Using Grammatical Relations for Document Retrieval and Analysis

Kenneth B. Haase
MIT Media Laboratory
20 Ames Street
Cambridge, MA 02139
(617)253-7418
haase@media.mit.edu

This article describes the use of grammatical information in information retrieval and extraction from a free text database. This information can be used both to increase the precision of recall in certain circumstances and for identifying analogous elements of documents. The identification of analogous elements allows content extraction with the need for hand-coded or automatically induced templates. Extracted content can then be used for automatic transcoding of information from textual to graphical forms.

1 Introduction

This article describes the use of grammatical information in information retrieval and extraction from a free text database. This information can be used both to increase the precision and recall in certain circumstances and for identifying analogous elements of documents.

The grammatical information is produced by a two-pass process and stored as *modification graphs* encoding relations between individual words and explicitly representing the ambiguity of preposition and clause attachment. A straightforward indexing algorithm is used for querying this database where the keys in the queries are words in particular grammatical relations rather than individual words alone.

While these sorts of queries appear to provide some advantage in conventional retrieval tasks, we use them primarily for determining *analogical correspondences* between texts in order to pull out particular content items without the need for hand-coding or automatically inducing templates. Such content extraction allows the transcoding of information from (for instance) textual to graphical forms. We present one example of this sort of operation.

2 Discriminating subject and content

In our work on information retrieval in general and text retrieval in particular, we distinguish the *subject* and *content* of documents. The subject of a document is what it is about: the individuals, organizations, locations, and kinds of events it describes. The content of a document is a richer property which includes how the document's subjects interrelate as well as the identity of those subjects.

Consider a story about a stock accquisition. Its 'subject' might be comprised of the companies involved in the accquistion, the market in which it occurred, and something about it being an accquisition. However the content, we would say, lies in who did the accquiring from whom of what: the roles which the subjects play. Further, the content includes particulars (like the purchase price) which would be useless unless both attached to particular events and distinguished from other similar particulars (like the originally offered price).

For many information retrieval tasks, discrimination on subject is all that is neccessary. Queries based on subject alone are often successful because there is only one 'content configuration' in the database corresponding to the subjects involved; a query on 'Kennedy assassination' has only one actual content interpretation in a news database.

But in some cases, this sort of content is vital, as in distinguishing things that 'Clinton said' from things that were said to or about Clinton, by Clinton staffers, or in Clinton, Massachussetts. We are interested in content understanding both as it improves information retrieval and allows for different sort of information functions to be implemented.

3 Building a relational textbase

We have chosen to address the discrimination of content by constructing what we call a 'relational textbase' comprised of free text documents indexed by both the words within

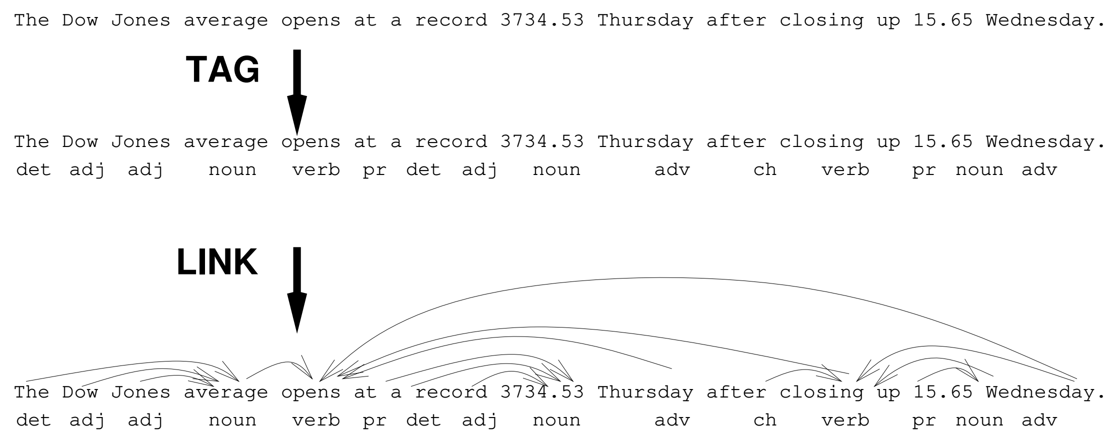

Figure 1: Modification graph representations are produced by a two-pass process: part of speech assignment based on a probabilistic grammar and a simple lexicon; and word linking based on simple linkage rules.

them and grammatical relations between these words. Queries to this database can take the form 'Get documents where the word 'Clinton' is the subject of the word 'said'." By parsing natural language requests and querying for documents with relations identical to those in the query, natural language queries to such a database are possible, since a parse of "What has Clinton said?" will include many of the same relations as "President Clinton, after meeting with French President Francois Mitterrand Tuesday, said the richest industrial nations...". In the next months, we will be combining the relational textbase with a thesaurus subcategorized by relational criteria; this allows us to identify synonymyous words only when they occur in particular relations to other words.

