An Approach towards Benchmarking of Table Structure Recognition Results

Thomas Kieninger
German Research Center for AI (DFKI)

Andreas Dengel
German Research Center for AI (DFKI),
CS Dept., University of Kaiserslautern, Germany

Abstract

After developing a model free table recognition system we wanted to tune parameters in order to optimize the recognition performance. Therefore we developed a benchmarking environment, including a user frontend to acquire ground truth and mechanisms to evaluate the quality of the recognition results. The tasks involved in the analysis systems were the locating of table regions, identification of cells and mapping of cells to rows and columns.

This paper presents our approach towards the comparison of recognition results with the ground truth. The established definitions of recall and precision did not meet our requirements, as we wanted to register even smallest improvements (or changes in general) in the results, even when both results were imperfect. We therefore extended the measures recall and precision in order to deal with recognition probabilities of objects rather than just with boolean values.

1. Introduction

In the recent years more and more researchers were addressing the topic of table analysis. This area yields a wide variety of facets and hence the early publications under this topic did not always describe comparable technologies. For instance, only a fraction of them was capable of locating a table within a document image and recognising the cells, rows and columns while others concentrated on higher level tasks such as the understanding or interpretation of the contents. A recent survey on table recognition approaches indicating the many facets and categorizing the different approaches is given in [1].

Once that a small fragment of tasks associated with that topic which was in common for two or more approaches was identified, it was still not possible to compare them against each other as there have yet not been any reference corpora or commonly agreed benchmarking approaches established.

Unlike for document analysis technologies that apply to non-tabular text (zoning, logical labeling, text categorization, content extraction etc.), there are little or no efforts towards a comparative evaluation of table recognizers. The reasons therefore are manifold: one might be, that only a little number of different table recognition approaches have been developed or published yet. These approaches themselves are not always to be compared. They either rely on different layout features (how can two systems be compared, if the one relies on table lines the other one relies on known column headers?) or they do not have the same input/output quality. But having an identical level of I/O-data is a prerequisite for any comparisons.

Benchmarking is a typicial activity when comparisons or quantitative statements about the quality of some analysis task are required. While for common Information Retrieval (IR) tasks, i.e. classification, a multitude of ground truth data and established measures [2] already exist, the field of table analysis is still under development. But even under the presence of sufficiently large document collections, table ground-thruthing has some problematic aspects as stated in [3].

The benefits of established benchmarking methods are manifold. First, it is possible to compare alternate approaches in a competitive way in order to find the best approach for a specific class of problems and/or a specific domain of documents. Return of Investment considerations heavily rely on benchmarking results in order to give measures to count on — a very important aspect when technology becomes mature and is about to be part of a product. Last but not least, the evolution and/or tuning of a specific approach wrt. parameter settings can strongly benefit, as different versions and parameter sets can automatically be compared to each other. Thus, benchmarking takes over the role of a supervisor or teacher while parameter sets are optimized.

Benchmarking itself is characterized by several subtasks: At first, one needs some *test-collections* which

ideally should be available to a larger community with no severe legal restrictions. Secondly, an adequate *representation* is needed in order to store the complete information that is produced from the analysis tasks — the ground truth data should be of the same (or a richer) quality and should as well be represented in that format. Third, an approriate *tool for acquiring ground truth data* is recommended. Such a tool should ideally consist of a GUI that displays the document and allows the definition of zones on top of the image. It it strongly advised to put much effort in such a user interface, as ground truth needs to be made by hand and consumes a lot of human ressources and thus is expensive. The fourth subtask is the *definition of appropriate measures* and *according algorithms* to evaluate these measures on the basis of ground truth and analysis results. Finally, it is desireable to have a suitable visualization of the comparison. Such a visual feedback of the comparison tells the developers more about the weaknesses of their approach (or a specific parameter setting) than just the plain values as it allows to precisely locate the errors made.

This paper concentrates on the explanation of a comparison measure that was developed in order to complement the development of the *T-Recs* and *T-Recs++* table recognizing systems. First results were given in [4] without explaining the methods and measures in detail.

2. Table-Specific Requirements

With the option to use benchmarking as a teacher for the training of optimized parameters, we encounter some specific requirements towards the measures and evaluation methods to be developed.

