Key Element Summarisation: Extracting Information from Company Announcements

Robert Dale[1,3], Rafael Calvo[2,3], and Marc Tilbrook[1]

[1] Centre for Language Technology, Macquarie University, Sydney
www.clt.mq.edu.au
[2] Web Engineering Group, University of Sydney, Sydney
www.weg.ee.usyd.edu.au
[3] Capital Markets Co-operative Research Centre, Sydney
www.cmcrc.com.au

Abstract. In this paper, we describe KES, a system that integrates text categorisation and information extraction in order to extract key elements of information from particular types of documents, with these informational elements being presented in such a way as to provide a concise summary of the input document. We describe the overall architecture of the system and its components, with a particular focus on the problems involved in handling the names of companies and individuals in this domain.

1 Introduction

Information Extraction (IE [1–3]) is concerned with the process of 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. During the 1990s, the DARPA-funded Message Understanding Conferences resulted in a number of systems that could extract from texts, with reasonable results, specific information about complex events such as terrorist incidents or corporate takeovers. These information extraction tasks are manageable because, in each case, some other process has determined that the document being analysed falls within the target domain, and the key information to be extracted is typically only a very small subset of the total content of the document. A major component task in information extraction is **named entity recognition** [4], 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 this paper, we describe KES, an experiment in information extraction where we first use text categorisation to determine the type of document being processed; given the document's type, we can make hypotheses about the kinds

of informational elements that can be extracted from that document. After extracting these key elements, we can then produce concise summaries of the input documents, thus saving the user the need to read the source documents in order to determine the central information they contain.

The KES project operates in the domain of financial information. In particular, we have been working with a data set from the Australian Stock Exchange (ASX). This data set consists of a large set of company announcements: these are documents provided by companies to the ASX, who subsequently make them available to users via the web. Many of these documents are required for regulatory purposes, and these regulations impose some requirements on the content of documents. The ASX categorises the documents into a large number of different types, including categories like 'change in shareholding', 'notice of intended takeover', 'statement of director's interests', and so on. Our goal is to take this data set (and similar data sets) and to add value to the documents by making use of language technologies.

In Section 2, we describe the characteristics of our particular problem scenario. Section 3 lays out our approach to solving this problem, and Section 4 elaborates on our approach to the handling of potentially unknown names of organisations and individuals. We conclude by summarising the results produced by our system so far, and pointing to further work in the area.

2 Background

2.1 The Problem

The corpus we are working with is the Signal G document set, a collection of corporate announcements made available by the Australian Stock Exchange (ASX) via their web site. These documents are provided to the ASX by companies in order to adhere to regulatory requirements; in various circumstances, companies are required to provide appropriate documents detailing, for example, changes in shareholding, or intentions to make takeover bids, and the ASX then makes this information available via their web site.[4]

The number of documents involved here is vast: over 100000 documents are submitted to the ASX each year and made available via the web site. This makes it difficult for a user to easily obtain and track information of interest: although the ASX web site permits searching by the stock exchange codes of the companies involved, this still provides only a very limited means of filtering the data.

In order to ease this difficulty, we set out to build a system which could extract key elements of information from this document collection: by deriving this set of structured data from the relatively unstructured data, we would be in a position to facilitate both the indexing and browsing of the documents by allowing a more structured search of their contents and a number of other services. For

[4] See `www.asx.com.au`. As of mid-2004, the ASX put in place constraints on the use of data gathered from their web site; the experiments reported here pre-date the imposition of those constraints.

example, when a company is in the process of taking over another company, it is required to issue frequent updates that report on the relevant changes in shareholding; by extracting this information from the source documents, we can track such events across multiple documents. We can also identify and track individuals across documents, so we can easily provide information about the involvement of particular individuals across a range of companies. With the additional structure added to the documents using IE, we can, for example, provide a messaging service that sends alerts (i.e., emails or SMS messages) based on specific triggers, or even generate speech summaries of important events which can be sent automatically.

