and a specific-purpose spotter could be generated automatically. In this paper, we describe a method of creating spotters for entities of a specified category given only initial seed examples, and using an unsupervised learning process to discover rules for finding more instances of the concept. At this time we place no limit on what kind of things one may want to build a spotter for, although our experiments thus far concentrated on entities customarily referred to with noun phrases, e.g., equipment (e.g., "gas turbine assembly"), tools (e.g., "adjustable wrench"), products (e.g., "canned soup", "Arm Hammer baking soda"), organizations (e.g., American Medical Association), locations (e.g., Albany County Airport), people (e.g., Bill Clinton), and so on. We view the semantic categorization problem as a case of disambiguation, where for each lexical entity considered (words, phrases, N-grams), a binary decision has to be made whether or not it is an instance of the semantic type we are interested in. The problem of semantic tagging is thus reduced to the problem of partitioning the space of lexical entities into those that are used in the desired sense, and those that are not. We should note here that it is acceptable for homonym entities to have different classification depending upon the context in which they are used. Just as the word ``bank" can be assigned different senses in different contexts, so can ``Boeing 777 jet" be once a product, and another time an equipment and not a product, depending upon the context. Other entities may be less context dependent (e.g., company names) if their definitions are based on internal context (e.g., "ends with Co.") as opposed to external context (e.g., "followed by manufactures"), or if they lack negative contexts.

The user provides the initial information (seed) about what kind of things he wishes to identify in text. This information should be in a form of a typical lexical context in which the entities to be spotted occur, e.g., "the name ends with Co.", or "to the right of produced or made", or ``to the right of maker of", and so forth, or simply by listing or highlighting a number of examples in text. In addition, negative examples can be given, if known, to eliminate certain `obvious' exceptions, e.g., "not to the right of made for ", "not toothbrushes". Given a sufficiently large training corpus, an unsupervised learning process is initiated in which the system will: (1) generate initial context rules from the seed examples; (2) find further instances of the sought-after concept using the initial context while maximizing recall and precision; (3) find additional contexts in which these entities occur; and (4) expand the current context rules based on selected new contexts to find even more entities.

We present and evaluate preliminary results of creating spotters for organizations and products.

What do you want to find: seed selection

If we want to identify some things in a stream of text, we first need to learn how to distinguish them from other items. For example, company names are usually capitalized and often end with `Co.', `Corp.', `Inc.' and so forth. Place names, such as cities, are normally capitalized, sometimes are followed by a state abbreviation (as in Albany, NY), and may be preceded by locative prepositions (e.g., in, at, from, to). Products may have no distinctive lexical appearance, but they tend to be associated with verbs such as `produce', `manufacture', `make', `sell', etc., which in turn may involve a company name. Other concepts, such as equipment or materials, have few if any obvious associations with the surrounding text, and one may prefer just to point them out directly to the learning program. There are texts, e.g., technical manuals, where such specialized entities occur more often than elsewhere, and it may be advantagous to use these texts to derive spotters.

The seed can be obtained either by hand tagging some text or using a naive spotter that has high precision but presumably low recall. A naive spotter may contain simple contextual rules such as those mentioned above, e.g., for organizations: a noun phrases ending with "Co." or "Inc."; for products: a

noun phrase following "manufacturer of", "producer of", or "retailer of". When such naive spotter is difficult to come by, one may resort to hand tagging. From seeds to spotters

The seed should identify the sought-after entities with a high precision (though not necessarily 100%), however its recall is assumed to be low, or else we would already have a good spotter. Our task is now to increase the recall while maintaining (or even increase if possible) the precision.

We proceed by examining the lexical context in which the seed entities occur. In the simplest instance of this process we consider a context to consist of N words to the left of the seed and N words to the right of the seed, as well as the words in the seed itself. Each piece of significant contextual evidence is then weighted against its distribution in the balance of the training corpus. This in turn leads to selection of some contexts to serve as indicators of relevant entities, in other words, they become the initial rules of the emerging spotter.

As an example, let's consider building a spotter for company names, starting with seeds as illustrated in the following fragments (with seed contexts highlighted):

... HENRY KAUFMAN is president of Henry Kaufman Co. , a ... Gabelli, chairman of Gabelli Funds Inc. ; Claude N. Rosenberg ... is named president of Skandinaviska Enskilda Banken ... become vice chairman of the state-owned electronics giant Thomson S.A. ... banking group, said the formal merger of Skanska Banken into ... water maker Source Perrier S.A., according to French stock ...

