An Architecture for Content Analysis of Documents and its Use in Information and Knowledge Management Tasks

Branimir Boguraev, Christopher Kennedy, & Sascha Brawer

Abstract

We present a generalised architecture for document content management, with particular emphasis on component functionalities and reconfigurability for different content management tasks. Natural language technologies are encapsulated in separate modules, which then can be customised and tailored for the specific requirements of the type of document, depth of analysis, and detail of output representation, of different document analysis systems. The versatility of the architecture is illustrated by configuring it for two diverse tasks: analysing technical manuals to instantiate databases for on-line assistance, and deriving topically-rich abstractions of content of arbitrary news stories.

Introduction

The natural language research program in Apple's Advanced Technologies Group (ATG) has been actively pursuing the automation of certain aspects of the information analysis and knowledge management tasks. The program's focus is on establishing a core set of natural language processing (NLP) technologies and defining application areas for these. Representative projects have investigated a range of issues including: optimal packaging of a substrate of NLP functionalities, with appropriate API's, embedded within the Macintosh Operating System (Mac OS); an architecture for text processing, configurable for different content analysis applications; studies of how NL technologies can be leveraged for further enhancing the user experience; and building several information management systems incorporating linguistic processing of text-based documents.

In particular, language-related work at ATG facilitates a number of information management tasks, including: semantic highlighting and indexing, topic identification and tracking, content analysis and abstraction, document characterisation, and partial document understanding. Given the broad base of Apple users, the emphasis has been on finding suitable tasks which can be enhanced by linguistic functionalities, on striking the right balance of scalable and robust technologies which can reliably analyse realistic text sources, and on developing algorithms for focused semantic analysis starting from a relatively shallow syntactic base.

This article highlights the core capabilities of an architecture for content analysis, within which a number of information processing applications have been implemented. The use of the architecture for application building is illustrated by two examples: domain acquisition and document abstraction.

SCOOP: An Architecture for Content Analysis

The underlying technology base for content analysis is encapsulated within an architecture for developing and utilising a set of tools and techniques for natural language processing and text analysis. As a base level toolset comprising a number of linguistic components, it is capable of lexical, morphological, syntactic, semantic and discourse analysis at various levels of depth and sophistication. This makes it possible to develop a range of readily configurable text analysis systems, which ultimately use the same operational components.

Different applications all rely on text stream analysis delivered by a *part of speech super-tagger* technology developed for the purposes of accurately identifying the correct part of speech of

words in context, as well as capable of assigning grammatical function (subject, object, adjunct, and so forth) to these [8]. A *morphological analysis* component reduces words to their base forms, also utilising a set of heuristics for guessing properties of unknown lexical items. This is augmented by a module for *identifying and analysing truly extra-lexical items*, such as proper names and abbreviations, which cannot be reasonably expected to be always found in a pre-compiled lexicon.

A pattern matcher, capable of interpreting finite-state specifications of phrasal units defined in terms of sequences of parts-of-speech, and constrained by morpho-syntactic features of words in the text stream, implements a general purpose *shallow syntactic analysis* capability. This mines the text for phrasal units of different types, each defined by its own grammar; grammars can be independently developed to extract, for instance, noun phrases in subject position, fully specified proper names, or verb phrases in which previously mentioned items are in object position.

Some *semantic level analysis* is carried out by an anaphora resolution component, capable of resolving anaphoric references (for instance, "it", "they", and "themselves") to previously mentioned objects [9]; this is part of a general mechanism for tracking references to the same object through an entire document [10]. This process fundamentally relies on a computational model of *salience*, which is intended to embody notions of relevance and topicality. In conjunction with a *discourse analysis* component, intended to follow major changes of topics in the body of a text document [5], the salience calculation underlies the incremental derivation of a model of discourse which reflects the flow of narrative in the document.

Within such an architecture, individual components can be customised for the specifics of a particular task, and configured as appropriate. For instance, different noun phrase grammars can be developed for technical terminology extraction (which might be required by an application for analysing technical reference materials), or for object identification (which would be an essential prerequisite for topic determination in arbitrary document sources); both would be realised by suitably programming the phrasal pattern matcher. While full anaphora resolution may not be required for some document analysis tasks, a model of co-reference, on the other hand, might necessitate some partial anaphora (such as proper name anaphora): the requisite components thus will be suitably configured for a different depth of analysis in different systems targeted, for instance, at software manuals as opposed to arbitrary news articles. Likewise, topic tracking, which relies on segmenting the body of a document into thematically coherent 'chunks', may need a semantically driven discourse segmentation component, or may be sufficiently informed by structural markup in the document (such as denoted by section and subsection headings, itemised lists, and so forth).

