SUMMARISING COMPANY ANNOUNCEMENTS

Robert Dale,[1,3] *Li Lei,*[2] *Hugo De Vries,*[1] *Mary Gardiner,*[1] *and Marc Tilbrook*[1]

[1]Centre for Language Technology, Macquarie University, Australia
[2]Center for Intelligence Science and Technology Research, BUPT, China
[3]Capital Markets Cooperative Research Centre, Sydney, Australia

ABSTRACT

This paper describes work that attempts to use language technology as a solution to the problem of information overload. The specific domain of application is the database of company announcements accessible via the web site of the Australian Stock Exchange: to meet regulatory requirements, over 100,000 documents a year are made available via this site, with only limited search facilities. We use a variety of techniques from language technology to make it easier to explore and manage the information in this data set. In this paper, we focus on our use of information extraction, which identifies and extracts important elements of information from a document, and text compaction, which applies linguistically-motived substitutions to reduce potential summary sentences to more compact forms. Together, these techniques provide a way of producing summaries of a significant proportion of the document base.

1. INTRODUCTION

Each year, over 100,000 company announcements are made available via the Australian Stock Exchange's website by publicly listed companies in order to keep the market informed of any activities or events that might be of importance with regard to the trading of shares in those companies.[1] The documents submitted to the ASX and made available via the web site can vary in length from short, one page announcements of the resignation of a director, to very long annual reports, often in excess of 100 pages.

Even limiting one's interest to a small number of companies still means that there are a substantial number of documents that must be monitored to keep track of what is happening in the market. The ASX's web site does not make this a particularly easy task: although it is possible to search the document database by the name of the issuing company, and by imposing date range restrictions on the search, the user is ultimately led to a page of potentially uninformative titles, along with links to the source PDF documents that contain the actual announcements. Determining what is contained in any given document generally requires the invocation of a PDF reader, and the downloading of the document in question.

As an alternative to this mode of delivery, we have been exploring the use of techniques from language technology as a way of extracting from these documents useful summaries of the information they contain, so that this can then be presented to the user more effectively, or repackaged for delivery in other media.

This paper provides an overview of the current status of our work in this area. In Section 2, we provide a description of the data set we are working with. Section 3 then provides an overview of the overall architecture of our system. Section 4 elaborates on two central components, these being the information extraction component, and a component which carries out what might be called text compaction, taking a natural language sentence and producing a compressed form of that sentence. Section 5 discusses the strategies used in our text compaction component in more detail. Section 6 concludes the paper by assessing the current state of the work and identifying some future directions.

2. THE COMPANY ANNOUNCEMENTS DATA

As noted above, each year the Australian Stock Exchange makes available over 100,000 announcements from publicly listed companies, in the form of PDF documents. The document collection has a property which turns out to provide extremely valuable leverage when it comes to applying language technologies: the documents are categorised by the ASX into a hierarchy of types depending on their content and purpose, and this gives us a way of targetting specific techniques to specific document types as appropriate.[2]

The ASX's document hierarchy encompasses 19 basic report types, shown in Table 1, and around 120 subtypes; as an example, the subtypes for report type 02 are shown in Table 2, along with counts indicating the distribution of these documents across the year 2000 data set. The document

[1]See `www.asx.com.au`.

[2]Another part of our project is concerned with the automatic categorisation of these documents within this report type hierarchy. See [2, 3] for more information.

	Category	Number	%age
01	Takeover Announcement	4616	3.38
02	Security Holder Details	25372	18.57
03	Periodic Reports	24323	17.80
04	Quarterly Activities Report	6617	4.84
05	Quarterly Cash Flow Report	383	0.28
06	Issued Capital	21785	15.94
07	Asset Acquisition Disposal	3832	2.80
08	Notice of Meeting	7381	5.40
09	Stock Exch. Announcement	2900	2.12
10	Dividend Announcement	1037	0.76
11	Progress Report	9169	6.71
12	Company Administration	7183	5.26
13	Notice of Call	11	0.01
14	Other	10481	7.67
15	Chairman's Address	1657	1.21
16	Letter to Shareholders	1999	1.46
17	ASX Query	1377	1.01
18	Warrants	5682	4.16
19	Commitments Qtrly Reps	825	0.60
	Total	136630	

Table 1. The 19 basic report types

Type	Security Holder Details	# Docs
02/001	Becoming a substantial holder	3763
02/002	Change in substantial holding	8249
02/003	Ceasing to be a substantial holder	1717
02/004	Beneficial ownership - Part 6C.2	5
02/005	Takeover update - Section 689 Notice	2314
02/006	Security holder details - Other	546
02/007	Section 205G - Directors Interests	8778
Total		25372

Table 2. The subtypes of report type 02

types fall into a number of different categories.

