An Annotated Corpus and Method for Analysis of Ad-Hoc Structures Embedded in Text

Eric Yeh, John Niekrasz, Dayne Freitag, Richard Rohwer
SRI International,
yeh@ai.sri.com,niekrasz@ai.sri.com,freitag@ai.sri.com,rohwer@ai.sri.com

Abstract

We describe a method for identifying and performing functional analysis of structured regions that are embedded in natural language documents, such as tables or key-value lists. Such regions often encode information according to ad hoc schemas and avail themselves of visual cues in place of natural language grammar, presenting problems for standard information extraction algorithms. Unlike previous work in table extraction, which assumes a relatively noiseless two-dimensional layout, our aim is to accommodate a wide variety of naturally occurring structure types. Our approach has three main parts. First, we collect and annotate a a diverse sample of "naturally" occurring structures from several sources. Second, we use probabilistic text segmentation techniques, featurized by skip bigrams over spatial and token category cues, to automatically identify contiguous regions of structured text that share a common schema. Finally, we identify the records and fields within each structured region using a combination of distributional similarity and sequence alignment methods, guided by minimal supervision in the form of a single annotated record. We evaluate the last two components individually, and conclude with a discussion of further work.

Keywords: table recognition, semistructured data, information extraction

1. Introduction

We address the problem of information extraction from structured regions embedded in unstructured documents. Whereas information extraction systems are generally designed to process sentential text, many important "unstructured" genres draw freely on a variety of structured idioms to communicate important information concisely, such as headings, lists, and tables. In general, a complete recovery of the information communicated in such genres requires that these structures be recognized and processed.

Although the extensive literature on table processing offers a number of algorithmic recipes, tables are only one important structuring idiom. Moreover, most such approaches make fairly strong assumptions about their inputs, and are therefore of limited general use. We are interested here in the general problem of embedded structure (in all its forms) and are in search of methods that can accommodate as wide a range of structuring phenomena as possible.

Consider the following excerpt from a law enforcement press release (we have changed names to preserve anonymity):

```
The six defendants arrested in the first
indictment are charged with....  They are:

  Don Juan Williams, a/k/a "Flav,"
    "Flava Flav," and "Flay,"age 32,
    of Laurel, Maryland;
  Andre T. Dupont, a/k/a "Dre," age 37,
    of Riverdale;

The indictment alleges that Williams
purchased large quantities ....
```

This is an example of a form that might be called a "loosely formatted two-dimensional sequence." Although it exhibits something of a tabular structure, its inter-cell and inter-row delimiters obey no positional regularity. This example also exemplifies the typical centrality of information communicated in structured idioms; often, such information represents the primary payload of a document.

Our target use case is that of a language engineer having to survey and develop an extraction strategy for a body of documents. Indeed, several of the authors have had to perform this task under tight time pressure, motivating a focus on rapid deployment against a variety of naturally occurring idioms. Two key desiderata arise from this use case: maximal generality and minimal supervision.

This paper offers three main contributions in support of these desiderata:

- We present a corpus of human-authored documents drawn from a number of sources, with annotations that identify and provide a functional analysis of embedded structures.
- We describe and evaluate a novel algorithm that identifies such structures, using a probabilistic segmentation algorithm over featurized lines of text.
- We describe and evaluate a novel lightly-supervised algorithm that identifies and aligns structure elements in a way that reflects functional roles.

2. Related Work

Work on the automated analysis of structured data intended for human consumption has focused primarily on *table recognition*, structured objects in which linear geometries reflect functional and semantic relations. Research in this area, which had its genesis in the document analysis community, is quite deep and is covered in several thorough surveys (Zanibbi et al., 2004; e Silva et al., 2006; Embley et al., 2006).

E Silva et al. (2006) place tables in a conceptual continuum between *lists* (repeating structures that lack vertical alignment) and *diagrams* (structures that use devices other than geometry, typically pictorial elements, to communicate key

relations). Our work can be viewed as investigating and elaborating lists, arguing for a practical treatment that unifies them with tables. We also study structuring idioms that are not lists, such as header-style structures, and that, to our knowledge, have not been studied.

Although we argue for a framing of tables as a special case of communicative written structure, much of the conceptual framework that the field of table recognition has developed applies to our more general class of structuring phenomena. Wang and Wood (1996) established an influential formalism that separates functional relations communicated by tables from the details of their presentation. Non-table structures can be subjected to the same analysis. Göbel et al. (2012) provide a helpful taxonomy of concerns that have been addressed in the work on table processing: *detection*, *segmentation*, and *interpretation*. Interpretation decomposes further into functional analysis (determining domain-independent relations among cells) and semantic interpretation. In common with this study, we consider only the concerns that can be treated in a domain-agnostic manner, i.e., all of the above concerns except semantic interpretation.

