Automatic Recognition of Generic Structure: Medical Discharge Notices

Maarten Van Mol & Mick O'Donnell
Language and Computing

1 Aims[1]

Written and spoken discourse is usually functionally structured, in that the discourse is composed of a number of stages, each serving a distinct function (or functions) towards the purpose of the discourse as a whole. A discourse is not an unordered succession of a number of data, but it is rather a structured whole, in which all information comes at a certain functional stage. A typical example is the functional structure proposed by Labov & Waletzky (Labov & Waletzky 1967) for fairy tales, which usually conform to a functional structure similar to:

Orientation ^ Complication ^ (Evaluation) ^ Resolution ^ (Coda)

Systemic linguists use the term 'Generic Structure' (GS) to describe those functional structures of texts which re-occur often in the society. The same structure is re-used by different individuals, the GS representing a socially-shared way of getting a task done. This structure adds to the meaning of the text, in that the way we interpret text depends on what stage of the text it is in (what function it is serving).

Because GSs are socially shared, readers are exposed to multiple instances of the GS, and learn to recognise their structure. Their reading effort is thus simplified – they know the structure of the text (to some extent) before they read, and have some idea as to what each section will try to do (its generic function). The more that a text conforms to a GS, the more this will be true.

If a human can use knowledge of GS to help them understand texts, then possibly computers can also. We are involved in a project involving the automatic semantic interpretation of medical texts by computers. We believe such automatic text understanding will be facilitated if the program knows what each particular section of a text is trying to do. Towards this end, we wish to provide our programs with knowledge of a particular GS, and the ability to recognise the various stages of the GS in real texts.

This paper will report on our experiments in automatic recognition of GS by computers. We will focus on one particular text type, that of a medical discharge notice (MDN), which will be described in section 2 below. Section 3 will outline the GS of MDNs. Section 4 will outline a methodology for automatically recognising the GS of texts, and section 5 will show results in applying this methodology to MDNs. We will then offer conclusions in section 6.

[1] We would like to thank our colleague Frederik Coppens, whose lexicon tool and segmenter/tagger have made this work possible.

2 What are Medical Discharge Notices?

A MDN is a letter written by one doctor to another, concerning the treatment of a patient in the hospital or by a private doctor. When the patient leaves a hospital, the attending MD has to write a report about the patient's stay in the hospital. Similarly, when a patient consults with an MD other than his GP, the MD usually writes a report about the consultation to the GP.

The MNN is either typed by the MD himself, or recorded on a dictaphone and then typed by a secretary. Sometimes the MD re-reads the MDNs typed by the secretary, sometimes they do not. The MDN thus is a document written in natural language as opposed to a formalised format. MDNs are usually typed on a computer, and are thus available for automatic processing.

To give you some idea of what MDNs look like, see Figure 1. As it is only intended to give a general idea, this letter has been abbreviated.

Opening-Words	Dear colleague,
Patient-Data	Regarding : Your patient Mary Smith, Born 11/11/1975, Address: 35 Whitlam Square, Fraserville
Admission-data	This patient presented at our consultation of January 6th 1993.
Antecedents	ANTECEDENTS : a pneumonia at the age of 12. Recurrent sinusitis.
Present-admission	PRESENT PROBLEMS : since August, the patient has been suffering recurrent common colds with a dry cough. … PHYSICAL EXAMINATION : lung auscultation reveals a normal vesicular breathing sound. The cardiac auscultation … CHEST X-RAY : normally clear lung fields, without pleuro-pulmonary lesions suspicious of evolution. … LABORATORY DATA : cf. enclosures. We draw the attention to a raised sedimentation. ALLERGOLOGIC INVESTIGATION OF THE SKIN : negative concerning the current inhalation allergens.
Conclusion	CONCLUSION : patient suffers a chronic dry cough, without a clear conclusion of pulmonary abnormalities, and without signs of a specific hypersensitivity. Probably ...
Therapy-proposal	AS A PRACTICAL ATTITUDE WE PROPOSE : * At first carry through a treatment with Lomudal spray 5 mg 4 X 2 deep inhalations each day. ...
Follow-up	A polyclinical re-evaluation is being planned on February 5th 1993 at 9h20 am.
Greeting	With fraternal greetings, John Smith

Figure 1: Example Medical Discharge Notice

Automatic processing of MDNs is made more difficult because they are usually produced in a rushed manner: writing or dictating an MDN is an administrative, non-medical task, and MDs try to minimise the time spent on them. MDNS thus frequently contain mistakes. Short sentences and fragments are quite common. When the MDN is typed by the secretary, misinterpretations can be added on top of this. On the other hand, because MDs and secretaries do not wish to waste time on them, the MDN format is usually fairly uniform, which makes it easier to process the texts automatically.

