Layout and Language: Integrating Spatial and Linguistic Knowledge for Layout Understanding Tasks

Matthew Hurst and Tetsuya Nasukawa
IBM Research, Tokyo Research Laboratory

Abstract

Complex documents stored in a flat or partially marked up file format require layout sensitive preprocessing before any natural language processing can be carried out on their textual content. Contemporary technology for the discovery of basic textual units is based on either spatial or other content insensitive methods. However, there are many cases where knowledge of both the language and layout is required in order to establish the boundaries of the basic textual blocks. This paper describes a number of these cases and proposes the application of a general method combining knowledge about language with knowledge about the spatial arrangement of text. We claim that the comprehensive understanding of layout can only be achieved through the exploitation of layout knowledge and language knowledge in an inter-dependent manner.

1 Introduction

There is currently a significant amount of work being carried out on applications which aim to deduce layout information from a spatial description of a document. The tasks vary in detail, however they generally take as input a document description which presents areas of text (including titles, headings, paragraphs, lists and tables) marked implicitly by position. A simple example is a flat text document which uses white space to demonstrate alignment at the edges of textual blocks and blank lines to indicate vertical spatial cohesion and separation between blocks.[1]

Rus and Summers ((Rus and Summers, 1994)) state that *"the non-textual content of documents [complement] the textual content and should play an equal role"*. This is clearly desirable: textual and spatial properties, as described in this paper, are inter-related and it is in fact highly beneficial to exploit the relationships which exist between them. In algorithmic terms, this implies implementing solutions which use both spatial and linguistic features to detect coherent textual objects in the raw text. Approaches to the problem are limited to those exploiting spatial cohesion. There are two techniques for achieving this. The first looks for features of space, identifying rivers of space which run around text blocks in some meaningful manner. The second looks at non-linguistic qualities of the text including alignment of tokens between lines as well as certain types of global interactions (e.g. (Kieninger and Dengel, 1998)). Although this second type focuses on the characters rather than the spaces in the text, the features that it detects are implications of the spatial arrangement of the text: judging two words to be overlapping in the horizontal axis is not a feature of the words in terms of their content, but of their *position*. Elements of the above basic methods may be combined and, as with any feature vector type of analysis, machine learning algorithms may be applied (e.g. (Ng et al., 1999)).

2 A New Method

The methods based on spatial cohesion outlined above make assumptions about the application of layout to the textual content of the document in order to derive features indicating higher order structure. These assumptions rely on the realisation of layout as space and do not always hold (see, e.g., Figure 4: Grid Quantization), and may result in ambiguities. However, there is another source of information which can be exploited to recover layout.

Though layout imposes spatial effects, it has little or no effect on low level language phenomena within the distinct layout document objects: *we do not expect the layout of text to render it ungrammatical*.[2] Conversely, we do not expect grammaticality to persist in an incorrect interpretation of layout. For example, applying this observation to the segmentation of a double column of text will indicate

[1] The term spatial cohesion is motivated by the work on lexical cohesion by Morris and Hirst ((Morris and Hirst, 1991)). Text which is cohesive is text which has a *quality of unity* (p. 21). Objects which have spatial cohesion have a quality of unity *indicated by spatial features*; in the words of Morris and Hirst: they "stick together".

[2] It is clear that layout does has very definite consequences for the content of textual document elements, however those features we are concerned with here are below even this rudimentary level of language analysis.

the line breaks, see Figure 4: Double Columns.[3] The application of a low level language model to the interpretation of spatially distinct textual areas can be applied in many cases where a purely spatial algorithm may fail. The following is an incomplete list of possible cases of application (concrete examples may be found in Figure 4):

Multi Column Text When the columns are separated by only one space, a language model may be applied to determine if and where the blocks exist. These may be confused with **False Space Positives** where, by chance, the text formatting introduces streams of white space within contiguous text.

Apposed/Marginal Material Text which is offset from the main body of text, similarly to multi column text, will contain its own line breaks.

Unmarked Headers Headers may be unmarked and appear similar to single line paragraphs.

Double Spacing The introduction of more than one line of spacing within contiguous text causes ambiguities with paragraph separation, headers and so on.

Elliptical Lists When text continues through a layout device, a language model may be used to detect it.[4]

Short Paragraphs When a paragraphs is particularly short, the insertion of a line break may cause problems.