Some documents are simply completed versions of forms, where the forms are either provided by the ASX, or are standard pro-formas developed for frequent reporting by the companies themselves.

A large number of document types, although not anything like as predictable as forms, consistently contain well-defined information elements. For example, documents of subtype 12001 (*Director Appointment/Resignation*) always contain information about the names of those who have been appointed or who have resigned as directors, but the way in which this information is expressed can vary widely.

Some document types, although concerned always with the same topic, may provide widely varying information with respect to that topic.

Finally, some document types cover a very wide range of topics: these can often be assigned to the general area covered by a specific top level report type, but within this type are categorised as being of subtype 'Other'.

These characteristics enable us to consider the application of specific techniques to particular document types, as outlined in the next two sections.

3. THE DOCUMENT PROCESSING ARCHITECTURE

3.1. Document Formats

Although our work has focussed on the ASX company announcements document set, we intend our techniques to be applicable to other kinds of document. To remove ourselves from the specifics of any given document format, our component technologies assume that the document to be processed has been first packaged up in what we call GSML (for GainSpring Markup Language). This is simply an application of XML that wraps up the text of the document in a BODY element, and provides a collection of management information in a HEADER element. The details of this representation are not particularly important here.

We assume that the client is responsible for converting the source document into GSML, and it should be capable of interpreting the GSML document returned after processing. Commercially available tools are available for the extraction of the content of PDF files as either plain ASCII text or HTML, the results of which can be wrapped up in GSML; for our experiments we have used both text extracted automatically from PDFs, and manually corrected versions as provided by ASX's Signal G electronic document feed.

3.2. Processing Strategy

The general document processing strategy pursued by our system is straightforward: information about the document's source and type is encoded in the GSML HEADER element, and so the system can use this information to determine what action to take. For some document types (about 35 subtypes in the ASX category hierarchy), we apply information extraction techniques, as described in Section 4.1; for another 50 or so document types, we apply more general text summarisation, as described in Section 4.2. Of the 120 document subtypes in the ASX type hierarchy, there are about 35 for which we currently attempt no processing at all; these documents fall into three categories:

- Some document types are very low frequency, and are therefore not worth attending to, or are not worthy of the development of specific processing.
- Some we have separated out as requiring a quite different approach: in particular, some types, such as periodic reports, often contain quite complex tabular

structures that require full-blown table interpretation. This is a focus of current work.

- Some document types are such that it makes little sense to summarise their content; meeting agendas are a good example of this category.

In practice, we carry out text summarisation for almost all document types, including those where we also carry out information extraction. This means that the client application is free to choose which result to use, perhaps on the basis of some assessment of the quality or consistency of information extraction across some document sample.

4. GENERATING SUMMARIES

4.1. Summaries via Information Extraction

Information Extraction (IE; [5, 1, 8]) is concerned with identifying a pre-specified set of key data elements from a free-text data source, and is widely recognised as one of the more successful spin-off technologies to come from the field of natural language processing. A major component task in information extraction is named entity recognition [9], whereby entities such as people, organizations and geographic locations are identified and tracked in texts; other processing can then take the results of the named entity recognition process to build higher order data structures, effectively determining who did what to who, when and where.

In each case, the general strategy is to construct a template that specifies the elements of information that need to be extracted from documents of a given type, and then to build shallow-processing natural language tools that extract these elements of information. These tools often use simple finite state parsing mechanisms: at the lowest level, the named entities—references to people, places and organisations—will be identified, along with dates, currency amounts and other numerical expressions; then, higher-level finite state machines may identify sentential or clausal structures within which these lower level elements participate. In many cases, the events of interest require the aggregation of information across multiple sentences.

Our information extraction techniques follow this pattern. As an example, documents of report type 02001 are quite predictable, and in many cases the required data is found by patterns of the following type:[3]

```
$Party became $shareholdertype
in $Company on $Date
with $interest of $sharespec
```

Here, `$Party`, `$Company` and `$Date` correspond to structures identified in a preprocessing step by our named entity recogniser; `$shareholdertype`, `$interest` and `$sharespec` are patterns that match the variety of ways in which information about the nature of the shareholding can be expressed. In practice, any document type requires a collection of such patterns that capture the range of different forms of expression used; we have developed a relatively simple rule language and associated interpreter that eases the production of these rules. Figure 1 shows the results for a sample document.