When comparing tables, it is not reasonable to make *true* or *false* decisions upon the recognition result as they are typical for e.g. benchmarking of classification results. If we would do so, only perfectly analyzed tables would yield a positive match with the ground truth while even smallest errors (e.g. missing assignment of a column-spanning cell to its different columns) result in a complete mismatch in the same way as a false spotting of the table boundary. Wrt. the intended learning of parameter sets, we are interested in much more distinguishing measures. Consider e.g. the results of two runs with different parameter settings for one document, both imperfect, we do like to know which of them has less errors.

With the above considerations in mind we decided to realize a bottom up approach. The main reason therefore is given by the nature of tables as they are aggregated objects with a complex structure. Tables consist of rows and/or columns and these in turn consist of cells. If the layout analysis fails at some point such as determining cells properly, these errors should be propagated to the next upper level aggregated objects. In order to account for this circumstance, it is neccessary to realize the evaluation methods as bottom-up approach.

Furthermore both systems *T-Recs* and *T-Recs++* that we implemented were designed as bottom-up approaches. They both start their computation on the same input data, namely word segments with their bounding box geometry. In order to best reflect our approach, which proceeds in consecutive steps, clustering word segments to higher level objects, we preferred a bottom-up approach for the benchmarking as well. If the benchmarking results drop significantly from one aggregation level to the next, this is an indicator for the developers to identify the responsible processing steps. Wrt. the automated learning, the responsible parameters can be isolated.

3. Representation Of Tables

A bottom-up design directly implies the choice of a tree structure as internal representation. But with tables there is another question to solve: Colums and rows are somewhat concurrent aggregations of cells. They both are made up of table cells as direct elements and build a table at the next aggregation level.

While the implementation of the benchmarking determines measures for both objects (columns *and* rows), we favoured the columns as relevant indicators when dealing with *T-Recs* and *T-Recs++* as this again reflects the analysis approach.

Thus, the internal representation of the tables carries redundant information for the sake of a hybrid evaluation of benchmarking results. The ground truth format itself solves the problem by specifying tables as a matrix of cells - a similar representation as proposed in [5].

The format is not only capable of storing sparse tables by using dummy identifiers for the empty cells but also allows the definition of row and/or column spanning cells.

The definition of ground truth data as well as the analysis runs are based upon the same input data, which is the output of some OCR (for scanned document images) or an ASCII-preprocessor (for plain text documents). The information items of relevance are the defined word segments and their bounding boxes.

4. Comparison Mechanisms

4.1. Word Segments

The comparisons are performed bottom-up, step by step. The word segments of both data sources ground truth (GT) and analysis result (AR) can be mapped based

on their position, as this is part of the input data to ground truth definition and analysis runs. We will refer to GT and AR as the two *representations* of a document. For the further explanations, we first like to define some notations that are used in the definition of the measures used:

Document: A document $\mathcal{D}$ consists of the hierarchical *word-* (W), *line-* (L) and *blocksegments* (B) such as occasionally their aggregations to *tables* (T). The objects of all levels are also characterized by their bounding box geometry. The respective coordinates will be refered to as $x_{0|1}[object]$ and $y_{0|1}[object]$ respectively.

Ground Truth: We refer to the human made Ground Truth representation as $\mathcal{G}$. Consequently, we denote the hierarchical objects as $\mathcal{G}^W$, $\mathcal{G}^L$, $\mathcal{G}^B$ and $\mathcal{G}^T$.

Analysis Result: In analogy to this we refer to the objects of the analysis results $\mathcal{H}$ (hypotheses) as $\mathcal{H}^W$, $\mathcal{H}^L$, $\mathcal{H}^B$ and $\mathcal{H}^T$.

Object Class: The function $class(object)$ returns for an arbitrary object of the document hierarchy its according level (W, L, B, T).

Document Version: The function $root(object)$ tells us, whether a given object belongs to the representation of the ground truth (GT) or to the analysis result (AR).