Another example, and a useful application, is that to the problem of table segmentation. Once a table has been located using this method or other methods, the cells must be located.

Multi-Column Cells A cell spans multiple columns. This may easily be confused with **Multi-Row Cells** where a cell contains more than one line and must be grouped according to the line breaks.

Elliptical Cell Contents Cells which form a disjunction of possible continuations to the content of another cell can be identified using a language model.

Grid Quantization When a plain text table contains cells which are not wholly aligned with other cells in the same grid row or column, it is difficult to associate the cells correctly.

Languages which permit vertical and horizontal orthography (such as Japanese) pose additional problems when extracting layout features from plain text data.

Orientation Detection With mixed orientation, a language model may be used to distinguish vertical and horizontal text blocks.[5]

We can hypothesise that spatially cohesive areas of the document are renderings of some underlying textual representation. If, at some level, the text is separated from the layout (the text is linearised by removing line breaks), then we may observe certain linguistic phenomena which are characteristic of the language. Reversing this allows us to identify the spatially cohesive objects in the document by discovering the transformation to the text (the application of layout, i.e. the insertion of spacing and line breaks) which preserve our observations about the language. One such observation is the ordering of words. Consequently, we can apply a language model to a line of text in a document to determine where line breaks have been inserted into the text for layout purposes by observing where the language model breaks down and where our simple notion of layout based on spatial features permits text block segmentation. This is an ideal. In fact, *knowledge of layout and language is required to overcome the short comings of each.*

There are many types of language model which may be applied to the problem being considered, ranging from the analytical - which provide an indication of linguistic structure), to the classifying - which indicate if (and to what extent) the input fits the model. The analytical, such as a context free grammar, are not appropriate for this problem as they require a broad input and are not suited to the fragments of input envisioned for this applications.

The prime purpose of the language model we wish to use is to provide some ranking of candidate continuations of a particular set of one or more tokens. A simple example is the bigram model. This uses frequency counts of pairs of words derived from a corpus. Although there are advantages and disadvantages to this model, it will serve as an example though other more sophisticated and reliable models may easily be applied.

[3] In Figure 4: Double Columns, we know, through the application of a language model, that there is a line break after `paragraph` as `a paragraph of text` is more likely than `a paragraph Applying`, and `Applying this of text is grammatically`.

[4] This bares similarities with a simple list, but the language is that of the textual list which uses functional words and punctuation to indicate disjunction.

[5] In Figure 4: Orientation Variation, the column of text on the left of the table is a vertically orientated label (`%w%m Ln5e7k2L`) whereas the remainder of the table is horizontally orientated. The apparent column on the right of the table is an artifact of the spacing and has no linguistic cohesion.

3 Basic Algorithm

The problem can be generally described in the following manner: given a set of objects distributed in a two dimensional grid, for each pair of objects, determine if they belong to the same cohesive set. The objects are tokens, or words, and the measure of cohesion is that one word follows from the other in accordance with the the nature of the language, the content of the document, and the idiom of the particular document element within which they may be contained *and* that the spatial model of the layout of the document permits cohesion. In summary, the cohesion is spatial and linguistic.

However, such a general description is not computationally sensible and the search space will be reduced if we consider the cases where we expect ambiguities to occur. This can be approached by recognising that when there is the potential for ambiguity there is often present some artifact which may well help identify the domain of the ambiguity: these are generally the markers of spatial cohesion; e.g., where there are double columns, we may also identify left justification. Consequently, for a given word in the the double column area, the ambiguity may be resolved by inspecting the word to the right, or the set of words which may be left justified with the line currently under inspection on the line below. Therefore, the application of the language model to the disambiguation problems mentioned above takes place between a small set of candidate continuation positions.

These continuation points are located as prescribed by the markers of the spatial layout of text. Consequently, any algorithm using linguistic knowledge must exploit layout knowledge in order to both arrive at an economic solution, and also to be robust to weaknesses in the language model. The general method described here relies on and determines both spatial and linguistic information in a tightly integrated manner. The algorithm falls in to the following broad steps:

1. detect potential for ambiguity.
2. compute the set of possible continuation points by using knowledge of spatial layout.
3. disambiguate using a combination of language and layout knowledge.