Much of the work on table or structural analytics explicitly targets specific formats, with PDF (Liu et al., 2007; Fang et al., ; Göbel et al., 2012; Klampfl et al., 2014) and HTML (Wang and Hu, 2002; Astrakhantsev et al., 2013) currently the most prominent and practically important formats. Both of these formats facilitate table recognition in particular ways, while posing certain technical challenges that are orthogonal to the problem of structure recognition as experienced by human readers. We note that most such formats can be converted to plain text—and often *are* processed in this fashion for the purpose of natural language processing. We argue therefore that a plain text treatment of the problem of structure analysis is both germane and practical.

Of course, ours is not the first format-neutral treatment of structure. Soderland (1999) introduces the notion of *semi-structured* data, by which was meant human written communication that is more condensed and telegraphic than sentential prose, typically involving certain stereotypical fields. The term "semi-structured" has since been applied to a wide variety of phenomena, thereby losing some of its usefulness. We do not address the processing of semi-structured data so much as *embedded* structured data.

Our work is distinct from work that targets purely structured data, data that is not mixed with unstructured data. A canonical domain is the bibliography, one that has generated a fair amount of technically relevant work (e.g., (Grenager et al., 2005; Poon and Domingos, 2007). Cortez et al. (2010b; 2011) are notable in this regard, proposing an approach using probabilistic finite automata and promising completely unsupervised extraction. However, *extraction* is an exercise in semantic interpretation, which imposes a penalty in the form of domain-specificity. Cortez et al. assume the existence of a library of purpose-built attribute detectors and reference data sources that are used to establish expectations about the constituency of extracted records. These are assumptions that are often reasonable to make, but they represent a barrier to deployment in new domains.

Finally, there is a fair amount of work on machine learning for table interpretation, work that assumes annotated data and pursues both functional analysis (Ng et al., 1999; Pinto et al., 2003; Fang et al., 2012) and semantic interpretation (Viola and Narasimhan, 2005; Govindaraju et al., 2013). Requirements for data annnotation, like the positing of a domain ontology or reference data set, limit generality and slow deployment to new domains. In constrast with such work, we imagine an application that detects, segments, and performs a functional analysis in a supervised fashion, relying on very light labeling to power semantic interpretation.

3. The Corpus

To acquire a broad empirical sample of structuring conventions, we collected a corpus of 227 documents downloaded from a variety of publicly accessible government websites. These consisted of a mix of pure text files, PDFs, and HTML pages. An assessment of the documents showed a variety of structured data types interspersed with regular text. For example, documents collected from the Bureau of Labor consisted of paragraph sized text descriptions, followed by lengthy tables, encoded in a variety of schemas. Conversely, "most wanted" documents consisted almost entirely of field-value pairs. Press releases consisted primarily of freetext, with structured information in the form of contact information in field-value pairs, and long list-like enumerations of properties. Many of these subcollections exhibited inconsistent adherence to any apparent structuring conventions, with missing data, ambiguous delimiters, etc. Following our use case, we retain the entire document, including any surrounding sentential text.

For this work, we focused on structures that are relatively dense in extractable information. We say that a particular region is *structured* if it conveys information extralinguistically (i.e., if subjecting it to a hypothetical perfect NLP engine would lose information), and if that information can be readily converted to propositional form. This formulation excludes elements of structure that have what we might call a *navigational* purpose (e.g., section headings).

Given this definition, any attempt to access the informational content of such structures is confronted with a series of technical challenges of increasing difficulty:

1. **Detection.** How do we distinguish structured fragments from other material in a document?
2. **External segmentation.** Which structured fragments in a document are properly part of the same structure?
3. **Internal segmentation.** What are the atomic elements of a given stuctured regions (i.e., the "cells")?
4. **Functional analysis.** How do these cells cohere horizontally (i.e., as rows or records) and vertically (i.e., as columns or properties)?
5. **Semantic interpretation.** What is the meaning of the various properties (columns) and of their assembly (i.e., how would we derive propositions from the data)?

One purpose of this study was to assemble and make available[1] a corpus that would allow us to begin to answer these questions empirically. Because the challenge as we have defined it is substantial, our empirical work to date has not touched on semantic interpretation.
We used Apache Tika[2] to convert any natively non-text documents to text files. We then manually generated stand-off annotations to capture the following kind of information:

- Structured regions. By policy, structured regions are unbroken sequences of lines, beginning and ending at line breaks.
- Cells, the putatively atomic elements out of which structured regions are built.
- Groups of cells corresponding to the same property (i.e., belonging to the same column, in the case of tables)
- Header cells, distinguished cells within a group intended to reflect the type, rather than the value, of a property.