Our relational textbase currently consists of 370,000 words of one-paragraph news summaries from Gannett's USA-TODAY, provided on-line via Clarinet. These summaries are parsed to produce a 'modification graph' representation of their grammatical content. Modification graphs are a variant of *dependency graphs* used in [Meĺčuk, 1988] and [Sleator and Temperly, 1991]. Modification graphs differ in allowing the explicit representation of ambiguity in preposition and dependent clause attachement.

The modification graphs for the texts in the database are stored and indexed in an in-house representation system called FRAMER [Haase, 1993]. FRAMER's algorithms allow the fast retrieval of structures with locally similar graph structure. This retrieval process is used in two ways: for simple relational queries (like those described above) and for analogical content extraction (which will be described below).

The modification graph representations are produced by a two-pass process: a first pass assigns part of speech information based on a probabilistic grammar; then a second pass uses this information to connect words together based on simple linkage rules (such as adjectives modify a subsequent noun, or nouns as subjects of subsequent verbs). Figure 1 illustrates this process. The resulting modification graph preserves ambiguity in preposition and dependent clause attachement while effectively identifying primary actors and events (and their possible relations). The parsing process is relatively efficient (about 12 sentences/minute for a LISP implementation running on a 30 MIPS Decstation 5000) and could easily be optimized by reimplementation in C.

The modification graphs are stored in FRAMER, a perisistent knowledge representation language developed and widely used at the Media Lab; an in-house database was chosen to allow experimentation in the design of indexing algorithms. The current indexing approach combines word-stemming with a simple inverted index where the keys are created for each $\langle root, relation, root\rangle$ triplet; similarity is scored based on the number of such keys in common between documents. The current representation allows the gradual replacement of word-stemming with 'meaning entries' based on further interpretation of the text.

4 Performance on Retrieval

Because most of our interest is in the extraction operations that the relational text base allows, we have only conducted rudimentary tests of retrieval performance. We will have more comprehensive tests for the final version of this paper (and would appreciate reviewers suggestions regarding important cases to try). We will describe one case in detail to illustrate the performance gains possible, but more exhaustive testing must precede any broader claims.

Responding to the query 'What has Clinton said?,' a simple boolean conjunction search identified 188 articles while the corresponding relational query yielded just 51 articles. For the keyword query, 133 of the articles were incorrectly selected (Clinton didn't say anything in them); for the relational query, 3 articles were incorrectly selected and 7 were missed.

Since these queries did not to use any sort of thesaurus, we count only literal identity of words as indicating meaning identity (for purposes of this evaluation). This means that the keyword retrieval had — in some sense — perfect recall but only 32% precision. Relational retrieval, on the other hand, had 92% recall and 94% precision. Most of the breakdowns in recall (5 of the 7) arose from the failure of the parser to handle pronominalization. By introducing an approach to pronominalization which uses the same representation of ambiguity used in preposition and clause attachment, we hope to improve recall with a minimal (but neccessarily still some) decrease in precision.

To assess how useful the grammatical analysis is, we can compare the relational query with simpler syntatic queries. For example, a query which just looks for the juxtaposition 'Clinton said' finds 43 of 51 documents found by relational retrieval. This excludes the false hits of the relational retrieval, giving 'perfect' precision but only 78% recall. On the other hand, a looser query, identifying occurrences of 'Clinton' and 'said' in the same sentence, finds 78 articles giving perfect recall with 71% precision. If we tighten this up to require that 'Clinton' occur before 'said' we get 60 articles with 92% precision.

However the relative success of these simple syntactic methods (particularly the latter one) is probably peculiar to the query 'Clinton said' since 'said' often directly follows its subject, and Clinton seldom occurs with any appositive elaboration (as in "Clinton, the U.S. President, said"). We expect more exhaustive testing to confirm this.

