Natural Language Engineering **2** (2): 137–160 © 1996 Cambridge University Press

An automated system that assists in the generation of document indexes

JULIA HODGES†, SHIYUN YIE†, RAY REIGHART‡ AND LOIS BOGGESS†

† *Box 9637, Mississippi State, MS 39762–9637, USA*
‡ *Department 7, Chemical Abstracts Service, 2540 Olentangy River Road, Columbus, OH 43202-1505, USA*

(*Received 11 January 1995; revised 5 August 1996*)

Abstract

In this article, we describe AIMS (Assisted Indexing at Mississippi State), a system intended to aid human document analysts in the assignment of indexes to physical chemistry journal articles. The two major components of AIMS are a natural language processing (NLP) component and an index generation (IG) component. We provide an overview of what each of these components does and how it works. We also present the results of a recent evaluation of our system in terms of recall and precision. The recall rate is the proportion of the 'correct' indexes (i.e. those produced by human document analysts) generated by AIMS. The precision rate is the proportion of the generated indexes that is correct. Finally, we describe some of the future work planned for this project.

1 Introduction

The KUDZU (Knowledge Under Development from Zero Understanding) project in the Intelligent Systems Research Laboratory of the Department of Computer Science at Mississippi State University addresses the extraction of knowledge from natural language text using methods that are as fully automated and domain-independent as possible (Boggess *et al*., 1991, 1995; Hodges and Cordova, 1993). Most recently, these efforts have been focused on the development of a system that aids human document analysts in the assignment of indexes to journal articles that have been published in the area of physical chemistry (Boggess and Hodges, 1994). This system is called AIMS (Assisted Indexing at Mississippi State). AIMS is being developed in collaboration with the Chemical Abstracts Service (CAS) of the American Chemistry Society in Columbus, Ohio. Approximately sixty of CAS's 400 document analysts read and index articles in the physical chemistry domain.

AIMS consists of two major components – the natural language processing (NLP) component and the index generation (IG) component. In this paper, we present an overview of the current version of AIMS, including a description of both major components. We begin with a brief discussion of other research projects that are

https://doi.org/10.1017/S1351324996001325 Published online by Cambridge University Press

related to this research. We then provide an overview of the KUDZU research project, followed by a brief discussion of the overall goal of the current research effort. Included in this section is a description of how AIMS can be used by human document analysts at CAS to improve the accuracy and efficiency with which they are able to generate the indexes for a physical chemistry journal article. Following that section is a description of the NLP component. Next, we describe the development of the IG component and how it generates indexes from the output generated by the NLP component. We then report the success rates of the AIMS system in producing indexes for a testbed of articles. Finally, we discuss the future work planned for this project.

2 Related work

A number of probabilistic models have been defined for the purpose of assigning indexes to documents. Some of these have estimated the probabilistic weights for various indexes by using the number of queries that relate to a specific document or document component as a way of determining relevance (Maron and Kuhns, 1960; Kwok, 1986). Fuhr and Buckley (1991) have described a probabilistic learning approach to the selection of indexes for documents. They introduced the notion of 'relevance description', which is an abstraction from term-document pairs, so that the parameters for their probabilistic model are not restricted to relevance information extracted from a specific document. A relevance description 'comprises a set of features that are considered to be important for the task of assigning weights to terms with relation to documents. So a relevance description $x(t_i, d_m)$ contains values of attributes of the term t_i, the document d_m and their relationship.' This approach is similar to those that have been used in pattern recognition. The Fuhr and Buckley model may be viewed as a learning method because relevance feedback data is used to derive the probabilities.

Kwok and Grunfeld (1994) have described PIRCS (Probabilistic Indexing and Retrieval – Components – System), an information retrieval system that uses Bayesian reasoning to determine those documents that are most relevant to a user's needs. One of the techniques used by Kwok and Grunfeld has been to pay attention to certain subparts of a document as being most likely to be important in determining the relevance of the document. As they have noted, '[m]any articles … introduce their themes by stating the most important concepts and contents at the beginning of a document. They also summarize at the end.' In the development of the AIMS project, we have found that focusing on certain subparts of an article can also be helpful in the determination of appropriate indexes for that article.

Ginsberg (1993) has described an experimental system called WorldViews, which provides automatic indexing and information retrieval by making use of 'a structured set of subject headings or descriptors', or what Ginsberg calls a 'thesaurus'. Thus the automatic indexing process consists of assigning thesaurus entries to documents so that these entries describe the contents of the documents.

The CLARIT System identifies candidate noun phrases from the parsed text of a document, maps them into a morphologically normalised form, and matches the resulting terms (or concepts) against a thesaurus of domain-specific terminology. It

https://doi.org/10.1017/S1351324996001325 Published online by Cambridge University Press

then ranks and weights the terms according to their distribution statistics in the document, a collection of domain-specific documents, and the Brown Corpus of English. Although CLARIT cannot generate indexes that consist of words or phrases that do not appear in the text, it does not rely on exact string matches of whole phrases (Evans *et al.*, 1991).

In an extension to CLARIT's indexing component, Evans and Zhai developed an approach that uses corpus-based distribution statistics as well as linguistic heuristics to extract subcompounds from noun phrases. They found that using the subcompounds rather than entire complex noun phrases aided in the problem of matching phrases which are syntactically different yet semantically similar. The resulting system showed improvement in both recall and precision (Evans and Zhai, 1996).

The Text REtrieval Conferences (TRECs) feature evaluations of numerous document retrieval systems, including systems which take advantage of natural language processing. While these latter face many of the same problems and make use of many of the same techniques that the AIMS system does, they operate in a somewhat different environment. An important difference, for example, is that these systems have access to both the user query and the documents. TREC competitors also are expected to operate in an interactive process, whereby a user might be presented with an initial set of documents proposed by the system and select some of the proposed documents as relevant, at which point the system can reweight the relevance of words and concepts from both the user's query and the retrieved documents to the documents in the database. The reassessment of relevance of terms as the user interacts with the retrieval system is called *late binding* by Lewis and Sparck Jones (1996), who stress the importance of late binding in user-centered interactive full-natural-language document retrieval systems. In contrast, the AIMS system is charged with producing indexes comparable to those produced by human document analysts, for documents in what Lewis and Sparck Jones would call a controlled language system: the indexed concepts must belong to a pre-specified concept hierarchy, though they may be accompanied by free text which is not from a restricted vocabulary.

For example, Strzalkowski *et al.* (1995) report on a text retrieval system which takes advantage of natural language processing techniques. Their system, like AIMS, uses a part of speech tagger and a parser, and identifies compound noun phrases. In contrast to AIMS, they also *stem* the terms used for indexing, both from a user query and from a database document. For example, in their system, the stem of 'enjoyment' would be 'enjoy', and the stem of 'induction' would be 'induce'. On the one hand, stemming reduces the need for an exact match in morphology between words in the query and words in a document. As we explain later in this paper, however, the indexes derived by the AIMS system may consist of words and phrases which never occur in the indexed document in the literal form of the index. As a consequence, some of the benefit of stemming is achieved in our system by other means.

Another issue which relates our work to other systems is the issue of multiple senses of a given word. As described below, not only can a word have more than one part of speech, but given that its part of speech in a sentence has been correctly deduced,

https://doi.org/10.1017/S1351324996001325 Published online by Cambridge University Press

it can have more than one sense, just as most words in a dictionary have more than one definition. Some of the automated elements of the AIMS system make semantic distinctions at a broad level, but none of these classification schemes embed into our lexicon the finer levels of distinction which are the focus of researchers who work with semantic lexicons. There is a growing body of such work (see, for example, Pustejovsky (1991, 1995); Boguraev and Pustejovsky (1996), and the proceedings of a workshop on that topic (Association for Computational Linguistics, 1996)). It will be evident to the reader that our work implicitly echoes the 'view in the field that time and effort notwithstanding, a substantial fragment of lexical knowledge can only be, and therefore should be, crafted by hand' (Boguraev *et al*., 1996: 5). In our own work, that has translated into (1) using past human experience in addition to semi-automated means to establish gross-grained semantic distinctions, (2) letting the part-of-speech tagging system use short-range context to hypothesize one of the coarser semantic classes for a given word or phrase, when appropriate, and (3) using a range of other tools such as the weighting scheme described in Section 6 to determine which of several indexable concepts a word with multiple senses is likely to indicate.