For example, the words marked with a clear box in Figure 2, upper, are those which, according to a naive spatial algorithm, are possibly in close proximity to the right edge of a text block. Having detected them, the possible continuation points, shaded boxes, are computed (here for a single word for illustration). A language model may then be applied to determine the most likely continuation.

Care must be taken when discovering equally likely continuations as opposed to a single most likely one. Figure 2, lower, contains two examples. The first illustrates the case when there is no continuation appropriate (there are three equally likely continuations; as none is the most likely, no continuation should be proposed). In the second example, a unique continuation is preferred. The general algorithm above provides annotation to the tokens in the document which may then be used to drive a text-block recognition algorithm.

Detecting the Potential for Ambiguity The potential for ambiguity occurs when a feature of the document is discovered which may indicate the immediate boundary of a text block. As we are dealing with the basic element of a token (or word), the potential for ambiguity may occur at the end of a word, or between any two words in a sequence on the line. However, we only need to consider those cases where a spatial algorithm may determine a block boundary (correctly or incorrectly). In order to do this we need a characterisation of a spatial algorithm in terms of the features it uses to determine text block boundaries.These are naturally related to space in the text, and so our algorithm will be concerned with the following three types of space: 1) Between words where there is a vertical river of white space which continues above and below according to some threshold; 2) Between words larger than a minimum amount of space; 3) At the right hand side of the document when no more tokens are found. These describe potential points for line break insertion into text and constitute a partial functional model of layout.

Computing the Set of Continuation Points The set of continuation points is computed according to the assumptions used to determine if there is the potential for ambiguity. The continuation point from a point of potential ambiguity are: 1) The next word to the right; 2) The first word on the next line; 3) All the continuation points on the next line which are to the left of the current word. These represent the complement to the above functional model of layout. Thus we have a model of layout which is intentionally over general as it uses local features which are ambiguous.

Disambiguation Disambiguation may be carried out in a number of ways depending on the extent required by the language model being employed. However, regardless of what range of history or lookahead is required by the language model, the process of disambiguation is not a simple matter of selecting the best possible continuation as proposed by the statistical or other elements of the language model. The interactions between layout and language require that a number of constraints be considered. These constraints model the ambiguities caused by

the layout and the language.

For any potential point of ambiguity, a single (or null) point of continuation must be found. And for any point of continuation, a single source of its history is required. If token A has potential continuation points X and Y, and token B has potential continuation points Y and Z, and the best continuation as predicted by the model for A is X and that for B is also X, then both A and B can not be succeeded by their respective best continuations. The selection of continuation points must be based on the set of possible continuation points for the connected graph in which a potential point of ambiguity occurs (see Figure 3). An additional constraint imposed by the layout of the text is that links representing continuation cannot cross. This constraint is a feature of the interaction between the spatial layout and the linguistic model.

3.1 Extensions

The above algorithm is not capable of capturing all types of continuation observed in the basic text blocks of certain document elements. Specifically, there is an implicit restriction on a unique continuation of the language through certain layout features. This may be called the one to one model of the interaction between layout and language. However, the less frequent, though equally important cases of one to many and many to one interactions must also be considered. In Figure 4: Many to One, examples of both are given. Significantly, these cases exists at the boundaries between basic textual components of large document objects (here tables). It is suggested, then, that the detection of equally likely continuation points may be used to detect boundaries where there is little or no spatial separation.[6]

3.2 Experimentation

In order to test the basic ideas described in this paper, a simple system was implemented. A corpus of documents was collected from the SEC archive (`www.sec.gov`). These documents are rich in various document elements including tables, lists and headers. The documents are essentially flat, though there is some amount of header information encoded in XML as well as a minimal amount of markup in the document body.

A simple bigram model of the language used was created. This was constructed partly from general texts (a corpus of English literature) of which it was assumed there was no complex content, and partly from the SEC documents.[7] A system was implemented which marked the potential points of ambiguity and the continuation points and then applied the cluster and selection algorithm to determine the presence of spatio-linguistically cohesive text blocks (see example output in Figure 1).

As yet, no formal evaluation of the implementation is available. It can be asserted, however, that the results obtained from this preliminary implementation indicate that the general method produces significant results, and that the basic notion of combining spatial and linguistic information for the determination of cohesive elements in a complex document is a powerful one.