3 Generic Structure of Medical Discharge Notices

Before we explore means of automatically recognising generic structure, we will first outline the generic structure of our target genre: MDNs. As a sub-genre of the letter genre the MDN has some features in common with other letters, but it also has some distinct features.

3.1 MDNs in relation to other Genres

The MDN is a sub-genre of the 'Letter' genre, which can be subdivided in 'Email' and 'Snail-mail'. Email is typically briefer, more to the point, and less formal than snail-mail. Because email is often produced in a rushed manner, it often contains more typing mistakes, and more fragments and syntactically incorrect clauses. Email shares this feature with MDNs.

Snail-mail consists of different sub-genres: a first distinction can be made between personal and professional letters, the latter being further sub-divided between business letters, medical letters, etc.

MDNs are one particular type of medical letters, in contrast to letters to advisory physicians of an insurance company or of the reimbursement division of the health insurance department. The last two letter-types have different GS from MDNs as they serve a completely different purpose. For instance, letters for drug reimbursement often consist of just a brief note asking for reimbursement, sometimes with arguments, such as results of technical investigations.

Not all MDNs follow the prototypical generic structure that we will describe below: a report of a consultation may for example only contain a short note on the present disease, with the main information being that the patient has been referred to another physician. A MDN about a patient who has died will also not contain the prototypical generic stages.

The common structure of all letters consists at a minimum of some opening words (e.g., 'Dear Sir', 'Hi Peter'), a date, the body of the text, a greeting formula (e.g., Yours faithfully, See you!), and the name of the writer. This generally is the complete structure of personal letters.

Business letters usually have some additional stages: a from-statement (name and address of the writer), a to-statement (name and address of the addressee), reference, a concerning-statement, a signature, and an enclosures-statement. MDNs for the most part have the same generic stages as business letters, except that they differ in the way that the stages are realised: e.g., wordings of formula may differ ('Dear Colleague' as opening words, 'With fraternal greetings' as greeting formula), or the content may differ (e.g., the 'concerning' statement in business letters contains a reference to what the letter is about, while in MDNs it contains the patient's name, age and address).

The ‘body’ element of a letter can itself be structured. However, the generic structure of the body varies across the sub-genres, as it is very purpose-dependent. In personal letters, the body structure is relatively free, although there may be minimal recurring structure (e.g., responding to last letter, then providing new news, then asking questions).

The structure of the body of MDNs also reflects its purpose: it reflects the way in which a medical consultation unfolds: the physician starts by asking the patient’s personal data, then looks at his antecedents (medical history). He then elicits the present complaint. This is followed by a clinical examination and physical investigations. The physician draws his conclusions and starts planning a therapy. The structural formula is thus chronologically iconic of the way of acting and thinking of the physician. The structure is thus a functional structure. It is also recurrent across different medical institutions, which makes it a generic structure.

3.2 The Structure of MDNs

To build our GSP for MDNs, we examined a random selection of MDNs drawn from a corpus of 35,000 MDNs in the field of pneumology. Note that the corpus is in Dutch, although examples thoughout this paper have been translated to English. We then checked our resulting structural formula against MDNs of other medical disciplines, and found that it in general holds true, with the exception that the more specific sub-sections (types of clinical and technical investigations) seem to be field specific: a pneumologist performs different tests that a neurologist, for example.

The resulting top-level structural formula is as follows:

```
(Opening-Words) ^ Patient-Data ^ (Admission-data)
    ^(Antecedents) ^ (Present-admission) ^ (Conclusion)
    ^(Therapy-proposal) ^ (Follow-up) ^ (Greeting)
    ^(Enclosures)
```

At this level, all of the sections appear (with very few exceptions) in a fixed order. However, every section, except for Patient-Data, is optional, meaning that many instantial structures are possible. In fact, the MDNs we have looked at do mostly exhibit different instantial structures, but they all fit this generic formula.

The structure above is, however, only the most general level of structure: several of these sections can in turn be broken into sub-structure. For example the Present-Admission section can consist of several further parts:

$$\text{(Anamnesis)} \;\hat{}\; \begin{array}{c}\text{(Clinical-}\\ \text{Examination)}\end{array} \;\hat{}\; \begin{array}{c}\text{Technical-}\\ \text{Investigations}\end{array} \;\hat{}\left\{\begin{array}{c}\text{Evolution}\\ \text{Therapy}\\ \text{-}\end{array}\right\}$$

In other words, the Present-Admission section is realised by an obligatory Technical-Investigations section, possibly with a prior Anamnesis and Clinical-Examination. The Technical-Investigations section can be followed by either Evolution or Therapy, both optional.

Figure 2 shows more fully part of the structural formula of MDNs, with layering.