The Dow Jones average of 30 industrials opens at a record **3734.53** Thursday after closing up 15.65 Wednesday. The New York Stock Exchange composite opens at **257.97**, down **0.01**. The American Stock Exchange market value opens at **468.87**, up 1.47. The NASDAQ OTC composite opens at **767.89**, down **1.46**.	The Dow Jones average of 30 industrials opens at **3685.34** Friday after closing down 19.01 Thursday. The New York Stock Exchange composite opens at **256.20**, down **0.61**. The American Stock Exchange market value opens at **469.54**, down 3.00. The NASDAQ OTC composite opens at **754.34**, down **8.02**.

Figure 2: These two texts have nearly identical grammatical structures, making it easy to identify corresponding elements between them.

5 Analogical Correspondence

The relational information produced by parsing free text can also be used in determining *analogical correspondences* between elements of the texts. For instance, given the texts in Figure 2, a match of grammatical structures sets up the following correspondences:

$$\begin{array}{lcl} 3734.53 & \Longrightarrow & 3685.4 \\ 257.97 & \Longrightarrow & 256.20 \\ 0.01 & \Longrightarrow & 0.61 \\ 468.87 & \Longrightarrow & 469.54 \\ 767.89 & \Longrightarrow & 754.34 \\ 1.46 & \Longrightarrow & 8.02 \end{array}$$

Readers may object that a trivial alignment algorithm could have also generated these correspondences. However, such algorithms would break down as surface structure variations grew more complicated while the use of grammatical graph structure allows correspondences to be determined even in the face of substantial variation.

While graph matching in general is intractable, we use a much simpler algorithm which is comfortably polynomial. The central idea is that we start with a set of unambigous 'linch analogies' and then use these to determine others. For instance, if we take as a set of linch analogies those words which occur exactly once in each document, we end up with matches for the bold faced words shown in Figure 3.

From this set of linch analogies, we can then look at the words modified by analogous pairs of words and if those words are identical, mark them as analogous. So, since the analogous occurrences of 'Jones' each modifies only one occurrence of 'average,' we can draw correspondences between the respective occurrences of 'averages'. The matches found in this way are shown italicized in Figure 3. We can continue to applying this algorithm recursively and generalizing the 'identical' constraint to identify different literal numbers as identical, we can then identify the corresponding ocurrences of 'opens' and finally the corresponding values listed previously.

The tractability of this algorithm relies on the fact that we are using literal identity (or a weakened version of it) to drastically reduce the space of possible matches. This

The **Dow Jones** *average* of 30 industrials opens at a record 3734.53 Thursday after **closing** up 15.65 Wednesday. The **New York** Stock Exchange *composite* opens at 257.97, down 0.01. The **American** Stock Exchange market *value* opens at 468.87, up 1.47. The **NASDAQ OTC** *composite* opens at 767.89, down 1.46.

The **Dow Jones** *average* of 30 industrials opens at 3685.34 Friday after **closing** down 19.01 Thursday. The **New York** Stock Exchange *composite* opens at 256.20, down 0.61. The **American** Stock Exchange market *value* opens at 469.54, down 3.00. The **NASDAQ OTC** *composite* opens at 754.34, down 8.02.

Figure 3: The matching process begins with certain 'linch analogies' (the words in bold face) and then uses the grammatical modification relations to identify secondary analogies (the italicized words). Applied recursively, the algorithm can quickly establish a complete match between the two texts.

President **Clinton** may have won a rare consensus Monday when he **nominated** Ruth **Ginsburg** to the Supreme Court. She has already won warm and widespread support from senators and special interest groups.. . .

President **Clinton nominated** outspoken Jocelyn **Elders** to be his surgeon general on Thursday. The move sets up what conservaties threan will be a bruising confirmation fight. . .

Figure 4: Despite quite different structures, we can identify corresponding 'nominees' in these two texts.

advantage should be retained even when we expand the 'literal identity' criteria to include reference to various thesauri. As an example of another case with less of a taste of bolier plate, the algorithm effectively identifies the analogy between 'Elders' and 'Gore' in the texts of Figure 4.

The matching criteria of literal identity is clearly too limiting. The matching mechanism includes the use of synonymy information from a (currently empty) thesaurus. In the near future, we will begin to build up this thesaurus; our hope is that the combination of structural constraints from the modification graph structure and knowledge in the thesaurus will significantly enlarge the range for which the matching algorithm will work.

6 Content Extraction

Analogical analysis allows the extraction of particular content items from within a document. The first example above was taken from our database; by querying for similar items and pulling out the element analogous to '3734.53' in each, the textual database was used to produce a graphical display of the quantity over time (Figure 5).