Such ready configurability allows for rapid development of text analysis systems with different functionality, aimed at processing different document types and genres. The following two sections illustrate, by example, two very different applications of the SCOOP architecture: instantiating databases for on-line assistance from technical manuals, and deriving topically-rich abstractions of content of arbitrary news stories.

WORDWEB: Domain Specification for On-line Assistance

Apple Guide is an integral component of the Macintosh operating system; it is a general framework for on-line delivery of context-sensitive, task-specific assistance across the entire range of software applications running under the Mac OS. The underlying metaphor is that of answering user questions like *"What is X?"*, *"How do I do Y?"*, *"Why doesn't Z work?"* or *"If I want to know more about W, what else should I learn?"*. Answers are 'pre-compiled', on the basis of a full domain description defining the functionality of a given application. For each application, assuming the existence of such a description in a certain database format, Apple Guide coaches the user through a sequence of definitional panels (describing key domain concepts), action steps (unfolding the correct sequence of actions required to perform a task or achieve a goal), or cross-reference information (revealing additional relevant data concerning the original user query).

WORDWEB is an instantiation of the SCOOP architecture for analysing the content of software instructional materials; the focus is on linguistically intensive analysis of suitable documentation sources (manuals, technical notes, help files, and so forth) in order to acquire knowledge essential to the process of assisting the user with their task.

An Apple Guide database is typically instantiated by hand, on a per application basis, by trained instructional designers. Viewed abstractly, the information in such a database constitutes complete domain specification for the application – as defined by the objects in the domain, their properties, and the relations among them. In order to answer user questions (like those above), certain aspects of the domain need to be identified: a kind of a domain object is a disk; there are several types of disk, including floppy disks, startup disks, internal hard disks; disks need to be prepared for use; floppy disks can be ejected; and so forth.

WORDWEB thus acts as an automatic domain acquisition system, whose application is illustrated in Figure 1.

The screen snapshot on the top is from the Macintosh Guide shipping with System 7.5; the database here is built, manually, by a team of instructional designers. The other snapshot displays, through the same delivery mechanism, a database constructed, fully automatically, by the WORDWEB system, following an analysis of the technical documentation (*Macintosh User's Guide*) for Macintosh systems. The *"How do I..."* lists (in the right-hand panels), display entry points to detailed instructions concerning common tasks with specific objects (in this example, disks) in the MacOS domain. Barring non-essential differences, there is a strong overlap between the two lists (*prepare a disk for use, eject a disk, test (and repair) a disk, protect a file/information on disk*, and so forth). [1] Moreover, WORDWEB has identified some additional action types, clearly relevant to this domain, but missing from the 'canonical' database: *share a disk, find items on a disk, protect information on disk.*)

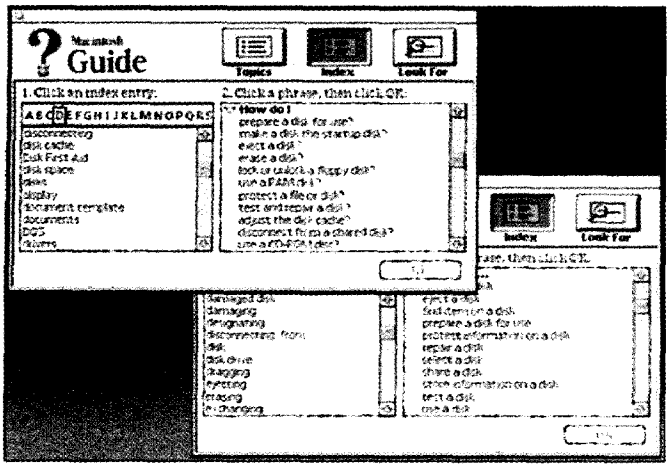

Figure 1: Comparison: Manual Domain Specification and Automatic Domain Acquisition

The value of such a system for instructional designers is obvious. At the very least, it offers a 'jump start' over the first step in a Guide database design: the complex process of introspection and domain mapping. Additionally, it generates a skeleton database which is close to optimal, offering substantial savings of time and effort. By the same token, the system would be helpful to third party developers who may not have the skills and/or resources required for instructional design.