OPENING WORDS
PATIENTDATA
ADMISSION DATA
ANTECEDENTS
Personal antecedents
Family antecedents
PRESENT ADMISSION
Anamnesis
Clinical examination
Clinical abdominal examination
Clinical cardiac examination
Clinical and technical cardiac investigation
Clinical cardiac third party investigation
Clinical and technical cardiac third party investigation
Technical investigation
Technical cardiac investigation
Clinical and technical cardiac investigation
Third party investigation
Cardiac third party investigation
Clinical cardiac third party investigation

Figure 2: Partial Structural Formula of MDNs

3.3 The use of GS in Medical Informatics

We have chosen the MDN as subject of research because it is a central point of reference in the whole of patient information documents: they are a means of communication between physicians; they are the basis on which a medical file is created or updated; they are the basis for charging costs to the health service. Automation of these uses would result in a more efficient use of information, in a lot of time saving and in less administrative workload for the MDs, of which they often complain. Let us look at these different uses in more detail.

Using Natural Language Understanding to analyse MDNs could result in a uniformed exchange of data between doctors. Doctor A can type in his data in natural language – as a MDN or in a computerised format. The computer will then read the text and create a meaning representation, which can then be converted to the format used by doctor B. To map from one format to another requires that the machine can interpret the generic structure of the input document.

Similarly, information can be automatically extracted from MDNs to create a new medical file, or to add to an existing one. Knowing which section of the MDN contained patient data will help in this regard. Other applications follow similar lines: one could automatically extract proposed medication so as to order drugs in the hospital pharmacy, or to print drug prescriptions. Thus the recognition of the therapy proposal section is useful. Important here also is to ignore the mentions of drugs and doses given in the patient history.

Perhaps the main application is in regard to how state-supported hospitals charge the government for services rendered. The government reimburses hospitals in regard to the overall seriousness of the diseases treated, and the procedures followed. Each type of diagnosis, procedure and treatment has a corresponding code, called an ‘ICD code’, and the hospital communicates its diagnoses and treatments to the government as a list of ICD codes. Typically, there is a team in the hospital which receives MDNs, and extracts out the set of ICD codes inherent in the MDN. One part of our work is to automate this process – getting the computer to read in MDNs and extract out the relevant ICD codes. Our aim is to provide automatic coding with a competitive error margin compared to that of the specialised coding teams.

For automatic ICD-coding, it is important to distinguish between relevant and irrelevant information, as not all diagnoses and therapies mentioned in the MDN are relevant for the present admission and thus for the charging the government. Using GS information one can identify which diagnoses and therapies are relevant and should be coded, and which are not to be coded. Acute past diseases or surgeries performed in the past for example are not at all relevant for the present seriousness of disease rating. The relevant information comes in particular sections. In the antecedents, only chronic diseases are relevant, while past acute diseases and surgical treatments performed in the past are irrelevant. Diagnoses and therapies mentioned in the present admission section and in the conclusion are relevant for the ICD-coding, as they are about the present illness. Explicit identification of the GS can thus improves the accuracy of the coding.

Ability to recognise GS can also form the basis of Text-type recognition: the automatic recognition of the document type from its generic structure. While various medical documents might include similar content, it is largely the text structure that defines being a MDN.

4 Methodology for Generic Structure Recognition

A human reader with experience of MDNs will be able to quickly identify the different sections of a MDN. This ability stems from their past exposure to a number of these documents. Their knowledge of general letter structure will also help, as will also knowledge of medical examination procedure, as the structure of MDNs to some extend follows this procedure.

A reader's ability to recognise the function of each section of a text stems from two directions:

1. Their knowledge of the generic structure tells them what sections to expect, in what order, and which sections are more or less obligatory, and which optional.
2. Their experience with reading particular sections makes them familiar with features of the text which typically constitutes the section. Such features including typical content (indicated by lexis), typical syntactic patterns (fragments may be more common in some sections), or orthography (bulleted lists common in Therapy Proposals, etc.).

So, the reader uses both their expectation of section sequence, and also the ability to identify a section by the text which realises it. Both are necessary skills: without the first, a reader could process a text just as well if the sections were realised in a random order, which is not true. Without the second, we could not explain how a reader recognises text sequence where optional elements are absent, or elements occur in variant orderings.

The central thesis of this paper is that computers too can be trained to recognise generic structure. While computers do not 'think' as we do, they are quite good at spotting patterns in data. Our plan was thus to provide enough examples of a text structure to a computer, and set it looking for patterns which help it identify the text structure in other texts.

The problem of automatic GS recognition can be broken up into two steps:

1. **Recognising section boundaries**: knowing where one stage ends and another begins.
2. **Assigning section labels**: assigning the most likely section label to a section of text.

