Natural Language Engineering **5** (1): 17–44. Printed in the United Kingdom

© 1999 Cambridge University Press

Applications of term identification technology: domain description and content characterisation

BRANIMIR BOGURAEV
IBM T.J. Watson Research Center, IBM Corporation, NY, USA
e-mail: bkb@watson.ibm.com

CHRISTOPHER KENNEDY
Department of Linguistics, Northwestern University, IL, USA
e-mail: kennedy@ling.nwu.edu

(*Received September 1997; revised 24 July 1998*)

Abstract

The identification and extraction of technical terms is one of the better understood and most robust Natural Language Processing (NLP) technologies within the current state of the art of language engineering. In generic information management contexts, terms have been used primarily for procedures seeking to identify a set of phrases that is useful for tasks such as text indexing, computational lexicology, and machine-assisted translation: such tasks make important use of the assumption that terminology is representative of a given domain. This paper discusses an extension of basic terminology identification technology for the application to two higher level semantic tasks: *domain description*, the specification of the technical domain of a document, and *content characterisation*, the construction of a compact, coherent and useful representation of the topical content of a text. With these extensions, terminology identification becomes the foundation of an operational environment for document processing and content abstraction.

1 Introduction

1.1 Technical terminology: properties and applications

Within the current state of the art of language engineering, there are few natural language processing technologies aimed at content analysis that are readily applicable to an open-ended range of texts and domains. Among those that do have this property, the identification and extraction of technical terms is arguably one of the better understood and most robust, as indicated by the literature on the linguistic properties of technical terms, and how these properties support the definition of computational procedures for terminology identification and extraction that both apply to a wide range of document types and maintain their quality regardless of domain or genre.

One of the best defined procedures for the identification and extraction of technical terms is the TERMS algorithm developed by Justeson and Katz (1995), which focuses

https://doi.org/10.1017/S1351324999002090 Published online by Cambridge University Press

on multi-word noun phrases occurring in continuous texts. A study of the linguistic properties of these expressions – preferred phrase structures, behaviour towards lexicalisation, contraction patterns, and certain discourse properties – leads to the formulation of a robust and domain-independent algorithm for term identification. The TERMS algorithm accomplishes high levels of coverage (both in comparison to data from terminology dictionaries and as evaluated by the authors of the technical prose processed by the algorithm), it can be implemented within a range of underlying NLP technologies including 'bare' morphologically enhanced lexical look-up (Justeson and Katz 1995), part-of-speech tagging (Dagan and Church 1995), or syntactic parsing (McCord 1990), and it has strong cross-linguistic application (see, for instance, Bourigault (1992)).

Despite the strength of its potential as a tool for NLP applications, however, terminology identification tends to be viewed as a technology that is primarily suited for a task outside the core domain of NLP, namely Information Retrieval (IR). The typical IR task views a document collection as a homogeneous corpus, and seeks to identify those items (words, base forms, stems, or occasionally phrases) that have distributional properties that *distinguish* a document from other documents in the collection. For example, in a set of articles exploring the common theme of programming languages, expressions like *'compiler'*, *'program'*, and *'assembly code'* are likely to be commonly mentioned in all articles and hence not characteristic of any single one of them. Terminology identification makes it possible to expand the granularity of the indexing task from tokens to larger phrasal units, permitting more refined distinctions among documents in a given domain, as well as more explicit listings of the words and phrases that uniquely identify a particular document (see, for example, Salton (1988) and Salton, Zhao and Buckley (1990)).

In the context of NLP, where the goal is not simply document identification, but rather (partial) understanding, the more general problem is to find those terms which are in some way *characteristic* of a particular document, as in a back-of-the-book indexing task. Quite often, these terms are not the same as those which uniquely distinguish a document from others in a similar domain. In the context of the articles on programming languages mentioned above, for example, an indexing task crucially requires the extraction of all the terms in a document, including those that might be ignored for the purpose of IR indexing because they are not sufficiently discriminating (e.g. *'compiler'*, *'program'*, *'assembly code'*, and so forth).

1.2 Higher-level applications

While text indexing – both the back-of-the-book variety and IR-style (Hodges, Yie, Reighart and Bogges 1996; Justeson and Katz 1995) – represents the most common application of term identification technology, more recent applications of the technology can be found in the areas of computational lexicology (e.g. for compilation of terminology for technical dictionaries and glossaries) and machine-assisted translation. It is still uncommon, however, to apply term identification technology to higher-level semantic tasks, such as the description of the specialised technical domain that a document is part of, or the generation of a representation

https://doi.org/10.1017/S1351324999002090 Published online by Cambridge University Press

statistical pattern classification: *'stochastic neural net', 'joint distribution', 'feature vector', 'covariance matrix', 'training algorithm'*, and so forth.

lexical semantics: *'word sense', 'lexical knowledge', 'lexical ambiguity resolution', 'word meaning', 'semantic interpretation', 'syntactic realization'*, and so forth.

liquid chromatography: *'mobile phase', 'surface area', 'packing material', 'preparative chromatography', 'displacement effect'*, and so forth.

Fig. 1. Term sets (scientific prose).

of its topical content. On the surface, this appears surprising: the (abbreviated) term sets derived by the TERMS algorithm for several articles (illustrated in Figure 1; see Justeson and Katz (1995, appendix)) suggest that a strong case could in fact be made for the application of term identification technology to these two tasks. The problem, however, is finding within such term sets the right kind of granularity of detail with which to enhance the semantic space defined by the terms, in order to adequately describe a technical domain or characterise the content of a document. While intuitively appealing, term sets like these are lacking in several important respects.

First, there is a substantial difference between identifying technical domain within which a document's subject matter belongs and generating a characterisation of its content. The former is, arguably, the more straightforward of the two tasks, as Figure 1 suggests. A term set for a domain can be incrementally developed, by cumulatively growing the set after analysing more than one of the technical documents relevant to the domain. Alternatively, given an appropriate source document – for instance one that is guaranteed to describe the domain fully, e.g. for instructional purposes (such as a user guide, or a reference manual) – its term set can be assumed to provide a near-complete description of the technical domain, unlikely to expand further by analysing additional sources.

In contrast, the mapping from a term set to an accurate and useful representation of a document's content requires a strong sense of topicality and relevance. A central aspect of the document characterisation task is that of data reduction: constructing a representation of document content that is smaller than the original document – such as a set of terms – and yet highly indicative of its central points. Completeness here is defined not in terms of the domain description; instead, the problem is that of compactness and coherence. Term-like entities are clearly a part of such a representation, as observed by Justeson and Katz (1995), and focusing on certain aspects of them – for example, exploiting those aspects of their lexical semantics that are brought in focus by the relational contexts in which they are found – is essential for supplying the right level of detail.

Furthermore, what is also essential is to identify just those terms and relations that carry the most, and most central, of the document content, a problem that does not arise in the domain description task. Again taking an article about compilers as an example, the domain description problem would require the identification of all the terms that appear in relational contexts such as *'design of', 'modularisation of',* and *'optimisation of'* in order to generate a complete description of the COMPILERS

https://doi.org/10.1017/S1351324999002090 Published online by Cambridge University Press

domain. However, in a given document about compilers, it might be the specific terms *'optimizing compiler'* and *'formal programming languages'* in the relational contexts '*interpreting*' and '*constraining the source languages to*', respectively, that would be most representative of its content.

Finally, it is important to note that while lists like those in Figure 1 may be topical within a particular source document, other documents within the same domain as the source are likely to yield similar, overlapping, sets of terms (this is precisely the reason why technical term sets are not necessarily readily usable for document retrieval). This means that if the content of any single document is to be characterised in terms of a representative set of phrasal units, then these units must include not only (some subset of) the technical terms identified for the document, but also additional information that is indicative enough to uniquely characterise that particular document. Structuring the term set according to the relative representativeness of its members provides one means of achieving this result, while the identification of the relational contexts in which the most representative terms appear provides another. For example, in the context of a set of documents about compilers, finer-grained distinctions among the individual texts can be made by observing that in some, compilers are being *'optimised'* and *'modularised'*, while in others they are being *'designed'* and *'built'*.