Note that all of these annotations are subject to various practical subtleties not always apparent on first examination, especially across different structuring conventions. For example, although cells are expected to contain indivisible values, it can often be difficult to determine whether a textual fragment is divisible. In the example listed above, do we have one column containing a defendant's full name, or a column for each of the first and last name? Because the structure in question does not provide strong two-dimensional signals to answer this question, it is more difficult to answer precisely than if presented in a standard table.

4. Method

Our primary technical contribution in this paper is the development of a novel approach to identifying structured information embedded within natural language texts. Our approach treats each occurrence of a structured region independently, breaking the problem down into two parts. First, we identify the location of each region within the corpus of text documents. Second, for each region, we identify the records, cells, and cell groupings associated with its ad hoc structure.
Our presentation follows our two-part breakdown of the problem. In section 4.1., we describe generic text preprocessing that is a prerequisite to the implementation. Then, in section 4.2. we present our approach to structured region identification. The approach centers on the use of a probabilistic text segmentation algorithm. Input to the algorithm is a token-level representation of the text built from spatial and token class features. As output, it produces a segmentation of the text that groups contiguous lines into either structured or unstructured segments. Finally, in section 4.3., we present our approach to identifying each cell in the region, along with its field assignment (i.e., its "column"). To do this, we assume that one record in the region has been annotated manually. We then combine a distributional representation of tokens with a sequence alignment algorithm to infer cell boundaries and field assignments for the entire region.

4.1. Preprocessing

Tokenization is the first preprocessing step. The input text is split into a sequence of contiguous substrings using the regular expression `([A-Za-z]+|[0-9]+|(.)\1*)`. The resulting tokens are either maximal sequences of alphabetic characters (e.g., `[YouTube]` or `[com]`), maximal sequences of digits (e.g., `[05]`), or maximal repetitions of any single non-alphanumeric character (e.g., `[@@@]` or `[\n]`). All subsequent processing ignores individual characters, instead considering these tokens as the atomic elements of the text.
We then apply a battery of tagging algorithms, each of which assigns a label to certain token sequences. Table 1 lists some of the taggers along with example labeled token sequences, implemented primarily with fast regular expressions. In addition to those shown, we also tag filenames, hostnames, currency, decimal numbers, percentages, fractions, phone numbers, paths, URLs, units, and xml tags. Our approach favors recall over precision, and we use as many taggers from as wide a range of domains as possible. We also augment our tag inventory with parts of speech, as extracted by a maximum entropy tagger (Toutanova et al., 2003).

Tagger	Example seq.	Label
dates	`[01][/][01][/][14]`	DATE
surnames	`[Smith]`	SUR
given names	`[John][ ][Joseph]`	GIVEN
initials	`[J][.][ ][J][.]`	INIT
abbreviations	`[U][.][S][.]`	ABRV
acronyms	`[AARP]`	ACRO
english words	`[this]`	ENG
emails	`[mj][@][cox][.][net]`	EMAIL
parts of speech	`[computer]`	NN
capitalization	`[CamelCase]`	CAMEL
named entities	`[Michael][ ][Smith]`	PERSON
char class	`[0123]`	NUMER

Table 1: A partial list of taggers used, along with example identified token sequences.

Tags always label a contiguous sequence of tokens, which we represent as an interval of token offsets. For example, consider the token sequence `[G][.][ ][Washington]` as an example input text, with the first token `[G]` taking the offset 0. Tagging this text would produce a set of overlapping tags like those shown in Table 2. As implied, tokens may be labeled with multiple tags.

4.2. Structured Region Identification

The next step is to separate structured from unstructured regions, as illustrated in Figure 1. In this study, we considered

[1]The annotated corpus is available at *http://www.ai.sri.com/ yeh/lrec-structure*
[2]http://tika.apache.org

Tag ($\langle$LABEL, interval$\rangle$)	Meaning
$\langle$ALPHA, 0$\rangle$	alphabetic
$\langle$OTHER, 1$\rangle$	non-alphanumeric
$\langle$SPACE, 2$\rangle$	space
$\langle$ALPHA, 3$\rangle$	alphabetic
$\langle$ACAPS, 0$\rangle$	all caps
$\langle$ICAPS, 3$\rangle$	initial caps
$\langle$INIT, 0 .. 1$\rangle$	initials
$\langle$ABRV, 0 .. 1$\rangle$	abbreviation
$\langle$SURNAM, 3$\rangle$	surname
$\langle$CITY, 3$\rangle$	city
$\langle$PERS, 0 .. 3$\rangle$	person

Table 2: An example list of token sequence tags, each represented by a label and a token offset interval $\langle$LABEL, interval$\rangle$.

entire lines as being structured or unstructured, a simplification that sufficed for most documents, leaving sub-line structure to future work.