Having "Co.", "Inc." to pick out ``Henry Kaufman Co." and "Gabelli Funds Inc." as seeds, we proceed to find new evidence in the training corpus, using an unsupervised learning process, and discover that ``chairman of" and ``president of" are very likely to precede company names. We expand our initial set of rules, which allows us to spot more companies:

... HENRY KAUFMAN is president of Henry Kaufman Co. , a ... Gabelli, chairman of Gabelli Funds Inc. ; Claude N. Rosenberg ... is named president of Skandinaviska Enskilda Banken ... become vice chairman of the state-owned electronics giant Thomson S.A. ... banking group, said the formal merger of Skanska Banken into ... water maker Source Perrier S.A., according to French stock ...

This evidence discovery can be repeated in a bootstrapping process by replacing the initial set of seeds with the new set of entities obtained from the last iteration. In the above example, we now have "Skandinaviska Enskilda Banken" and "the state-owned electronics giant Thomson S.A." in addition to the initial two names. A further iteration may add "S.A." and ``Banken" to the set of contextual rules, and so forth. In general, entities can be both added and deleted from the evolving set of examples, depending on how exactly the evidence is weighted and combined. The details are explained in the following sections.

Text preparation

In most cases the text needs to be preprocessed to isolate basic lexical tokens (words, abbreviations, symbols, annotations, etc), and structural units (sections, paragraphs, sentences) whenever applicable. In addition, part-of-speech tagging is usually desirable, in which case the tagger may need to be re-trained on a text sample to optimize its performance. Finally, a limited amount of lexical normalization, or stemming, may be performed. The entities we are looking for may be expressed by certain types of phrases. For example, people names are usually sequences of proper nouns, while equipment names are contained within noun phrases, e.g., `forward looking infrared radar'. We use part of speech information to delineate those sequences of lexical tokens that are likely to contain `our' entities. From then on we restrict any further processing on these sequences, and their contexts.

These preparatory steps are desirable since they reduce the amount of noise through which the learning process needs to plow, but they are not, strictly speaking, necessary. Further experiments are required to determine the level of preprocessing required to optimize the performance of the Universal Spotter.

Evidence items

The semantic categorization problem described here displays some parallels to the word sense disambiguation problem where homonym words need to be assigned to one of several possible senses. There are two important differences, however. First, in the semantic categorization

problem, there is at least one open-ended category serving as a grab bag for all things non-relevant. This category may be hard, if not impossible, to describe by any finite set of rules. Second, unlike the word sense disambiguation where the items to be classified are known apriori, we attempt to accomplish two things at the same time: discover the items to be considered for categorization; actually decide if an item belongs to a given category, or falls outside of it. The categorization of a lexical token as belonging to a given semantic class is based upon the information provided by the words occurring in the token itself, as well as the words that precede and follow it in text. In addition, positional relationships among these words may be of importance.

Experiments and Results

We used the Universal Spotter to find organizations and products in a 7 MBytes corpus consisting of articles from the Wall Street Journal. First, we pre-processed the text with a part-of-speech tagger and identified all simple noun groups to be used as candidate phrases. 10 articles were set aside and hand tagged as key for evaluation. Subsequently, seeds were constructed manually in form of contextual rules. For organizations, these initial rules had a 98% precision and 49% recall; for products, the corresponding numbers were 97% and 42%. No lexicon verification has been used in order to show more clearly the behavior the learning method itself (the performance can be enhanced by lexicon verification). The seeds that we used in our experiments are quite simple, perhaps too simple. Better seeds may be needed (possibly developed through an interaction with the user) to obtain strong results for some categories of concepts.

For organization tagging, the recall and precision results obtained after the fourth bootstrapping cycle are 90% and 95%, respectively. Examples of extracted organizations include: "the State Statistical Institute Istat", "Wertheim Schroder Co", "Skandinaviska Enskilda Banken", "Statistics Canada" .

The results for products tagging are at 80% recall at 85% precision, and 75% recall at 90% precision. Examples of extracted products include: "the Mercury Grand Marquis and Ford Crown Victoria cars", "Chevrolet Prizm", "Pump shoe", "AS/400" .

ACKNOWLEDGEMENTS

This paper is based upon work supported by the Advanced Research Projects Agency under Tipster Phase-2 Contract 94-F-133200-000 to Lockheed Martin Corporation, under a subcontract to GE Corporate Research and Development.

REFERENCES

[1] Brown,P., S. Pietra, V. Pietra and R. Mercer. 1991. Word Sense Disambiguation Using Statistical Methods. Proceedings of the 29h Annual Meeting of the Association for Computational Linguistics , pp. 264-270.

[2] Gale, W., K. Church and D. Yarowsky. 1992. A Method for Disambiguating Word Senses in a Large Corpus. Computers and the Humanities , 26, pp. 415--439.

[3] Harman, D. 1995. Overview of the Third Text REtrieval Conference. Overview of the Third Text REtrieval Conference (TREC-3) , pp.1-20.

[4] Strzalkowski, T. 1995. Natural Language Information Retrieval. Information Processing and Management, vol. 31, no. 3 , pp. 397-417.

[5] Yarowsky, D. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics , pp. 189-196.