1.3 Overview of the paper

The goal of this paper is to show that the identifiable, and reliably computable, discourse properties of technical terms and term-like phrases can be used as the primary information base for both higher-level semantic tasks discussed in the previous section, which we will refer to as *domain description* and *content characterisation.* Domain description is the process of identifying the technical domain within which a particular document's subject matter belongs; a description of a technical domain is an object that supports such an identification by providing a complete specification of the objects in the domain, the properties they possess, and the relations that hold between them. Content characterisation, on the other hand, is the construction of a concise, coherent, and useful representation of the core information content of a text. This representation is not a 'summary' in the traditional sense, as it does not attempt to restate the main idea of a text in well-formed prose (generated or extracted). Instead, it is an overview of document content at a higher-level of abstraction and finer granularity (phrases and relational contexts), geared towards providing a broad overview of the topical content of a text (see Boguraev and Kennedy (1998) and below for additional discussion).

The different natures of the domain description and content characterisation tasks raises at least two questions for us. With respect to domain description, and assuming strictly technical domains, the issue is to what extent a term set identified by methods similar to the one developed by Justeson and Katz (1995) can be leveraged for the purposes of *complete* domain description. With respect to content characterisation, we are concerned with a suitable generalisation – and relaxation – of the notion of a term, which still involves identification and extraction of phrasal

https://doi.org/10.1017/S1351324999002090 Published online by Cambridge University Press

units by essentially the same procedure, but generates a data structure that provides the basis for a more definitively characteristic representation of the core content of a document. This concern leads to a third, more general question: to what extent does the generalisation of the notion of technical term that is required to solve the problem of content characterisation provide a means of stretching the applicability of the technology beyond technical prose to documents from arbitrary domains and genres? These questions are addressed in the remainder of this paper.

2 Domain description

The primary problem with using term sets as domain descriptions is that they tend to be incomplete: at the very least, a description of a domain based on objects encountered in it ought to provide not only a listing of the objects in the domain, but also some specification of the various relations that hold between those objects. For instance, in the domain of lexical semantics mentioned earlier (see Figure 1), we would like to know that the notions of *'word sense'* and *'lexical ambiguity'* are related, or that a *'semantic interpretation'* can be *'derived'* or *'constructed'*.

As it turns out, acquiring such relational information is not dissimilar to the process of lexical acquisition, where the acquisition is done dynamically, on the basis of linguistic analysis of text sources (see Boguraev and Pustejovsky (1996)). Indeed, it is just a matter of emphasis whether, given a text source, essentially the same operational mechanisms are applied for the purpose of lexicon acquisition alone, or for the purpose of analysing that source. Johnston, Boguraev and Pustejovsky (1994) discuss, at some length, the similarities between the acquisition of relational information for domain objects (denoted by technical terms) and the identification of functional information about a domain by means of mapping out the linguistic analysis of a closed corpus describing this domain. We further develop this line of argument here, by describing a fully implemented procedure for domain description, which takes as input a source that provides a suitably complete description of the domain in question: the technical documentation associated with the reference guide for the Mac OS operating system.[1]

2.1 Automatic generation of an on-line assistance database

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?". For each application, assuming the existence of a database in a certain format, Apple Guide coaches the user through a sequence of definitional panels (describing key domain concepts), action steps (unfolding the

[1] Apple, Macintosh, and Mac OS are registered trademarks of Apple Computer, Inc.

https://doi.org/10.1017/S1351324999002090 Published online by Cambridge University Press

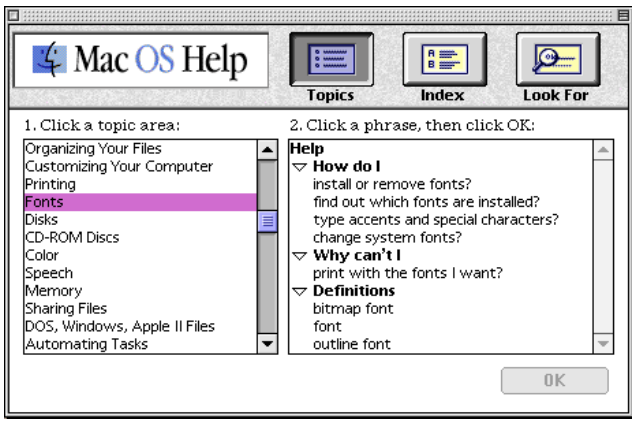


Fig. 2. Apple Guide.

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). The general ideas behind Apple Guide, illustrating some of the ways in which a domain concept (e.g. 'fonts') is described to a user, are illustrated in Figure 2.

An Apple Guide database is typically instantiated by hand, for individual applications, by trained instructional designers. Viewed abstractly, however, the information in such a database constitutes a complete domain description, insofar as it contains references to all and only the objects relevant to the domain of the application, as well as information about the properties they possess and the relations that hold between them. To answer questions like "What is a startup disk?", "How do I prepare a disk for use", or "Why can't I eject the floppy disk?", certain aspects of the domain (in this case the general domain of operating system level activities) need to be identified; this is accomplished by locating discussions of the relevant concepts in a technical manual. Consulting the manual will reveal that the term *'disk'*, for example, refers a kind of a domain object; that there are several types of disk, including *'floppy disks'*, *'startup disks'*, *'internal hard disks'*, etc.; that disks need to be *'prepared for use'*; that floppy disks can be *'ejected'*; and so forth.

Clearly, a terminology identification component, applied to suitably chosen technical documentation, could be utilised for generating a list of the objects (introduced by the technical terms) that form the foundation of a structure like the Apple Guide database. What is more important for our purposes is that by augmenting the same technology with an additional degree of linguistic analysis (incorporating knowledge of notions like derivational morphology, nominalisation, grammatical function, predicate-argument structure, and compounding), the set of technical terms derived from a source text can be refined to include not just all (and only) the domain objects, but also the relational context they appear in. This extra level of information provides exactly the enrichment of the basic term set required not only for the construction of the Apple Guide database, but also for the generation of a complete domain description.

https://doi.org/10.1017/S1351324999002090 Published online by Cambridge University Press

In essence, the process is not unlike one of generating lexical entries for a dynamically induced lexicon. However, we make a strong assumption about the completeness of the technical documentation, namely: the text source for the domain description (e.g. a reference manual) specifies everything that is essential and characteristic. This assumption is common for the purpose of concept learning in fixed domains, where it is conventional to assume some version of the 'closed world hypothesis'. From such an assumption, following a process of lexical context acquisition, feature induction can be limited to the literal contexts within which a word, or a term phrase, occurs. This is a standard assumption in the field of corpus-based lexical acquisition; it turns out, however, that when dealing with *closed* corpora (such as technical manuals or help guides), the special style of writing – aimed at a particular knowledge level of the reader, with a specific depth of description of the information included, and explicit with respect to all relevant details concerning a complete term set – offers an especially rich source for lexical induction and, ultimately, the construction of a full domain description (Johnston, Boguraev and Pustejovsky 1994).

