

# Text Classification of Formatted Text Documents

**Oliver Carr and Dominique Estival**

Human Systems Integration Group

Command and Control Division

Defence Science and

Technology Organisation

{Oliver.Carr, Dominique.Estival}@dsto.  
defence.gov.au

## Abstract

We describe a multiclass text classification system for formatted text messages contained in the Rich Text Format fields of a structured database of military documents. This system uses a Part-Of-Speech tagger and a Rule-Based Classifier to classify 80 different types of formatted messages.

Keywords: Text Classification, Rule-Based Classification, POS Tagging

## 1 Introduction

The document classification system we describe in this paper is the first stage of a sub-task on document classification within the larger speech and natural language processing task at DSTO.<sup>1</sup> We are concerned with automating and facilitating information retrieval and extraction from large document databases used within the headquarters of the Australian Defence Force. Most of these documents are formatted, but many of them also contain free text fields, which may themselves contain other documents or attachments. The aims of this project are to ultimately reduce the workload of operators managing tactical or operational information and thus to increase their efficiency, especially in high tempo operations such as peace keeping or natural disaster relief. Thus this project must result in a practical system for the management of electronic information.

---

<sup>1</sup> This project is funded through a DSTO task JNT 01/092 entitled "Human Computer Interaction Research, including Speech and Natural Language Approaches, for Command Support" managed by Dr. Ahmad Hashemi-Sakhtsari and sponsored by Commander Deployable Joint Force Headquarters, MAJGEN Mark Evans and Head Knowledge Systems, RDM Peter Clarke.

## 2 Electronic Information Management in the ADO

### 2.1 Databases used in the ADO

Lotus Notes is a suite of collaborative database tools, which represent information in a structured manner. IBM is convinced Lotus' COTS software with appropriate military amendments can play a significant role in military crisis applications (Fournery and Sorensen, 2000), and there are already 1300 applications using Lotus Notes for messaging and collaboration in the Australian Defence Organisation (Maple, 2001).

For instance, the Deployable Joint Force Headquarters (DJFHQ) uses Lotus Notes for e-mail and to log operational events. These "logs" are Lotus Notes databases, which contain several forms with mostly fixed fields and one or two Rich Text Format (RTF) fields.<sup>2</sup> The RTF fields in the Log Entry forms may contain any combination of free text, attached Microsoft Office documents and links to other Lotus Notes documents. During an operation, around 200 documents will be "logged" in a Lotus Notes database per day, but the actual number will vary depending on the operational tempo.

Administrative tasks slow down the operational effectiveness of any headquarters both in peacetime and when deployed into a tactical area of operations. One task which we aim to automate is the classification of the

---

<sup>2</sup> The forms are called Log Entry, Action, Comment, Shift Handover and SITREP (Situation Report). Log Entry forms are the most common forms and will result in an Action form being generated. Comment forms are generated from Log Entry forms. SITREP forms summarise the events of the last 24 hours. Shift Handover forms summarise the events and outstanding actions of the previous shift. All of these forms may contain one or more instances of a SOP document, as described below.

Standing Operating Procedure (SOP) documents entered in the logs.

## 2.2 Standard Operating Procedure Documents

The SOPs are all formatted text documents that are contained in the RTF fields of log documents. They describe strategic, tactical and operational events that range from casualty evacuation to situation reports. There are 88 different SOPs defined for use by the operators at DJFHQ and at the units reporting to DJFHQ. The definitions of these SOPs are provided in separate Microsoft Word files, some of which contain examples of the SOP they describe. An example of a Casualty Evacuation SOP document is shown in Fig 1 below.

```
UNCLAS
FROM 3 BDE
TO DJFHQ
DUSTOFF 6/00
A. MAP AUST GR 231456
1. DUSTOFF
2. 1 RAR
A. 1 Apache
B. ASAP
3. Gunshot wounds to chest and
limbs.
4. Total wt 100lbs
5. N/A
```

**Figure 1. Example of a SOP Document**

Every year the Information Manager at DJFHQ reviews all of the SOPs for that year. Our aim is to produce a system that automates sorting of these documents in the Lotus Notes database into classes corresponding to the different SOP documents and free text. Note that we are not trying to either summarize or route those documents (Jackson and Moulinier, 2002).