[3]This is a much-simplified representation of a rule in our system.

Element	Contents
DocumentCategory	02001
AcquiringParty ASX Code	TCN
AcquiringParty	TCNZ Australia Pty Ltd
AcquiredParty ASX Code	AAP
AcquiredParty	AAPT Limited
DateOfTransaction	4/07/1999
NumberOfShares	243,756,813
ShareType	ordinary shares
PercentageOfShares	79.90%

Fig. 1. Extraction results for a document of type 02001

A number of report types exhibit similar simplicity; others, however, are more complex, with the information we need to find being much more dispersed around the document. Figure 2 shows an example of a document of type 02002, where the information to be extracted—regarding both the change in holdings, and the total holdings that result from the change—is spread over several sentences.

```
This is to advise that Toll Group (NZ) Limited
has today announced to the New Zealand Stock
Exchange that its holding in Tranz Rail
Holdings Limited has increased by an additional
20,800,000 common shares at NZ Dollars $0.94
per share. This represents a further
consideration of NZ Dollars $19,552,000. These
additional shares now increase Toll Group (NZ)
Limited's holding in Tranz Rail Holdings
Limited to a total of 42,034,153 common shares,
representing a 19.99% shareholding in Tranz
Rail.
```

Fig. 2. A change in shareholdings

4.2. Text Bite Summaries

If we do not have a set of information extraction rules for a given document type, or if the extraction process delivers no results, or if we do not recognise the type of the document, then we pass the document to a more generic text summarisation process.

Text summarisation [10] is a widely explored topic in

natural langauge processing. Historically, a distinction can be drawn between relatively simple approaches that are based on the extraction of sentences that might serve as components of a summary, and more sophisticated 'knowledge-based' approaches that try to achieve some understanding of the text in order to then regenerate a summary from some derived representation. In practice, approaches of the latter kind are still only really explored in research laboratories, and tend to suffer from the kind of domain-specificity problems that one would expect. Real functioning summarisation systems, especially when broad coverage is required, are of the sentence extraction type.

Our summarisation process, built around the idea of constructing what we call 'text bites', consists of two steps.

- First, we use a number of heuristics to identify a sentence from the document that might serve as a good summary of the document's content.
- Then, we apply a number of 'compaction techniques' to compress the content of this sentence: these sentences are typically quite lengthy and often contain material that is not particularly important; given that a number of our possible delivery mechanisms require short, concise statements, we want to remove as much unnecessary material as possible.

For each document type in the hierarchy, we make use of keywords that are good indicators of important sentences for that document type. The top level types make use of fairly general terms such as *announce*; the lower level types make use of terms that are specific to those subtypes. A number of other heuristics, such as the appearance within a sentence of one or more named entities, can aid in selecting from amongst multiple candidates for a given document.

5. PRODUCING COMPACT SUMMARIES

There are two key aspects to generating the kinds of summaries we need. First, they need to be appropriately indicative of the content of the document being summarised; and second, they need to be concise and to the point.

The first requirement is addressed by the sentence selection mechanism outlined in the previous section. The second requirement is met by the component described in this section: given a candidate sentence to be used as a document summary, we apply a number of heuristics to produce a shortened version of the sentence. The general idea of 'text compaction' has been explored elsewhere in the literature. For example, [4] describes a mechanism that takes a text and turns it into a densely packed and abbreviated form that maximises the amount of information that can be delivered as an SMS message, making use of many of the abbreviatory conventions of text messaging; and [7] describes a system which removes parts of sentences on the basis of

```
Yesterday <ENAMEX Type="Company"
ID="ASX-CBA">the Commonwealth Bank of
Australia Pty Ltd</ENAMEX> announced
...  This is not the first time
<ENAMEX Type="Company" ID="ASX-CBA">the
Commonwealth Bank</ENAMEX> has made
...  The remaining original director,
<ENAMEX Type="Person" ID="JM001">James
Merriott PhD</ENAMEX>, has yet to sell
his interest in the company.  When
asked about this, <ENAMEX Type="Person"
ID="JM001">Merriott</ENAMEX> said ...
```

Fig. 3. Named entity recognition

syntactically-driven heuristics. In our case, we don't currently require the degree of compacting provided by either of these approaches; but we do need to reduce the sometimes overwordy sentences extracted from our documents to something that is more easily assimilable by the reader. For this reason, we refer to the results of this process as 'text bites', by analogy with the concept of a sound bite.