WORDWEB carries out normalisation and data reduction over text, by configuring the SCOOP architecture into a linguistic cascade of lexical, morphological, and syntactic processing, followed by mining for core domain objects, relations they participate in, and properties they possess. Mining for objects is conceptually identical to extracting domain-specific vocabulary of *technical terms*; this is done by suitably deploying a term grammar [7] in the pattern matcher. Mining for relations is identical to instantiating contextual (semantic) information for such terms. The core set of terms is, in effect, refined not only to include all (and only) domain objects, but also to enrich the descriptions of these domain objects by deriving relational structures for each of them: what relations they participate in, and what properties they possess. Ultimately, a conceptual map of the technical domain is derived. Such a domain specification may be represented as a domain catalog, a small fragment of which is illustrated in Figure 2.

Mapping from such a domain description to an Apple Guide database is relatively straightforward. A term identified as a salient domain object clearly ought to be in the database – in its simplest form, by providing a definition for it. The same applies to a relation, which easily maps onto a *"How do I ...?"* panel. For the example fragment above, this would mean definition entries for *startup disk, network connection, System Folder*; and action sequence panels for *How do I specify a startup disk?, How do I use a hard disk as a startup disk?*, and so forth. The definitions and task sequences would still have to be supplied externally,[2] but the generation of the database is fully automatic.

Term: startup disk
  Relations:
    conserve space on []
    correct a problem with []
    look for []
    recognize []
    specify []
    use hard disk as []
  Objects:
    System folder

Term: icon
  Relations:
    assign color to []
    .....
  Objects:
    Apple printer
    network connection
    startup disk
    Trash
    .....

Figure 2: Domain Catalog: A Fragment

The identification of terms, relations, properties, and so forth exemplifies the use of SCOOP's shallow syntactic parser to map syntactic patterns (noun phrases with certain lexical, syntactic, and discourse properties; verb phrases with certain terms in selected argument positions; and so forth) onto conceptual categories. Thus a "hard disk" emerges as an important domain object from repeated observations of the phrase, and its variants, in the text; and the fact that a hard disk can be used as a startup disk (see above) can be deduced from observing salient co-occurrences of the phrase *hard disk* and *startup disk* as direct and indirect objects to *use*. Suitably defined phrasal grammars drive SCOOP's pattern matcher to seek, and record, just these kinds of observations which embody the set of salient facts about the domain. An ordered set of observations, each of which has been brought into a canonical form, ultimately becomes the domain catalog.

A comparison between the database derived by WORDWEB from the *Macintosh Reference* manual, and the manually compiled *Macintosh Guide* database in System 7.5, shows that the analysis system described here is very accurate. The evaluation focuses on precision and recall in coverage; [1] discusses at some length how the sets of terms and relations for the two databases are adjusted to take into account both similarities and differences in the data. Following an evaluation methodology, also presented in [1], recall and precision rates are calculated for the automatic database construction system. *Recall* is defined with respect to the 'reference' database, as a normalised count of how many items defined manually have also been identified automatically by WORDWEB. *Precision* is defined as the ratio of 'good' hits, relative to all items identified by the automatic extraction methods. Table 1, which gives a quantitative account of the performance of the WORDWEB system,

1. Actions associated with specific types of disk (e.g. startup disk, floppy disk) appear elsewhere in the automatically generated database.

2. In fact, the process of technical documentation analysis maintains a complete 'audit trail', relating items in the domain catalog to text source where information concerning them has been found; a prototype implementation augments the database with pointers into the on-line version of the manual

demonstrates the practical viability of the document analysis techniques described here for the purposes of domain acquisition and specification.