The analysis framework is organised as a cascade of linguistic processing modules, which carry out lexical, morphological, and syntactic analysis of the source text, as well as term identification and additional processing targeted at relation and property identification. Base-level linguistic analysis is provided by a supertagger (LINGSOFT – Karlsson, Voutilainen, Heikkilä and Antilla 1995), which provides information about the part of speech, number, gender, and grammatical function (as well as other morpho-syntactic and lexical features) of expressions of English. Term identification involves running the tagged text through a parsing engine implemented as a pattern matcher. Patterns are written in a language that ranges over the lexical, morphological, and syntactic tags assigned by LINGSOFT, and are constructed to identify sequences of tokens that correspond to a slightly relaxed definition of technical term: whereas technical terms in the strict sense are complex nominals (containing at least two tokens) that are repeated in the text (see Justeson and Katz (1995) for details), we not only allow for single occurrences of complex nominals to 'count as' terms, but also single nominal tokens. The latter are subsequently filtered by a procedure that retains only those single-token nominals that occur as co-arguments with a true technical term in an identified relation (see below). The assumption here is that having identified *'access privilege'* and *'shared item'* as technical terms, an observed relation such as *'setting access privileges for a shared item over a network'* in the document context supports the inference that *'network'* should also enjoy term status.[2]

Once the set of domain objects has been established through the identification and extraction of technical terms, individual terms are used as anchor points for

[2] This sort of inference could be further strengthened by observations that indicate lexical relations between single-token nominals and true terms: in this case, the fact that in addition to *'network*, the term vocabulary includes expressions such as *'AppleTalk network'*, *'TokenTalk network connection'*, *'Token Ring network'*, *'network cable'*, *'networking expansion card'*, and so forth. Although we have developed a clustering engine that is able to make observations of this sort, it has not been incorporated into the implemented system described here. It is clear, however, that such an addition would be an important and useful addition to the procedure.

https://doi.org/10.1017/S1351324999002090 Published online by Cambridge University Press

```
Term: startup disk                Term: icon
  Relations:                        Relations:
     conserve space on []              assign color to []
     correct a problem with []         ........
     look for []                    Objects:
     recognize []                      Apple printer
     specify []                        network connection
     use internal hard disk as []      startup disk
  Objects:                             Trash
     System Folder                     ........
```

Fig. 3. Domain catalogue.

the identification of the relations and properties that are characteristic of the object interactions in the domain. Like term identification, relational expressions (and property-denoting expressions) are extracted by filtering the tagged text through a second set of phrasal patterns which target specifically the types of relations that we would expect to find in the domain: property-denoting predicates, transitive and ditransitive verb phrases, purpose clauses, and so on. The end result is that the technical vocabulary is expanded to include both object terms and action verbs, as well as common modifiers and other collocates for them. Due in part to the special nature of the particular genre, and in part to the closed nature of the text corpus, the entire procedure can be carried out to an extremely high degree of precision and recall, even though the initial linguistic analysis employs very shallow parsing methods. Evaluation is discussed in more detail in section 2.3.

In essence, the analysis procedure outlined here represents a progressively refined process of normalisation and data reduction over the source text, in which staged lexical acquisition ultimately derives a conceptual map of the technical domain by 'mining' for core domain objects, the relations they participate in, and the properties they possess. Mining for objects corresponds to the extraction of a domain-specific vocabulary of technical terms; mining for properties and relations corresponds to the instantiation of lexical semantic information for such terms. The domain model emerges from refinement of the initial set of core technical terms in the document, and the incremental construction of a *domain catalogue*: a set of terms (representing domain objects) with specifications of the relational contexts in which they appear and the properties that are predicated of them. Figure 3 illustrates the domain catalogue entries for the terms *'startup disk'* and *'icon'*.[3]

Mapping from domain catalogue 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 naturally 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 internal hard disk as a startup disk?"*, and so forth. The definitions and task sequences still have to be supplied externally, but the generation

[3] A complete catalog incorporates several different types of link between terms: `"objects"`, as exemplified here, `"properties"`, `"definition"`, `"used-for"`, and others. For the sake of simplicity, we focus here on terms and relations only.

https://doi.org/10.1017/S1351324999002090 Published online by Cambridge University Press

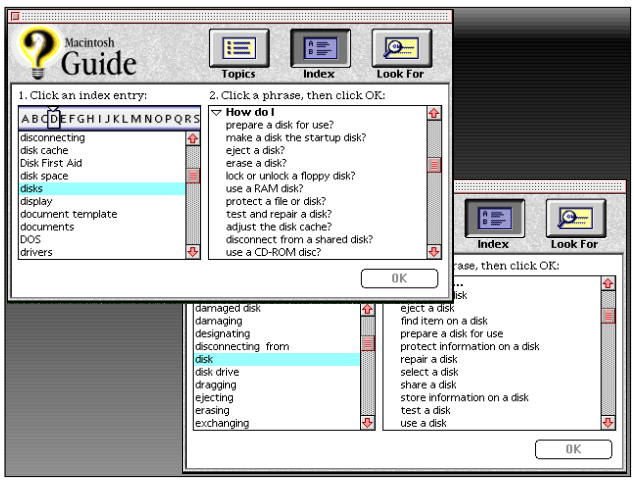


Fig. 4. Guide screens constructed from manually (upper left) and automatically generated databases.

of the database is fully automatic. In fact, the process of technical documentation analysis maintains a complete 'audit trail', relating items in the domain catalog to the relevant fragments in the 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.

The outcome of the analysis framework described here are illustrated in Figure 4. The upper left screen snapshot is from the Apple Guide shipping with the standard Mac OS configuration, which has been constructed manually by a team of professional instructional designers. The second snapshot displays, through the same delivery mechanism, a database which has been constructed fully automatically by the system described above, on the basis of an analysis of the *Macintosh User's Guide* (MacRef 1994), the primary technical documentation for Mac OS. Note, in particular, the *"How do I..."* lists, with entry points to detailed instructions concerning common tasks with specific objects (in this example, disks) in the Mac OS 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. (Actions associated with specific types of disk, e.g. *'startup disk'*, *'floppy disk'*, and so forth, appear elsewhere in the automatically generated database; see the discussion below.) Moreover, some additional action types have been identified, which are clearly relevant to this domain, but missing from the manually generated database: *'share a disk'*, *'find items on a disk'*.

2.2 From local ontologies to domain descriptions

As noted above, the procedure for automatic construction of help guides introduced in the previous section crucially relies on the assumption that the technical prose that provides the source material for the analysis is itself a *complete* description of

https://doi.org/10.1017/S1351324999002090 Published online by Cambridge University Press

domain (the 'closed world hypothesis'). It is this hypothesis that allows the induction of the structure of fixed domains to be anchored in the process of identifying the core set of objects in the domain, growing from that a set of associated, and relevant, relations and properties, and seeking closure over these sets. The reason for this is that for every assertion which is not explicitly given in the knowledge base, we may assume that its negation is true; this guarantees completeness of the model. For such closed worlds, closure on literals that are not axioms in the domain is a valid method for completing the theory. In general, feature induction without assuming a closed world model is intractable. Fortunately, the particular environment in which we seek structure to be imposed on the term sets (e.g. software applications in finite domains) falls under that definition: everything of import is described in the manual, and there is nothing essential that is not to be found somewhere in the manual.

As a result, the database acquisition procedure outlined in section 2.1 lends itself naturally to the induction of local ontologies, which organise related objects in the domain, and semantic frames, which provide internal structure for these objects. Local ontologies are hierarchies which capture subdomains of related objects within the content of a particular corpus. In other words, they group together domain objects of closely related types. For example, in the *Macintosh User's Guide* domain there is a local ontology of disks; the most basic type in this ontology is *'disk'*. The subtypes of *'disk'* are *'floppy disk'*, *'RAM disk'*, and *'hard disk'*. The type *'hard disk'* is further subtyped into *'internal hard disk'* and *'external hard disk'*. These relations are illustrated in Figure 5.[4]

The structuring of objects into local ontologies is accomplished by a two-step procedure. First, terms are clustered according to a simple notion of head identity, as illustrated by the groupings in Figure 5. Secondly, shared properties are identified by an algebra over relation sets, where such sets are loosely interpreted as meanings (extensional definitions) of the newly acquired domain objects, and established as high in the hierarchical structure as possible. This type of structuring is possible precisely because of the validity of the closed world hypothesis: in the domain of personal computer operating systems, a floppy disk is an object which can be initialised, ejected, protected, and so forth.[5]

[4] In addition to the hierarchical relations between terms in a local ontology such as the one illustrated in Figure 5, a number of different functional and semantic relations may hold between the sub-constituents of terms corresponding domain objects. For example, in a form like *'hard disk'* the component description *'hard'* acts as a sub-typing modifier, and specifies one of the *formal properties* of the disk. In a form like *'startup disk'*, the lexical item in the same position specifies the *purpose* to which the disk is put. The inferences required for making such distinctions should be based on a close inspection of the relational contexts in which the terms occur, which are already identified by our system. Deducing inferences of this sort is a hard problem, and beyond the current capabilities of our system. See Johnston, Boguraev and Pustejovsky (1994) for relevant discussion.

