Extracting hierarchical data points and tables from scanned contracts

Jan Stadermann, Stephan Symons, and Ingo Thon

Recommind Inc., 650 California Street, San Francisco, CA 94108, United States
{jan.stadermann,stephan.symons,ingo.thon}@recommind.com
http://www.recommind.com

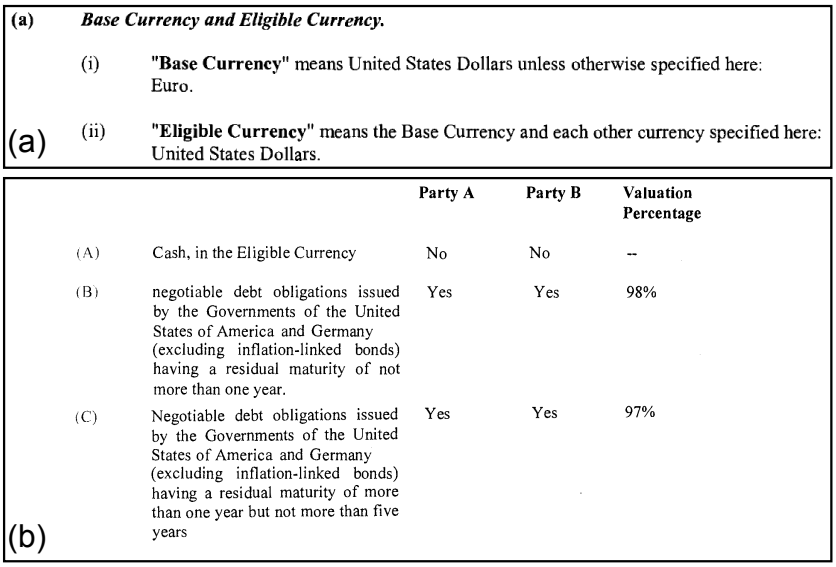
Abstract. We present a technique for developing systems to automatically extract information from scanned semi-structured contracts. Such contracts are based on a template, but have different layouts and client-specific changes. While the presented technique is applicable to all kinds of such contracts we specifically focus on so called *ISDA credit support annexes*. The data model for such documents consists of 150 individual entities some of which are tables that could span multiple pages. The information extraction is based on the Apache UIMA framework. It consists of a collection of small and simple *Analysis Components* that extract increasingly complex information based on earlier extractions. This technique is applied to extract individual data points and tables. Experiments show an overall precision of 97% with a recall of 93% regarding individual/simple data points and 89%/81% for table cells measured against manually entered ground truth. Due to its modular nature our system can be easily extended and adapted to other collections of contracts as long as some data model can be formulated.

Keywords: OCR robust information extraction, hierarchical taggers, table extraction

1 Introduction

Despite the existence of electronic document handling and content management systems there is still a large amount of paper based contracts. Even when scanned and OCRed the interesting data contained in the document is not machine-readable as there is no semantic attached to the text. Especially in the banking domain it is necessary to have the underlying information available, e.g., for risk assessment. Until now, the information has to be extracted by human reviewers. The goal of the system presented here is to automatically obtain the relevant information from OTC (over-the-counter) contracts which are based on a template provided by the ISDA[1]. The data is given in the form of image-embedded pdf documents. Each contract contains around 150 data points organized in a complex hierarchical data model. A data point can be either a (possibly multi valued) simple field or a table. The main challenges of such a system are:

[1] International Swaps and Derivatives Association, www.isda.org



(a)

(a) *Base Currency and Eligible Currency.*

(i) **"Base Currency"** means United States Dollars unless otherwise specified here: Euro.

(ii) **"Eligible Currency"** means the Base Currency and each other currency specified here: United States Dollars.

(b)

		Party A	Party B	Valuation Percentage
(A)	Cash, in the Eligible Currency	No	No	--
(B)	negotiable debt obligations issued by the Governments of the United States of America and Germany (excluding inflation-linked bonds) having a residual maturity of not more than one year.	Yes	Yes	98%
(C)	Negotiable debt obligations issued by the Governments of the United States of America and Germany (excluding inflation-linked bonds) having a residual maturity of more than one year but not more than five years	Yes	Yes	97%

Fig. 1. (a) Example simple valued fields (`base_currency` and `eligible_currency`). Note that for the `eligible_currency` one or more currencies can be specified. (b) Example table (collateral eligibility).