	Terms	Relations
Recall	94.0%	91.1%
Precision	89.5%	88.5%

Table 1: Recall and Precision for Automatic Domain Acquisition

Beyond automatic generation of Apple Guide databases, the WORDWEB technology can also be applied to: general purpose indexing, (e.g. for glossary construction); mapping the functionality of applications, (e.g. for scripting situations where an application needs to publish its range of capabilities); hypertext delivery of technical documentation; and so forth. This analysis framework is as powerful as it is because of a fundamental assumption implicit in these applications: the closed nature of technical domains and documentation. "Closeness" here is taken to mean that everything that is relevant in the domain, is mentioned in the document and that there is nothing that is essential that is not to be found somewhere in the document.

An interesting question is that of the applicability of linguistic processing for term identification, relation extraction, and object cross-classification to the larger space of document management and access tasks: how similar phrasal identification techniques could be exploited for the purpose of content characterisation of arbitrary text documents. A different application of the SCOOP architecture addresses this question.

FACTFINDER: Capsule Overviews of News Articles

Several problems arise with scaling up the WORDWEB technology to an open-ended set of document types and genres such as press releases, news articles, web pages, mail archives, and so forth. The domains are not closed any more. The 'clean', typically regular, language of technical documentation dissolves into a variety of writing styles, with wide variation in formality and grammaticality. A system seeking to characterise a document analogously to a domain being characterised by its terms, typically will find a large number of term-like phrases, certainly larger than a user can absorb without too much cognitive overhead.

The SCOOP architecture has therefore been reconfigured to identify the core content-bearing units in arbitrary texts, with a focus on smaller documents, and allowing for wide diversity of genre. This work is centered around the development of a set of sophisticated text processing tools, again based on a shallow syntactic analysis of the input stream, but incorporating more intensive semantic processing. The architecture now supports the identification, extraction, classification and typing of proper names, technical terms, and other complex nominals from text, by extending the core phrasal analysis engine originally developed for term identification. New modules also embody a stronger notion of discourse modelling, in particular: maintaining co-referentiality among objects discussed in the document [10], deriving a salience measure for these objects [9], and determining topically significant segments and objects in the text [3]. A particular instantiation of the architecture, FACTFINDER, now performs the task of salience-based content characterisation of text documents.

Below we briefly present highlights of FACTFINDER, which seeks to derive document content characterisations as collections of highly salient topical phrases, embedded in layers of progressively richer and more informative contextualised text fragments, with contexts calculated as meaningful fragments defined by a containment hierarchy of information-bearing phrasal units, and organised as capsule overviews which track the occurrence of topical phrases and other discourse referents across the document discourse.

Capsule Overviews of Documents

The notion of *capsule overviews* as content abstractions for text documents is explicitly designed to capture "aboutness" ([3]). This is represented as a set of *highly salient*, and by that token most representative, phrases in the document. Viewing topicality in its linguistic sense, we define *topic stamps* to be the most prominent of these phrases, introduced into, and then elaborated upon, the document body. On the basis of this definition, an algorithmic procedure has been developed for generating a set of abstractions for the core meaning in the document, ultimately resulting in a *capsule overview* of the document based upon suitable presentation [4] of the most representative, and most contentful, expressions in the text. These abstractions comprise layered and inter-related phrasal units at different levels of granularity, following a containment hierarchy (illustrated below) which relates the different information levels in a document together: topic stamps are embedded in (contextualised to) more informative relational phrases; these are further elaborated by the sentences in which they appear; sentence contents are further elaborated in the enclosing paragraphs, themselves contextualised to topically coherent document segments (Figure 3).

Document Characterisation by Topics

Topic stamps are certain noun phrases judged to be "topically-relevant" in the text. The function of a noun phrase in telling a story is to introduce a new entity – a *discourse referent*, typically an object, a concept, an event, a participant in an action – into the story. *Topical relevance* is defined as a feature of a discourse referent, which marks it as being subsequently elaborated upon in the course of story-telling. Following processes of identification and normalisation, discourse referents are ranked according to a global measure of *salience*. Salience of a topic is defined as a single numeric parameter, which embodies a number of semantic criteria. Some of these are: how prominently the topic is introduced into the discourse; how much discussion is there concerning the topic; how much is the topic mentioned throughout the entire document, as opposed to e.g. in just one (or some) sections of the text. The full set of semantic factors embodied in the salience weight is described in detail elsewhere ([9], [10], [3]); these factors are calculated on the