[5] In contrast, the standard assumptions and procedures in corpus-based lexicon acquisition typically restrict the types of induction techniques that are applicable to the corpus only to the literal contexts within which a word occurs, namely, the positive instances. As we have seen, however, the documents from which the domain characterising term sets are extracted are examples of *closed* corpora, with the result that learning is not restricted to positive data: the absence of a particular pattern with a specific lexical item can be interpreted as a negative instance for that word.

https://doi.org/10.1017/S1351324999002090 Published online by Cambridge University Press

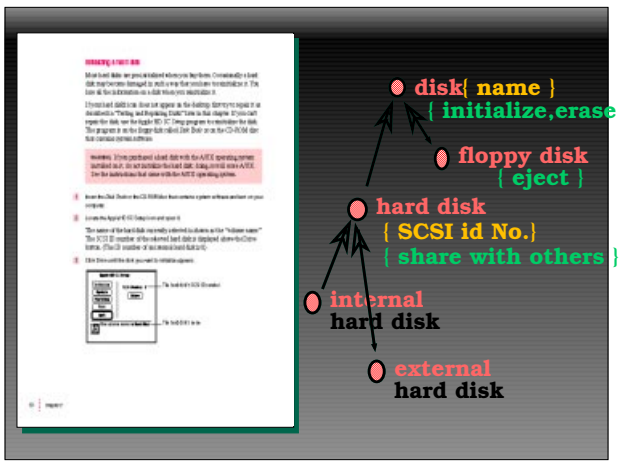


Fig. 5. A local ontology.

For illustration, consider the two terms near the top of the lattice in Figure 5: *'floppy disk'* and *'hard disk'*. The relation sets for these terms derived from the document are shown in Figure 6.

floppy disk: name [], erase [], access [], initialize [], use [], test [], save file on [], repair [], eject [], insert [] into disk drive, lock [], unlock [], protect []

hard disk: name [], save file on [], repair [], erase [], access [], initialize [], use [], test [], attach [] to printer, reinstall system software on [], copy program disk to []

Fig. 6. Relation sets.

As the predicates in the relation sets (Figure 6) clearly indicate, hard disks and floppy disks share certain properties: they can be named, initialised, accessed, and tested. On the other hand, floppy disks are things which can be inserted into a disk drive, ejected, locked, unlocked, and protected; no such actions can be applied to hard disks. Furthermore, hard disks are objects that can be attached to printers, have system software reinstalled on them, and have program disks copied to them; floppy disks are not. It is this kind of analysis which suggests that shared properties should be established as high in the ontology as possible: in this case, we have reliably learned some general information that applies to the whole class of disks, as well as more specific information, applicable to specific sub-classes.

A final principle for organising the objects in the domain catalogue emerges from the notion that semantically related object terms are found in similar contexts, as well as that a family of objects closely related by function in the domain will be classified in a tight category. Thus, while methods for elaborating the local ontology for *'disk'* identify, among others, *'RAM disk'* as a sub-type, clustering techniques similar to those described by Waterman (1996), but applied to phrasally tagged

https://doi.org/10.1017/S1351324999002090 Published online by Cambridge University Press

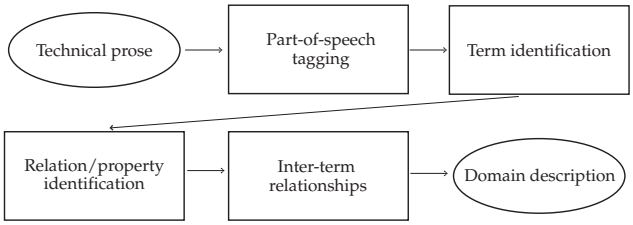


Fig. 7. Term-based domain description.

syntactic contexts, place *'RAM disk'* in a category together with *'battery power'*, *'powerbook computer'*, *'temporary storage device'*, and *'minimal system folder'*. This is precisely the type of information which, while hard to formalise, is very useful for building up the cross-reference links in the on-line assistance framework of Apple Guide.[6]

This is also the type of information that, when combined with the structure already imposed on individual local ontologies, ultimately leads to a strikingly accurate description of the technical domain from which the source document was drawn. The domain description takes the form of a set of terms which denote the domain objects, specified for the relations and properties with which they are associated, and interrelated by the relational connections among the local ontologies. Most importantly for our purposes, this domain description is constructed using simple and computationally non-intensive extensions to basic term identification technology, taking advantage of the distributional characteristics of technical terms within the restricted domain of technical documentation. The overall structure of the analysis framework for domain description is illustrated in Figure 7.

2.3 Evaluation

The question of appropriate evaluation metrics and methodology for applied language engineering systems is always a hard one, as often it is not entirely clear just what the right criteria should be, or what standard a system should be compared against. However, in the case of the approach to domain description described here, an evaluation procedure is relatively easy to define, as there exists a convenient – and appropriate – reference point: automatically derived databases for on-line help guides can be directly compared to their manually crafted counterparts, which are designed from the ground up by experienced instructional designers. Focusing on the specific example under discussion here, a comparison of the automatically derived database discussed in the previous section and the manually compiled Apple Guide database for Mac OS, as it ships with every computer, shows that the procedure for domain description described here is extremely accurate.

[6] In the Apple Guide delivery metaphor (see Figure 2), these links are used as an additional representational device ("See also...") at the same level as the "How do I?" and "Why can't I?" descriptors.

https://doi.org/10.1017/S1351324999002090 Published online by Cambridge University Press

Table 1. *Raw coverage data for automatically derived database*

	MANUAL normalised	AUTO normalised	AUTO true total	AUTO errors
Terms	84	79	464	41
Relations	57	51	260	30

The comparative analysis focuses on precision and recall in coverage by identifying both those elements that are common to both databases, and those elements which are unique to either database. Before moving to the actual numbers, two important aspects of the calculations should be clarified. First, although the manually generated reference guide provides a uniquely appropriate standard against which to judge the automatic system, direct comparison of the terms and relations in the two databases is not possible. This is because the database constructed by the instructional designers contains a number of expressions that do not appear anywhere in the source text from which the automatic database is derived (the *Macintosh User's Guide*). Since the automatic system should not be penalised for failing to identify terms and relations that do not appear in its source, we use as a reference value just those terms and relations that appear in both the manual database and the *Macintosh User's Guide*. Second, the total number of expressions identified by the automatic system is significantly larger than the number of terms in the manual guide. Not all of these constitute what human judges consider to be legitimate terms and relations, a point which also must be taken into account when devising an evaluation strategy.

With these considerations in mind, we have adopted recall and precision measures with the following definitions. *Recall* is defined with respect to the reference database as the normalised count of how many items included by the instructional designers have also been identified automatically as characteristic of the domain. *Precision* is defined as the ratio of 'good' hits, relative to all items identified by the automatic extraction methods.

Table 1 presents the raw data. The first column indicates the number of terms and relations that occur in both the manually generated database and the *Macintosh User's Guide*. The second column indicates the number of terms and relations identified by the automatic system that also occur in the manual database. Column three lists the total number of terms and relations automatically identified, and column four lists the number of expressions identified that do not represent legitimate objects and relations in the domain.

Table 2 summarises the overall quantitative analysis of the performance of the domain description system, using the definitions of recall and precision given above.

These results clearly demonstrate the practical viability of the domain acquisition techniques described here. By extension, these figures also quantify the recall and precision rates for the specific task of technical terminology identification, as applied to a very real problem.

While the results are strikingly good, there is also room for improvement. A very

https://doi.org/10.1017/S1351324999002090 Published online by Cambridge University Press

Table 2. *Recall and precision of term and relation identification*

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

important set of mismatches between the two databases, which do not manifest themselves in the numbers in Table 2, are those that go beyond simple lexical non-identity and into the realm of conceptual structure. For instance, the lack of automatic conceptual grouping in the domain description system fails to assign a category label like NETWORK SERVICES to the set *'network service'*, *'network access'*, *'network administrator'*, *'network cable'*, *'network connection'*, *'network control panel'*, *'Network Identity section'*, *'network software'*, *'network user'*, *'networking expansion card'*, *'networking software'*, all of which it identifies as individual domain objects. Such deficiencies could be at least partially overcome with the addition of a clustering engine, which would allow the system to at least assign the simpler, but less descriptive, category label NETWORK to this set (cf. footnote [2]).