The first of these tasks is very important, but quite difficult. Often, in some written text-types, section beginnings are well-marked by use of section titles, with embedding of stages within stages indicated by sub-headings, etc.

However, even when well marked in this way, not all sections are marked by headings. For instance, in an academic paper, the introduction section often breaks up into several sections like:

```
Aim of this paper ^ Lacks in Prior Approaches ^
                                        Structure of this Paper
```

These sections are rarely marked by titles, but are clearly separate in terms of their content.

In MDNs, also, we occasionally have section headings, often with the heading corresponding to a section label we would provide (e.g., 'Therapy Proposal'). However, in many cases, there is no header to separate sections of the MDN. Sometimes, two or more sections may even co-exist in the same paragraph. Also, there is a certain degree of variability in the words used in titles -- different doctors may provide different titling to the same section. Titles can also be ambiguous: for instance, "Medical History" can be used for antecedents, for anamnesis, or for both together.

So section heading by itself is not a reliable means of recognising section boundaries. While there are methods to make reliable guesses where the section boundaries are (see, e.g., Choi 2000), we will not concern ourselves with this problem here, but explore this at a later date. We will thus concern ourselves with the second problem: assigning of section labels to sections. We will assume that the section boundaries of the texts are first indicated by hand.

Our methodology thus consisted of the following:

1. **Research Stage**: identify a number of text features which can help a computer categorise a given body of text as belonging to one generic section or another.
2. **Data Preparation**: prepare a 'training corpus', that is a number of randomly-chosen instances of the text-type, where each text has the various sections clearly labelled. For this purpose, we inserted by hand section labels such as "#patient_data" at the start of each section.
3. **Training Phase**: ask the computer to process the training corpus, looking for particular recurrent patterns which strongly correlate with each section label.
4. **Testing Phase**: provide some previously unseen texts to the computer, and ask it to correctly label each section.

The rest of this section will explore this methodology.

4.1 Research Stage

We looked at various aspects of text to see which would best serve automatic text classification of sections. These features can be situated on four levels: Semantics, Syntax, Lexis and Typography.

4.1.1 Semantics

We can talk of the 'register' of an MDN, the common register of the text as a whole, being a document with Field of medicine, Tenor, being from one doctor to another, and Mode being a written document, produced in somewhat of a rush. However, each section of an MDN is serving a somewhat different function within the text, and as such, there are registerial differences between these sections. For instance, while Opening-Words and Greetings are somewhat personal (although formulaic), the other sections tend to be impersonal. In regards to Field, the Present-Admission section talks more of medical procedures performed, while Therapy-Proposal may list drugs to be given, and Antecedents will talk about past diseases. Mode is constant throughout the text.

The differences in register are realised by different meanings being used in the text. The differences in Field will result in different experiential meanings (different configurations of things, processes and circumstances), and the differences in Tenor will result in differences in speech act (greetings vs. proposed-actions vs. informatives).

We are not yet at the point where we can perform automatic semantic analysis of text, although we are working on it. We thus decided to ignore semantic patterns at this stage. However, in the future, we will return to the use of semantics. Our company owns the largest medical ontology in the world, which is organised on top of the Generalised Upper Model (Bateman *et al.* 1994), a Systemic-based organisation of 'things', 'processes', and 'qualities'. The goal of our project is to parse text to this representation. This will then allow us to check whether certain process-types, circumstances, etc. are more common in certain stages of MDNs. The clinical examination section could for example typically contain various sub-types of medical examination processes. However, this is for the future.

4.1.2 Syntax

The patterns of semantics are realised partially through the syntactic patterns used, and partially through the lexis. Certain syntactic patterns are more or less indicative of certain sections. For instance, tense may vary between past-tense (for the case history), to present tense (current symptoms) to future tense (for treatment proposals). Fragments are also more common in some sections over others. Imperatives typically appear in the Therapy_proposal section. We explored a number of syntactic patterns to see which were more helpful in identifying section category of text.

4.1.3 Lexis

Certain words are more or less common in certain sections. Some are indexical - unique for a certain section - while others are only indicative – more common in certain sections.

To some degree, this correlation between words and sections relates to the use of formulas - some sections are realised formulaically, such as "Dear Colleague" in the Opening-Words section, and "With fraternal greetings" in the Greetings section.

Another explanation of the correlation is due to differences in Field. A section serves a particular generic purpose, and this may activate a particular Field. For instance, because the purpose of the Present-Admission section is to describe the medical procedures performed on the patient, the Field will focus on 'medical procedures'. As a result, words and phrases which express medical procedures will be strongly indicative of this section. Likewise, Therapy_proposal may list drugs to be given, and Antecedents will talk about past diseases.