For each tagged token produced by the previous step, we record the associated set of *spatial skip bigrams*. These skip bigrams encode both the horizontal and vertical character offsets between a given anchor token and each of its target neighbors, and their tag assignments. The intent here is to simply capture how semantic categories of tokens are spatially arranged with each other.

As we have found that structured regions tended to both be contiguous and exhibit content and spatial regularities that differ significantly from unstructured text, we identified them using a labeled variant of a maximum-likelihood segmentation algorithm (Utiyama and Isahara, 2001). Our decision to base our approach upon this particular segmentation algorithm is rooted in a previous study that found it produced the most accurate and unbiased results in text segmentation (Niekrasz and Moore, 2010). For a given document, we aim to identify a segmentation and labeling per segment that maximizes the likelihood,

$$\arg\max_{S,L} \Pr(S, L|W) \quad = \quad \prod_{i=1}^{m} \frac{\Pr\left(W^i|L,S\right) \Pr\left(L\right) \Pr\left(S\right)}{\Pr\left(W\right)}$$

Here, S refers to a segmentation of the document and L refers to the labeling of each of those segments as structured or unstructured. S is a vector of integer pairs, each pair consisting of a start and stop sentence index that describes the segment. L is a corresponding vector that identifies each segment as being *structured* or *unstructured*. W^i represents the pool of spatial skip bigrams within the ith segment. The conditional $\Pr\left(W^i|L,S\right)$ expresses the probability of observing the collection of spatial skip bigrams within segment number i, as computed by treating the skip bigrams as being mutually independent and dependent only upon the segment label. In this model, labelings are treated as being i.i.d.

While prior work treated $\Pr(S)$ as a penalty on segment size, we found that a uniform $\Pr(S)$ yielded better performance. Our corpus includes a number of documents that have very large segments, drawing into question the validity of any fixed expectations about segment size.

4.3. Segmentation and Functional Analysis

The next step is to decompose structured regions into their primitive constituents (cells) and to determine the functional relations over these cells. Our process, for a given structured region, is illustrated in Figure 2. Using the set of overlapping tags for a given token in the region, we generated lattices describing that token's context. We then apply *information theoretic co-clustering* (Dhillon et al., 2003) to generate a distributed representation of these contexts, which are used to compute a similarity measure between tokens. We then solicit supervision in the form of a single seed sequence $A_{k,l} = (a_k, a_{k+1}, ..., a_{k+l})$ where each $a_i \in F$ represents the assignment of a token x_i to a member of the set of possible record fields F (e.g., "surname", "zipcode", or even "field 3"). The token similarity measure is used by a soft sequence aligner to match candidate tokens c_m in the rest of the structured region with the seed tokens a_k. Candidate tokens are then assigned the record fields of their seed tokens.

4.3.1. Tag Contextual Similarity

In order to establish a basis for computing soft sequence alignments, we assembled "tag lattices," the set of overlapping tags to the left and right of a token, as a source of information about a each token's context. The tags used are given in Table 2. Lattices are acyclic directed graphs where each tag is represented as a node and two nodes are connected by an edge if and only if they are contiguous in the text with one another.

We then traverse the lattice and recorded all possible unidirectional paths up to a specified maximum length (7 is used as a default). We accumulated context statistics for each token by considering all paths that start or end on a tag in which the token is included. For example, the token `[.]` from our example has (among others) the contexts (0, OTHER), (0, INITIAL), (-1, ALLCAPS), and (+2, SURNAM). Here, the numeral represents the length of the path and the sign represents whether the contextual tag occurs at the end or beginning of the path. The result is a co-ocurrence matrix where rows represent tokens and columns represent unique contexts, e.g. (+1, SURNAM).

Our soft sequence aligner required a measure of the similarity between two arbitrary tokens, which itself may be impacted by sparsity in the raw contextual features. To alleviate this, we induced a dense representation of these contexts by applying information theoretic co-clustering to the token to tag context co-ocurrence matrix.

The co-clustering procedure is described as follows. Let $X = \{x_1, x_2, ..., x_L\}$ represent the set of L tokens in the text. And let $Y = \{y_1, y_2, ..., y_{N_Y}\}$ represent the set of $N_Y = wN_T$ unique context features, where N_T is the number of unique context feature labels T, and w is the maximum path length used during lattice traversal. The co-occurrence matrix is thus a set of non-negative integers n_{x_i,y_j} for every pair of symbols (x_i, y_j) in $X \times Y$. The output of co-clustering is two partitions $X^* = \{x_1^*, x_2^*, ..., x_{N_{X^*}}^*\}$ and $Y^* = \{y_1^*, y_2^*, ..., y_{N_{Y^*}}^*\}$ of the