In terms of an overall assessment of the approach, however, the numbers in Table 2 are encouraging, not only for the viability of inducing domain descriptions from technical documentation, but also for the utility of terminology identification technology for tasks much stronger than the relatively bland, and unfocused, indexing in general.

3 Content characterisation

The processing environment described in section 2 clearly demonstrates that the task of domain description can be effectively tackled by basic terminology identification technology, augmented with the richer toolset of methods and techniques for text-based lexicon acquisition. This section discusses the issues which arise when this technology is applied to the problem of content characterisation: generating a suitable representation of the 'aboutness' of a document.

The content characterisation task is of particular importance to applications where the goal is partial understanding of arbitrary texts (e.g. press releases, news articles, web pages, mail archives, and so forth); partiality being a necessary constraint in environments where no simplifying assumptions can be made concerning domain dependence and task specificity. Without such assumptions, and in particular, without the closed nature of technical domains and documentation, it is not clear what use can be made of term sets derived from arbitrary documents. Indeed, once we expand our attention to arbitrary types of texts, we cannot even talk of 'technical terms' in the narrower sense used so far. However, starting from the position that terms are phrasal units of a certain (very well defined) type and with some degree of topicality (as pointed out by Justeson and Katz (1995)), we may well ask whether the application of similar phrase identification technology generates phrase sets which

https://doi.org/10.1017/S1351324999002090 Published online by Cambridge University Press

can be construed as more broadly characteristic of the topical content of a document, in much the same way in which a term set can be used to derive a description of the domain to which technical prose belongs. The question concerns the wider applicability of linguistic processing targeted at term identification, relation extraction, and object cross-classification: can a set of phrases derived in this way provide a representational base which enables rapid, compact, and accurate appreciation of the information contained in an arbitrarily chosen document?

3.1 Technical terms as content indicators

The primary problem facing the application of terminology identification technology to arbitrary domains is one of a loss of robustness: outside of the strict confines of technical genres, there is a deterioration in the reliability of the linguistic analysis which underlies the high accuracy and precision of the TERMS algorithm of Justeson and Katz (1995). Several specific problems arise from scaling up term identification technology to an open-ended set of document types and genres.

First, since the domains are not closed, there is no strong sense in which notions of sublanguage can be leveraged. The 'clean', typically regular, language of technical documentation dissolves into a variety of writing styles, with wide variation in formality and grammaticality. Secondly, instead of working with a relatively small number of substantially sized documents, the data sets may be made up of a large number of small documents. As a result, the assumptions about how the entities discussed in a document lexicalise do not hold in the way which enables the identification of core technical terms. In addition to these problems, which concern the actual identification of terms, three specific problems arise when 'vanilla' term sets are considered as the basis for a content characterisation task.

Undergeneration The first is, broadly construed, a problem of undergeneration. For a set of phrases to be truly representative of document content, it must provide an exhaustive description of the entities discussed in the text. That is, it must contain not just those expressions which satisfy the strict phrasal definition of 'technical term', but rather every expression which mentions a participant in the events described in the text. Such broad coverage is precisely *not* the goal of canonical term identification, which extracts only those expressions that have a suitably rich amount of descriptive content (compound nominals and nominals plus modifiers), ignoring e.g. pronouns and reduced descriptions. Phrasal analysis must therefore be extended to include (at least) all the nominal expressions in a text.

Overgeneration Extending phrasal analysis in this way, however, results in a different kind of problem: without the constraints imposed by domain and genre restrictions, a full listing of all the terms that occur in the text is typically too large to be usefully presented as a representation of a document's content, even when attention is restricted to technical terms in the strict sense. Thus the second problem when using term sets as a basis for content characterisation is one of overgeneration: presentation of a list of phrases whose size rapidly leads to information overload.

https://doi.org/10.1017/S1351324999002090 Published online by Cambridge University Press

A system that extracts phrases on the basis of relaxed canonical terminology constraints, without recourse to domain or genre restrictions that might help to limit the size of the term set, will generate a term set far larger than a user can easily absorb. What is needed, then, is some means of establishing referential links between phrases, thereby reducing a large phrase set to just those that *uniquely* identify the participants in the discourse.

Differentiation The final problem is one of differentiation. While lists of terms such as the ones presented in section 1.2 (see Figure 1) might be topical for the particular source document in which they occur, other documents within the same domain are likely to yield similar, overlapping sets of terms. (This is precisely the reason why technical term sets are not necessarily readily usable for document retrieval.) The result is that two documents containing the same or similar terms could be incorrectly classified as 'about the same thing', when in fact they focus on completely different subtopics within a shared domain. In order to resolve this problem, it is necessary to differentiate term sets not only according to their membership, but also according to the relative representativeness (of document content) of the terms they contain.

These considerations point to the fact that a phrasal grammar alone is a weak technology for the task of domain-independent content characterisation. This is not particularly surprising, and certainly explains the fact that terminology identification is not widely considered an appropriate tool for content analysis. On the other hand, for any set of term-like phrases extracted from a given text, some are more representative of the document's content than others. What we need, then, in order to make a phrasal approach more viable, is a procedure for selecting precisely those expressions that are most representative of the topical content of the documents in which they occur. The remainder of this paper describes such a procedure, which we refer to as *salience-based content characterisation* (see also Boguraev and Kennedy (1998) for additional discussion of this type of approach). The ultimate goal of salience-based content characterisation is to identify and present in context a set of *topic stamps*: phrasal units that are most representative of the content of a text. Specifically, we identify topic stamps as term-like phrases that manifest a high degree of topical prominence, or *salience*, within contiguous discourse segments. As we will show in section 3.3, salience (in this sense) is directly computable as a function of the grammatical distribution and discourse properties of a phrase.

3.2 Salience-based content characterisation

The hypothesis underlying salience-based content characterisation is that the task of content characterisation can be defined as one of identifying phrasal units with lexico-syntactic properties similar to those of technical terms and with discourse properties that signify their status as most salient. This approach tackles the problems that arise when a large term set is used to characterise a document's content by identifying a function that is capable of ordering all terms with respect to some measure of their prominence in the discourse (the salience function), and selecting

https://doi.org/10.1017/S1351324999002090 Published online by Cambridge University Press

from this set only those that are most prominent (the topic stamps). This type of filtering is instrumental in overcoming the lack of coherence in the term sets, which are by definition somewhat diffuse for arbitrary documents.

Before moving to more detailed discussion of the procedure underlying salience-based content characterisation, it is worth considering an obvious, and arguably more straightforward, alternative to a salience-based approach, namely one that relies solely on the frequency of occurrence of terms. Such an approach starts from the assumption that the most important terms in a text, and the ones that are most indicative of its content, are just those that are mentioned most frequently. Clearly, it would be a simple matter to filter a term set for its N frequent members; why, then, do we adopt a more complex salience function?

The problem is that arbitrary documents – and certainly the kinds of documents we want to be able to handle (general news articles, arbitrary press releases, information digests, web pages, and so forth) – are typically smaller than the technical prose for which the canonical terminology identification procedures have been developed. Compounded by effects of wide diversity of genre, this makes frequency-based measures largely unusable for single document analysis. Our decision to use a salience feature, rather than a frequency measure, as the basis for determining which members of the large lists of content-bearing phrases identified by the technology constitute the most representative elements of the text, is an attempt to solve this problem.

Salience is a measure of the relative prominence of an object in a discourse: objects with high salience are the center of attention; those with low salience are at the periphery. Although salience loosely correlates with frequency, it provides a considerably more refined measure of the topicality of a particular phrase than what frequency counts reveal, as it takes into account additional factors such as syntactic and discourse prominence. By evaluating the members of a term set according to their salience, a partial ordering can be imposed which, in connection with an appropriate choice of threshold value, provides the basis for a reduction of the entire term set to only those terms which identify the most prominent participants in the discourse. This reduced set of terms – the topic stamps for the document – together with relational information of the sort discussed in section 2, can be folded into an appropriate presentation metaphor and presented as a characterisation of a document's content. In essence, we replace a weak selection procedure (a frequency-based analysis) which does not 'scale down' well for smaller and arbitrary documents with a more informed choice of representative phrases, which leverages semantic and discourse factors for reliable selection of salient objects. The fact that the higher-level semantic and discourse analysis is constructed directly from the base-level linguistic analysis (as will be made clear in section 3.3), rather than indirectly from inferences over frequency counts and probability distributions, makes it applicable to individual documents, even documents of small size and diverse nature.