Thus, while we cannot use Field itself as a means of identifying section titles, we can use lexis, which realises Field, as a partial replacement. Note however that exploring the correlation of individual words and phrases with sections is not a total replacement - a semantic approach would catch the fact that a number of words all experientially related should have more effect than individual words.

4.1.4 Typographical features

Typographical features are most useful to recognise section boundaries, e.g., new sections are sometimes started by a heading, indicated in bold or (in our texts) in upper-case letters. Headings are sometimes by themselves on a line, or sometimes at the start of a paragraph, followed by a colon. Paragraphs also sometimes separate different topics.

Some typographical features also correlate with section name. For instance, list structures indicated by bulleting are common in the therapy proposal section, in which the prescribed drugs are listed.

We have however decided to ignore typograpical features in our work, due to lack of time.

4.2 Methodology: Summary

We wish to replicate a human's ability to recognise the generic structure of a text by combining two abilities: a sense of what sections appear in what order, and an ability to recognise the function a body of text is serving just from its own contents.

We have chosen to ignore the problem of recognising section borders, by simply providing these by hand in our texts (both training and testing corpora).

The first step in our methodology is to identify linguistic features of a text which can help us label that text, ignoring its position in the text. To this end, we chose to ignore typography and semantics at this point, and focus on lexis and syntax, which are more accessible for machines.

The second step is to use the words and syntactic features of our training corpus to build a model which can help us label the sections of other MDNs. This will be explored in section 5.1 and 5.2 below.

The third step in our methodology involves using this ability to label sections from content in conjunction with a model of the sequence of generic elements. This will be explored in section 5.3 below.

5 Experiments in Automatic Recognition of Generic Structure

5.1 Sectional Labelling: Lexis

A word is indicative of a particular section if that word appears more frequently in that section than in the text as a whole. For instance, the word "year" is far more common in both Antecedents and Anamnesis sections, dealing with the past, than in other sections. Some words are also counter-indicative, in that they occur less often in a section than in normal text.

Single words by themselves cannot identify the label for a section. However, if we combine the levels of indication from all the words in the section, together they provide a good prediction of the section label.

For this purpose, we used the multinomial model, which, given the set of words within a section, assigns a probability for each of the section labels. We have used the formulas as set out in McCallum & Nigam (1998) for this task. Where they applied their formulas to identifying the class of a document based on its vocabulary, we have adapted their approach to classifying sections. The formulas for this work are rather complex, and possibly not of interest to this audience, so we will not detail them here. Refer to the source if interested.

5.1.1 Data Preparation

To prepare the corpus, we firstly stripped out all excess white spaces, and punctuation. Then, all numbers, dates, ratios, names and addresses (where automatically recognisable) were replaced with symbols, e.g., %date, %ratio, etc., as the exact number or date is not important to our study, only that one occurs. We then decapitalised all words, so that words at beginning of sentences would not be treated distinctly from other occurences (this might cause some confusion when a proper name shares a common spelling with another word, but this is a minor loss compared to that which it avoids).

5.1.2 Training Phase

In the training phase, our program reads in this stripped training corpus, and statistically analyses it. The result is a set of values, for each section, the probability of each word occurring in that section. For our training corpus of 40 texts, this process takes less than a second. This data is stored, and used in the testing phase.

5.1.3 Testing Phase

To test how well lexis alone can be used to identify a section's generic function (its label), we provided 10 MDNs, randomly selected, which were not included in our training corpus. They were however drawn from the same corpus of MDNs, which are all concerned with Pneumology. Each MDN had marks placed to indicate section boundaries, but not otherwise edited.

The statistical model built in the training phase is applied to each section in turn, producing a degree of probability of the section having a particular label. For instance, for section 11 of Text 1, the results were as follows (actual Label: Follow_up):

P(Follow_up)	95.69%
P(Admission_data)	4.11%
P(Therapy_proposal)	0. 14%
etc.	

Our program then chooses the prediction with the highest probability.

5.1.4 Results

Of the 110 sections of the test corpus, 102 were correctly labelled on the basis of their lexis alone. This is 92.7% accuracy, quite a strong result, and points to a reasonably strong indexicality between lexis and sections.

However, 8 sections were mislabelled. Some reasons for these errors are:

1. **Section not in Training corpus**: in one case, the section to be classified, Cranial_sinuses_x_ray, did not occur in the training corpus. There was thus no way it could be correctly labelled. Chest_x_ray was predicted, which partially overlaps in Field with Cranial_sinuses_x_ray (they both involve x-ray).
2. **Common Lexis**: sometimes two sections overlap in field, and thus share certain common terms. For instance, both Lung_function_investigation and Chest_x_ray involve investigation of the lung, but using different techniques. However, some terms are common to both sections, and it is possible a mistake can be made.
3. **'Summarising' type errors**: in some cases, a section is mislabelled because it uses lexis typically associated with another. For instance, the Antecedents section might include reports of past chest x-rays, thus being confused with a technical investigation. Also, a Conclusion section might summarise a relevant technical investigation and thus be similarly mislabelled.
4. **Authorship problems**: in one case the Doctor who wrote the MDN wrote a Conclusion, according to an independent MD, reads more like an Anamnesis. A human Doctor shown the text in isolation would make the same mistake.