Soderland and Lehnert (1994) have described information extraction systems as systems that analyze 'real world text such as newswire stories'. Such a system 'returns a structured representation of just the information from the text that is relevant to a user's needs, ignoring irrelevant information.' An information extraction system must first determine the objects being discussed in a particular text. It must then identify relationships among the objects that may be explicitly stated in or inferred from the text. This second step, referred to as discourse analysis, is very similar to the identification of objects and their relationships that is done by the KUDZU KE component. Soderland and Lehnert have reported the initial development of Wrap-Up, a trainable discourse component for an information extraction system. Wrap-Up makes use of machine learning techniques to discover a set of classifiers for a training set of texts. In addition, Wrap-Up suggests a feature set for each classifier it discovers. The intent of this system is to reduce the amount of hand crafting required in information extraction systems to provide domain-specific information and to make the information extraction systems more portable from domain to domain.

3 The KUDZU research project

Originally, the KUDZU project addressed the problem of developing a computer system capable of extracting knowledge by processing natural language text that is in machine-readable form. The primary testbed was *The Merck Veterinary Manual* (Fraser and Mays, 1986), although some testing was also done with technical material from the field of geology. An important capability of the system was that it could bootstrap itself. That is, beginning with only a description of the types of objects and relationships to be stored in the knowledge base, the system could start with an empty knowledge base and build the knowledge base as it processed the text (Cordova and Hodges, 1992, 1992a; Hodges and Boggess, 1992; Hodges *et al*., 1993; Boggess *et al*., 1995). The techniques that were used are similar to what other researchers have called the automatic generation of sublanguage semantic patterns (e.g. Hirschman, 1986;

https://doi.org/10.1017/S1351324996001325 Published online by Cambridge University Press

Sager, 1987). We have adapted the components of the KUDZU system to address the current problem of identifying indexes for scientific articles.

The NLP component of the KUDZU system processes text by labeling or tagging each word with syntactic and semantic information such as 'noun/body part'. It then makes use of a simple parser to produce flat parse structures. Specialists are included in the NLP component to handle specific tasks such as the attachment of prepositional phrases to the sentential components that they modify and the identification of those sentential elements that make up the conjuncts of a conjunction. More detailed information about the NLP component of the KUDZU system has been reported elsewhere (Boggess *et al*., 1991; Boggess *et al*., 1995).

The knowledge extraction (KE) component of the KUDZU system uses the syntactic structures produced by the NLP component to extract from the text the objects and relationships among objects being described in the text. For example, the object class 'symptom' and the object class 'disease' may be related by a relationship called 'is a symptom of'. The KE component stores the extracted objects and relationships in an object-oriented knowledge base. Initially, the knowledge base is empty. The KE component builds the knowledge base 'from scratch' using only the parsed text and a schema-like description of the different classes of objects and relationships among objects found in the application domain. The schema is the only domain-specific information used by the KE component.

The KE hypothesizes what relationship appears to be described in a portion of the text. It builds a frame-like structure called a *template*, where various slots of the template represent various pieces of information which, according to the schema, are part of the description of the hypothesized relationship. Some of the slots are mandatory, i.e. they must be filled before the hypothesized relationship can be affirmed. Other slots are optional. If the KE is able to fill all of the mandatory slots with the appropriate information, then it creates an instance of the hypothesized relationship and stores this instance in the knowledge base. More detailed information about the process by which the KE component recognizes relevant objects and their relationships has been reported elsewhere (Boggess *et al*., 1995; Cordova *et al*., 1992, 1992a; Hodges *et al*., 1993).

4 Goal of the AIMS project

The goal of the current research effort is to develop an automated system that, by processing articles in a scientific domain, can assist in the rapid determination of the key topics found in those articles and the production of appropriate phrases to describe the contents of the articles. To assist in training and evaluation, a set of computer files consisting of 2000 articles from the *Journal of Physical Chemistry* (about 65 million words of text) and the associated indexes produced by trained document analysts was provided. Unlike most commercially available information retrieval systems, AIMS does not index only on words, but does phrase-based indexing. Currently, the AIMS system processes the articles and generates indexes for them without any human intervention or interaction. Thus none of the indexes in the index set produced by AIMS have been manually generated. Our intent is that the

https://doi.org/10.1017/S1351324996001325 Published online by Cambridge University Press

production version of AIMS will be able to generate at least 80% of the correct indexes for a given article, with a document analyst providing the rest.

The process of assigning key topics and descriptive phrases to the articles is referred to as *concept indexing*. Currently, the indexing is done by hundreds of human document analysts. For example, CAS employs about 400 document analysts to index technical articles appearing in the chemistry literature. Each year document analysts at Chemical Abstracts Service index 600,000 articles from journals, books and conference proceedings in all areas of chemistry in 50 languages. The indexes are both published and added to STN International, a network of approximately 175 scientific and technical databases.

A document analyst primarily looks for new and novel information in journal articles. The analyst maps this new information into CAS's controlled vocabulary along with context information that shows how it fits into known chemical knowledge. Indexing is divided into two major tasks, subject matter indexing (concept indexing) and chemical substance indexing. The AIMS project deals with concept indexing.

A *concept index* is hierarchical and contains two to seven levels. Depending on the major heading, called the *concept heading*, some of the levels are required, some are optional, and some are not allowed. The top six levels are selected from controlled vocabulary phrases. These are approximately 10,000 first level phrases, such as 'Laser induced flash photolysis', 'Photochemical electron exchange' and 'Kinetics'. The number of controlled phrases in levels two through six varies from a handful to more than 100,000. At the bottom is an uncontrolled phrase that gives the context of the index entry.

For example, from the sentence

> The triplet (T_1) state of acridine-9-carboxylate ($AcrCOO^-$) is shown by laser flash photolysis to be an efficient mediator of electron transfer from both a water-soluble analogue of vitamin E (Trolox) and ascorbic acid to methyl viologen (MV^{2+}) in aqueous 0·1 M $NaHCO_3$ solution (pH 8·3) at 22 °C.

two concept indexes are generated:

1. Photolysis,
   flash, laser-induced,
   triplet state of acridine carboxylate as efficient mediator of electron transfer from Trolox and ascorbic acid to Me viologen.
2. Electron exchange,
   photochemical,
   from water-soluble analog of vitamin E and ascorbic acid to methyl viologen in aqueous solutions, mediated by triplet excited acridine carboxylate.

These levels correspond to the slots of the templates maintained by the KE component of the KUDZU system. Similar templates are used in the IG component of the AIMS system. In describing this component, we shall use the term 'slot' rather than 'level'.

https://doi.org/10.1017/S1351324996001325 Published online by Cambridge University Press

5 The AIMS NLP component

AIMS consists of two major components: the natural language processing (NLP) component and the index generation (IG) component. An overview of the architecture of AIMS is given in figure 1. The NLP component processes the machine-readable English text of an article from a physical chemistry journal. The parsed text produced by the NLP component is then processed by the IG component, which generates a set of indexes for the article. Currently there is no human intervention in the generation of the indexes. In the production version of AIMS, however, we will provide a graphical user interface through which a document analyst may review the set of indexes generated by AIMS, adding and deleting indexes as needed.

The NLP portion of the project can be thought of as having three major objectives: (a) the identification of 'lexical units', (b) the identification and labeling of significant, meaningful terms in the text, and (c) the identification of 'phrasal units'.

The 'lexical units' which need to be identified are word collocations such as 'atmospheric pressure'. The 'phrasal units' which need to be identified include phrases such as prepositional phrases and noun phrases.