1. The complex legal language used in the contracts.
2. Despite existing contract templates, the wording varies across customers.
3. The layout varies. Especially tables can be represented in various forms.
4. The scanning quality of the contracts is often poor, especially in old contracts or documents sent by fax. Still the remaining information needs to be extracted correctly.

Figure 1 shows examples of two simple data points (a), and a table (b).

In general, on the one hand, there are a lot of sophisticated entity extraction systems that try to find flat entities only ("Named entity extraction") [9]. These systems sometimes use hierarchical information, like Tokens, Part-Of-Speech-Tags, Sentences, but only on a linguistic level without collecting and combining this information. These approaches work well on well-defined and general entities such as persons or locations. However, they are difficult to adapt to a new domain since a new classifier needs to be created which requires huge amounts of labeled training data which is expensive to produce.

On the other hand, there are systems that use a deep hierarchical structure, e.g. represented using Ontologies, but still do the classification in one single, flat step [1]. This approach is not as flexible and extensible compared to the presented one since in general it requires a re-training or re-building of the classifier if layers within the hierarchy are changed. An early solution for dealing with scanned forms was presented by Taylor et al., who used a model-based approach for data extraction from tax forms [12]. Semi structured texts have been analyzed using

rule based approaches [10] or discriminative context free grammars [13]. Closest to our solution is a system described by Surdeanu et al. [11]. They employ two layers of extraction using Conditional Random Fields [5], and deal with OCR data. For table extraction, heuristic methods [8] have been proposed as well as Conditional Random Fields [7].
In contrast, our system uses a theoretically unlimited number of layers with separate classifiers for each piece of information, including tables, on each level. Instead of processing the whole text at once, our classifiers just collect the information they require, and decide only on that data. Therefore, they allow for better performance and extensibility, as additional data does not affect the existing classifiers. Our work follows strategies commonly used in spoken dialogue systems [4] and uses a set of small classifiers which is inspired by the boosting idea [6]. In addition, we use automatically extracted segmentation information and cross-checks between our classifiers to increase the precision of the extracted data. From a UI standpoint there is a similar application called GATE [2] which extracts entities based on given rule-sets. This application provides a hierarchical organization of entities and the architecture seems to be very similar to the UIMA framework. However, GATE has no special provisions to deal with noise from due to the OCR step and it only allows to specify simple extraction rules. Furthermore there is no direct way that the entity extraction works hierarchically but only the result can be organized in a hierarchical way.

2 Information extraction

An overview of or system's architecture is shown in figure 2. Prior to information extraction, the OmniPage[2] OCR engine is used to convert the image to readable text. However, many character level errors, and layout distortions remain which need to be dealt with in the following processing steps. The overall strategy is based on the idea that small pieces of relevant text can be extracted quite accurately even in the presence of OCR errors. On top of these pieces we build several layers of higher level extractors – here called "experts" – that combine these small pieces to decide on a final data point. The extraction of tables works in a similar fashion by first trying to extract small pieces that form table cells. Then stretches of cells are collected, trying to deduce a layout from order and type of the pieces. Finally, an optimal result table is selected (see section 2.2).

Our solution is based on the UIMA framework [3]. Each type of expert is implemented as a configurable annotation engine. The overall extraction system consists of a large hierarchy of analysis engines, encompassing several hundred elements. The type system, in contrast, only consists of three principal types, i.e. for simple fields, tables and table rows. Annotation types, extracted values, etc. are stored as features. Both final and intermediate annotations are represented by these types.