Another experiment investigated the utility of the methods described in this paper. We wanted to determine how often ambiguities occurred and how important correct resolution was. Looking at the ambiguity in table stub cells - the ambiguity between multi-row cells and multiple cells below a header - resulted in some significant results. For a sample of 28 tables (1704 cells); in the 131 stub cells we found 68 examples of multi-row cells, and 35 of headers to multiple cells (note that these are not disjoint sets). Using the SEC bigram model, the cases were disambiguated by hand, resulting in a 74 % success rate. This simple investigation demonstrates that the disambiguation is required and that linguistic information can be applied successfully.

4 Conclusions

This paper has outlined a set of problems particular to the encoding of complex document elements in flat or partially marked up files. The application of a simple language model in conjunction with algorithms sensitive to the layout characteristics of the document elements in terms of spatial features is proposed as a general solution to these problems. The method relies on the persistence of the language in which the document is written in terms of the model used to recognize it.

In the future, we intend to apply this approach to the implementation of a general layout analysis preprocessor. An interesting feature of the interaction between the language model and the layout of the document is that the performance of a system is only sensitive to the quality of the language model at the points at which it interacts with the layout of the document. Consequently, a general purpose model built from a corpus of marked up documents may be used to determine a subset of the cohesive textblocks in a document. Those blocks may then be used to derive more language data, possibly specific to the document, and then the process repeated until no more interactions are left ambiguous.

[6] This begs a definition of equally likely - which would be dependent on the language model and implementation.

[7] An important process in the creation of a language model for layout problems is the identification of usable language in the corpus. To these ends, the SEC documents were marked up by hand to identify paragraph text. These text blocks were then used for the creation of a simple bigram model.

References

T. Kieninger and Andreas Dengel. 1998. A paper-to-html table converting system. In *Proceedings of Document Analysis Systems (DAS) 98*, Nagano, Japan, November.

Jane Morris and Graeme Hirst. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Liguistics*.

Hwee Tou Ng, Chung Yong Lim, and Jessica Li Teng Koo. 1999. Learning to recognize tables in free text. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 443–450, Maryland, USA, June.

Daniela Rus and Kristen Summers. 1994. Using white space for automated document structuring. Technical Report TR94-1452, Cornell University, Department of Computer Science, July.

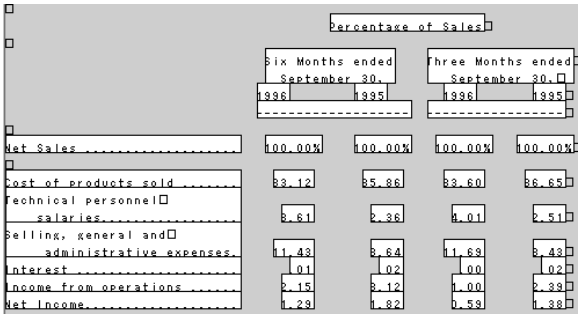

Figure 1: Example portion of output from prototype system

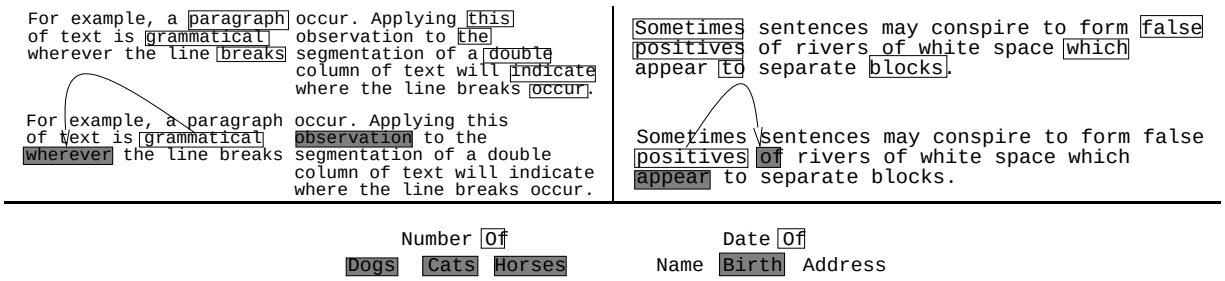

Figure 2: Locating Potential Ambiguity and Computing Continuation Points

If a bigram model is used, the probability that word w is followed by word w' may be expressed as a probability as $p(w' \mid w)$ and assigned a value between 0 and 1. If the probabilities are those shown in to the right then the continuation for A would be X and the continuation point for B would be Y.