Consider the former: in any domain there are multiple-word sequences which appear to be treated as a single entity in the mind of the writer and the intended audience. 'Atmospheric', for example, is an adjective which often has to do with weather (e.g. 'atmospheric conditions') and 'pressure' is a property of gasses and liquids. However, 'atmospheric pressure' is generally a noun of measure, in the same category with liters and moles.

The identification of phrasal units is important, in part, so that the heads of these phrases may be examined. If *dissociation* is repeatedly mentioned as the main noun of phrases that occur in subject or object position of sentences of an article, this is supporting evidence for a hypothesis that *dissociation* is worthy of being indexed to the article. On the other hand, if it is repeatedly mentioned in modifying phrases such as '… was maintained to prevent the dissociation of ionic clusters,' then the fact that the substance occurs only in modifying positions in the text suggests that it is not central to the concepts being indexed.

The other major activity of the NLP portion of the AIMS project is the attaching of labels to significant terms in the text. Words such as 'actinometry', 'calibration' and 'spectroscopy' are labelled not only with their syntactic category, noun, but also with a semantic label, 'method', to identify them as activities associated with experiments. 'Kinetic energy' and 'viscosity' are labelled as nouns of property – that is, nouns that refer to properties of substances.

The sections that follow will describe these three kinds of activities: (a) identification of significant lexical units, (b) determination of reasonable semantic classes for the domain and the process of labeling a word of the text with appropriate semantic class, and (c) identification of significant phrasal units.

https://doi.org/10.1017/S1351324996001325 Published online by Cambridge University Press

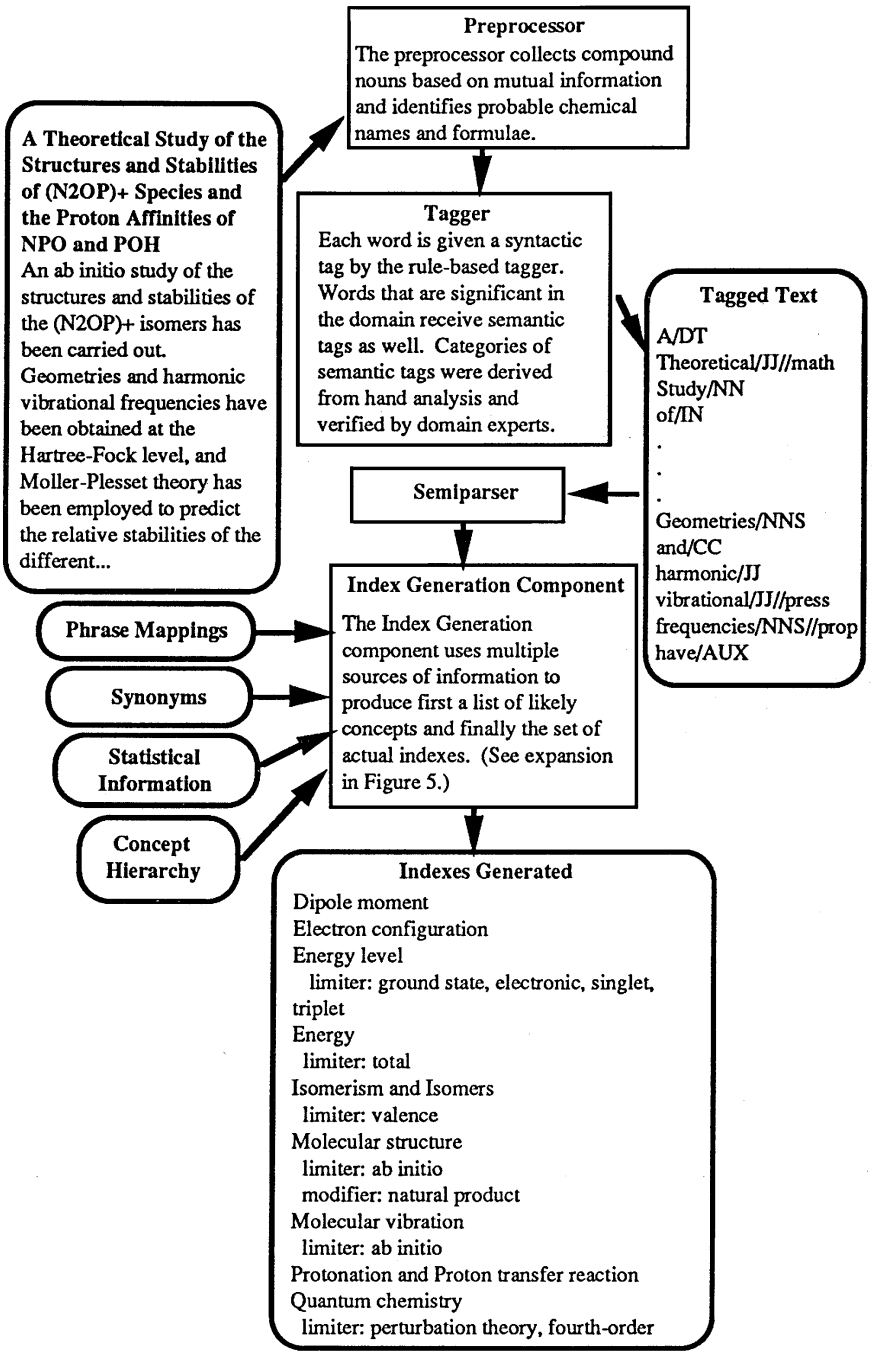


Fig. 1. Overview of AIMS architecture.

https://doi.org/10.1017/S1351324996001325 Published online by Cambridge University Press

24.420 Supplementary Material	9.996 initial condition
23.250 poly(acrylic acid)	9.988 heavy water
21.420 irreducible representations	9.986 salt concentration
20.691 deterministic chaos	9.985 relative importance
20.633 nickel silicide	9.985 equal amounts
20.453 digital oscilloscope	9.982 thermal equilibrium
20.371 conductometric titration	9.975 additional evidence
19.787 circular dichroism	9.970 structural changes
19.648 technical assistance	9.969 free base
19.618 refractive index	9.950 vibrational levels
19.397 Basic Energy	9.949 linear response
19.328 fractal dimension	9.949 double minimum
19.162 systematic sample-size	9.947 micellar core
19.150 steric hindrance	9.937 little effect
19.149 isothermal compressibility	9.933 IR spectra
19.002 p-doped silicon	9.924 different types
18.654 helpful discussions	9.921 quantitative agreement
18.632 refractive indices	9.919 dynamic processes
18.166 benzil ketyl	9.916 isotopic shift
18.096 cyclic voltammetry	9.914 electronic states
18.026 hot spot	9.908 critical points
17.986 personal computer	9.907 acidic solutions
17.946 supercritical fluids	
17.884 Experimental Section	
17.875 ultrahigh vacuum	
17.783 Molecular Dynamics	
17.765 supersonic jet	
17.511 Research Council	
17.501 angular momenta	
17.458 molar absorptivity	
17.428 natural abundance	
a) Top thirty adjective/noun collocations based on mutual information	b) All twenty-two adj/noun collocations with mutual informations from 9.9 to 10.0

Fig. 2. Some adjective/noun collocations and their mutual information statistics.

5.1 Word collocations

In any domain, there are numerous phrases which appear to have component parts and yet are treated almost always as units, and whose meaning may in fact differ from the literal interpretation of the component parts. Identification of these 'multiwords' is useful not only because the semantic label of the entire unit is often different from the labels of the component parts, but also because these compounds are usually significant concepts in the domain.

If a large body (tens of millions of words) of syntactically labelled text from the domain is available, then a very simple approach will unveil a large number of such multiwords. One may collect all adjective/noun pairs (such as 'kinetic energy') and noun/noun pairs (such as 'bond energy') that occur more than some threshold limit (say, 100 times). Not all such pairs are true multiwords, but eliminating those that are not is fairly straightforward.