- TOPIC STAMP: "desktop machines"
- TOPIC IN RELATIONAL CONTEXT:
  "Tomorrow's machines must accommodate rivers of data,"
- TOPICAL SENTENCE:
  "Tomorrow's machines must accommodate rivers of data, multimedia and multitasking (juggling several tasks simultaneously)."
- SENTENCE IN PARAGRAPH CONTEXT:
  "Lots of "ifs", but you cannot accuse Amelio of lacking vision. Today's desktop machines, he says, are ill-equipped to handle the coming power of the Internet. Tomorrow's machines must accommodate rivers of data, multimedia and multitasking (juggling several tasks simultaneously)."
- PARAGRAPH WITHIN TOPICALLY COHERENT DISCOURSE THEME:
  "Lots of "ifs", but you cannot accuse Amelio of lacking vision. Today's desktop machines, he says, are ill-equipped to handle the coming power of the Internet. Tomorrow's machines must accommodate rivers of data, multimedia and multitasking (juggling several tasks simultaneously).
  We're past the point of upgrading, he says. Time to scrap your operating system and start over. The operating system is the software that controls how your computer's parts (memory, disk drives, screen) interact with applications like games and Web browsers. Once you've done that, buy new applications to go with the reengineered operating system."

Figure 3: Contextualisation of Topical Highlights

basis of SCOOP's discourse model. The intent is to be sensitive to a number of linguistic and stylistic devices employed in text-based discourse for the purposes of introducing, defining, refining, and re-introducing discourse referents. The set of such devices is large, and it is precisely this richness which enables finer distinctions concerning content elaboration to be observed and recorded. What is of particular interest is that even from the shallow syntactic base of SCOOP's analyser, the specially configured co-reference module can derive reliable judgements concerning such semantically relevant factors.

Encoding the results of such analysis into a single parameter provides the basis of a decision procedure focusing on discourse referents with high salience weight. These are the *topic stamps* for the document. While simple, the decision procedure is still remarkably well informed, as the salience weight calculation by design takes into account the diverse manifestation of topicality in written prose.

The final set of topic stamps is representative of the core document content. It is *compact*, as it is a significantly cut-down version of the full list of document topics. It is *informative*, as the topics in it are the most prominent discourse referents in the document. It is *representative* of the whole document: in breadth, as a separate topic tracking module effectively maintains a record of where and how discourse referents occur in the entire span of the text, and in depth, as each topic stamp maintains its relational and larger discourse contexts. As topics are the primary content-bearing entities in a document, the topic stamps offer *accurate* approximation of what that document is about.

Topic Stamps and Capsule Overviews

Capsule overviews of documents take the form of a set of topic stamps, enriched by the textual contexts in which they are encountered in the source. The topic stamps are organised in order of appearance, and, as exemplified earlier, are 'superimposed' onto progressively more refined and more detailed discourse fragments: relational contexts, sentences, paragraphs, and ultimately discourse segments.

Discourse segments reflect (dis-)continuity of narrative and the way in which focus of attention/description changes with the progress of the text story. 'Chunking' the document into more manageable units is not just for convenience. Discourse segments correspond to topically coherent, contiguous sections of text. The approach to segmentation SCOOP implements uses a similarity-based algorithm along the lines of the one developed by Hearst [5], which detects changes in topic by using a lexical similarity measure. By calculating the discourse salience of referents with respect to the results of discourse segmentation, each segment can be associated with a listing of those expressions that are most salient within the segment, i.e., each segment can be assigned a set of topic stamps. The result of these calculations, a set of segment-topic stamp pairs ordered according to linear sequencing of the segments in the text, can then be returned as the capsule overview for the entire document. In this way, the problem of content characterisation of a large text is reduced to the problem of finding topic stamps for each discourse segment.

Capsule Overview: An Example

The SCOOP architecture has been configured to make use of the following components for salience-based content characterisation: discourse segmentation; phrasal analysis (of nominal expressions and relations); anaphora resolution and generation of a referent set; calculation of discourse salience and identification of topic stamps; and enriching topic stamps with information about relational context(s). Some of the functionality derives from phrasal identification, suitably augmented with mechanisms for maintaining phrase containment; in particular, both relation identification and extended phrasal analysis are carried out by running a phrasal grammar over a stream of text tokens tagged for morphological, syntactic, and grammatical function (this is in addition to a grammar mining for terms and, generally, referents). Base level linguistic analysis is provided by a supertagger, [8]. The later, more semantically-intensive algorithms are described in detail in [9] and [10].