Overlapping Objects: In order to determine mutually overlapping objects (of any class) we define the relation $coinc^B(a, b)$ (coincide) that solely relies on the bounding box geometry as follows:

$$\begin{aligned} &coinc^B(a,b) \Leftrightarrow \\ &\quad (x_1[b] > x_0[a]) \quad \wedge \quad (x_0[b] < x_1[a]) \quad \wedge \\ &\quad (y_1[b] > y_0[a]) \quad \wedge \quad (y_0[b] < y_1[a]) \end{aligned}$$

Corresponding Word Segments: For each word segment a_i the corresponding word segment b_j of the complementary representation will be given through the following relation:

$$\begin{aligned} &\mathrm{corr\_word}(a_i, b_j) \Leftrightarrow \\ &\quad class(a_i) = W \quad \wedge \quad class(b_j) = W \ \wedge \\ &\quad root(a_i) \neq root(b_j) \quad \wedge \quad coinc^B(a_i, b_j) \end{aligned}$$

Existence of a Corresponding Word Segment: The existence is based on the fact that analysis and ground truth definition rely on the same data. We formaly denote this as follows:

$$\forall a_i \in \mathcal{G}^W \ \exists^1 b_j \in \mathcal{H}^W \ : \ \mathrm{corr\_word}(a_i, b_j)$$

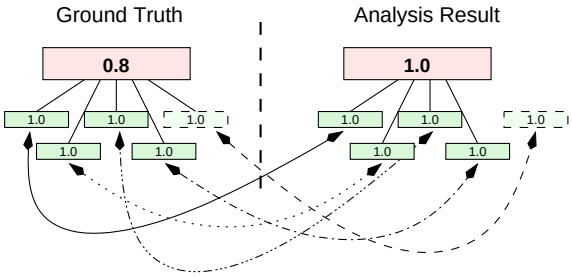

Figure 1. Recognition Probability of Corresponding Objects

Consequently, there exists a bijective transformation idx_map of the indices with:

$$\mathrm{corr\_word}(a_i, b_j) \ \Leftrightarrow \ j = idx\_map(i)$$

Index Transformation: This bijective transformation idx_map allows a rearrangement of the Word indices i of $b_i \in \mathcal{H}^W$ in a way that we can define:

$$b'_i := b_{idx\_map(i)} \Rightarrow \mathrm{corr\_word}(a_i, b'_i) \quad a_i \in \mathcal{G}^W$$

This allows the use of identical indices for corresponding word segments of different representations.

Recognition Probability: The probability $P^W(w)$ with which a word w can be assigned to a corresponding word of the other representation, can be defined as:

$$P^W(w) \ := \ \begin{cases} 1 & \text{if} \quad \exists^1 w' \ \mathrm{corr\_word}(w, w') \\ 0 & \text{otherwise} \end{cases}$$

Figure 1 gives an example of some corresponding word segments and the differing aggregation to line segments between GT and AR and the effect upon $P^L(x)$

4.2. Aggregated Objects

Comparing two objects wrt. geometry information is possible by means of the $coinc^B(a, b)$ relation for objects a and b of all hierarchy levels. As bounding boxes of aggregated objects are determined through the minima and maxima of the contained elements, identical aggregations within both representations GT and AR lead to identical bounding boxes of these objects.

At the other hand, identical bounding boxes do not imply identical elements of both objects, i.e. for all the elements that do not define maximum and minimum of

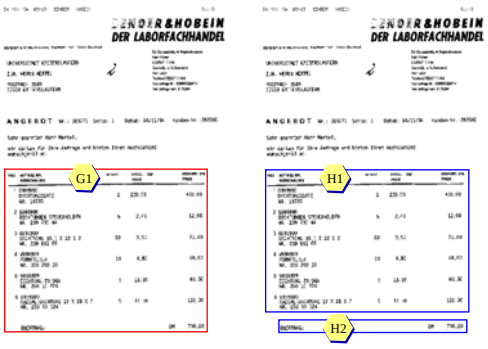

Figure 2. Regions differing between the GT and AR representation.

the set of bounding boxes, it cannot be decided whether or not they are part of the object.

For the identification of corresponding objects, we first make use of the $coinc^B(a,b)$ relation as a preselection. But a precise comparison requires a closer look at the contained elements in the representations.