[2] `http://www.nuance.com/omnipage`

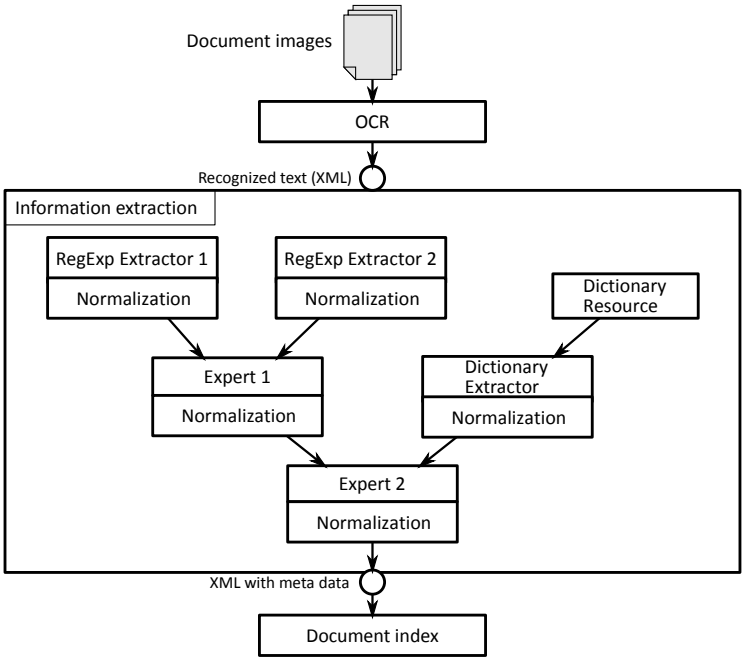


Fig. 2. Extraction architecture

2.1 Extraction of simple-valued fields

We use the term "simple-valued fields" for data points, where one key has one or more values. They differ from named entities as they may include multi-valued data. Figure 1(a) shows an example of the key `eligible_currency` with the (normalized) values "USD" and "Base currency". Fields are extracted layer-wise. On the lowest layer, all instances of the identifying term "Eligible currency", are captured, as well as the different currency expressions, including the special term "Base currency", which refers to another simple field. On this level we typically use annotators based on dictionaries and regular expressions, where variations due to OCR errors are reflected in dictionary variants, respectively the regular expressions. All such annotators are implemented as analysis engines. On the next level, so-called "expert-extractors" combine the existing annotations to a new one. An expert is a rule, defined as a set of slots for annotations of specific types, and a definition of which slots form a new annotation if the rule is satisfied, i.e. if all slots are filled. To allow for fine tuning the experts, slots can be configured, e.g. by indicating certain slots as optional. Furthermore, it is possible to specify the order of annotations in slots appearing in the document. It is also possible to specify a maximum distance. If the distance between two found annotations exceeds the defined threshold for this expert, the expert assumes to be in the wrong area of the document and clears its internal state to start all over

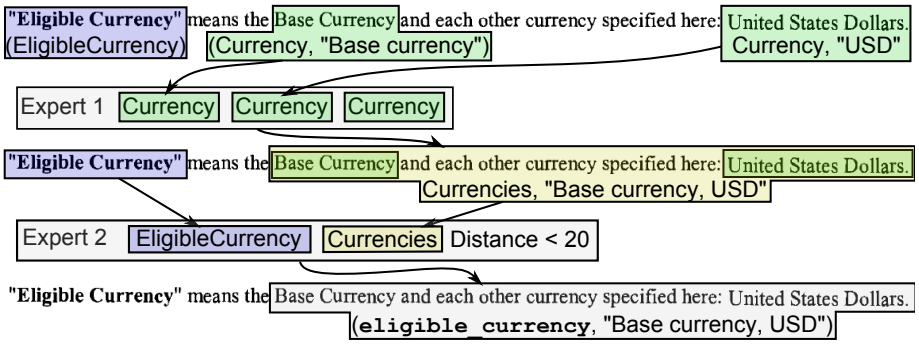


Fig. 3. Extraction of a simple field. First level components have tagged the "Eligible Currency" phrase and the different variants of currencies. Expert 1 collects two or more currencies (the third slot is optional). The resulting annotation is used by Expert 2 to build the final annotation. All elements are represented in UIMA as simple fields types.

again. Finally, slots can be write-protected, accepting only the first occurrence of the configured annotation.