However, this simple technique misses a large number of multiwords. If the threshold is lowered, many more multiwords surface, but the proportion of word pairs that are not unit entities rises. Sifting through all adjective/noun and noun/noun pairs that occur at least 25 times in a large corpus of text has a much

https://doi.org/10.1017/S1351324996001325 Published online by Cambridge University Press

worse effort-to-benefit ratio than scanning pairs that occur at least 100 times. Far too many of the pairs are unwanted.

Instead, the AIMS project collects the pairs that occur with frequency at or above some threshold and then computes the mutual information of that pair. The pairs are then presented to a human in order of decreasing mutual information, not decreasing frequency of occurrence.

The value computed for a pair is not its mutual information in the strict definition of mutual information. If words a and b occur in a body of text of k words with frequency $f(a)$ and $f(b)$, we estimate the probability that word a will occur, $P(a)$, as $P(a) = f(a)/k$. Similarly, we estimate the probability that word b will occur as $P(b) = f(b)/k$. If the occurrence of each word is independent of the other, we would expect the probability of their occurring together to be

$$P(a)P(b) = \frac{f(a)f(b)}{k^2}.$$

The mutual information of a and b is computed as the logarithm of the ratio of their probability of occurrence as a pair to the product of their independent probabilities,

$$\log_2\left(\frac{P(a,b)}{P(a)P(b)}\right) \approx \log_2\left(\frac{f(a,b)/k}{f(a)f(b)/k^2}\right) = \log_2\left(k\frac{f(a,b)}{f(a)f(b)}\right).$$

A strict mutual information value would be computed irrespective of the order in which a and b occurred. However, in our case order matters, so that the value we calculate is for the sequence ab, and would certainly differ from the value calculated for the sequence ba.

Example

Suppose that in a text of 1000 words the word *mutual* occurred four times, *information* occurred eight times, and *mutual information* occurred three times. Then for the two-word sequence the formula gives a measure of $\log_2[1000*3/(4*8)] = \log_2[93{\cdot}75] = 6.55$.

If the ratio of the probabilities is greater than one, then the sequence occurs more often than random chance would lead one to expect. If the ratio is less than one, then the occurrence of one of the words tends to predict that the other will *not* occur. Positive values of the logarithm indicate ratios larger than one. It has been suggested that mutual information values of 3 or greater are 'interesting' (Church and Hanks, 1989). Figure 2(a) and figure 2(b) show word collocation pairs with mutual information figures for adjective/noun pairs that occurred at least 25 times in a body of 8·8 million words (as large a fraction of our 65 million words of available training text as could be handled at one time by the equipment). Figure 3(a) shows noun/noun pairs with mutual information values 18·0 to 19·0 and figure 3(b) shows present-participle/noun pairs with mutual information values in the range from 10·0 to 15·0. Several of the latter are three-word collocations, produced by having first found a pair with high mutual information which was labeled as a noun and then in a second pass having discovered that the just-defined multiword had high mutual information with another word.

https://doi.org/10.1017/S1351324996001325 Published online by Cambridge University Press

18.948 friction kernels 18.921 confidence intervals 18.882 square root 18.837 DAF class 18.774 infinite dilution 18.658 Laplace transform 18.644 diode array 18.604 TPA aluminosilicate 18.511 random walk 18.432 miscibility gap 18.425 DODAC vesicles 18.350 UHV chamber 18.316 CdS colloids 18.198 Energy Sciences 18.192 refractory period 18.081 naphthoate anions 18.074 Fourier transform 18.003 selection rules	14.945 working electrode 14.350 branching ratio 13.528 preceding paper 13.527 exciting light 13.465 stretching vibration 13.120 preceding section 13.038 quenching rate constant 13.035 bending moment 12.745 fitting procedure 12.588 quenching rate constants 12.475 ordering information 12.420 stretching vibrations 12.210 bending vibration 12.168 bending mode 12.064 stretching mode 11.752 stretching modes 11.428 limiting cases 11.293 emitting state 11.254 stretching frequencies 11.228 scattering length 11.063 scattering chamber 11.024 bending vibrations 10.817 bending modes 10.529 bending motion 10.413 scattering cross section 10.360 stretching frequency 10.207 stirring effects 10.175 scanning electron 10.115 quenching cross section 10.025 bending wave
a) Eighteen consecutive noun/noun collocations with mutual informations from 18.0 to 19.0	b) All present-participle/noun collocations with mutual informations from 10.0 to 15.0

Fig. 3. Some noun/noun and present participle/noun collocations and their mutual information statistics.

The most important reason for using the mutual information statistic is that it allows the collection of pairs of words which occur relatively infrequently in a large corpus, without overburdening the person who must screen them for inclusion in the system at large. But because the AIMS project then forces the compound word to be treated as a single word, it is also the case that some word sequences of length greater than two are identified. We ran the mutual information program in two passes before screening the results. For example, the sequence 'electron transfer' was identified in a first pass as having high mutual information. Consequently, it was marked as a single word, and on the second pass the sequence 'intramolecular electron transfer' was found to have high mutual information. In figure 3(b) the phrase 'scanning electron' may look anomalous, but it is in fact the first two words of 'scanning electron microscope' and allows the system to find the longer phrase on a next pass.

5.2 Semantic classes

The KUDZU system makes use of semantic as well as syntactic labels on text. One of the tasks of moving to a new domain, then, was determining the relevant semantic classes of the new domain. An important source of information at this stage was a

https://doi.org/10.1017/S1351324996001325 Published online by Cambridge University Press

large collection of journal articles and associated indexes which had been determined by document analysts and domain experts. Some of the noun classes derived from inspection of the indexes included nouns naming substances, nouns of method (having to do with conducting experiments), nouns of process (having to do with changes undergone by substances), and nouns of property (of substances), among others.

We next developed a training set of 48,000 words with the proper syntactic and semantic labels, on which to train a tagging program. To create this training set, we ran a syntactic tagger on the text which was to become our training set. We then took the several hundred words which we knew to belong to the classes identified from inspection of articles and indexes, mentioned above, and blindly attached the corresponding semantic labels to the syntactic ones for those words. We then laboriously hand-corrected the resulting text, looking all the while for words which seemed to fulfill the same functions as the ones which already had semantic classification, and for significant roles in the text that might suggest classes not yet identified.

In the process, we adopted the Penn Treebank syntactic labels (Marcus *et al.*, 1993), plus a few of our own – subordinate conjunctions and auxiliary verbs, for example. We also switched to the Brill tagging program (Brill, 1992). Of course, the tagger had to be trained on something that already had Penn Treebank syntactic labels before it could label our training set. And that available text was not from the domain of physical chemistry. Between the change in domain and the lack of part of the target syntactic label set, the resulting tagged text had a relatively high error rate. As mentioned, however, in the process of correcting the syntactic labels, we also proposed new classes of words (verb of study, for example, or mathematical nouns) to identify important themes in sections of text ('we examine …' 'we show that …', or extended discussion of elements of equations or graphs). Many of the proposed new classes were speculative. In a later stage, all proposed classes were reexamined, and some were merged or eliminated.

Once the training set was corrected and appropriately labelled, the Brill tagger was retrained on it for general use in labelling future text to be indexed by the AIMS system.

A fair number of words belong to more than one semantic class. Counting syntactic variations, it is possible for a few words to have as many as five possible labellings. During training, the Brill tagger inspects the errors that it makes on a text sample and infers rules that help it to avoid further errors of the same kind. However, such rules may be based on a single instance in the training set; consequently it was possible (even probable) that a careless error left in the training data could propagate to all text labelled thereafter. Labelling the training text demanded *great* care. On the other hand, the rules inferred by the tagger can be examined and modified, so it is possible to interdict 'bad' rules if they are sufficiently obvious.