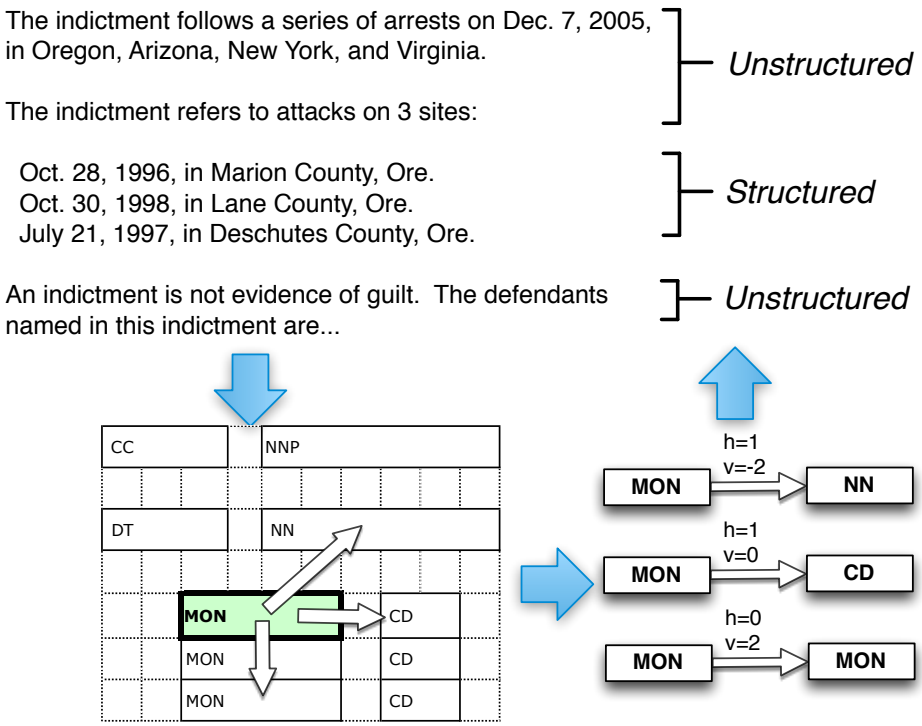

Figure 1: For a given document (upper left), we label each token by its word class, drawn from a mix of part-of-speech and word categories such as month indicators (lower left). For each token we encode its spatial skip bigrams, describing both the content and spatial arrangement with its neighbors (lower right). Using these features, we label each line as being governed by a *structured* schema, or as regular *unstructured* text (upper right).

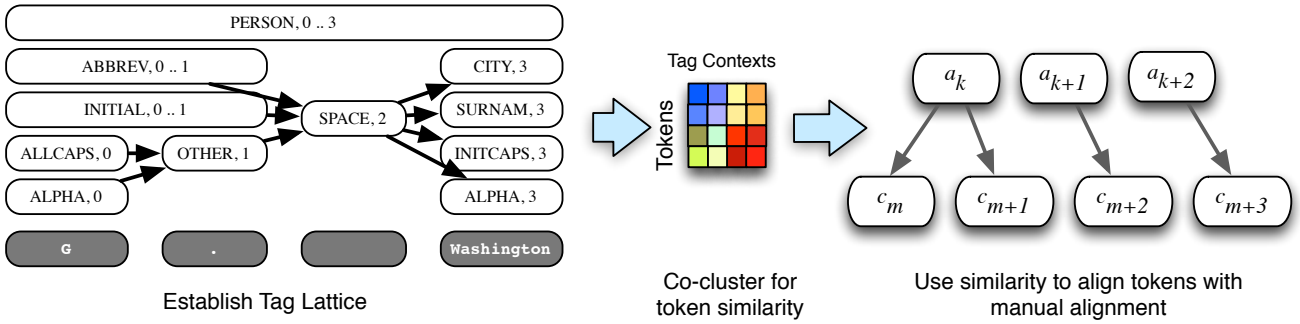

Figure 2: Overview of the alignment procedure. We first establish tag lattices, which capture the tag context for a given token. We then generate the basis for a contextual similarity measure by co-clustering the token and context co-occurrence matrix. This is then used to align tokens in the structured region with tokens in the annotated sequence.

sets X and Y. Co-clustering seeks to maximize the mutual information between a X^* and Y^* given constraints N_{X^*} and N_{Y^*}, making it a sensible method for compressing our context features while preserving as much information as possible distinguishing the types of tokens. Namely, it allows us to represent each token x_i in the text as a categorical distribution over N_{Y^*} clustered context features $p_{x_i} = (n_{x_i,y_1^*}, n_{x_i,y_2^*}, ..., n_{x_i,y_{N_{Y^*}}^*})$ where $n_{x_i,y_j^*} = \sum_{y_j \in y_j^*} n_{x_i,y_j}$. These vectors are the basis for measuring token similarity as discussed in the next section.