Aggravating, several objects of one representation may share their contained elements with the same object of the other representation. This is the case, when the borders of an object could not be determined properly. An injective mapping of objects in between the representations is then not directly possible. Such an example is given in Figure 2, where the area $G1$ is shared by $H1$ and $H2$.

In order to achieve an injective mapping we determine the *best matching* counterpart to a given object which is the one that has the most corresponding elements in common (weighted by their recognition probability).

Wrt. the ability to train optimized parameters, it is desireable to have a more differenciating measure. Even marginal changes in the result should have an effect upon the benchmarking measure. Therefore we decided to follow the *best matching* approach.

Under the assumption that for all parts of an aggregated object the elements a_i themselves have at most one corresponding counterpart b'_i, we determine the corresponding higher level object as follows:

Overlapping Aggregated Objects: The binary relation $coinc^A$ defines for a given aggregated object all objects of the other representation which have at least one corresponding element in common [1].

$$coinc^A(a,b) \Leftrightarrow class(a) = class(b) \wedge root(a) \neq root(b) \wedge \exists v,w \ : \ \mathrm{corr\_word}(v,w) \wedge v \in^* a \wedge w \in^* b$$

[1]The symbol $\in^*$ denotes the transitive hull if the part-of relation.

Common Elements: As the elements a_i and b_j of two objects a and b that are to be compared have differend roots (GT or AR), the intersection cannot be processed right away. We assume, that in the next lower object-level the corresponding objects can be mapped through the injective function $idx\_map(i)$ [2]. The set of common elements for two aggregated objects can thus be described by means of their indices. For the following definition we assume to work with a mapped index ($b'_i := b_{idx\_map(i)}$):

$$idx\_set(a,b) := \{i \mid a_i \in a \wedge b'_i \in b \wedge \mathrm{corr\_obj}(a_i, b'_i)\}$$

Overlap Measure: There are two alternate ways to compute the degree of congruence for two aggregated objects a and b: First, sum up the products of the recognition probability of all common elements, or second, sum up their arithmetic mean:

$$sim^{mult}(a,b) := \sum_{i \in idx\_set(a,b)} P(a_i) \times P(b'_i)$$

$$sim^{add}(a,b) := \sum_{i \in idx\_set(a,b)} \frac{P(a_i)+P(b'_i)}{2}$$

Generally, we refer to the overlap measure as $sim(a,b)$. Which computation is used, has strong impacts towards the overall values of the defined measures: While sim^{mult} reacts very sensitive upon small changes of the recognition probability of underlying parts, sim^{add} is less sensitive but therefore yields higher values when working on identical data. Wrt. the learning of parameters, one might use sim^{add} first and once the results are getting better use sim^{mult} as it still unveils slightest improvements.

Corresponding Object: Here, we search for the best-match for a given object a. Consider the following expression:

$$\mathrm{corr\_obj}^{\mathrm{asym}}(a,b) \Leftrightarrow coinc^A(a,b) \wedge \neg\exists i \ : \ sim(a,i) > sim(a,b)$$

This relation is not symmetrical, as can easily be seen at the example of Figure 2: Here we would expect the relations $\mathrm{corr\_obj}^{\mathrm{asym}}(G1,H1)$ and $\mathrm{corr\_obj}^{\mathrm{asym}}(H1,G1)$ to be valid. Furthermore, $\mathrm{corr\_obj}^{\mathrm{asym}}(H2,G1)$ is given while $\mathrm{corr\_obj}^{\mathrm{asym}}(G1,H2)$ is not.

To obtain a bijective mapping, we need to assure symmetry. This is done by excluding the

[2]This is inductively given, as such a mapping is given for the word segments as the lowest level objects of our hierarchy.

case $\mathrm{corr\_obj}^{\mathrm{asym}}(H2, G1)$ for which the symmetric counterpart $\mathrm{corr\_obj}^{\mathrm{asym}}(G1, H2)$ does not exist. Therefore, we extend the above relation as follows:

$$\mathrm{corr\_obj}(a, b) \Leftrightarrow \mathrm{corr\_obj}^{\mathrm{asym}}(a, b) \wedge \mathrm{corr\_obj}^{\mathrm{asym}}(b, a)$$

4.3. Comparison of Aggregated Structures

For the evaluation of the overlap measure $sim(a, b)$ we require the recognition probability P, which is so far defined for corresponding word segments. Based on this value we will then define *table recall* and *table precision*.