Although the Brill tagger is normally used to label with syntactic tags only, relatively little effort is required to adapt it to the task of applying the combination syntactic/semantic tags used by the AIMS project, and it is successful in labelling new text with the combination tags. The tagger uses a lexicon, and many of the words in

https://doi.org/10.1017/S1351324996001325 Published online by Cambridge University Press

our lexicon have a single applicable label. Furthermore, a preprocessor scans the text for chemical formulae and substance names, so that many 'new' words which are nothing more than previously unencountered substances are properly labelled before the tagger comes into play. The Brill tagger can ignore already-labelled words.

The rule-based tagger is a mixed blessing when it comes to recognising the relationship between suffixes and appropriate tags. In moving from our previous stochastic tagger to the rule-based one, we gave up the processor that deduced root forms by using common morphological rules related to suffixes. However, the Brill tagger readily deduces such rules as the following, drawn from the versions of the tagger trained on our data: 'If a word has the suffix *-ly*, it is highly likely to be an adverb.' 'If adding the suffix *-ly* to the current word produces a known word, then the current word is highly likely to be an adjective.' With somewhat less confidence, it associates the endings *-ic*, *-ble* and *--ar* with adjectives, in our data. Of significance to AIMS, with its semantic tags, is that the Brill tagger also associates the endings *-ide* and *-ne* with the tag NN//subst, and the ending *-vity* with the tag NN//prop (i.e. noun relating to a property of a substance), among many examples that go well beyond the morphological processing of our previous system. Ironically, one of the first rules that the tagger learns is 'if removing an *s* from the end of an unknown current word produces a known word, with extremely high confidence the current word is a noun.' This rule is so frequently successful that it statistically drowns an important class of errors – unknown verbs ending in *s* because they are in the third person singular form.

The substance recognizer is a preprocessor that runs before the Brill tagger. It can tag chemical substance names, chemical formulas, acronyms, and numbers. It uses word lists and pattern matching. It sequentially looks up each word from an article in four hash tables built from four word lists containing non-technical English words, common chemical substance names, substance name parts, and chemical formulas.

If the word is found to be non-technical English, it is ignored. If it is found to be a common chemical substance name, it is tagged as a substance. If it is not found in either of the first two lists, it is broken at non-alphabetics and each part, starting with the longest, is looked up in the substance name parts list.

For example,

p-Aminobenzeneazodimethylaniline

breaks into 'p' and 'Aminobenzeneazodimethylaniline'. 'Aminobenzeneazodimethylaniline' is in the substance name parts list, so p-Aminobenzeneazodimethylaniline is tagged as a substance.

If a word gets no hits, it is probably a chemical formula. It is passed to a pattern matcher that looks for subscripts and superscripts. If it finds any, the word is tagged as a chemical formula. The recognizer also uses pattern matching to tag numbers and acronyms.

The AIMS project has available to it files of words and phrases collected by document analysts familiar with the indexing task. These experts have found the provided words and phrases to be significant indicators of the presence of indexable concepts. We transformed the phrases into multiwords, and all of the resulting words

https://doi.org/10.1017/S1351324996001325 Published online by Cambridge University Press

Property	Change		Process
density	change	minimum	relaxation
energy gap	loss	motion	reaction
dipole moment	increase	variation	energy transfer
polarization	shift	singlet	substitution
energy	transition	overtone	exchange
line width	effect	splitting	transition
rate constant	transformation	progression	emission
response	deformation	vibration	torsion
separation	enhancement	volume	case
	distortion	expansion	
		mode	

Fig. 4. Three classes proposed by the semi-automated classifier when given the 400 most frequent nouns from a body of 1·5 million words of text in the domain of physical chemistry.

were given semantic labels and merged with the existing lexicon. The resulting lexicon was many times richer in significant vocabulary than would normally be the case from a 48,000 word training set.

Because of the effort required to develop and use the semantic classes, investigations were made into extracting the semantic classes semi-automatically. This work (Agarwal, 1995) clusters nouns and verbs by looking at partially parsed sentences and extracting the subject-verb-object triples in clauses. A noun, for example, might be clustered with other nouns that have the same most frequent verbs and objects as it does when it occurs in subject position, as well as the same most likely verbs and subjects that it does when it occurs in object position. Clusters of three to ten nouns are presented to a human, who can discard a presented cluster as useless or keep it. If the cluster is kept, the human names the class to which the cluster belongs, and indicates whether the cluster is 'strong' (all nouns presented actually belong to the named class) or 'weak' (one or more of the nouns do not belong to the named concept. Ordinarily, a class will be represented in numerous clusters; in the final stage, all clusters given the same name are merged into a single class. This classifier has been applied only to the most frequent nouns and verbs of the domain at this time. It does suggest a couple of classes from the corpus that were not identified by the human research team that developed the initial semantic classes. Some of the classes that it discovered from text with syntactic tags alone are found in figure 4.

5.3 Phrasal units

Once development was complete, the tagging program was trained on the 48,000 word training text. Thereafter, to process a text, the following steps are followed:

- A preprocessor is run to identify formulae and substance names. Any that are found are labelled as nouns, with semantic labels indicating substance or formula.
- The tagger is run on the resulting text. The lexicon used by the tagger includes the vocabulary encountered in the 48,000 word training set, multiwords extracted from an 8·8 million word body of text, and a very large list of words and phrases provided by the domain experts.

https://doi.org/10.1017/S1351324996001325 Published online by Cambridge University Press

- The labelled text is given to a semi-parser which does little more than identify noun phrases, prepositional phrases, infinitive phrases and gerund phrases. No conventional grammar is used, and most of the power of the semi-parser lies in its noun phrase recogniser, which is unabashedly *ad hoc*. On the other hand, it is dependent only on syntactic part of speech tags, and as such is domain independent, and it has served us well in two domains.
- A prepositional phrase handler attaches prepositional phrases to the sentential elements that those phrases modify. The current version of the prepositional phrase attachment specialist pays attention to syntax only, having no domain-specific rules.

The result is a structure which is relatively flat. Almost all structural depth has to do with nested prepositional phrases. This structure is passed to the IG component.

6 The AIMS IG component

The general approach used by the IG component has been adapted from the KE component of the KUDZU project (Hodges *et al*., 1993). In this section, we provide an overview of how the IG component works, as illustrated in figure 5. This has been reported in greater detail elsewhere (Hodges *et al*., 1996).

The IG component currently makes use of only the title, the abstract, the conclusion and the introduction for an article, in conjunction with specific domain information supplied by CAS. Since the indexes produced by the document analysts at CAS must be uniform, CAS has created a number of files containing information to help the various analysts adhere to the accepted terminology of the physical chemistry domain. CAS made these files of domain-specific information available to the members of the research group working at Mississippi State University (MSU). The files were not necessarily organised in a way that facilitated their use in an automated system, since they had been used only as resource material by the document analysts at CAS. The MSU researchers modified these files so that they were structured in a way that they could be efficiently accessed by AIMS.

One of the files provided by CAS contains a *concept hierarchy*. Each concept in the hierarchy has associated with it the information that must be present before a document analyst would create an index for that concept. For example, there is a file that contains *phrase mappings*, i.e. phrases whose presence in an article suggests the possibility of a particular concept being referenced. Since a phrase may map to more than one concept, and a concept may have more than one associated phrase, other information that can disambiguate the concept references must be used. CAS has provided a variety of disambiguating information such as synonyms for the concepts and modifying information that may indicate a more specific concept in the hierarchy.