The ability to do this sort of extraction relies on the success of the matching process described above which in turn requires the background of matching knowledge for a particular domain.

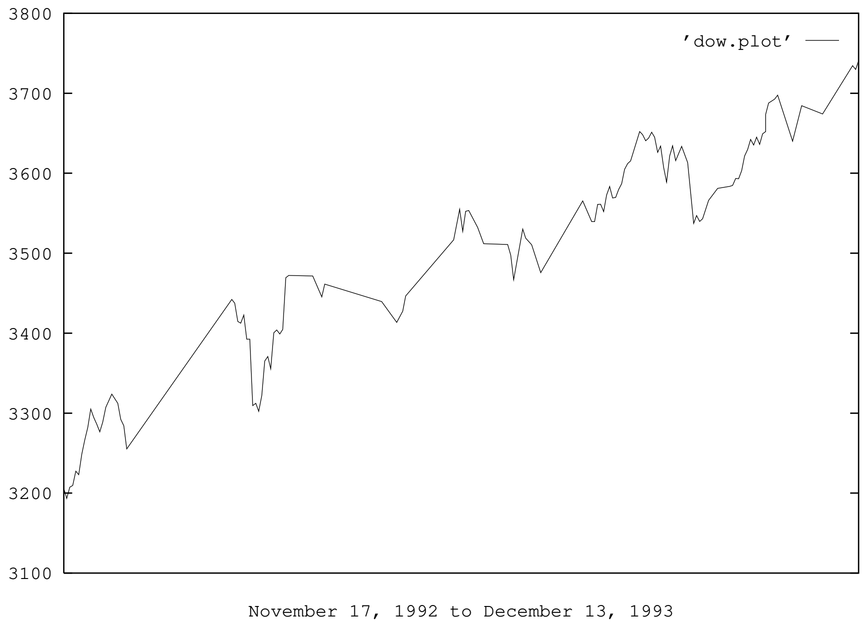

Figure 5: Analogical content analysis can be used to extract a quantity over time from a textual database. In this plot, data was extracted from parsed free text and then displayed graphically. The suspicious looking linear stretches indicate gaps in the record drawn as linear in the plot.

7 Conclusions, Related, and Future Work

We have described the generation and use of a 'relational textbase' for certain information retrieval and extraction tasks. The database was produced by a multi-pass parsing process and indexed by a combination of stemming words and creating keys based on the relations produced by parsing. The precision and recall of this indexing approach was compared to other approaches for extracting relations automatically. We then described the use of the relational information to construct analogical correspondences between texts and the use of these corresondences to extract particular information from texts.

The database preparation and retrieval components of this work have clear analogs in efforts to extract and use syntagmatic relationships in various ways [Croft *et al.*, 1991, Strzalokowski, 1992]. The innovations in our approach are the use of a link grammar to produce parses and the explicit representation of ambiguity to avoid premature exclusion of relations. The content extraction work's most obvious precedents are in template extraction [Jacobs and Rau, 1990] especially the host of programs involved in the ONR-sponsored 'MUC' competitions; these approaches generally use explicit 'trope matchers' to pull certain kinds of assertions out of the database. We differ in relying on direct text-to-text matches to extract information, an approach we hope will be more likely to move easily across domains.

In terms of future work, our highest priority is the extension of the contextual thesaurus; such an extension would improve both retrieval and the deterimination of analogical correspondences. This would then allow more interesting and significant content extraction and transcoding.

8 References

9 References

[Croft *et al.*, 1991] W. Bruce Croft, Howard R. Turtle, and David D. Lewis. The use of phrases and structured quiereis in information retrieval. In *Fourteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 32–45. SIGIR, 1991.

[Haase, 1993] Kenneth B. Haase. FRAMER: A Portable Persistent Representation Library. In *Proceedings of the AAAI Workshop on.* American Association for Artificial Intelligence, 1993. (Submitted).

[Jacobs and Rau, 1990] Paul Jacobs and Lisa Rau. SCISOR: Extracting information from on-line news. *Communcations of the ACM*, 33(11), November 1990.

[Meĺčuk, 1988] I.A. Meĺčuk. *Dependency Syntax: Theory and Practice.* State University of New York Press, 1988.

[Sleator and Temperly, 1991] Daniel D.K. Sleator and Davy Temperly. Parsing english with a link grammar. Technical Report 91-196, Carnegie Mellon Computer Science Department, October 1991.

[Strzalokowski, 1992] Tomek Strzalokowski. Tpp: a fast and robust parser for natural language. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, pages 198–204, 1992.