Part of the problem here is that we are labelling sections free of position in the text. We hoped that sequence information would improve the results.

5.2 Section Labelling: Syntax

5.2.1 Data Preparation

We initially intended to use the Syntactic parser we are developing to syntactically tag the MDNs. However, while the parser works, it produces multiple alternative parses for each sentence (due to lexical and attachment ambiguity), and as such it was unusable for this task at present. We are currently working to fix this problem.

To get around this problem, we hand-tagged a 20 text training corpus with syntactic features that we thought might be important. We used a qualitative tagging tool, Systemic Coder[2] (O'Donnell 1996).

[2] Available free from: http://www.wagsoft.com/Coder/index.html.

The features we included in the coding include, among others (54 in all):

- fragment vs. clause;
- finite vs. nonfinite;
- active vs. passive;
- declarative vs. imperative vs. interrogative;
- past_tense vs. present_tense vs. future_tense vs. modal;
- topicalised_clause vs. nontopicalised_clause;

5.2.2 Training Phase

The training corpus of 20 texts, syntactically tagged with the tool, was then fed into the same tool which built the training model for lexis, except this time it was dealing with syntactic features, rather than words. The training model produced in this way thus tells us the probability of each feature in each section.

5.2.3 Testing Phase

We then tested the results on a test corpus consisting of 3 MDNs from the Pneumology domain. These texts were prepared in a similar manner as the training corpus.

5.2.4 Results

10 of the 25 sections (45%) were correctly labelled. This is a far worse result than that using lexis, but note the lexis trails used a corpus double the size.

We conclude that syntactic patterns are not as valuable as lexical patterns in identifying generic stage labels.

In a future study, we will experiment with using both lexis and syntax together to see if we can improve on the results for lexis by itself.

5.3 Incorporating Sequence Information

Sections 5.1 and 5.2 established that a computer can reasonably well provide section labels, just on the basis of the text itself, at least for this fairly formulaic genre. We will now explore if this ability can combine with knowledge of generic structure potential to produce better results.

To use this information we have to calculate with what probability each section can follow another section. The result is a flow chart as in Figure 3. For instance, at the start of a MDN, 95% of the times the first section is an Opening_words section, while in the other 5% of cases, this section is skipped, and the MDN opens directly with Patient_data. From there, various options are available, amounting to skipping various sections.

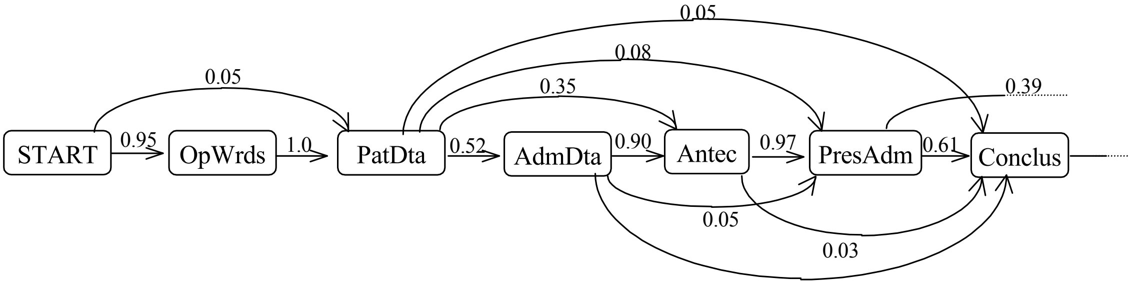

Figure 3: Section sequence

One of these sections, Present_admission, is realised by another graph such as this one, our generic model thus makes use of constituency in the generic structure. One of the elements of Present_admission, Technical_investigations, also has sub-structure.

Each path through the diagram assigns a probability to a particular sequence. For instance, The probability of the sequence: Patient_data ^ Antecedents ^ Present_admission is thus 0.05 * 0.35 * 0.90 = 0.01575, or 1.575%.

The section labeller of section 5.1, using just lexis, provides a probability for each label for each section of the text. Sometimes the label with the highest probability is actually unlikely if sequence is taken into account -- for instance, if we assigned the first section of a MDN the label "Anamnesis" we can see that this is not possible on the basis of sequence.