To extract `eligible_currency`, two experts are employed (see figure 3). The first expert collects adjacent currency annotations. The second one combines the "Eligible Currency" term, and the collected currencies found by expert one, if both annotations are found within a short distance. The resulting annotation will span the relevant currency terms. This modular design allow us to reduce the number of extractors and re-use the already made annotations for completely different data points. In general, the information found in the examined contracts is not independent of each other. We use business rules and other constraints to validate and normalize the found results, e.g., the set of currencies is well-defined. If the validation fails or the normalization repairs some value due to business rules a corresponding message can be attached to the annotation to inform the reviewer.

2.2 Extraction of tables

We define a table as multi-dimensional, structured data present in a document either in a classical tabular layout, or defined in a series of sentences or paragraphs in free text form (like in figure 4). We aim at extracting tables of both structure types and intermediate formats (e.g. as in figure 1(b)) only from the document's OCR output at character level. In our application, table extraction extends the simple valued field extraction: The basic input for a table expert is a document annotated with simple value fields and intermediate annotations. The experts attempt to match sequences of simple annotations to a set of table models. A table model is user-defined and describes which columns the resulting extracted table should have. Each column can contain multiple types of simple

fields. Furthermore, columns can be configured to be optional and to accept only unique or non-overlapping annotations. This allows for both more general models with variable columns and fine-tuning the accepted annotations.

The process of detecting tables by the table expert (see figure 4 for an example) begins with collecting all accepted annotations for a model, within a predefined range or until a table stop annotation is found, into a list sorted by order of appearance. For each such list, several *filling strategies* are employed. A filling strategy addresses the problem that multiple columns may accept the same types of annotations. If elements appear row-wise, or column-wise, the corresponding strategies will recover the correct table, also compensating for some errors from omitted table elements. In mixed cases, adding a new table cell to the shortest relevant column is used as a fall back strategy. Each strategy is evaluated, using the fraction of cells filled in the resulting table c and the filling strategy specific score s. The latter score measures how well the annotations match the expectations of the filling strategy. The table which maximizes $s_f = c \cdot s$ is annotated as a candidate, if s_f is above a predefined threshold. The table expert is implemented an analysis engine. Configuration encompasses the columns describing the table model, distance and scoring threshold, and the set of filling strategies to be evaluated. The output is a table type annotation, which in turn contains several table rows, each containing simple fields as cells.

Multiple table experts may be used to generate candidate tables for a single target, and candidates may occur in several locations in a document. Usually, the correct location gives raise to tables with certain properties, e.g. short, dense tables. This is used by a feature-based selection of the optimal table candidate. We model this using both general purpose features (e.g. size, and number of empty cells) as well as domain specific features. The table with the highest weighted sum of score features is selected as the final output. The weights can either be user defined or fitted using a formal optimization model.

3 Experiments

We composed a document set containing 449 documents[3] to measure the extraction quality of our system. These documents are from various customers and represent as many variants of different wordings and layouts as possible.

With our customers we agreed upon certain quality gates that the automatic extraction system has to meet. Due to the nature of the contracts it is much more important to achieve a high precision of the extracted data instead of recall. For simple fields the gate's threshold is 95% precision and 80% recall. Table cells are more difficult to extract since the OCR component not only mis-recognizes individual characters but makes errors on the structure of a table. For table cells, our goal is to have a high recall since errors within a structured table are easier to detect and correct than simple field errors by a human reviewer. Table 1 shows our results against a manually created ground truth. The numbers represent the

[3] see `tinyurl.com/csa-example` for a public sample document.

	Insertions	Deletions	Substitutions	Correct	Precision	Recall
Simple fields	375	1267	330	20519	0.966	0.928
Table cells	1492	3563	906	18838	0.887	0.808

Table 1. Results for simple fields and table cells on our document corpus, shown as absolute numbers of (in)correct data points and values for precision and recall.