We already have a search tool developed in-house called the Query Building Interface (QBI) (Broughton et al, 2002). This tool was developed to improve the capabilities of operators to search the Lotus Notes databases. Lotus Notes uses full text indexing combined with Boolean search and Ranked Retrieval techniques as described in Jackson and Moulinier (2002). We are currently integrating our text classification system with QBI. We expect that this addition to the QBI will improve the response time for searches concerning the SOP documents in the database.

More specifically, the current project aims are to:

- Provide an efficient means of searching for information in the Lotus Notes duty logs at DJFHQ.

- Classify incoming messages into these databases.

- Use the Word files which define the Standard Operating Procedures in the classification process.

- Use the above classification in the QBI system already developed at DSTO.

This project will thus add a classification scheme to the QBI and associate each document to one of the classes defined by the SOPs, plus the “Free Text” and “Unclassifiable Formatted” classes.

## 3 Text Classification of SOP Documents

There are many different approaches to text classification of documents. Our system could have used Multiclass, Multilabel, Ordinal, Hierarchical, Probabilistic or Graded classification techniques (Lewis, 2002). Another approach is to use Hidden Markov Models to represent the structure of the SOP documents by keyword relevance. In this approach, the SOP document definitions are not used and documents with similar features are grouped or categorized together (Camilleri, 2002). Note that although some researchers make a distinction between the terms classification and categorisation, we use these terms interchangeably (Jackson and Moulinier, 2002).

We chose to use a Multiclass technique where each document is classified into exactly one of three or more classes (Lewis, 2002), because we made the assumption that each formatted message either belonged to exactly one SOP document or was free text. However, a document in the database may in fact contain a combination of free text, SOP document, link to another document containing a formatted message, free text incorporated in a numbered list and other unknown (to the system) types of documents. At a later stage in the project, we intend to deal with this issue by incorporating the Multilabel technique, which allows a document to be assigned to zero, one or more classes, and would provide a multilabel classification for documents containing attached or linked documents.

The advantage of both the Multiclass and Multilabel approaches is that there is a simple “text in, text out” interface, a finite set of outputs for count, correlation or conditioning a specific action (Lewis, 2002).

For now, the system we have developed is simply a rule-based multiclass classifier, which we call SOP-MRC (Standard Operating Procedures Multiclass Rule-based Text Classifier).

#### 4 SOP-MRC: System Description

The SOP-MRC contains two main components: the POS Tagger and the Rule-Based Classifier. It is best to describe the system in two stages: when it is used to train the POS Tagger, and when it is used at run-time. Fig. 2 shows a component level description of the training stage.

##### Training

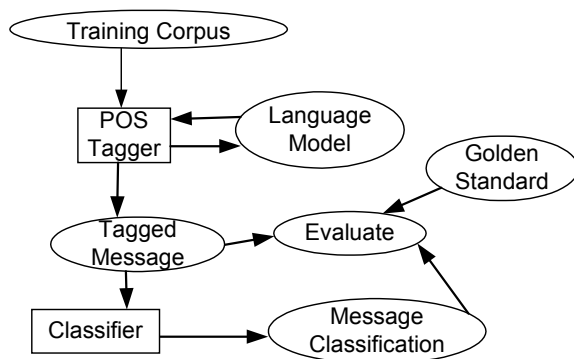


Figure 2. SOP-MRC: Training

The initial training corpus for the POS Tagger was the tagged lexicon for Industrial Parsing of Software Manuals (IPSM) from the AMALGAM project (Atwell et al, 2000) site at Leeds University. In addition to the 227 tags found in the Brown Corpus Tagset (Greene and Rubin, 1981), we created 52 new tag types to recognize formatting characters, Date Time Groups and some other words in the SOP documents.

For example, the tag **Frm1** is used for the text string “1.” (see Fig.1). Similarly, **Frm** is used to tag “FROM” and **FrmB** for “B.”.