The intuitions underlying our approach are best illustrated in the context of a specific example. Consider the following news article:[7]

[7] Adapted from an example of S. Nirenburg; the italics, whose relevance will become clear in section 3.3, are ours.

https://doi.org/10.1017/S1351324999002090 Published online by Cambridge University Press

Priest Is Charged with Pope Attack

A Spanish Priest was charged here today with attempting to murder the Pope. *Juan Fernandez Krohn*, aged 32, was arrested after *a man armed with a bayonet* approached the Pope while he was saying prayers at Fatima on Wednesday night.

According to the police, *Fernandez* told the investigators today that *he* trained for the past six months for the assault. *He* was alleged to have claimed the Pope 'looked furious' on hearing *the priest's* criticism of his handling of the church's affairs. If found guilty, *the Spaniard* faces a prison sentence of 15–20 years.

The kind of content characterisation we would seek from this text is best illustrated by its title, *"Priest Is Charged with Pope Attack"*. As a title, it encapsulates the essence of what the story is about; as a condensation of the article content, it highlights the essential components of the discourse: there are two actors, identified by their most prominent characteristics; one of them has been attacked by the other; the perpetrator has been charged; there is an implication of malice to the act. As an instance of content characterisation – brief *and* informative – this works by bringing an adequate set of salient facts together: the priest and the Pope are the main actors of an *"attempt to murder"* event.

In the general case of document analysis, where no *a priori* domain knowledge is assumed, the task of automatically generating this kind of phrasal summary can be arbitrarily complex; it is certainly beyond the capabilities of present day natural language processing (both in terms of depth of the analysis, and sophistication of the generation). However, an accurate approximation of the content of the title can be delivered by identifying the core components of this abstraction: *'priest'*, *'pope attack'*, *'charged with'*. It is from these components that, eventually, a message template would begin to be constructed, and it is these components, construed as topic stamps plus the relational contexts in which they appear, that a salience-based procedure for content characterisation seeks to identify and extract directly from the text. In many respects, the outcome of such a process is rather akin to a 'telegraphic' summary, as illustrated in the following extract from a novel, *Heart's Journey in Winter* (Buchan 1996), where a protagonist is summarising a diplomatic brief:

"... an intellectual and moral renewal," he said. Negotiations in Geneva. Ambassadors Polk and Kvertsovsky. Towards a Zero Solution. Strongest efforts. Reliable alliance partner. Fellow Germans on the far side of the Wall and barbed wire.

Our approach to content characterisation attempts to formalise these intuitions, by developing a technology for mining a document for the most salient – and by hypothesis, the most representative – phrasal units, plus the relational contexts in which they appear, with the goal of establishing precisely the kind of core specification of content that is captured in these examples.

This strategy for elaborating the phrasal analysis which underlies standard term identification thus relies heavily on notions like discourse structure and contextual factors – both of which are missing from the traditional technology for 'bare' terminology identification. The questions for this kind of approach to content characterisation can be summarized as follows. First, what formal definition of salience can be applied to a arbitrary sets of phrasal units in order to generate an ordering which accurately represents the relative prominence of the objects referred to in a document? Secondly, what linguistic information, available through

https://doi.org/10.1017/S1351324999002090 Published online by Cambridge University Press

scalable and robust identification technologies, can be levered to inform such a notion of salience? Third, what aspects aspects of discourse structure, also robustly computable, can be most effectively exploited for the dual purposes of calculating the topical prominence of expressions across an entire document and of constructing a coherent and concise representation of document content? These questions arise as we consider the overriding practical constraints imposed by the completely open-ended nature of the language, domain, style and genre of the texts we want to be able to handle; we answer them in the following section.

3.3 Computing salience: From term sets to topic stamps

Like the domain description procedure discussed in section 2, our approach to content characterisation is built around term identification technology. The term identification component, however, requires a number of extensions and enhancements to make it appropriate for the tasks discussed here, and to resolve the problems of overgeneration, undergeneration, and differentiation discussed in section 3.1. Four enhancements in particular play fundamental roles in this process: the extension of phrasal analysis beyond canonical technical terms, anaphora resolution, the salience calculation itself, and the identification of higher levels of discourse structure.

3.3.1 Term sets, coreference classes, and local salience

The problem of undergeneration is resolved by implementing a suitable generalisation and relaxation of the notion of a term, so that identification and extraction of phrasal units involves a procedure essentially like TERMS (Justeson and Katz 1995), but results in an exhaustive listing of all of the nominal expressions in the text. This is accomplished by running a phrasal grammar over text that has been analyzed by the LINGSOFT supertagger (Karlsson et al. 1995). The phrasal grammar targets expressions that consist of a head noun preceded by some number (possibly zero) of pre-nominal modifiers (nouns or adjectives). As a result, it extracts not just the complex nominals that meet the formal definition of technical terms, but reduced descriptions and pronouns as well. Phrasal analysis yields the set of all nominal expressions occurring in the text, which we refer to as an *extended phrase set*. As with the domain description procedure, the 'terms' extracted by extended phrasal analysis are used as anchors for a second stage of processing targeted at the identification of the relational and clausal contexts in which they appear.

In order to eliminate the problem of overgeneration, it is necessary to reduce the extended phrase set to a smaller set of expressions which uniquely identify the objects referred to in the text, hereafter a *referent set*. We make the simplifying assumption that every phrase identified by extended phrasal analysis constitutes a 'mention' of a participant in the discourse (see Mani and MacMillan (1996) for discussion of the notion of mention in the context of proper name interpretation); in order to construct a referent set, it is necessary to determine which expressions constitute mentions of the same referent.

Coreference is established largely through the application of an anaphora resolu-

https://doi.org/10.1017/S1351324999002090 Published online by Cambridge University Press

tion procedure that is based on the algorithm developed by Lappin and Leass (1994). The fundamental difference between our algorithm (described in detail elsewhere (Kennedy and Boguraev 1996a, 1996b)) and the one developed by Lappin and Leass (1994) is that it is designed to provide a reliable interpretation from the shallow linguistic analysis of the input provided by the LINGSOFT tagger (the Lappin–Leass algorithm works from the analysis provided by the McCord Slot Grammar parser (McCord 1990)). The basic approach to anaphora resolution, however, is the same. The interpretation procedure involves moving through the text sentence by sentence and analysing the nominal expressions in each sentence from left to right (expressions identified by the phrasal grammar are marked both for overall position in the text and for the sentence in which they occur). There are two possible outcomes of this examination. Either an expression is identified as a mention of a new participant in the discourse, or it is taken to refer to a previously mentioned referent, i.e. it either introduces a new referent or is identified as coreferential with some other expression in the text.

Coreference is determined by a three step procedure. First, a set of candidate antecedents is collected, which includes all nominals within a local segment of discourse. Secondly, those expressions with which an anaphoric expression cannot possibly corefer, by virtue of morphological mismatch or syntactic restrictions, are eliminated from consideration. (See Kennedy and Boguraev (1996a) for a discussion of how syntactic relations can be inferred from a shallow linguistic analysis.)

Finally, the remaining candidates are ranked according to their relative *salience* in the discourse (see below), and the most salient candidate is selected as the antecedent for the anaphor. (In the event that a coreference link cannot be established to some other expression, the nominal is taken to introduce a new referent.) Linguistic expressions that are identified as coreferential are grouped into equivalence classes, or *coreference classes*, and each coreference class is taken to represent a unique referent in the discourse. For any text, the set of such coreference classes constitutes its reference set.