5.1. Named Entity Abbreviation

By the stage at which our summarisation component is invoked, the text has already been passed through our named entity recognizer, and this enriches the text with additional information that we can use to compress the text intelligently. The named entity recognizer identifies and marks a wide range of named entity types, but the most important for our present purposes are the following:

- Companies which are included in the comprehensive list of publicly-listed companies used by the system have already been tagged with their stock codes, as exemplified in Figure 3.
- Where a named person is someone explicitly listed in our lexicon of 'known individuals', their full name and possibly other attributes are marked up in the text.
- Dates are marked up in a standard format, with some resolution of temporal reference being carried out.
- Monetary amounts and numeric values of various kinds are identified and normalised.

This markup makes it relatively trivial to carry out some simple but high-value forms of abbreviation, in terms of the amount of reduction in length they provide.

- Companies can be reduced to their three-character ASX stock codes; this is probably the most space-saving of the abbreviation techniques we use.
- Named entities that correspond to person names can be reduced to surnames.

- Dates are reduced to their minimal canonical forms: currently we reduce all dates to strings of the form *YYYY-MM-DD*, but any form of date can be output.
- Monetary amounts and numeric values can be provided in abbreviated form, with rounding being carried out if this seems desirable: for example, an announcement that a particular entity has acquired *1,450,000 shares* can trivially be reduced to *1.5m shares*.

There are of course some limitations here. For example, not all companies that are mentioned in these documents are listed in our company name list; in such cases, we cannot provide an ASX stock code. However, we can still save some characters in many such cases by removing corporate designators such as *Pty Ltd* and *Inc*, leaving just the company name.

5.2. Participant Substitution

The other substitutions and abbreviations we carry out can be viewed as string replacement approximations of what are really more sophisticated syntactic and semantic transformations. The need for our system to process documents quickly (we generate a summary of a document in well under one second) means that, given current technology, we cannot rely on using sophisticated parsing techniques even if tools with the required coverage could be found. Consequently, our approach is to identify candidate methods for compacting text from a linguistic perspective, and then to implement shallow finite-state approximations of these linguistically motivated substitutions.

The first and simplest of these is what we refer to as 'participant substitution'. This technique takes advantage of observations like the following: *If an executive of a company makes an announcement, then it is reasonable to say that the company has made the announcement.* Syntactically, one can think of this rule of thumb motivating syntactic transformations that take structures of the form

```
[S [NP [Det The]
       [N Directors]
       [PP [Prep of]
           [NP [Det the]
               [N Company]]]]
   [VP [V announced] ...
```

and return structures of the form

```
[S [NP <CompanyName>]
   [VP [V announced] ...
```

This substitution makes use of the observation that mentions of *the Company* in the first few sentences of a document generally refer to the company issuing the document, information that we already have available in the document headers.

We have characterised the substitution here as a semantically-driven syntactic transformation: given a particular syntactic structure, we can replace a participant in a sentence by 'upwards delegation'. However, although the transformation may be justified in this way, it can trivially be implemented as a simple string replacement process, which is precisely what our summarisation component does.

This rule does not, of course, hold of text in general. It is easy to find examples, even in the domain of company announcements, where the application of this rule would impact on the veracity of the resulting text. For example, applying the rule to a sentence like *The directors of Acme Engineering announced that they have all resigned* would produce nonsense. However, in our corpus, the rule works reliably for the restricted range of document types to which it is applied.

5.3. Formality Removal

A feature of our extracted sentences that contributes significantly to their length is their inclusion of what we refer to here as 'formalities'. Some examples of these are as: *X wishes to advise that . . .* ; *Notice is given that . . .* ; and *I have been instructed to advise you that . . .* . Linguistically, these are structures where the important content is buried within an embedded sentence; the embedding sentential material can in these cases be trivially removed.

Again, we can characterise the transformation required as an editing operation on syntax trees, but in the absence of a syntactic parse, the same result can be achieved by a string substitution.

5.4. Implication

A form of transformation closely related to the above is what we refer to here as 'implication'. This covers cases where some state or event is described, as a consequence of which some other state or event can be assumed to hold or take place; for example, the state of affairs described in a sentence like *CBA has concluded arrangements to purchase 50% of ANZ* also follows from *CBA to purchase 50% of ANZ*. There are a wide range of constructions and associated verbs that have this form, capturable by means of transformations like the following:

[. . . have concluded arrangements to . . .] $\rightarrow$ *[. . . will . . .]*
[. . . have entered into an agreement to . . .] $\rightarrow$ *[. . . agree to . . .]*
[. . . have entered into an agreement under which . . .] $\rightarrow$ *[. . . agree . . .]*

As above, these transformations are implemented using reasonably straightforward string substitution.