The procedure is illustrated by highlighting certain aspects of a capsule overview of a recent *Forbes* article ([6]). The document focuses on the strategy of Gilbert Amelio (former CEO of Apple Computer) concerning a new operating system for

"ONE DAY, everything Bill Gates has sold you up to now, whether it's Windows 95 or Windows 97, will become obsolete," declares Gilbert Amelio, the boss at Apple Computer. "Gates is vulnerable at that point. And we want to make sure we're ready to come forward with a superior answer."

Bill Gates vulnerable? Apple would swoop in and take Microsoft's customers? Ridiculous! Impossible! In the last fiscal year, Apple lost $816 million; Microsoft made $2.2 billion. Microsoft has a market value thirty times that of Apple.

Outlandish and grandiose as Amelio's idea sounds, it makes sense for Apple to think in such big, bold terms. Apple is in a position where standing pat almost certainly means slow death.

It's a bit like a patient with a probably terminal disease deciding to take a chance on an untested but promising new drug. A bold strategy is the least risky strategy. As things stand, customers and outside software developers alike are deserting the company. Apple needs something dramatic to persuade them to stay aboard. A radical redesign of the desktop computer might do the trick. If they think the redesign has merit, they may feel compelled to get on the bandwagon lest it leave them behind.

Lots of "ifs," but you can't accuse Amelio of lacking vision. Today's desk-top machines, he says, are ill-equipped to handle the coming power of the Internet. Tomorrow's machines must accommodate rivers of data, multimedia and multitasking (juggling several tasks simultaneously).

We're past the point of upgrading, he says. Time to scrap your operating system and start over. The operating system is the software that controls how your computer's parts (memory, disk drives, screen) interact with applications like games and Web browsers. Once you've done that, buy new applications to go with the reengineered operating system.

Amelio, 53, brings a lot of credibility to this task. His resume includes both a rescue of National Semiconductor from near-bankruptcy and 16 patents, including one for coinventing the charge-coupled device.

But where is Amelio going to get this new operating system? From Be, Inc., in Menlo Park, Calif., a half-hour's drive from Apple's Cupertino headquarters, a hot little company founded by ex-Apple visionary Jean-Louis Gassee. Its BeOS, now undergoing clinical trials, is that radical redesign in operating systems that Amelio is talking about. Married to hardware from Apple and Apple cloners, the BeOS just might be a credible competitor to Microsoft's Windows, which runs on IBM-compatible hardware.

Figure 4: Sample document, with segmentation

the Macintosh. Too long to quote here in full (approximately four pages in print), the sample passage from the beginning of the article contains the first three segments, as identified by the discourse segmentation component; in the example (Figure 4), segment boundaries are marked by extra vertical space (of course, this demarcation does not exist in the source, and is introduced here for illustrative purposes only).

The relevant sections of the overview (for the three segments of the passage quoted) are shown in Figure 5. The listing of topic stamps in their relational contexts provides the core data for the capsule overview; while not explicitly shown here, the capsule overview data structure fully maintains the layering of information implicit in the containment hierarchy.

1: APPLE; MICROSOFT
APPLE would swoop in and take MICROSOFT'S customers?
APPLE lost $816 million;
MICROSOFT made $2.2 billion.
MICROSOFT has a market value thirty times that of APPLE
it makes sense for APPLE
APPLE is in a position
APPLE needs something dramatic

2: DESKTOP MACHINES; OPERATING SYSTEM
Today's DESKTOP MACHINES, he [Gilbert Amelio] says
Tomorrow's MACHINES must accommodate rivers of data
Time to scrap your OPERATING SYSTEM and start over
The OPERATING SYSTEM is the software that controls
to go with the REENGINEERED OPERATING SYSTEM

3: GILBERT AMELIO; NEW OPERATING SYSTEM
AMELIO, 53, brings a lot of credibility to this task
HIS [Gilbert Amelio] resumé includes
where is AMELIO going to get this NEW OPERATING SYSTEM?
radical redesign in OPERATING SYSTEMS that AMELIO is talking about

Figure 5: Capsule Overview

The division of this passage into segments, and the segment-based assignment of topic stamps, exemplifies a capsule overview's "tracking" of the underlying coherence of a story. The discourse segmentation component recognizes shifts in topic – in this example, the shift from discussing the relation between Apple and Microsoft to some remarks on the future of desktop computing to a summary of Amelio's background and plans for Apple's operating system. Layered on top of segmentation are the topic stamps themselves, in their relational contexts, at a phrasal level of granularity.