A crucial component of this anaphora resolution procedure is the computation of a salience measure for terms that are identified as candidate antecedents for an anaphoric expression. This measure, which we refer to as *local salience*, is straightforwardly determined as a function of how a candidate satisfies a set of grammatical, syntactic, and contextual parameters, or 'salience factors' (this term is borrowed from Lappin and Leass (1994)). Individual salience factors are associated with numerical values, as shown in Figure 8.[8] The local salience of a candidate is the sum of the values of the salience factors that are satisfied by some member of the coreference class to which the candidate belongs; values may be satisfied at most once by each member of the class.

The most important consequence of this characterisation of local salience is that the numerical values associated with the salience factors correspond to a relational

[8] Our salience factors mirror those used by Lappin and Leass (1994), with the exception of POSS, which is sensitive to possessive expressions, and CNTX, which is sensitive to the discourse segment in which a candidate appears (see section 3.3.3).

https://doi.org/10.1017/S1351324999002090 Published online by Cambridge University Press

SENT(*term*) = 100 iff *term* is in the current sentence
CNTX(*term*) = 50 iff *term* is in the current discourse segment
SUBJ(*term*) = 80 iff *term* is a subject
EXST(*term*) = 70 iff *term* is in an existential construction
POSS(*term*) = 65 iff *term* is a possessive
ACC(*term*) = 50 iff *term* is a direct object
DAT(*term*) = 40 iff *term* is an indirect object
OBLQ(*term*) = 30 iff *term* is the complement of a preposition
HEAD(*term*) = 80 iff *term* is not contained in another phrase
ARG(*term*) = 50 iff *term* is not contained in an adjunct

Fig. 8. Salience factors.

structure that is directly computable on the basis of grammatical information about particular terms.[9] This relational structure in turn provides the basis for ordering candidate antecedents according to their relative salience in some local segment of discourse, and (by hypothesis) their likelihood as antecedents for a pronoun. Moreover, the overall success of the anaphora resolution procedures built on top of the salience measures derived from these values – Lappin and Leass (1994) report 85% accuracy; our system, which uses a shallower linguistic analysis, runs at 75% accuracy (see Kennedy and Boguraev (1996a) for details) – provides clear evidence for its accuracy and appropriateness as a representation of the prominence of expressions in a discourse.

3.3.2 Discourse salience and topic stamps

Anaphora resolution solves a number of the problems that arise when term identification technology is extended to work on arbitrary texts. First, it reduces the total list of terms identified by extended phrasal analysis to just those that uniquely identify objects in the discourse. Second, it establishes crucial connections between text expressions that refer to the same entities. This latter result is particularly important, as it provides a means of 'tracking' occurrences of prominent expressions throughout the discourse (see Kennedy and Boguraev (1996b) for discussion of this point). A much broader consequence of the approach to anaphora resolution outlined in section 3.3.1, however is that it introduces both a working definition of salience and a mechanism for determining the salience of particular linguistic expressions based on straightforwardly computable grammatical properties of terms. Our proposal is that the principles underlying the computation of salience for anaphora resolution can be extended to the computation of a more global measure of salience, reflecting the prominence of expressions across the entire discourse, which can be used as the basis for the identification of topic stamps.

The hypothesis underlying the anaphora resolution procedure is that the local salience of a referent, computed as a function of its frequency of mention and the grammatical distribution of the terms that refer to it, reflects its prominence within

[9] Note that the relational structure imposed by the salience factors is justified both linguistically, as a reflection of the functional hierarchy discussed in Keenan and Comrie (1977), as well as by experimental results (Lappin and Leass 1994).

https://doi.org/10.1017/S1351324999002090 Published online by Cambridge University Press

some local segment of discourse. An important feature of local salience is that it is variable: the salience of a referent decreases and increases according to the frequency with which it is mentioned (by subsequent anaphoric expressions). When an anaphoric link is established, the anaphor is added to the equivalence class to which its antecedent belongs, and the salience of the class is boosted accordingly. If a referent ceases to be mentioned in the text, however, its local salience is incrementally decreased. This approach works well for the purpose of anaphora resolution, because it provides a realistic representation of the antecedent space for an anaphor by ensuring that only those referents that have mentions within a local domain have increased prominence.

In contrast, the goal of salience-based content characterisation is to generate a picture of the prominence of referents across the entire discourse. In order to generate this broader picture of discourse structure, we introduce an elaboration of the local salience computation described in section 3.3.1 that uses the same conditions to calculate a non-decreasing salience measure, which we refer to as *discourse salience*. Specifically, discourse salience is calculated for a coreference class by computing the sum of the salience factors satisfied by the members of the class.[10] The discourse salience of a particular referent is thus a function not only of its frequency of mention – the more terms in a coreference class, the higher the discourse salience of the entire class – but also of its grammatical distribution: as the values in Figure 8 indicate, some syntactic positions (e.g. subject, head of a phrase) correspond to higher salience values than others.

Since coreference classes are composed of terms that have been identified as coreferential, discourse salience provides a means of 'tracking' salient expressions as they occur in the text. In conjunction with such tracking of referents, made possible by anaphora resolution, discourse salience provides the basis for a representation of discourse structure that indicates the topical prominence of specific terms across the text. In particular, by associating every term with a discourse salience value, a (partial) ordering can be imposed on the members of a term set according to their relative prominence in the discourse. This ordering provides the additional structure necessary to use term sets for the purpose of content characterisation (see section 3.2), because it can be sued to determine which phrases should be selected as broadly representative of the content of a document, i.e. which phrases should be identified as topic stamps. We turn to the final details of the selection procedure in the next section.

3.3.3 Discourse structure and content characterisation

As the example article discussed in section 3.2 demonstrated, two well chosen phrases may be a sufficient indicator for the core content of two paragraphs. Clearly, a two page document is unlikely to be equally well served by two phrases alone, no matter how salient they might be. In order to address the problem of variability in

[10] This calculation excludes the two factors designed to indicate the proximity of a candidate antecedent to an anaphor, SENT and CNTX.

https://doi.org/10.1017/S1351324999002090 Published online by Cambridge University Press

document size, we incorporate a process of *discourse segmentation*, which creates a representation of the overall discourse structure of a text by identifying those points that correspond to shifts in topic. Segmenting a long document into a number of text fragments – where each fragment is characterised by relatively high degree of internal coherence, and segment boundaries are defined as the points in the narrative where there are relatively abrupt shifts in topicality and, hence, flow of discourse – reduces the content characterisation task to that of finding topically coherent segments of text, and then identifying an appropriate number of topic stamps for each segment.

The approach to segmentation we adopt implements a similarity-based algorithm based on Hearst's TextTiling procedure (Hearst 1994), which identifies topically coherent discourse segments by comparing adjacent blocks of text for overall lexical similarity. By calculating the discourse salience of referents with respect to the results of discourse segmentation, each segment can be associated with a listing of the most salient expressions within the segment, i.e. each segment can be assigned a set of topic stamps. Specifically, we identify the topic stamps for every segment of text S as the n highest ranked referents in S, where n is a scalable value.

The result of these calculations, the set of segment-topic stamp pairs, ordered according to linear sequencing of the segments in the text, is the data structure on the basis of which the content analysis of the entire document is constructed. The problem of content characterisation of a large text is thus reduced to the problem of finding topic stamps for each discourse segment.

3.4 Capsule overviews

The end result of the document analysis process is a compact text-based object, which encapsulates via its pivotal points, the topic stamps, the core content of the source. In order to distinguish this type of object from what current work in the field refers to as a 'document summary', we adopt the term *capsule overview*. In essence, a capsule overview is simply a structured listing of the linguistic expressions referring to the most topically prominent objects in each segment of discourse (the topic stamps), and a specification of the relational contexts (verb phrases, minimal clauses, etc.) in which these expressions appear. (More detailed discussion concerning the positioning of this research into the space of document summarisation work can be found in Boguraev and Kennedy (1998).)

We illustrate the approach highlighting certain aspects of the analysis of a recent *Forbes* article (Hutheesing 1996). The document is of medium-to-large size (approximately four pages in print), and focuses on the strategy of Gilbert Amelio (Apple Computer's former CEO) concerning a new operating system for the Macintosh. Too long to quote here in full, Figure 9 reproduces a passage from the beginning of the article that contains the first, second and third segments, as identified by the discourse segmentation component (see section 3.3.3; in the example, segment boundaries are marked by extra vertical space).