Figure 3 shows the result of applying our tagging and co-clustering steps to an example text. The colorization of the tokens indicates their assignment to a particular token cluster, which each token cluster x_i^* being assigned a unique color.

4.3.2. Functional Analysis

Our procedure for aligning tokens with record fields is illustrated in Figure 4. We first solicit a manually annotated seed representing the field assignments for a single record from the user. This is represented as a contiguous sequence

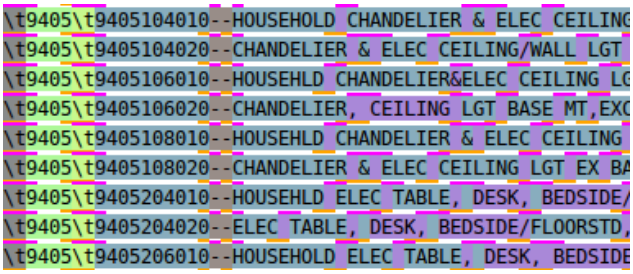

Figure 3: An example of token clustering performed on a text. Each token is assigned to a cluster. Each cluster is represented by a unique color. Magenta and orange bars indicate the first and last characters of each token.

of tokens from the structured region, with the field assignments for each token represented as an integer. The field values have no semantics other than to indicate which tokens belong to which field in the underlying schema.

We use the manually annotated seed record and apply sequence alignment and distributional similarity measures to identify other occurrences of records in the text. This is done by iterating a sliding window, of the same length l as the annotated sequence, through the entire text token by token. At each iteration $m \in (1, 2, ..., L - l + 1)$ (where L is the length of the text in tokens), the sequence $C_{m,l} = (p_{x_m}, p_{x_{m+i}}, ..., p_{x_{m+l}})$ of clustered context distributions is aligned with the sequence $C_{k,l}$ corresponding to the annotated record.

Alignment is performed using the Needleman-Wunsch sequence alignment algorithm (Needleman and Wunsch, 1970), where the score (penalty) for insertions and deletions is defined as -1 and the score for paired elements is defined as $1 - D(p_{m,l}, p_{k,l})$, where $D(p, q)$ is the Hellinger distance between distributions p and q. Alignment produces a set of pairs mapping each token in some subset of tokens in one sequence to a subset of tokens in the other. Tokens may remain unmapped, resulting in gaps. Pairings must also be sequential, so they cannot cross.

At each iteration of the window, if the resulting alignment score is positive, a field value is assigned from the annotated sequence $A_{k,l}$ to each of the tokens in the window that are part of an alignment pair. This is represented graphically in figure, by following the arrows from the top row to the bottom row.

Since each window iteration overlaps with the previous one, any single token may have multiple field values assigned to it. Therefore, we consider each of these assignments as a "vote" and chose the majority vote as the final field assignment for the token.

5. Evaluation and Results

Here we present the outcomes of the structured region detection and alignment phases.

5.1. Structured Region Detection

We evaluated the performance of our structured segmentation algorithm against several baselines. We measured performance on this task, which amounts to labeling each

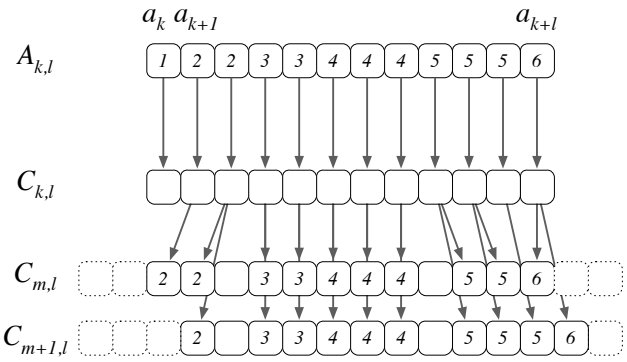

Figure 4: Example alignment between annotated tokens to candidate windows. The annotation maps field assignments $A_{k,l}$ to a sequence of tokens from the record, $C_{k,l}$. Two windows over the tokens in the region are given by $C_{m,l}$ and $C_{m+1,l}$. Each token is aligned with the annotated sequence $C_{k,l}$ according to the similarity score, and is given a field assignment based off the match in $C_{k,l}$. Final assignment is based on the majority vote. The alignment procedure permits gaps, displayed as unaligned and unassigned tokens.