The index generator processes the tagged article supplied by the NLP component. If it encounters one of the CAS phrases, and if this phrase has an appropriate semantic tag (i.e. 'substance', 'property', 'instrument', 'process', or 'method'), then the IG chooses the most likely of the concepts to which this phrase can map. This choice is based upon statistics produced by a modified version of a statistical program supplied by CAS. These statistics include the number of 'good hits' and 'false hits'

https://doi.org/10.1017/S1351324996001325 Published online by Cambridge University Press

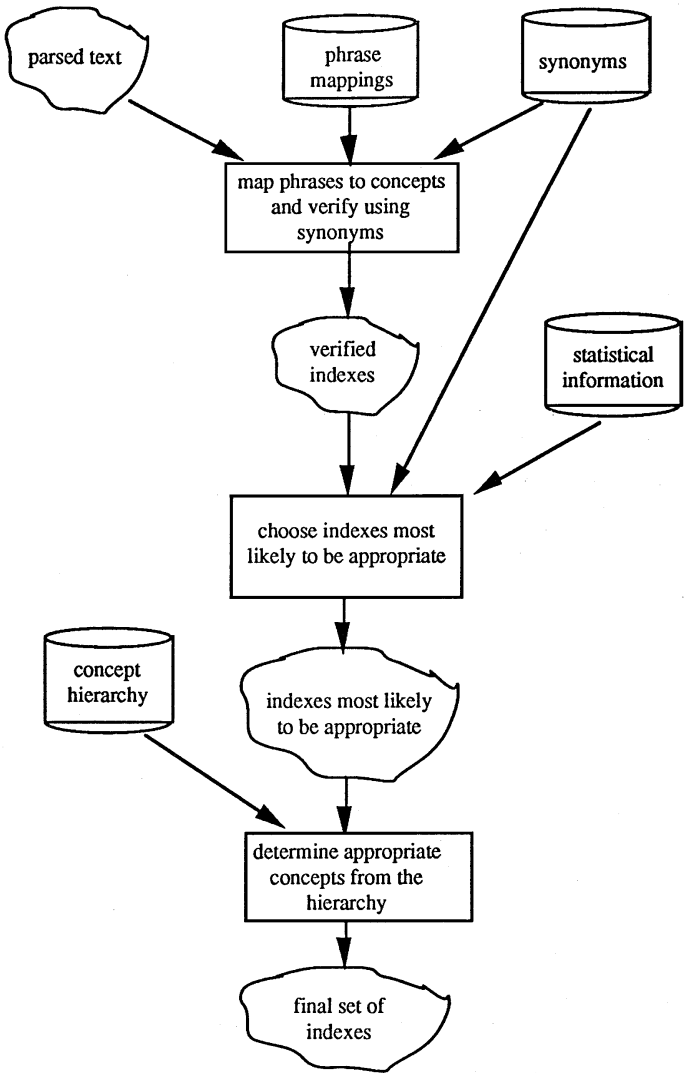


Fig. 5. Overview of the IG component.

for the phrases computed over a collection of 2000 physical chemistry articles. A 'good hit' indicates that the presence of the concept was correctly indicated by a phrase; a 'false hit' indicates that the presence of the concept was incorrectly indicated. The modified statistical program computes a weighted sum of the good hits and false hits for each concept in each of the major portions of the article: the title, the abstract, the body (including the introduction) and the figures. The final weight assigned to a particular concept is the ratio of the number of good hits to the sum of the number of good hits and false hits. So the final weight is the proportion of the total weight assigned to a concept that is the result of good hits recognized by the statistical program.

Currently, the IG keeps an ordered list (ordered by weight) of all the concepts that

https://doi.org/10.1017/S1351324996001325 Published online by Cambridge University Press

have been generated because of the presence of one of the CAS phrases. It then attempts to verify each concept by using the synonyms file. The phrase mappings file lists each concept that can be generated by a phrase. The synonyms file provides more specific information so that the generation of certain irrelevant concepts can be avoided. For example, the phrase 'decomposition' maps to each of these concepts: decomposition, dissociation, photolysis, proteins, radiolysis. However, the synonyms file indicates that 'photolysis' may not be generated from the phrase 'decomposition' unless there is also at least one of the following sub-modifiers present: laser-induced, light-induced, photochemistry, photo-, laser. So unless one of these sub-modifiers is present in the article, 'photolysis' will be eliminated as a possible index. During the verification process, the IG makes use of the structure of the parsed text produced by the NLP. For example, the modifiers for a noun which may be an indexable concept frequently appear in an embedded prepositional phrase.

Because of the variety of ways in which a concept may be expressed in an article, the IG makes extensive use of information provided by CAS about synonyms, acronyms and abbreviations.

The IG also pays attention to the structure of the concept hierarchy. If three or more sibling concepts in the hierarchy are chosen, then the IG recommends the parent concept as an index. Therefore, the recommended indexes remain at a specific level in the hierarchy unless there is a strong indication that the appropriate index is the more general concept.

When the document analysts at CAS produce an index for an article, they produce not only the concept name, but what is referred to as the *context*. Thus the complete structure of a CAS concept index entry is:

CTH: the concept heading (exactly 1)
HOM: homograph (0 or 1)
MDF: modifier (0 or 1)
QLF: qualifier (0 or 1)
CAT: category (0 or 1)
LTR: limiter (0 or 1; usually 1)
free text: uncontrolled vocabulary (1)

For example:

CTH: Abstraction reaction
HOM: nil
MDF: nil
QLF: nil
CAT: nil
LTR: intramolecular, photochemical
free text: of hydrogen, in polymethylene linked xanthone and xanthene, mechanism of

CAS provided a list of the 208 most frequently-occurring concept headings in physical chemistry. Some examples are: electric potential, energy level transition, fluorescence and kinetics.

https://doi.org/10.1017/S1351324996001325 Published online by Cambridge University Press

A homograph distinguishes between concept headings which are the same but mean different things in different contexts. For example:

CTH:	molds	CTH:	molds
HOM:	fungus	HOM:	forms

Homographs rarely appear and so have been ignored by the index generator.

Included in the file of synonyms for concepts is a list of the allowed modifiers and limiters for each concept. Examples of modifiers are: gas, column and liquid, thin-layer, mechanical, electric, flash, laser-induced.

Qualifiers refer to what the article says happened to a substance class; categories refer to the types of substance classes. Qualifiers must be inferred from what the author of the article did to the substance class. For example:

analysis (The substance class was detected, determined, or analysed.)
biological studies (The substance class was studied in connection with a living thing, such as a study of nutrition.)
preparation (There was some synthesis, distillation, purification, or separation of a chemical in one of the substance classes.)
properties (The article reported on the determination of chemical or physical properties such as the determination of an atomic structure or crystal structure.)

We have not yet addressed the production of qualifiers for the concepts. We anticipate inferring this information from the presence of certain verbs or nouns that can indicate specific processes.

The free text is uncontrolled vocabulary that gives the context of the index entry. We intend to investigate the use of the free text in order to limit the context of an article and thus reduce the number of incorrect indexes being generated. We anticipate that the use of free text for this purpose will be limited, however, because of its uncontrolled, rather vague nature.

A graphical user interface has been developed for AIMS. The interface allows the user to display the indexes generated for a particular article, to highlight the phrases within the article that caused the indexes to be generated, and to modify any of the index information (such as adding an additional limiter in the LTR slot). In addition, the user may request that the interface display all of the candidate entries that were considered for the various slots of an index. Cut and paste operations are allowed so that the user can select text from the article and paste it into a slot. Detailed information about the graphical user interface may be found in Kulkarni (1995).

7 Results

To evaluate how successful our AIMS system is in the generation of indexes for physical chemistry articles, we generate the indexes for articles for which we already have the indexes that were generated by the human document analysts. The indexes generated by the document analysts are considered to be the 'correct' indexes. We realise that the AIMS system is being penalised by using the document-analyst-

https://doi.org/10.1017/S1351324996001325 Published online by Cambridge University Press

generated indexes as a standard because frequently the indexes generated by AIMS are quite reasonable indexes even though they were not generated by a particular document analyst. It has long been recognised that human indexing is 'plagued by incompleteness and inconsistency' (Evans *et al.*, 1991). It is possible that we may eventually devise some other method of comparison, such as having indexes for a given article generated by more than one document analyst, then performing comparisons among the sets of indexes generated by the document analysts as well as those generated by AIMS. Or we could have AIMS generate indexes for a collection of documents used in the Text REtrieval Conference (TREC) evaluations and compare the results to those of other systems that have participated in the TREC evaluations. Another possibility is to have a document analyst review the indexes generated by AIMS, then record the number of indexes that the document analyst adds to and deletes from the set generated by AIMS. However, at this stage of our project, and given that the indexes generated by AIMS must follow the same CAS guidelines that the document analysts use, we find it more appropriate to use the set of indexes generated by a single document analyst for a particular article as the article's correct set of indexes.