#### 4.1 Part-Of-Speech Tagger

The Part-Of-Speech Tagger we are currently using is the QTAG POS Tagger developed by Oliver Mason (Tufis and Mason, 1998). After investigating a number of different POS taggers, the two which stood out were QTAG and Thorsten Brants’ (1998) TnT. QTAG is a purely probabilistic tagger and combines two sources of information, a dictionary of words and their possible tags and a matrix of tag sequences with corresponding probabilities (Tufis and Mason, 1998). In our case, these probabilities were generated from our pre-tagged corpus, which, as described above, was a combination of the IPSM corpus tagged with the Brown tagset and our specialized corpus with our 52 new tag types.

The training corpus used in the experiment we report here consists of 118 messages, representing 88 different message types. These messages came from the definitions of the SOPs provided by DJFHQ (see section 2.2).

#### 4.2 Rule-Based Classifier

At run-time, the system processes one input file at a time, using the Language Model created during training, as shown in Fig.3. A Python script was written to read in a file, convert the text to uppercase and process the output (Lutz, 2002).<sup>3</sup> This was required to correctly tag military acronyms and abbreviations. Most of the text in the Lotus Notes databases is written in uppercase, which is probably due to the amount of acronyms and abbreviations used in military messages.

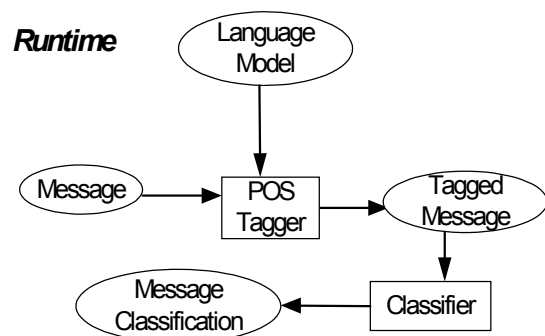


Figure 3. SOP-MRC: Runtime

<sup>3</sup> Python is used widely for language and text processing and was developed for rapid application development.

The Classifier module, also written in Python, uses a Rule-Based technique to classify the document into one of the SOP document classes or as free text.

The classification rules were first determined from an analysis of the 118 training documents described above, and they still need to be refined after examination of the larger test corpus. The most important, i.e. the most discriminating, features for classifying our documents rely on the POS sequences at the beginning and end of the document.

There are 15 different possible start POS tag sequences for the 88 SOP document types, but after considering the end POS tag sequences, only 13 combinations of start and end POS tag sequences remain ambiguous. Out of the 88 document types, 64 can thus be correctly identified simply by the start and end POS sequences. For the remaining 24 document types, the Classifier needs only to disambiguate between 2 and 6 different document types in each ambiguous case.

To perform this disambiguation, the Classifier relies on the contextual information provided by the POS tags which follow the new tags (such as **Frm1**, **FrmA**, etc., see section 4.1) that were added to the tagset. In most cases, the disambiguating tag either also belongs to this set of new tags, or is the CD tag (i.e. number).

While the rules in the Rule-Based Classifier were written by hand,<sup>4</sup> we intend to automate the generation process for the classification rules by an automated partially supervised technique.

## 5 Results

At this stage, we can only give very preliminary results, from the first run of SOP-MRC with a batch of 2331 real documents from the logs of two military exercises.

From this preliminary experiment, we found that only 19 different document classes out of the 88 classes defined at DJFHQ and known by the SOP-MRC were actually represented in the corpus. Of these, the largest was the class of “Free Text” (1456 out of a total of 2331 documents) and, unsurprisingly, this is the class that was recognised most accurately by SOP-MRC (1019 out of 1546 were correctly

recognised, giving a recall of 78.26%, and precision of 65.87%).

The preliminary results also show that for the other classes, the performance was much lower, and the total recall is only 53%, with total precision also about 53%.

We can explain this poor performance in terms of the discrepancies between the training corpus and the actual corpus used for testing. In particular, the training corpus was small and it did not contain enough instances of “Date Time Group” (DTG) to allow the POS tagger to learn to reliably recognise them in the test corpus. An example of a DTG expression is given in (1).