Method	Macro F1
OddsRatio, Unigram	0
OddsRatio, Bigram	0
OddsRatio, SkipBigram	0.6868
MaxLikelihoodSeg	**0.8655**
PixEntropyLabeler	0.602

Table 3: Region Detection, showing macro F1 for structured regions computed using five-fold cross validation.

line as either *structured* or *unstructured*, in terms of F1 over structured regions. We employed five-fold cross validation over our corpus to evaluate each method, with results listing the macro F1 for detecting structured lines given in 3.

The methods compared were simple odds-ratio comparisons over unigrams and sequential bigrams of the tags, odds-ratio using the skip-bgrams, a simple "pixel" entropy measure, and the maximum likelihood segmentation strategy. The odds-ratio methods trained a simple "bag of words" model using the available training data, to score the likelihood of the line being structured or unstructured. The label itself was selected based upon the log-ratio of probability of the line being structured over unstructured. To enforce a light-supervision policy, all training folds were winnowed to just one percent of available data, selected at random. The pixel entropy measure is a visually motivated baseline that treated the line as binary pixel array, setting non-whitespace characters as 1, and all else as 0. A threshold for labeling was tuned over the pixel entropies in the training data.

As shown in Table 3, maximum likelihood segmentation over spatial skip bigrams dominates the baselines. We attribute the failure of the baselines using tag unigrams and bigrams to a paucity of training data enforced by the light-

	PMI	Tag	Distributional
Kappa	0.3902	0.5260	**0.5787**

Table 4: Alignment evaluation results, measuring macro kappa of per-token field agreement across the evaluted regions. Methods were the pointwise mutual information baseline, aligning using the token tags, and aligning using the distributed representation.

supervision regime.

5.2. Alignment Evaluation and Results

For a given structured region in our alignment experiments, the seed sequence representing the user annotation is selected at random from the ground truth label set, with the remaining used for evaluation. The task is then to produce field assignments that best match the gold field assignments for that region, leveraging that seed sequence.

We experimented with three methods for generating hypotheses for a given region: the soft sequence aligner using just the token tags, the previously described aligner with similarity derived from the Hellinger distance over the distributed representations, and a non-aligning pointwise mutual information baseline. The tag based aligner uses the same alignment procedure as the distributional method, except we use the token tags as a basis for similarity.

The pointwise mutual information method is a baseline method that used a clustering over tag types to determine token field assignments. We first performed a clustering over the tag lattices, setting the number of clusters to be a value larger than the number of possible field types. Each token in the seed annotation was given a cluster label, and then we mapped field values to each cluster label. The mapping that gave the best pointwise mutual information score with the original field assignments in the seed was then applied to each of the remaining tokens in the region.

Evaluation was made by comparing the kappas of the per-token field assignments against the gold assignments for that region. Results listing the macro kappas across evaluated structured regions are given for each method are given in Table 4, with the best alignments resulting from use of the distributional representation.

6. Discussion and Conclusions

In this paper we have described methods motivated by a need for rapid, format-neutral, broad-domain harvesting of information communicated in structured regions embedded in unstructured documents. We have approached this need empirically, assembling an annotated multi-source corpus, and devising unsupervised and lightly supervised algorithms for structured region detection and parsing. The methods we have presented impose very modest computational requirements, relying on a small library of efficiently executable token "taggers."

Observed algorithm accuracies leave some room for improvement. In particular, the macro-kappa of 0.6 that we measured in our functional analysis experiments bears further study. An obvious route to improvement is to loosen the rather stringent limitation of one labeled row that we imposed, and to study how performance increases with more training examples. Although we view a reliance on more supervision as an impediment to broad-scale use, there are use cases in which the labor investment might be warranted.

In particular, we treated every structured region as being independent of all others, when in reality structuring conventions are often repeated within documents and across collections. Thus, effort devoted to getting a single table right could be rewarded by accurate analysis across a whole collection. We leave multi-region and multi-document extensions of the algorithms presented here to future work.

We have adopted a one-size-fits-all approach to the recognition and parsing of structured regions, but in fact our solutions may require different parameterizations or variants for different structuring idioms. For example, we have largely represented these regions as sequences of lines, a representation in which the functional relations among cells depends on their position in a horizontal sequence, but is such a representation appropriate for structures resembling email headers? Presumably not, leading us to suppose that elaborations of our annotation in which distinct idioms are distinguished would be relevant. It may be, for example, that our sub-optimal performance in functional analysis arises from a mixing of conventions, hiding strong performance on certain idioms.

Another area of improvement would be integration of the kind of spatial-semantic information covered by spatial skip bigrams into the extraction step. Although we have found the this kind of information useful for identifying structured regions, how to properly apply it in an extraction system remains an open question.

Our work appears against a backdrop of related work that relies on reference data and a library of semantically suggestive features as a surrogate for human supervision (Cortez et al., 2010a). Obviously, the existence of such data and features greatly facilitates the exploitation of embedded structures. We have assumed that such information is absent in our study, but a potentially powerful middle way could lie in *induction* of reference tables directly from data (Agichtein and Ganti, 2004).