2.2 The Corpus

Our corpus consists of a set of 136,630 documents from the year 2000. The ASX categorises the documents it receives into a large set of categories: there are 19 basic report types, subdivided into 176 subtypes. The 19 basic report types are shown in Table 1, with a breakdown showing how many documents from our sample fall into each category.

	Category	Number
01	Takeover Announcement	4616
02	Security Holder Details	25372
03	Periodic Reports	24323
04	Quarterly Activities Report	6617
05	Quarterly Cash Flow Report	383
06	Issued Capital	21785
07	Asset Acquisition Disposal	3832
08	Notice of Meeting	7381
09	Stock Exchange Announcement	2900
10	Dividend Announcement	1037
11	Progress Report	9169
12	Company Administration	7183
13	Notice of Call (Contributing Shares)	11
14	Other	10481
15	Chairman's Address	1657
16	Letter to Sharareholders	1999
17	ASX Query	1377
18	Warrants	5682
19	Commitments Test Entity Quarterly Reports	825
	Total	136630

Table 1. The 19 basic report types in the Signal G Data

For our experiment, we focussed on the sub-types of report type 002, as shown in Table 2. We focussed on this category of reports for three reasons:

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 Notice - Directors Interests	8778
Total		25372

Table 2. The subtypes of report type 002

- First, the category represents a significant proportion (18.5%) of the documents in our corpus.
- Second, our categorisation technology, as discussed below in Section 3.1, worked very well on these categories.
- Third, the documents in this category are relatively predictable in terms of their content, and often quite short.

Figure 1 shows an example of the text contained in a document of type 02/002. This is a very simple example, as are most in this category; by way of contrast, Figure 2 shows a more complex document that reports a change in shareholdings.

```
Document date: Fri 20 Jun 2003 Published: Fri 20 Jun 2003 14:37:46
Document No: 298709  Document part: A
Market Flag: N
Classification: Change in substantial holding
SYDNEY AQUARIUM LIMITED                        2003-06-20  ASX-SIGNAL-G

HOMEX - Sydney

++++++++++++++++++++++++++
Commonwealth Bank of Australia decreased its relevant interest in
Sydney Aquarium Limited on 17/06/2003, from 3,970,481 ordinary shares
(18.49%) to 3,763,203 ordinary shares (17.29%).
```

Fig. 1. A simple document of type 02/002

In the current ASX web site, accessing this information requires considerable effort on the part of the user. First, they must search the document set by entering the ASX code of the company they are interested in; this results in a list of documents individuated by their titles, with hyperlinks to the PDF and text versions of these documents. It then takes two more clicks to access the text

```
Document date: Fri 27 Jun 2003 Published: Fri 27 Jun 2003 08:55:58
Document No: 205676  Document part: A
Market Flag: N
Classification: Security holder details - Other , Asset Acquisition
TOLL HOLDINGS LIMITED                          2003-06-27  ASX-SIGNAL-G

HOMEX - Melbourne

++++++++++++++++++++++++++
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.

B McInerney
COMPANY SECRETARY
```

Fig. 2. A more complex document of type 02/002

that makes up the document, as shown in Figures 1 and 2; and then, of course, the user has to read through the document to find the information of interest.

2.3 Our Goals

Our goals, then, were to develop techniques for finding the key elements of information in the documents in our corpus. The identification of key elements is essentially an information extraction problem. The idea is that, for certain kinds of documents, we can determine ahead of time specific items of information that those documents contain. In the case of a change of shareholding, for examples, the key elements of information would be the name of the shareholding company, the number of shares it now holds, the company the shares are held in, and the date of the change.

Information that has been extracted from a document can be used in a variety of ways; for example, it can be used to populate a database, and the database might then be searched or analysed in various ways. The focus we have taken so far in KES is that we can add value to documents by producing short summaries of what those documents contain, and we generate those summaries using the key elements we have extracted. By finding the key information and then presenting it in a summarised form, we make it much easier for a user to find information.

3 Our Approach

Our system consists of three processes:

Text Categorisation: We use a text categoriser, trained on human-annotated documents from the Signal G corpus, to determine the report type of each document.

Information Extraction: Given the document's type, we then apply a collection of information extraction routines that are adapted to the specific document types; these locate the key elements of information that are relevant for that document type.