(1) AUG021730Z

The Rule-Based Classifier component of the SOP-MRC relies heavily on recognizing DTGs correctly, in particular for the 39 SOP document types which need to be further disambiguated. For example, the definition of the OPSTAT REPT SOP document does not rely on the recognition of DTG fields by the POS tagger, and this class was recognised more accurately (159 out of 217) than any of the other class, for which classification depends on recognition of DTGs.

On the other hand, the definitions of NOTICAS, CASEVAC/DUSTOFF and SITREP SOP documents are identical in structure, but the NOTICAS SOP contains a DTG after the tag **Frm2**, while CASEVAC/DUSTOFF contains a DTG after **Frm3**. The small training corpus and the resultant difficulty for the POS Tagger to recognise the DTGs help explain that there were 24 false positives and 11 false negatives for the CASEVAC/DUSTOFF class (for which only 1 out of 51 documents was correctly recognised), and 10 False Negatives for the NOTICAS class (for which no document was recognised).

Similar explanations can be given for the errors in the other classes, all stemming from the fact that the training corpus was too small and contained example messages rather than real messages.

- The real messages contain a larger preamble to the SOP documents. Since our classifier relies on recognising the sequence of POS tags at the start of a formatted message, this preamble resulted in a larger sequence of POS tags which were not handled by the Classifier.

- The real messages also contain messages received from other units reporting to

---

<sup>4</sup> The Rule-Based Classifier is an 800 line Python script.

the DJFHQ, and they do not always conform to the SOP definitions. Those real messages also contain a large number of other SOP documents not described by the SOP definitions provided by DJFHQ.

These problems should be remedied as soon as we use the larger corpus for training. We will also address the problem of correctly tagging DTGs more robustly by using the time and date information from the Lotus Notes document and then retagging the text from the RTF field, and by creating a separate POS tagger for DTGs.

## 6 Conclusions and Future Work

### 6.1 Improvements to the current SOP-MRC

From the discussion of the results presented above, it is clear that the first efforts must be directed at training the POS Tagger with more, and more realistic, data. This will lead to improved recognition of DTGs in particular, and to improvements in classification for certain document types.

Another area for immediate improvements concerns the hand-written rules used by the Classifier, which can now be refined after a closer analysis of the errors.

A third area is to make use of the probabilities associated with the POS tags in the classifier rules. With more data at our disposal, we will start using the probabilistic information from the commercial version of QTAG to create a numeric classifier to provide relevance to the result set in the QBI.

The combination of efforts in these three directions should lead to marked improvements in performance without changes to the current SOP-MRC system. New experiments will then give an indication of whether the addition of the SOP-MRC is of benefit to operators of the Lotus Notes databases.

### 6.2 Additions to SOP-MRC

The development of the Rule-Based Classifier and the ease with which new tagsets can be added to a POS Tagger such as QTAG suggest that the SOP-MRC will allow us to implement a more automated process for document management in DJFHQ. However with the current system, this process still requires the intervention of the Information Manager at DJFHQ when new document types are

introduced, or when the SOP definitions are revised. The next step is then to make the Classifier component trainable, and to allow the system to automatically learn the classification rules. The use of Self-Organising Map algorithm as used by Kohonen et al (2000) for massive document collections will be investigated for its application to automatic rule generation for the Rule-Based Classifier.

## 6.3 Integration with other techniques

### 6.3.1 XML

We are investigating the use of XML for document management and for interfacing with other information systems. The SOP-MRC could be used to create extra XML tags for the Lotus Notes database documents. Lotus began to introduce XML support in Domino version 5.02b. The Lotus XML Toolkit contains C++ and Java libraries to access Domino databases using the Domino eXtension Language (DXL) (Lotus, 2000). The current release of the XML Toolkit does not fully support all the Domino Design elements; however, the XML tag `<richtext>` is the only Domino Design element the SOP-MRC would be concerned with, and the extra XML tags would be placed in between the `<richtext>` start and end tags.

### 6.3.2 Acronym Management

In a project run in parallel with this one, an Australian Defence Force Acronym Management tool is currently being developed. This tool allows the many tens of thousands of acronyms and abbreviations used in the ADF to be stored in the same database as the documents. database in a categorized fashion. Therefore the information provided by the classification of the acronyms found in the documents could be used to limit the search space.