In an earlier version of our work we never assigned a zero possibility to any sequence, but allowed for sequences which were not in our training corpus with a probability of 0.01%. However, we found that our results improved when we removed this allowance. The exception here is for the technical investigations (Chest_x_ray, etc.). There is such a variety of these, and in fact, there is no clear ordering between them. They basically appear in a semi-random sequence. To allow for this possibility, we equalised the sequence probabilities for all technical investigations, such that if one technical investigation follows another element, then all technical investigation are possible in that position.

5.3.1 Data Preparation

The data as prepared for the Lexis trails was just reused. The text includes section tags, so our training program simply scans the file to extract the occurring section sequences.

5.3.2 Training Phase

During the training phase, the program looks at the training file, and extracts out the section labels of each document. The program keeps track of the range of sections which can follow any given section. These counts are then converted to percentages, as in figure 3.

We provide the system with some knowledge of constituency, e.g., a list of the sections which together constitute the section. For example, Present_admission is not tagged in our training set, only the sections which compose it. Our program takes this into account, and builds a sequence graph (such as in figure 3) for each level of the hierarchy.

As stated above, we treat the technical investigations differently, each individual technical investigation getting the same probability as each other of occurring in a particular position. See figure 4.

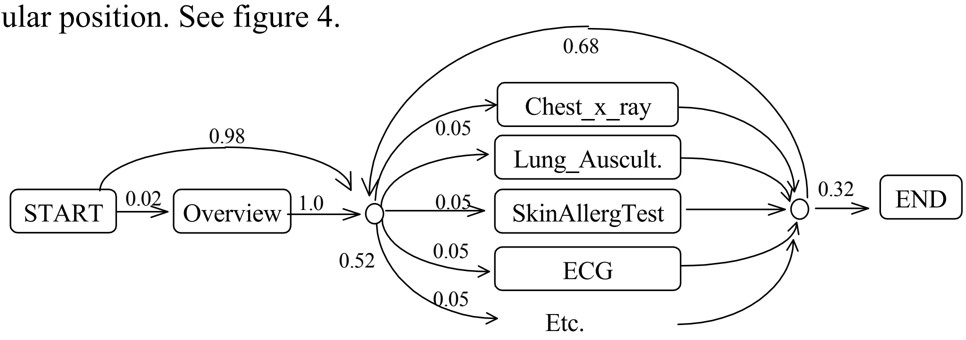

Figure 4: The Technical Investigations layer.

5.3.3 Testing Phase

In the testing phase, we started as in the classification with lexis trails. However, rather than just accepting the label with the highest probability, we combined the content-based probability with the sequence-based probability of each label. For instance, if the sequence data tells us that there is a 95% chance of starting a document with Opening_words, and a 5% chance of starting with Patient_data, and the word count gives us the following predictions for the first section:

```
Opening_words    0.34
Patient_data     0.55
Conclusion       0.11
```

...then our assessment of the document structure up to this point is:

```
START ^ Opening_words :      0.95 * 0.34 = 0.32
START ^ Patient_data :       0.05 * 0.55 = 0.0275
START ^ Conclusion :         0.00 * 0.11 = 0.00
```

As these probabilities do not ad up to 1.0 (100%), we need to normalise the data, and drop candidates with 0 probability, which produces:

```
START ^ Opening_words :      0.95 * 0.34 = 0.921
START ^ Patient_data :       0.05 * 0.55 = 0.079
```

We then read in the next section, assess its labelling based solely on words, and multiply out again. For example, the probability of the START ^ Opening_words path being continued with Patient_data is:

1) the probability of the path so far;
2) multiplied by the probability of the sequence Opening_words ^ Patient_data
3) multiplied by the probability of this section being Patient_data based on words alone.

At the end of a text, the path with the biggest probability is chosen.

5.3.4 Results

Applying this method to our pneumology testing corpus, we correctly labelled 106 out of 110 (96.4%) compared with 102 correct without sequence information. Clearly, the sequence information has helped the computer, and thus our hypothesis is verified.

Using lexical information and sequence information together, the program still makes four mistakes. The reasons for three of these were discussed in the Lexis section above: use of a section label not in the training corpus (1 case) and use of lexis more similar to another section (2 cases).

One other error was due to using sequence information: we are using only a relatively small training corpus (40 texts), and as such, some possible section sequences may not be covered. This was true in one case, where we had a Clinical_examination followed directly by a Conclusion, without any technical investigations. This did not occur in the training corpus, and thus our sequence rules did not allow for this. The program thus classified the Conclusion as a Chest_x_ray. Note that on lexical information alone, the program provided a 99.9996% probability of the section being a Conclusion.

As we stated for the lexis trails, a larger training corpus will fix these problems.

5.4 Applying the Generic Model to Related Genre

Our program was trained on MDNs from Pneumology. We were curious to see how well the model developed on this data would apply to MDNs from different Fields.