The first segment sets up the discussion by positioning Apple opposite Microsoft in the marketplace and focusing on their major products, the operating systems. The topic stamps identified for this segment, APPLE and MICROSOFT, together with their local contexts, are both indicative of the introduc-

tory character of the opening paragraphs and highly representative of the 'gist' of the first segment. Note that the apparent uninformativeness of some relational contexts, for example, '... APPLE *is in a position ...*', does not pose a serious problem. An adjustment of the granularity – at capsule overview presentation time (see [4] for detailed discussion of mechanisms of delivering of capsule overview to users) – reveals the larger sentential context for the topic stamp, which in turn inherits the high topicality ranking of its anchor: 'APPLE *is in a position where standing pat almost certainly means slow death.*'

For the second segment of the sample, OPERATING SYSTEM and DESKTOP MACHINES have been identified as representative. The set of topic stamps and contexts illustrated provides an encapsulated snapshot of the segment, which introduces Amelio's views on coming challenges for desktop machines and the general concept of an operating system. Again, even if some of these appear under-specified, more detail is easily available by a change in granularity, which reveals the definitional nature of the even larger context '*The* OPERATING SYSTEM *is the software that controls how your computer's parts...*'

The third segment of the passage is associated with the stamps GILBERT AMELIO and NEW OPERATING SYSTEM. The linguistic rationale for the selection of these particular noun phrases as topical is closely tied to form and function of discourse referents in context. Accordingly, the computational justification for the choices lies in the extremely high values of salience, resulting from taking into account a number of factors: co-referentiality between *'Amelio'* and *'Gilbert Amelio'*, co-referentiality between *'Amelio'* and *'His'*, syntactic prominence of *'Amelio'* (as a subject) promoting topical status higher than for instance *'Apple'* (which appears in adjunct positions), high overall frequency (four, counting the anaphor, as opposed to three for *'Apple'* – even if the two get the same number of text occurrences in the segment) – and boost in global salience measures, due to "priming" effects of both referents for *'Gilbert Amelio'* and *'operating system'* in the prior discourse of the two preceding segments.

Conclusion

The tasks exemplified in the previous two sections are clearly very different, yet we have been able to reuse a number of components within the SCOOP text processing architecture. Typically, this has required some reconfiguring: for instance, the different tasks enforce different decisions about what constitutes a 'token', what kinds of phrasal units are particularly rich with information about core content, and how much anaphora is needed to get a sense of the distribution of an object across the entire document. We have been able to account for such differences by appropriately readjusting parameters like granularity of analysis, coverage of phrasal grammars, and input-output constraints for anaphora resolution; the architecture supports precisely these kinds of adjustments. The architecture also supports reconfigurability in a different sense: where necessary, new modules can be developed, operating over the same underlying document representation as an annotated text stream, and easily 'inserted' within the flow of control. This has been the case with identifying discourse segments, and contextualising topical phrases to levels of sentences and paragraphs. Natural language processing is, in general, a very hard problem. This makes a capability, by a computer, to understand language a very long way away. It is thus necessary to have access to an inventory of NLP technologies, which can be easily and quickly configured for a variety of tasks, where text processing is only a part of the larger operational context, and where some capability to extract document fragments related to its meaning can be effectively leveraged for information management tasks. The SCOOP architecture meets this need: the content analyses derived through its varying configurations have always been defined by the needs of such tasks, as exemplified by the domain catalogs derived by WORDWEB and utilised by Apple Guide, and by the capsule overviews abstracted by FACTFINDER and utilised by a variety of document content viewers (separately discussed in [2] and [4]).

References

[1] B. Boguraev. *WORDWEB and Apple Guide: a comparative evaluation.* Technical report, Internal Report, Advanced Technologies Group, Apple Computer, 1995.