Information Rendering: Once extracted, this information can then be represented to users in a variety of forms.

Each of these processes is described in more detail below.

3.1 Text Categorisation

Our text categoriser is described elsewhere [5, 6], and so we will restrict ourselves here to some comments relevant to the present discussion.

The categoriser is trained on the human-categorised data set, and the results for a test set of 7620 documents are shown in Table 3.

Category	# Docs	Precision	Recall	F_1
02/001	1109	0.975	0.951	0.963
02/002	2457	0.96	0.957	0.958
02/003	517	0.959	0.972	0.966
02/004	0	0	1	0
02/005	702	0.971	0.984	0.978
02/006	184	0.260	0.586	0.361
02/007	2651	0.986	0.952	0.968

Table 3. Categoriser performance on Report Type 02

As can be seen from the table, the categoriser works well on all the subtypes of category 02 except for 02/004 (*Beneficial ownership*), for which there were no documents in our set, and 02/006 (*Security holder details–Other*), which is effectively a 'miscellaneous' category. It should also be noted that these results are produced by comparing the categoriser's assignment of report types to that of the human annotators; some preliminary analysis, however, has determined that the human annotators make mistakes, and so it is possible that our categoriser is performing better than these numbers suggest.

3.2 Information Extraction

Information extraction is now a well-developed body of techniques that has been applied in a wide range of contexts. 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. The templates for two report subtypes are shown in Figures 3 and 4.

Field	Contents
AcquiringParty	a Company or Individual named entity
AcquiredParty	a Company named entity
DateOfTransaction	the date of transaction
NumberOfShares	the number of shares owned as a result of the transaction
PercentageOfShares	the percentage of shares owned as a result in the transaction
ShareType	one of {ordinary, voting, ...}

Fig. 3. Extraction template for 02/001, *Becoming a Substantial Shareholder*

Field	Contents
Director	the name of the director
Company	the Company in which the director has an interest
PreviousNotificationDate	the date of previous notification of interest
InterestChangeDate	the date of change of interest
CurrentNotificationDate	the date of current notification
Holding	a structure consisting of **HoldingCompany**, **NumberOfShares**, and **ShareType**

Fig. 4. Extraction template for 02/007, *Section 205G Notice—Director's Interests*

Documents of report type 02/001 are quite predictable, and in many cases the required data is found by pattern of the following type:[5]

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

[5] This is a simplified representation of a rule in our system.

Here, `$Party` and `$Company` are complex patterns used to identify persons and companies; `$shareholdertype`, `$interest` and `$sharespec` are patterns that match the variety of ways in which information about the nature of the shareholding can be expressed; this is typically distributed over a number of nominal elements separated by prepositions, so we use this fact in anchoring the pattern matching. Table 4 shows the results for a sample document.

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

Table 4. Extraction results for a document of type 02/001

A number of report types exhibit similar simplicity; others, however, are more complex, with the information we need to find much more dispersed around the document. In the case of report subtype 02/007, for example, the information is often presented in the form of a table; however, since this table is rendered in plain ASCII text, we need to parse the table to identify the required information. A number of researchers have worked on this particular problem: see, for example, [7, 8]. Our techniques for doing this are still being refined, as demonstrated by the significantly poorer extraction performance on this category: Figure 5 shows results for a random test sample of 20 documents of each of the five 02 subtypes, demonstrating that we do significantly worse on this category.[6] In general, however, the accuracy is high, largely because of the predictability of the documents. The major problems we face are in handling variety and complexity in proper names, a point we return to below in Section 4.

3.3 Information Rendering

Once we have extracted the information, we need to present it to the user in a maximally useful way. We are experimenting with a number of ideas here, including voice synthesis of short sentences that contain the key elements; on the web, we have implemented a mechanism that pops up a box showing the key

[6] The '# slots' column indicates the total number of extractable slot fills available in the sample selected; 'Found' indicates the number of slots extracted; 'R' and 'P' provide the recall and precision figures respectively. The f-score shown is calculated as 2PR/(P+R).