Acknowledgments. This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the Federal Bureau of Investigations, Finance Division. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

7. Bibliographical References

Agichtein, E. and Ganti, V. (2004). Mining reference tables for automatic text segmentation. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 20–29, New York, NY, USA. ACM.

Astrakhantsev, N., Turdakov, D., and Vassilieva, N. (2013). Semi-automatic Data Extraction from Tables. In *RCDL*, pages 14–20.

Cortez, E., da Silva, A. S., Gonçalves, M. A., and de Moura, E. S. (2010a). Ondux: On-demand unsupervised learning for information extraction. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, SIGMOD '10, pages 807–818, New York, NY, USA. ACM.

Cortez, E., da Silva, A. S., Gonalves, M. A., and de Moura, E. S. (2010b). Ondux: on-demand unsupervised learning for information extraction. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 807–818. ACM.

Cortez, E., Oliveira, D., da Silva, A. S., de Moura, E. S., and Laender, A. H. (2011). Joint unsupervised structure discovery and information extraction. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 541–552. ACM.

Dhillon, I. S., Mallela, S., and Modha, D. S. (2003). Information-theoretic co-clustering. In *Proceedings of The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD-2003)*, pages 89–98.

e Silva, A. C., Jorge, A. M., and Torgo, L. (2006). Design of an end-to-end method to extract information from tables. *International Journal of Document Analysis and Recognition (IJDAR)*, 8(2-3):144–171.

Embley, D. W., Hurst, M., Lopresti, D., and Nagy, G. (2006). Table-processing paradigms: a research survey. *International Journal of Document Analysis and Recognition (IJDAR)*, 8(2-3):66–86.

Fang, J., Gaoa, L., Bai, K., Qiu, R., Tao, X., and Tang, Z.). A Table Detection Method for Multipage PDF Documents via Visual Seperators and Tabular Structures. In *Document Analysis and Recognition, 2011., Proceedings of the 2011 International Conference on*.

Fang, J., Mitra, P., Tang, Z., and Giles, C. L. (2012). Table Header Detection and Classification. In *AAAI*.

Göbel, M., Hassan, T., Oro, E., and Orsi, G. (2012). A methodology for evaluating algorithms for table understanding in PDF documents. In *Proceedings of the 2012 ACM symposium on Document engineering*, pages 45–48. ACM.

Govindaraju, V., Zhang, C., and R, C. (2013). Understanding Tables in Context Using Standard NLP Toolkits. In *ACL (2)*, pages 658–664.

Grenager, T., Klein, D., and Manning, C. D. (2005). Unsupervised learning of field segmentation models for information extraction. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 371–378. Association for Computational Linguistics.

Klampfl, S., Jack, K., and Kern, R. (2014). A Comparison of Two Unsupervised Table Recognition Methods from Digital Scientific Articles. *D-Lib Magazine*, 20(11):7.

Liu, Y., Bai, K., Mitra, P., and Giles, C. L. (2007). Tableseer: automatic table metadata extraction and searching in digital libraries. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pages 91–100. ACM.

Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453.

Ng, H. T., Lim, C. Y., and Koo, J. L. T. (1999). Learning to recognize tables in free text. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 443–450. Association for Computational Linguistics.

Niekrasz, J. and Moore, J. D. (2010). Unbiased discourse segmentation evaluation. In Dilek Hakkani-Tr et al., editors, *Proceedings of SLT 2010*, pages 43–48.

Pinto, D., McCallum, A., Wei, X., and Croft, W. B. (2003). Table extraction using conditional random fields. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 235–242. ACM.

Poon, H. and Domingos, P. (2007). Joint inference in information extraction. In *AAAI*, volume 7, pages 913–918.

Soderland, S. (1999). Learning information extraction rules for semi-structured and free text. *Machine learning*, 34(1-3):233–272.

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *IN PROCEEDINGS OF HLT-NAACL*, pages 252–259.

Utiyama, M. and Isahara, H. (2001). A statistical model for domain-independent text segmentation. In *ACL*, pages 491–498.

Viola, P. and Narasimhan, M. (2005). Learning to extract information from semi-structured text using a discriminative context free grammar. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 330–337. ACM.

Wang, Y. and Hu, J. (2002). Detecting tables in html documents. In *Document Analysis Systems V*, pages 249–260. Springer.

Wang, X. and Wood, D. (1996). *Tabular abstraction, editing, and formatting*. Citeseer.

Zanibbi, R., Blostein, D., and Cordy, J. R. (2004). A survey of table recognition. *Document Analysis and Recognition*, 7(1):1–16.