Recognition Probability: Analog to the recognition probability of word segments $P^W(word)$, the function $P^O(x)$ will give a confidence measure for the proper mapping of the corresponding object for x. It is defined as the overlap measure to the corresponding object divided by the number of elements of x:

$$P^O(x) := \begin{cases} \frac{sim(x,y)}{|\{x_i\}|} & \text{if} \quad \exists y : \mathrm{corr\_obj}(x, y) \\ 0 & \text{else} \end{cases}$$

The values of P^W and P^O range inside the intervall $[0, 1]$. For P^W this is given by its definition.

For $sim(x, y)$ holds: the accumulated values themselves range within $[0, 1]$. Thus, $sim(x, y) >= 0$ and $sim(x, y) <= |idx\_set(x, y)|$ (the number of common elements). We can furthermore state that $\{x_i | i \in idx\_set(x, y)\} \subseteq \{x_i\}$ which leads to: $sim(x, y) <= |\{x_i\}|$ and consequently to $0 <= P^O(x) <= 1$.

Table Recall and Table Precision: To avoid any confusion with the original definitions as establisched in IR we like to name our functions *table recall* and *table precision* instead. Just as above, we simplify from the specific object-level O and define:

$$\text{Table Recall:} \quad \mathrm{TRcl}^O := \frac{\sum_{i \in \mathcal{G}^O} \mathrm{P}^O(i)}{|\{\mathcal{G}_i^O\}|}$$

$$\text{Table Precision:} \quad \mathrm{TPrec}^O := \frac{\sum_{i \in \mathcal{G}^O} \mathrm{P}^O(i)}{|\{\mathcal{H}_j^O\}|}$$

The original definitions of recall and precision determine the fraction of *correctly recognized objects* in relation to *all correct objects* (= recall) or in relation to *all recognized objects* (= precision). In order to depict why our definitions of TRcl and TPrec correspond to the original definitions we like to match the individual parts of the functions:

The cardinality of set $\mathcal{G}_i^O$ (number of objects of class O as defined in the ground truth) corresponds to the number of *all correct objects* in the definition of recall.

The cardinality of set $\mathcal{H}_j^O$ (number of objects of class O as recognized by the analysis procedure) corresponds to the number of *all recognized objects* in the definition of precision.

Finally, for all objects i of class O that are defined in the ground truth ($i \in \mathcal{G}^O$) we sum up their recognition probability $\mathrm{P}^O(i)$ which indicates to what degree of confidence this object has been correctly recognized. This corresponds to the number of *correctly recognized objects* in the definition of table recall and table precision.

5. Conclusion

The proposed approach was successfully applied to the results of our table recognition systems *T-Recs* and *T-Recs++* and we could use it to compare both systems and quantify the improvements. But it is neither limited to these specific systems nor to table recognition systems at all. In principle it can be applied to all IR tasks whoose results are represented as a hierarchical structure, no matter whether the analysis procedures follow a bottom-up approach or not. For those systems that are addressing table location and structure recognition, this might be a first attempt to establish mechanisms to compare systems with each other.

In our current implementation, ground truth needs to be defined based on the ouput of the used OCR (= input to *T-Recs*). In order to evaluate more complex systems (e.g. with an integrated OCR) ground truth should be defined on the images. This only requires a more sophisticated mapping procedure for the word segments to be applied. Such technologies have e.g. been successfully applied in the area of forms identification.

References

[1] Zanibbi et al., "A survey of table recognition: Models, observations, transformations, and inferences." in *IJDAR*, 2004.

[2] Lewis, "Evaluating and optimizing autonomous text classification systems," Special Issue of the SIGIR Forum, 1995.

[3] J. Hu et al., "Why table ground-truthing is hard." in *ICDAR*, 2001.

[4] Kieninger, Dengel, "Applying the T-Recs table recognition system to the business letter domain." in *ICDAR*, 2001.

[5] Arias et al., "Interpreting and representing tabular documents." in *CVPR*, 1996.