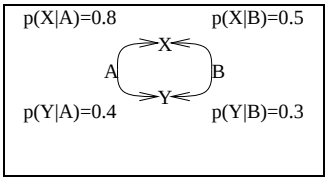

Figure 3: Sorting continuation depends on the potential layout of the document

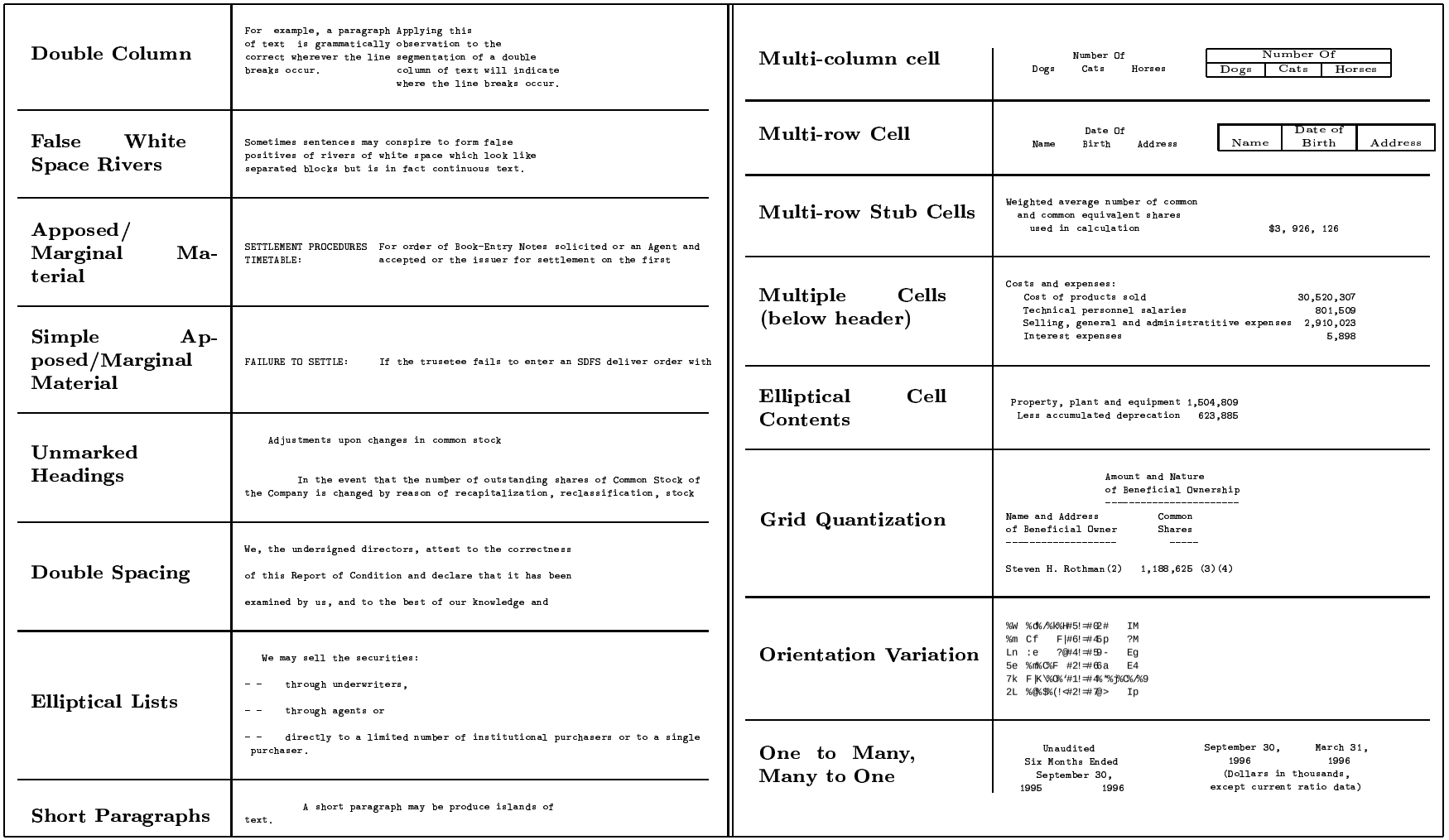

Figure 4: Layout and Language Effects