7.1 Recall and precision

We compute the values of performance evaluation metrics commonly used in the evaluation of text understanding and message understanding systems – *recall* and *precision* (Lehnert and Sundheim, 1991). Recall is the proportion of the correct indexes which are generated. Precision is the proportion of the generated indexes which are correct.

recall = number of correct indexes generated/total number of correct indexes
precision = number of correct indexes generated/total number of indexes generated

In recent tests, AIMS generated indexes for 116 of the articles previously indexed by CAS (Hodges *et al.*, 1996). The articles ranged in length from 19 words (a correction to a previously published article) to 9112 words, with an average length of 3401 words. Using the 116 articles, we defined five test cases in which we varied the parts of the articles that AIMS used during index generation: (1) only the abstracts of the articles, (2) only the abstracts and introductions, (3) only the abstracts and parts of the introductions, (4) only the abstracts and conclusions, and (5) only the abstracts, parts of the introductions, and the conclusions. For test cases 3 and 5, where only part of the introduction section of an article was used, we manually filtered out 'incidental references' – references to other publications or projects which appeared in the article (e.g. as part of the article's literature review), but which could result in the generation of indexes that were not directly related to this article. The results of these experiments are summarised in Table 1 (from Hodges *et al.*, 1996). The precision rates for the test cases ranged from an average of 61·71 per cent (test case number 2) to an average of 66·45 % (test case number 1). The recall rates ranged

https://doi.org/10.1017/S1351324996001325 Published online by Cambridge University Press

Table 1. *Experimental results.*

Experiment	Precision (%)	Recall (%)	Overgeneration (%)
1	66·45	71·79	33·55
2	61·71	85·50	38·29
3	64·42	84·76	35·58
4	63·37	87·36	36·63
5	62·17	90·08	37·83

from an average of 71·79 % (for test case 1) to an average of 90·08 % (for test case 5). AIMS enjoyed its best precision rates when we have filtered out the incidental references.

The low precision rates reflect a decision that was made early in the development of AIMS. Specifically, we preferred to have the system generate more possible indexes for an article than would be generated by a document analyst rather than to have the system fail to generate all of the desired indexes. In many cases, increasing the precision will require the AIMS system to have the ability to recognise that the presence of certain key phrases in an article may not constitute more than a passing mention of a concept. Consider, for example, the phrase 'spectroscopy'. This is one of the key phrases identified by the CAS document analysts as being an indicator that the concept being discussed in the article may be 'mass spectroscopy'. Yet its presence alone is not sufficient for a document analyst to choose 'mass spectroscopy' as an index. The current AIMS system, however, will suggest 'mass spectroscopy' as an index if 'spectroscopy' appears in an article.

Also, there are many cases where the 'extra' indexes generated by AIMS are reasonable indexes for the article even though they were not suggested by a human document analyst. The analyst attempts to keep the number of indexes to a minimum to keep the printed product as small as possible. The current shift at CAS from printed indexes to on-line indexes is likely to remove this restriction so that reasonable indexes will not be counted as being incorrect.

7.2 Discussion of the results

In our analysis of the indexes generated by AIMS, we found that AIMS's failure to generate some of the indexes that were generated by a document analyst is a result of characteristics of the synonyms file. The synonyms have a relatively restricted format. They tend to be very long phrases, and those phrases generally appear with only one specific ordering of the words in the phrase. Even though the appropriate key words may appear in an article, the AIMS IG component will not recognise them because of the word order.

Another characteristic of the synonyms file is that the synonyms are very general phrases that may map to many different concepts. Although the AIMS IG component makes use of the semantic tags supplied by the NLP component, many times the different concepts to which a phrase maps belong to the same semantic

https://doi.org/10.1017/S1351324996001325 Published online by Cambridge University Press

category. (This is not surprising given that all of the articles are from the same domain.) The result is that a concept for which AIMS should have generated an index may be dropped during the index generation process simply because its statistical rating is lower than some other concept.

Our analysis of the indexes generated by AIMS indicates that polysemy is the major reason for generating indexes which should not have been generated. Consider, for example, that a given phrase in an article maps to two different concepts, only one of which should be indexed. In most such cases, the semantic tags of the two concepts are the same, and their statistical ratings are also very close to the same. If the difference in their ratings falls within a given threshold, then AIMS generates both concepts. This is a trade-off that we currently make between recall and precision. That is, we prefer to tolerate a lower precision rate rather than to have a lower recall rate (especially since we are aware that some of our 'incorrect' indexes are, in fact, quite reasonable).

Another cause of the generation of incorrect indexes is the incidental references that may appear in an article. There may be phrases that appear frequently throughout an article, yet will be ignored by a document analyst because he/she does not think it represents a significant aspect of the article. But AIMS will generate an index from such phrases because they are phrases that map to concepts, and the frequency of their occurrence in the article results in a relatively high statistical rating.

Some of the errors in the indexes produced by AIMS are due to tagging errors made by the NLP component. The most common error is for the tagger to mistag a verb, changing the entire structure of the sentence. This error has an impact on the process that verifies a potential index by comparing it to the entries in the synonyms file. Most of the synonyms consist of several words, so that the verification process must verify the 'head' noun as well as each of its modifiers. Frequently, the modifiers appear in prepositional phrases that modify a noun. While it is difficult to determine the precise influence of tagging errors on the performance of the system, we have estimated that approximately 10–12 % of the errors made by AIMS are due to tagging errors.

A similar problem that we have encountered is having the NLP component group into phrases individual words each of which appears in the phrase mappings file. In an earlier version of AIMS, we used word-based indexing rather than phrase-based indexing, and we found that the use of phrase-based indexing resulted in a dramatic increase in our precision rates. However, there are a few cases in which the IG component does not recognise the presence of certain phrases that appear in the phrase mappings file because they have been made a part of a larger phrase by the NLP component. This has a relatively small effect on the system's performance, however. We estimate that about 3–4 % of the index generation errors are due to the inability to recognise important phrases within phrases.

As we have described, AIMS does phrase-based rather than word-based indexing. Also, AIMS is not restricted to producing only indexes which are string-identical to phrases that appear in the text. CLARIT, which also does phrase-based indexing without being restricted to exact string matches, claims to be closer to 'concept-based' indexing than those systems that are limited to word-based indexing (Evans *et*

https://doi.org/10.1017/S1351324996001325 Published online by Cambridge University Press

al., 1991). Unlike CLARIT, however, AIMS is not restricted to producing indexes that consist of words or phrases that actually occur in the text. We have not restricted ourselves to using only a fairly simple thesaurus as CLARIT does. Our use of a richer knowledge base of domain-specific expert information has allowed us to develop a system that more closely approaches the goal of 'concept-based' indexing.

8 Future work

There are several steps that we can take which we expect to improve our precision rates. One of the first things that we plan to do is to have one of the CAS scientists examine the phrase mappings file in order to identify some phrase mappings that may be too general to be helpful in generating accurate indexes. A human document analyst simply 'knows' to ignore such phrases, but AIMS does not. Another approach that we intend to investigate in the immediate future is to try to recognise what we earlier referred to as 'incidental references' in an article. For example, an article may refer to some key concept only as a part of a literature review or a reference to work previously accomplished. Such a reference should not generate an index for that concept. Also, we expect to improve our precision rate once we begin to identify the contents of other slots in the concept index entries.