We prepared a test corpus of 10 MDNs from Neurologists. We then ran them through the Labeller program, using the training data from Pneumology. The results were somewhat disappointing:

Using just Lexis, we managed to correctly label 45 out of the 115 sections (39.1%). However, one needs to take into account that 36 of the sections were not in the training corpus (they are specific to the Field). If we consider only the sections it knew about, the performance was 45 out of 79, or 57%. This is still quite poor compared to the results for Pneumology.

Using Lexis and Sequence information together, the results get even worse. Only 31 of the 115 sections are correctly labelled. These Neurology MDNs vary slightly in structure from the Pneumology ones, and thus the model of GS for Pneumology did not always apply.

Most mistakes are made in labelling the Greeting section (all labelled as Antecedent) and in Patient_data (all labelled as Admission_data). The incorrect labelling of Greeting is due to the low occurrence of this section in the training data and to the completely different lexis used in the Pneumology training set than in the Neurology testing data ("Hopend U met dit schrijven van dienst te zijn geweest" versus "Met kollegiale groeten"). In the Patient_data of the testing data only "Uw patient" (your patient) is useful for classification based on lexis, but precisely these words are also typical of Admission_data.

We thus draw the conclusion that for a hospital to use our system to label their MDNs, they should either use specific training data for each distinct Field, or else base their training model on MDNs from a wide variety of Fields.

6 Conclusions

This paper has shown that the automatic recognition of generic structure is possible for at least one text-type, medical discharge notices. Our approach, however, has assumed that section boundaries are already indicated. Our next phase of work will attempt to label unsegmented texts.

We concluded that recognition of generic structure is best done using two sources of information i) knowledge of the sequence of generic elements, and ii) knowledge of how to recognise the function of text from its contents alone.

In regards to sequencing, we found that a mixture of human and automatic construction of the generic structure potential works best -- a human simply specifies constituency of elements, while the computer works out their probable order. The human needs also to specify where a number of sections should be unordered in respect to each other (as with the technical investigations).

In regards to labelling, we decided that use of semantics, while ideal, is not currently in a state to be automated. Typography was also ignored due to lack of time. We thus focused on lexis and syntax. We found that lexis by itself produces very good results, 92.7% correctness without sequence information. Syntax by itself was not so helpful. In future work, we will experiment with labelling on the basis of syntax and lexis combined.

Our hypothesis that best results would be obtained by using both sequence and internal criteria to label documents proved correct: we achieved a 96.4% level of accuracy, which we believe will improve if a larger training corpus is used.

We tested the robustness of our approach by using a testing corpus drawn from a different sub-genre of MDNs, but found our results quite poor. If labelling of MDNs from a number of fields is required, we would need to introduce more variety into our training corpus.

A point which remains to address is: how extendable is our methodology to other genres, either written or spoken? We leave this question for later work, but we guess that where there is a degree of correlation between lexis and field, and some regularity of generic sequence, then our results could prove useful.

We are currently incorporating the above technology into products of our company, which provides 'intelligent' text processing tools for medical applications.

7 References

Bateman, J. A., B. Magnini, and F. Rinaldi (1994) "The Generalized Italian, German, English Upper Model", *Proceedings of the ECAI94 Workshop: Comparison of Implemented Ontologies*, Amsterdam.

Choi, Freddy Y. Y. 2000. "Advances in domain independent linear text segmentation*"*, *Proceedings of NAACL'00*, Seattle, USA, April 2000. ACL.

Hasan, R. (1978): "Text in the systemic-functional model", in W. Dressler (eds.): *Current Trends in Text Linguistics*, Berlin, de Gruyter.

Labov, W. and J. Waletzky (1967) "Narrative analysis: Oral versions of personal experience", in Ed. Helm, J. (ed.): *Essays on the Verbal and Visual Arts*. San Francisco: American Ethnological Society.

Martin, J.R. (1992) *English Text: System and Structure*, Amsterdam, Benjamins.

McCallum, A. & K. Nigam. (1998): "A Comparison of Event Models for Naive Bayes Text Classification", *AAAI-98 Workshop on "Learning for Text Categorization".* (Also: http://www-2.cs.cmu.edu/~knigam/papers/multinomial-aaaiws98.pdf).

O'Donnell, Michael (1995) "From Corpus to Codings: Semi-Automating the Acquisition of Linguistic Features", *Proceedings of the AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, Stanford University, California, March 27 - 29.

Simone Teufel and Marc Moens (1999) "Discourse-level argumentation in scientific articles: human and automatic annotation", *Proceedings of the ACL '99 Workshop "Towards Standards and Tools for Discourse Tagging"*, pp84-93. June 22, 1999.