Category	# slots	Found	R	True +ves	P	False +ves	f-score
02001	119	119	1.000	118	0.992	1	0.996
02002	189	188	0.995	188	1.000	1	0.997
02003	60	57	0.950	56	0.982	1	0.966
02005	117	101	0.863	100	0.990	1	0.922
02007	129	69	0.535	68	0.986	1	0.693
Total	614	534	0.870	530	0.993		0.927

Fig. 5. Success rates in extracting key elements

data fields whenever the mouse is scrolled over the title of the document, thus avoiding the need for several mouse clicks to see what the document contains. The same information could be presented in tabular form to allow sorting or comparison by specific fields and values.

4 Issues in Handling Named Entities

In our domain, we have so far focussed on documents whose structure is relatively well-behaved, so that we achieve high accuracy in extracting the key elements. However, making sense of some of these key elements proves to be a little harder; in particular, the variety of forms of proper names that we find introduces some difficulties into the task. This is particularly the case since we want to resolve, where possible, company names to stock exchange codes, so simply identifying that we have a named entity is not enough; we need to be able to work out what that named entity is.

In this section, we describe some of the problems we face in handling proper names, and outline the solutions we have developed so far.

4.1 Variations in Proper Names

Variations in proper names fall into two broad categories: legitimate variations and misspellings.

Legitimate Variations Legitimate variations cover cases where multiple names are used for the same entity. In the case of companies, for example, both *Broken Hill Proprietary Limited* and *BHP Ltd* refer to the same organisation. In KES, we attempt to resolve company names to their ASX codes, and so determining that these two terms refer to the same entity is important. Currently, we achieve this by using a small number of heuristics in conjunction with a large manually constructed table of known company names that maps these to their stock codes. One heuristic, for example, looks for substring matches on the 'content-bearing' elements of names, ignoring corporate designators like *Limited* on the basis that these are frequently omitted. There are other heuristics that might be used: in

the example just cited, we might employ a mechanism that, on finding an all-caps string like *BHP* in the text, looks for names whose element begin with the letters that make up the abbreviation; however, some initial experiments suggest that this is not very robust.

Legitimate variations are also common in person names: so, for example, *A Smith*, *Mr Smith*, *Alexander Smith* and *Alex Smith* might all refer to the same person. Clearly, the confidence with which identity of reference can be claimed for any two of these strings varies, with the third and fourth being closest, and any assumed co-reference with the second being the riskiest. The problem here, however, is much worse than with company names, since it is not uncommon to find different people sharing exactly the same name.

Misspellings Misspelled names are rife in our corpus, in large part because of the way in which the document set is constructed: documents are received from companies by fax, and these faxes are scanned, OCRed and then manually edited.[7] Quite apart from spelling errors that might be present in the source document, clearly each stage of this process has the chance of adding additional errors. The list below provides a sample set of misspellings of *Perpetual Trustees Australia Limited* found in our corpus:

Perpectual Trustees Australia Limited
Perpetual Trustee Australia Limited
Perpetual Trustee Company Limited
Perpetual Trustees Astralia Limited
Perpetual Trustees Australian Limited
Perpetual Trustes Australia Limited

Our heuristics currently accommodate a range of common misspellings, although clearly there are other methods that might be explored. Integration of a variation of Knuth's Soundex algorithm would address one type of error, where misspellings arise at source; a different approach would be required to handle errors which are clearly introduced by the OCR process.

4.2 Conjunctions of Names

The analysis of conjunctions is a long-standing problem in parsing. The problem is, quite simply, working out what the conjuncts are, given that in many cases some information is assumed to be shared between both conjuncts. This problem surfaces in our corpus in the form of examples like the following:[8]

1. Advent Investors Pty Limited; Enterprise Holdings Pty Limited; Leadenhall Australia Limited; Koneke Pty Limited; Noble Investments Pty Limited; Advent Accounting Pty Limited; Chi Investments Pty Limited

[7] As of mid-2003, the ASX has required companies to provide documents as PDF files rather than faxes, but a proportion of these are still produced by scanning.

[8] Each of these examples fills a single **AcquiringParty** slot in our extraction process, and so has to be decomposed into its constitiuent elements.