The IG component is not yet capable of appropriately filling every slot of an index entry. Although the particular information needed to generate a slot varies from slot to slot because of the different kinds of information represented by the slots, we anticipate that we will continue to use a modified version of the KUDZU KE component to fill the slots. We believe that as we modify AIMS so that it can ignore an 'incidental reference' to a concept and fill more of the slots of a concept index entry, we will be able to improve the precision of the system.

The KUDZU system relies heavily on the presence of mandatory slots (i.e. slots for which there must be a value) in hypothesizing about the relationship being described in some portion of text. The lack of mandatory slots for the concept index entries in the AIMS system has made the automated recognition of incorrect indexes more difficult. The only slots that must be filled are the concept heading (CTH) slot and the free text slot. The current version of AIMS does not fill the category, qualifier and free text slots. We intend to investigate the possibility of inferring this information from the presence of certain verbs or nouns that can indicate specific processes.

9 Summary

We have described an automated system that aids human document analysts in the generation of indexes for articles in the physical chemistry domain. We have provided brief descriptions of other research projects whose goals and/or techniques are similar to ours. In our overview of the AIMS system, we have summarised the basic approach used in the development of each of the main components – the NLP component and the IG component. We have also reported the results of the success rates of the current version of AIMS. Finally, we have suggested future modifications or enhancements to the AIMS system to improve its precision.

https://doi.org/10.1017/S1351324996001325 Published online by Cambridge University Press

Acknowledgements

The authors are particularly indebted to the work of Lynellen Smith, Rajeev Agarwal and Sonal Kulkarni. Lynellen was instrumental in developing the semantic categories, not only from inspection of thousands of indexes from previously indexed articles but also from thoughtful analysis of the 48,000 word training data, most of which she hand-labelled. She also developed the collocation specialist. Rajeev oversaw almost all aspects of the tagger and parser and the many facets of their interaction with the system as a whole. Sonal developed the graphical user interface. All three contributed numerous ideas at the weekly AIMS meetings, and they are part of the body and soul of AIMS. We are also indebted to the American Chemical Society, and to Chemical Abstracts Service in particular, for making available to us over 2000 indexed articles from the *Journal of Physical Chemistry*, lists of useful phrases, their concept hierarchy, and the time and considerable patience of their experts in answering our questions and providing feedback on various stages of the AIMS system.

This work is supported in part by grant number IRI-9314963 from the Interactive Systems Program of the National Science Foundation and the Advanced Research Projects Agency (NSF/ARPA Human Language Technology Initiative).

References

Agarwal, R. (1995) Semantic feature extraction from technical texts with limited human intervention. PhD dissertation. Mississippi State University.

Agarwal, R. and Boggess, L. (1992) A simple but useful approach to conjunct identification. *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, Newark, DE.

Association for Computational Linguistics (1996) *Breadth and Depth of Semantic Lexicons*. Association for Computational Linguistics.

Brill, E. (1992) A simple rule-based part of speech tagger. *Proceedings of the Third Conference on Applied Natural Language Processing*, Trento, Italy.

Boggess, L., Agarwal, R. and Davis, R. (1991) Disambiguation of prepositional phrases in automatically labelled technical text. *Proceedings of the Ninth National Conference on Artificial Intelligence*, Anaheim, CA.

Boggess, L. and Hodges, J. (1994) A knowledge-based approach to indexing scientific text. *Workshop Notes: ARPA workshop on human language technology*, Merrill Lynch Conference Center, Princeton, NJ.

Boggess, L., Hodges, J. and Cordova, J. (1995) Automated knowledge derivation: Domain-independent techniques for domain-restricted text sources. *International Journal of Intelligent Systems* (accepted for publication).

Boguraev, B. and Pustejovsky, J. (eds.) (1996) *Corpus Processing for Lexical Acquisition*. Cambridge, MA: MIT Press.

Church, K. and Hanks, P. (1989) Word association norms, mutual information, and lexicography. *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, Vancouver, B.C.

Cordova, J. L. and Hodges, J. E. (1992) The automated building of a knowledge base through natural language text analysis. *Southeast Cognitive Science Conference*, January 16–17, 1992, Atlanta, GA.

Cordova, J. L. and Hodges, J. E. (1992a) The automatic initialization of an object-oriented knowledge base. *Proceedings of the 30th Annual Southeast Conference of the ACM*, 401–404.

https://doi.org/10.1017/S1351324996001325 Published online by Cambridge University Press

Evans, D. A., Ginther-Webster, K., Hart, M., Lefferts, R. G. and Monarch, I. A. (1991) Automatic indexing using selective NLP and first-order thesauri. In A. Lichnerowicz (ed.), *Intelligent Text and Image Handling*. Amsterdam, NL: Elsevier.

Evans, D. A. and Zhai, C. (1996) Noun-phrase analysis in unrestricted text for information retrieval. *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics* (*ACL-96*), San Diego, CA.

Fraser, C. M. and Mays, A. (eds.) (1986) *The Merck Veterinary Manual*, *6th ed*. Rahway, NJ: Merck & Co.

Fuhr, N. and Buckley, C. (1991) A probabilistic learning approach for document indexing. *ACM Trans. Information Systems* **9**(3): 223–248.

Ginsberg, A. (1993) A unified approach to automatic indexing and information retrieval, *IEEE Expert* **8**(5): 46–56.

Hirschman, L. (1986) Discovering sublanguage structures. In R. Grishman and R. Kittredge (eds.), *Analyzing Language in Restricted Domains: Sublanguage description and processing*. Hillsdale, NJ: Lawrence Erlbaum.

Hodges, J. and Boggess, L. (1992) Extracting information from technical text. *Southeast Cognitive Science Conference*, January 16–17 1992, Atlanta, GA.

Hodges, J. E. and Cordova, J. L. (1993) Automatically building a knowledge base through natural language text analysis. *International Journal of Intelligent Systems* **8**(9): 921–938.

Hodges, J., Yie, S., Kulkarni, S. and Reighart, R. (1997) Generation and evaluation of indexes for chemistry articles. *Journal of Intelligent Information Systems* (to appear).

Kulkarni, S. (1995) Indexer: A tool to access index information from an object-oriented knowledge base. Master's project. Mississippi State University.

Kwok, K. L. (1986) An interpretation of index term weighting schemes based on document components. *Proceedings of the 1986 ACM Conference on Research and Development in Information Retrieval*, 275–283.

Kwok, K. L. and Grunfeld, L. (1994) Learning from relevant documents in large scale routing retrieval. *Workshop Notes: ARPA workshop on human language technology*, Merrill Lynch Conference Center, Princeton, NJ.

Lehnert, W. and Sundheim, B. (1991) A performance evaluation of text-analysis technologies. *AI Magazine* **12**(3): 81–94.

Lewis, David D. and Jones, K. S. (1996) Natural language processing for information retrieval. *Communications of the ACM* **39**(1): 92–101.

Marcus, M., Santorini, B. and Marcinkiewicz, M. (1993) Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics* **19**(2): 313–330.

Maron, M. E. and Kuhns, J. L. (1960) On relevance, probabilistic indexing, and information retrieval, *Journal of the ACM* **7**: 216–244.

Pustejovsky, J. (1991) The generative lexicon. *Computational Linguistics* **17**(4): 409–441.

Pustejovsky, J. (1995) *The Generative Lexicon*. Cambridge, MA: MIT Press.

Sager, N. (1987) Computer processing of narrative information. In *Medical Language Processing: Computer management of narrative data*. Reading, MA: Addison-Wesley.

Soderland, S. and Lehnert, W. (1994) Wrap-Up: A trainable discourse module for information extraction. *Journal of Artificial Intelligence Research* **2** (December): 131–158.

Strzalkowski, T., Carballo, J. P. and Marinescu, M. (1995) Natural language information retrieval: TREC-3 report. *Proceedings of the Third Text Retrieval Conference* (*TREC-3*), NIST Special Publication 500–225, 39–53.

https://doi.org/10.1017/S1351324996001325 Published online by Cambridge University Press