[2] B. Boguraev and R. Bellamy. Dynamic Presentation of Document Abstractions. *SIGCHI Bulletin*, 1998. In this issue.

[3] B. Boguraev and C. Kennedy. Salience-based content characterisation of text documents. In *Proceedings of ACL'97 Workshop on Intelligent, Scalable Text Summarisation*, Madrid, Spain, 1997.

[4] B. Boguraev, Y. Y. Wong, C. Kennedy, R., Bellamy, S. Brawer, and J. Swartz. Dynamic presentation of document content for rapid on-line browsing. In *Proceedings of AAAI Spring Symposium on Intelligent Text Summarisation*, Stanford, CA, 1998.

[5] M. Hearst. Multi-paragraph segmentation of expository text. In *32nd Annual Meeting of the Association for Computational Linguistics*, Las Cruces, New Mexico, 1994.

[6] N. Hutheesing. Gilbert Amelio's grand scheme to rescue Apple. *Forbes Magazine*, December 16, 1996.

[7] J. Justeson and S. Katz. Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1(1):9-27, 1995.

[8] F. Karlsson, A. Voutilainen, J. Heikkila, and A. Antilla. *Constraint grammar: A language-independent system for parsing free text.* Mouton de Gruyter, Berlin/New York, 1995.

[9] C. Kennedy and B. Boguraev. Anaphora for everyone: Pronominal anaphora resolution without a parser. In *Proceedings of COLING-96* (16th International Conference on Computational Linguistics), Copenhagen, DK, 1996.

[10] C. Kennedy and B. Boguraev. Anaphora in a wider context: Tracking discourse referents. In W. Wahlster, editor, *Proceedings of ECAI-96* (12th European Conference on Artificial Intelligence), Budapest, Hungary, 1996. John Wiley and Sons, Ltd, London/New York.

About the Authors

Branimir Boguraev holds degrees in Computer Science and Applied Mathematics from the Higher Institute for Mechanical and Electrical Engineering in Sofia, Bulgaria. He obtained a Ph.D. in Computational Linguistics in the Computer Laboratory at the University of Cambridge, UK. Since then, he has

held several research award positions, under the UK Information Technology Initiative, and has been broadly involved in work on developing a broad natural language technology base. In 1988, he joined IBM's T.J. Watson Research Center, focusing on data-intensive methods for constructing rich computational lexicons. More recently, he has managed research efforts in natural language processing, first at IBM Research, and following that, at Apple's Advanced Technology Group. His interests are in linguistically intensive methods for document content analysis, and their deployment in information and knowledge management tasks.

Christopher Kennedy holds degrees in linguistics; most recently, he obtained a Ph.D. on the syntax and semantics of comparative structures from the University of California at Santa Cruz. He has been closely associated with Apple's natural language program, working in the Advanced Technology Group in different capacities since 1995. His general linguistics research interests cover syntax, semantics, the syntax-semantics interface, comparatives, lexical semantics, anaphora and ellipsis; from a different, natural language processing, perspective, he is also interested in applying strong linguistic notions to the content analysis task. He recently joined, as a professor, the Linguistics Faculty at Northwestern University.

Sascha Brawer has studied German and Computer Science at the University of Zurich. Subsequent involvement with the Swiss Federal Institute of Technology (ETH Zurich) exposed him to the challenges of linguistic engineering, and consequently he joined one of the first European Computational Linguistics M.Sc. programs, in the German University of Saarbruecken. In the interim after completing the courses required by the program, he joined the natural language effort at Apple's Advanced Technology Group, where he was re-engineering a substrate of linguistic technologies into an architecture for content analysis. His research interests are in robust engineering of linguistically intensive analysis components. He is about to start work on his Ph.D., in the interdisciplinary field of Computer Science, Linguistics, and Artificial Intelligence.

Authors' Addresses

Branimir Boguraev
Advanced Technology Group
Apple Computer, Inc.
Cupertino, CA 95014, USA
bkb@cs.brandeis.edu

Christopher Kennedy
Department of Linguistics
Northwestern University
Evanston, IL 60208, USA
kennedy@ling.nwu.edu

Sascha Brawer
Department of Computer Science
University of Zurich, Switzerland
brawer@coli.uni-sb.de