The relevant sections of the capsule overview for this document (corresponding to the three segments of the passage quoted) are shown in Figure 10; this overview

https://doi.org/10.1017/S1351324999002090 Published online by Cambridge University Press

"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 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.

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.

Fig. 9. Example: (segmented) document.

was automatically generated by a fully implemented and operational system, which incorporates all of the processing components identified in section 3.3. 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. Note that the listing of topic stamps and contexts shown here is only the core data out of which a capsule overview is constructed – such a listing is arguably not the most effective and useful presentation of this information. Questions pertaining to the right presentation metaphor (or metaphors, allowing for task-dependent differences

https://doi.org/10.1017/S1351324999002090 Published online by Cambridge University Press

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*

Fig. 10. Example: capsule overview.

in delivery) are addressed in detail by Boguraev, Wong, Kennedy, Bellamy, Brawer and Swartz (1998).

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 introductory 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 – reveals the larger context in which the topic stamp occurs (e.g. a sentence), 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 four phrases 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 are somewhat 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 exemplified above is associated with the stamps GILBERT AMELIO and NEW OPERATING SYSTEM. The reasons, and linguistic rationale, for the selection of these particular noun phrases as topical are essentially identical to the intuition behind *'priest'* and *'Pope'* being the central topics of the example in Section 3.2. The computational justification for the choices lies in the extremely high values of salience, resulting from taking into account a number of factors: coref-

https://doi.org/10.1017/S1351324999002090 Published online by Cambridge University Press

erentiality between *'Amelio'* and *'Gilbert Amelio'*, coreferentiality 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. Even if we are unable to generate a single phrase summary in the form of, say, *"Amelio seeks a new operating system"*, the overview for the closing segment comes close; arguably, it is even better than *any* single phrase summary.

As the discussion of this example illustrates, a capsule overview is derived by a process which facilitates partial understanding of the text by the user. The final set of topic stamps is designed to be representative of the core of the document content. It is *compact*, as it is a significantly cut-down version of the full list of identified terms. It is highly *informative*, as the terms included in it are the most prominent ones in the document. It is *representative* of the whole document, as a separate topic tracking module effectively maintains a record of where and how referents occur in the entire span of the text. As the topics are, by definition, the primary content-bearing entities in a document, they offer *accurate* approximation of what that document is about.

3.5 Summary

At its core, 'salience-based content characterisation' involves defining a selection procedure, operating over a larger set of phrasal units than that generated by a base-level term identification procedure, that makes informed choices about the degree to which each phrase is representative of the text as a whole, and presents its output in a form which retains contextual information at different levels of granularity. This process adapts the formal characterisation of salience used in anaphora resolution, building on and extending an independently needed component of any higher-level approach to text analysis. Moreover, it presupposes very little in the way of linguistic processing, working solely on the basis of the shallow analysis provided by a (super-)tagger. It thus meets the important requirements of domain independence and robustness of performance, extending the applicability of the technology to texts across a wide range of genres and styles. Figure 11 provides a schematic illustration of the organization of the content characterisation procedure.

4 Conclusion

The particular strength of terminology identification lies in the guaranteed robustness of the technology, which is itself due to the clear understanding of how the linguistic properties of technical terms manifest themselves via a variety of lexical, syntactic and discourse factors. Within the confines of a specific genre, namely scientific prose, this makes for a powerful tool for certain types of content management.

What we have shown in this paper is that such a methodology, which maps certain

https://doi.org/10.1017/S1351324999002090 Published online by Cambridge University Press

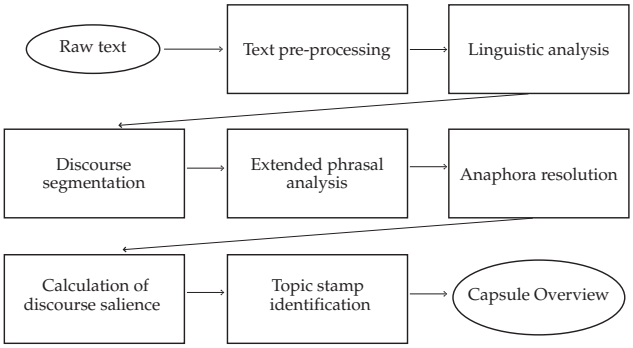


Fig. 11. Term-based content characterisation.

syntactic, semantic and discourse properties of a larger set of information-bearing units onto their local contexts in text documents, enables the definition of new class of content analysis algorithms, targeted at very different information management tasks and document genres. These can be viewed as a natural extension of term extraction, but result in term-based representations of document content that are nevertheless considerably more coherent than simple term enumeration.

References

Boguraev, B. and Kennedy, C. (to appear 1999) Salience-based content characterisation of text documents. In Mani, I. and Maybury, M., editors, *Advances in Automatic Text Summarization*. MIT Press.

Boguraev, B. and Pustejovsky, J., editors (1996) *Corpus Processing for Lexical Acquisition*. MIT Press.

Boguraev, B., Wong, Y. Y., Kennedy, C., Bellamy, R., Brawer, S. and Swartz, J. (1998) Dynamic presentation of document content for rapid on-line browsing. *AAAI Spring Symposium on Intelligent Text Summarization* , pp. 118–128. Stanford, CA.

Bourigault, D. (1992) Surface grammatical analysis for the extraction of terminological noun phrases. *15th International Conference on Computational Linguistics*. Nantes, France.

Buchan, J. (1996) *Heart's Journey in Winter*. Harvill Collins.

Dagan, I. and Church, K. (1995) Termight: identifying and translating technical terminology. *4th Conference on Applied Natural Language Processing*. Stuttgart, Germany.

Hearst, M. (1994) Multi-paragraph segmentation of expository text. *32nd Annual Meeting of the Association for Computational Linguistics*. Las Cruces, NM.

Hodges, J., Yie, S., Reighart, R. and Bogges, L. (1996) An automated system that assists in the generation of document indexes. *Natural Language Engineering*, **2**(2): 137–160.

Hutheesing, N. (1996) Gilbert Amelio's grand scheme to rescue Apple. *Forbes Magazine*.

Johnston, M., Boguraev, B. and Pustejovsky, J. (1994) The structure and interpretation of compound nominals. *AAAI Spring Symposium on Generativity and the Lexicon*. Stanford, CA.

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

Karlsson, F., Voutilainen, A., Heikkilä, J. and Antilla, A. (1995) *Constraint grammar: A language-independent system for parsing free text*. Mouton de Gruyter.

Keenan, E. and Comrie, B. (1977) Noun phrase accessibility and universal grammar. *Linguistic Inquiry*, **8**: 62–100.

https://doi.org/10.1017/S1351324999002090 Published online by Cambridge University Press

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

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

Lappin, S. and Leass, H. (1994) An algorithm for pronominal anaphora resolution. *Computational Linguistics*, **20**(4): 535–561.

MacRef (1994) *Macintosh User's Guide*. Apple Computer, Inc., 20525 Mariani Avenue, Cupertino, CA 95014–6299, USA.

Mani, I. and MacMillan, T. (1996) Identifying unknown proper names in newswire text. In Boguraev, B. and Pustejovsky, J., editors, *Corpus Processing for Lexical Acquisition*, pp. 41–59. MIT Press.

McCord, M. M. (1990) Slot grammar: a system for simpler construction of practical natural language grammars. In Studer, R., editor, *Natural language and logic: international scientific symposium: Lecture Notes in Computer Science*, pp. 118–145. Springer-Verlag.

Salton, G. (1988) Syntactic approaches to automatic book indexing. *26th Annual Meeting of the Association for Computational Linguistics*. Buffalo, NY.

Salton, G., Zhao, Z. and Buckley, C. (1990) A simple syntactic approach for the generation of indexing phrases. *Technical Report 90-1137*, Department of Computer Science, Cornell University.

Waterman, S. (1996) Distinguished usage. In Boguraev, B. and Pustejovsky, J., editors, *Corpus processing for domain acquisition*, pp. 143–172. MIT Press.

https://doi.org/10.1017/S1351324999002090 Published online by Cambridge University Press