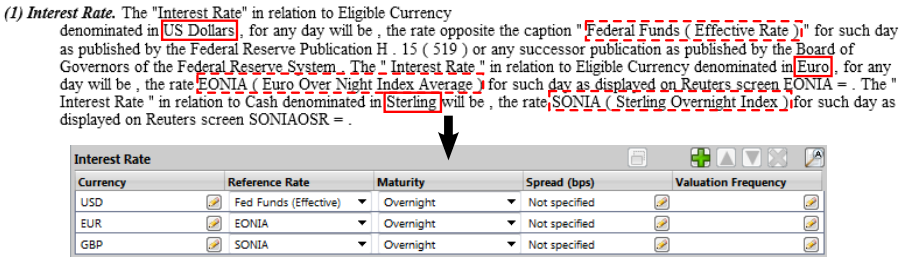


Fig. 4. Textual source (OCR output rendered as rich text) and extraction result of an interest rate table. The tabular result is extracted from the paragraph with three similar sentences which contain currencies (solid frame) and interest rate names (dashed frame). Domain knowledge is used to fill the maturity and spread columns.

total number of data points and errors respectively over all of our documents. In total, we meet our gate criterion for simple fields. Precision can be as low as 33% for rare fields, where fitting appropriate data experts is hard. In contrast, for frequent fields, precision may exceed 99%. In principle, the same is true for recall, with both maximum and minimum lower, due to our target criteria. For table cells, the precision needs improvement mainly due to the OCR's structural errors like swapping rows within a table or switching between row-wise and column-wise recognition in one table. This is especially true for tables which are complex with respect to both lay-out and contents, like the collateral eligibility table in figure 1(b). Here, precision and recall are 84.4% and 80.2%, respectively. In contrast, structurally simple tables, like the interest rate table (see figure 4 for an example) can be extracted with much higher confidence (97.4% precision and 90.8% recall).

4 Conclusion and outlook

This article presents a system to automatically extract simple data points and tables from OTC contract images. The system consists of an OCR component and a hierarchical set-up of small modular extractors either capturing (noisy) text or combining already annotated clues using a slot-filling strategy. Our experiments are conducted on a in-house contract collection resulting in a precision of 97% (recall 93%) on simple fields and a precision of 89% (recall 81%) on table cells. While the evaluation we conducted is limited, we expect overfitting to be

moderate. The legal nature of the contracts limits the layout and wording options. Our next steps include the introduction of a confidence score on data-point level and the use of statistical classification methods for selecting the best-suited table model.

Acknowledgement. We would like to thank our partner Rule Financial for providing the data model and for their assistance in understanding the documents.

References

1. Paul Buitelaar and Srikanth Ramaka. Unsupervised ontology-based semantic tagging for knowledge markup. In *Proceedings of the Workshop on Learning in Web Search at the International Conference on Machine Learning*, 2005.
2. Hamish Cunningham. Gate, a general architecture for text engineering. *Computers and the Humanities*, 36(2):223–254, 2002.
3. David Ferrucci and Adam Lally. Uima: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3-4):327–348, 2004.
4. Kyungduk Kim et al. A frame-based probabilistic framework for spoken dialog management using dialog examples. In *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*, 2008.
5. John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, 2001.
6. Ron Meir and Gunnar Rätsch. An introduction to boosting and leveraging. In *Advanced lectures on machine learning*, pages 118–183. Springer, 2003.
7. David Pinto, Andrew McCallum, Xing Wei, and W Bruce Croft. 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, 2003.
8. Pallavi Pyreddy and W Bruce Croft. Tintin: A system for retrieval in text tables. In *Proceedings of the second ACM international conference on Digital libraries*, pages 193–200, 1997.
9. Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the thirteenth conference on Computational Natural Lanugage Learning*, pages 147–155, 2009.
10. Stephen Soderland. Learning information extraction rules for semi-structured and free text. *Machine learning*, 34(1-3):233–272, 1999.
11. Mihai Surdeanu, Ramesh Nallapati, and Christopher D. Manning. Legal claim identification: Information extraction with hierarchically labeled data. In *Proceedings of the LREC 2010 Workshop on the Semantic Processing of Legal Texts*, 2010.
12. Suzanne Liebowitz Taylor, Richard Fritzson, and Jon A Pastor. Extraction of data from preprinted forms. *Machine Vision and Applications*, 5(3):211–222, 1992.
13. Paul Viola and Mukund Narasimhan. 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, 2005.