Some cases are a little more complex; in particular, we have to watch out for pronominal usage as in *CBA is pleased to report that it will issue ...*. In these cases, our transformation ensures that the pronominal form is replaced by the more complete antecedent nominal; thus we would produce a summary sentence of the form *CBA will issue ...*.

5.5. Other Abbreviatory Devices

There are a collection of other strategies we use beyond those described above. For example, roles in companies, such as *Chief Executive Officer*, can be replaced by abbreviations; and we have also experimented with syntactically-motivated tactics like removal of sentential adjuncts or prepositional phrases. However, there is a limit to what can be achieved here safely using purely superficial techniques; in particular, when removing material from a sentence, there is always a danger that the truth conditions of the original sentence may be lost. The potential for damage can be limited here by appropriate restrictions on the lexical elements that are elided: for example, it would be inappropriate to remove the sense of allegation in *Smith alleged that he did not kill Jones*, since the reduced form *Smith did not kill Jones* is not a logical implication of the original. However, not all such dangerous transformations are so easily identified.

6. CONCLUSIONS AND FUTURE WORK

6.1. Evaluation

So far, we have only carried out formal evaluation of the results of our information extraction technology, and even here only on a small subset of document types. For documents which behave quite predictably (as is the case for the majority of the 02 report types, for example), we achieve extremely high accuracy; but there are many other report types where an informal analysis suggests we still have some way to go in terms of the coverage of our extraction rules.

The text bite summarisation technique is much harder to evaluate. The two questions one would want to ask of any such approach are (a) whether it conveys truth, and (b) whether it conveys what the document is really about. The first of these could be determined by an appropriately set-up experiment, without too much subjectivity entering into the results; however, the second property is much more open to subjective interpretation. In this regard, we are in no worse a situation than other work in text summarisation.

6.2. Extensions and Future Work

Currently, our system has the status of a working prototype: it operates over a document database of some 200,000 documents, and, as noted above, produces good results for some document types, but performs less well on other types. A clear short-to-medium-term goal for us is to find some manageable way of assessing the quality of the system's results.

There are also subsets of the data, noted earlier, for which we do not yet have a viable summarisation solution. Many of these require sophisticated table processing; this is an area we are currently addressing, with the hope that bringing domain knowledge to bear on the task of table interpretation will provide a way of achieving high quality results.

7. REFERENCES

[1] D Appelt, J Hobbs, J Bear, D Israel, M Kameyana and M Tyson. Fastus: a finite-state processor for information extraction from real-world text. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'93)*, 1993.

[2] Rafael A. Calvo. Classifying financial news with neural networks. In *Proceedings of the 6th Australasian Document Symposium*, December 2001.

[3] Rafael A. Calvo and Ken Williams. Automatic categorization of announcements on the Australian Stock Exchange. In *Proceeedings of the 7th Australasian Document Computing Symposium*, 2002.

[4] Simon Corston-Oliver. Text compaction for display on very small screens. In *Proceedings of the Workshop on Automatic Summarization, NAACL 2001*, Carnegie Mellon University, Pittsburgh, USA, 2001.

[5] J. Cowie and W. Lehnert. Information extraction. *Communications of the ACM*, Volume 39, Number 1, pages 80–91, 1996.

[6] Robert Dale, Rafael Calvo and Marc Tilbrook. Key element summarisation: Extracting information from company announcements. In *Proceedings of the 17th Australian Joint Conference on Artificial Intelligence*, 7th-10th December 2004.

[7] G. Grefenstette. Producing intelligent telegraphic text reduction to provide an audio scanning service for the blind. In *AAAI Spring Symposium on Intelligent Text Summarization*, pages 111–117, Stanford, 1998.

[8] P Jackson and I Moulinier. *Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization*. Benjamins, Amsterdam, 2002.

[9] Andrei Mikheev, Claire Grover and Marc Moens. XML tools and architecture for named entity recognition. *Markup Languages*, Volume 1, Number 3, pages 89–113, 1999.

[10] C. Paice. Constructing literature abstracts by computer: Techniques and prospects. *Information Processing and Management*, Volume 26, pages 171–186, 1990.