Another direction to investigate is the combination of different classifiers. Finally, a combination of all these techniques should allow us to build a robust system, and the operators of the Lotus Notes databases at DJFHQ will be provided with a search tool that increases their efficiency and reduces their workload.

## References

Eric Atwell, George Demetriou, John Hughes, Amanda Schiffirin, Clive Souter, Sean Wilcock

- (2000). *A Comparative Evaluation of Modern English Corpus Grammatical Annotation Schemes*, Centre for Computer Analysis of Language and Speech, School of Computer Studies, University of Leeds, England, International Computer Archive of Modern and Medieval English Journal, Vol. 24., pp. 7-23, HIT Centre, Bergen.
- Thorsten Brants (1998). *TnT -- Statistical Part-of-Speech Tagging*, Universität des Saarlandes, Computational Linguistics, Saarbrücken, Germany, October. <http://www.coli.uni-sb.de/~thorsten/tn/>
- Michael Broughton, Linton Henderson, and Ahmad Hashemi-Sakhtsari (2002). *Query Building Interface to a Lotus Notes Command Support System Database*, DSTO Technical Report, DSTO-TR-1264, Electronics and Surveillance Research Laboratory, Defence Science and Technology Organisation, Department of Defence, Salisbury, Australia.
- Sion Camilleri (2002). *Document Classification and Information Retrieval Using Hidden Markov Models*, Masters Thesis, Department of Computer Science, University of Adelaide, to be published.
- P. Fournery and U. Sorensen (2000). *Lotus White Paper on COTS for Military Crisis Applications*, Information Systems Technology Symposium on Commercial Off-the-Shelf Products in Defence Applications – The Ruthless Pursuit of COTS, RTO MP-48, Research and Technology Organisation, NATO, Belgium, April. <http://www.rta.nato.int/Activities.asp?Panel=IST&pg=2>
- Barbara Greene and Gerald Rubin (1981). *Automatic Grammatical Tagging of English*, Providence, R.I., Department of Linguistics, Brown University.
- IBM (2000). *Lotus XML Toolkit Release 1.0*, Lotus Development Corp., San Francisco, USA. <http://www.lotus.com/xml>
- Peter Jackson and Isabelle Moulinier (2002). *Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization*, Natural Language Processing, Vol. 5., John Benjamins Publishing Company.
- Teuvo Kohonen, Samuel Kaski, Krista Lagus, Jarkko Salojärvi, Jukka Honkela, Vesa Paatero and Antti Saarela (2000). *Self Organisation of a Massive Document Collection*, Neural Networks Research Centre, Helsinki University of Technology, Espoo, Finland, IEEE Transactions on Neural Networks, Vol. 11, No. 3, May.
- David Lewis (2002). *Machine Learning for Text Classification Applications*, Tutorial of the 40<sup>th</sup> Anniversary Meeting of the Association of Computational Linguistics, University of Pennsylvania, Philadelphia, PA, USA. <http://www.DavidLewis.com>
- Lotus (2000). *Domino eXtension Language*, Lotus Developer Domain, <http://www.notes.net/dxl>
- Mark Lutz (2001). *Programming Python*, 2<sup>nd</sup> Edition, O'Reilly & Associates, Inc. <http://www.oreilly.com/catalog/python2/>
- Tony Maple (2001). *Lotus & Defence – Lotus Software in the Australian Defence Organisation*, Lotus Products in Defence, IBM, Australia, August.
- Tony Patton (2000). *Lotus's XML Plans*, Article ID 54, TIS Worldwide, July 2000. <http://www.e-promag.com>
- Dan Tufis and Oliver Mason (1998). *Tagging Romanian Texts: a Case Study for QTAG, a Language Independent Probabilistic Tagger*, Proceedings of the First International Conference on Language Resources & Evaluation (LREC), Granada, Spain, 28-30 May 1998, p.589-596. [http://www.racai.ro/~tufis/Selected\\_Papers/LREC98-TT.pdf](http://www.racai.ro/~tufis/Selected_Papers/LREC98-TT.pdf)