2. James Brenner Skinner, Janice Ivy Skinner, Topspeed Pty Limited, and GN, AW, CM, SM, and SB Skinner

In case (1), we have a long, semi-colon separated list of company names. This case is quite well behaved, with the punctuation effectively removing any problems in parsing. However, it is very common for more ambiguous conjunctions, such as the comma and *and*, to be used in separating elements, as in example (2): here, we need to determine that the first *and* separates complete names, whereas the second *and* separates sets of initials, each of which must be paired-up with the surname *Skinner*. Similar problems occur with proper names like *Centaur and Mining Exploration Limited*, *Investors Trust and Custodial Services*, and *Graham Whelan and G Whelan Pty Limited*: if these strings, or substrings contained within them, appear in our table of company names, we will recognise them appropriately; but there are many company names (for example, of overseas companies) which are not present in our database, and in such cases there is no reliable way of determining whether we have one entity or two.

Currently, we have a prototype parser based around a Prolog Definite Clause Grammar which returns all possible parses of conjunctions, applying heuristics that govern the forms of person names and company names; the results are then filtered these against the known names in the company database. In many cases, however, this still produces more than one possible parse, and so further heuristics are required in order to choose the best parse.

5 Conclusions and Further Work

We have described the components of a system which takes corporate announcements, categorises these into a number of report types, and then uses information extraction techniques to identify specific predefined elements of information in each of these report types. The information so extracted can then be provided to the user in a way that facilitates efficient searching or browsing of the document set, or exported from the database in a variety of other formats.

Our initial prototype performs well for the subset of document types that we have focussed upon, but there are a number of clear directions that we need to pursue next:

1. Although our categoriser performs well, it could still be improved. We are exploring how feedback from the information extraction mechanisms might help the categoriser. For example, if we are unable to identify the information elements for the report type that is ascribed by the categoriser, we can try to extract the elements required for the other report types, and if this is successful, provide this information back to the categoriser. The precise details of how the categoriser can use this information to revise its subsequent categorisation activity remain to be worked out.
2. Our information extraction techniques need to be extended to other report types, and the performance on some of the existing report types needs improvement. The ways forward here are reasonably clear, with a need for more

sophisticated mechanisms that can take account of widely dispersed information within a document. Our currently high values for extractions in the simpler document types are obtained largely because the documents' predictability means we can use quite tightly constrained patterns; as we relax these patterns, the scope for error in what we extract increases, and multiple possible values will be found for slots. We then need to develop techniques for choosing amongst alternative solutions.
3. Our handling of the problems posed by variations in proper names and conjunctions of proper names is still relatively limited.

For the last two of these extensions, we are exploring how the system can learn from its previous experience. Some of the techniques required here are quite simple: for example, if reliable data for most slot fills in a template means that an unknown string must correspond to a variant of a particular company name, then this hypothesis about the unknown name can be added to the database so that it is available for the processing of subsequent documents. We are in the process of exploring how best to integrate this kind of information into the system's data sources.

References

1. Cowie, J., Lehnert, W.: Information extraction. Communications of the ACM **39** (1996) 80–91
2. Appelt, D., Hobbs, J., Bear, J., Israel, D., Kameyana, M., Tyson, M.: 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)
3. Jackson, P., Moulinier, I.: Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization. John Benjamins, Amsterdam (2002)
4. Mikheev, A., Grover, C., Moens, M.: XML tools and architecture for named entity recognition. Markup Languages **1** (1999) 89–113
5. Calvo, R.A.: Classifying financial news with neural networks. In: 6th Australasian Document Symposium. (2001)
6. Calvo, R.A., Williams, K.: Automatic categorization of announcements on the Australian Stock Exchange. (2002)
7. Pinto, D., McCallum, A., Wei, X., Croft, W.B.: Table extraction using conditional random fields. In: Proceedings of SIGIR'03, Toronto, Canada., ACM (2003) 235–242
8. Hurst, M.: The Interpretation of Tables in Texts. PhD thesis, University of Edinburgh, School of Cognitive Science, Informatics (2000)