

Extracting structured data from invoices

Xavier Holt *
Sypht
xavier@sypht.com

Andrew Chisholm *
Sypht
andy@sypht.com

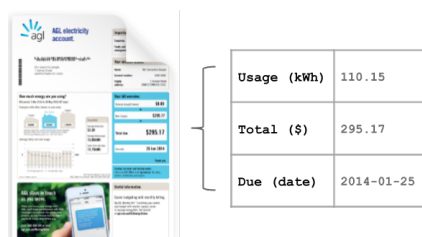
Abstract

Business documents encode a wealth of information in a format tailored to human consumption – i.e. aesthetically disbursed natural language text, graphics and tables.

We address the task of extracting key fields (e.g. the amount due on an invoice) from a wide-variety of potentially unseen document formats. In contrast to traditional template driven extraction systems, we introduce a content-driven machine-learning approach which is both robust to noise and generalises to unseen document formats. In a comparison of our approach with alternative invoice extraction systems, we observe an absolute accuracy gain of 20% across compared fields, and a 25%–94% reduction in extraction latency.

1 Introduction

To unlock the potential of data in documents we must first interpret, extract and structure their content. For bills and invoices, data extraction enables a wide variety of downstream applications. Extraction of fields such as the amount due and biller information enable the automation of invoice payment for businesses. Moreover, extraction of information such as the daily usage or supply charge as found on an electricity bill (e.g. Figure 1) enables the aggregation of usage statistics over time and automated supplier switching advice. Manual annotation of document content is a time-consuming, costly and error-prone process (Klein et al., 2004). For many organisations, processing accounts payable or expense claims requires ongoing manual transcription for



Usage (kWh)	110.15
Total (\$)	295.17
Due (date)	2014-01-25

Figure 1: Energy bill with extracted fields.

verification of payment, supplier and pricing information. Template and RegEX driven extraction systems address this problem in part by shifting the burden of annotation from individual documents into the curation of extraction templates which cover a known document format. These approaches still necessitate ongoing human effort to produce reliable extraction templates as new supplier formats are observed and old formats change over time. This presents a significant challenge – Australia bill payments provider BPAY covers 26,000 different registered billers alone¹.

We introduce SYPHT – a scalable machine-learning solution to document field extraction. SYPHT combines OCR, heuristic filtering and a supervised ranking model conditioned on the content of document to make field-level predictions that are robust to variations in image quality, skew, orientation and content layout. We evaluate system performance on unseen document formats and compare 3 alternative invoice extraction systems on a common subset of key fields. Our system achieves the best results with an average accuracy of 92% across field types on unseen documents and the fastest median prediction latency of 3.8 seconds. We make our system available as an API² – enabling low latency key-field extraction scalable to hundreds of document per second.

* Authors contributed equally to this work

¹www.bpay.com.au

²www.sypht.com

2 Background

Information Extraction (IE) deals broadly with the problem of extracting structured information from unstructured text. In the domain of invoice and bill field extraction, document input is often better represented as a sparse arrangement of multiple text blocks rather than a single contiguous body of text. As financial documents are often themselves machine-generated, there is broad redundancy in this spatial layout of key fields across instances in a corpus. Early approaches exploit this structure by extracting known fields based on their relative position to extracted lines (Tang et al., 1995) and detected forms (Cesarini et al., 1998). Subsequent work aims to better generalise extraction patterns by constructing formal descriptions of document structure (Coüasnon, 2006) and developing systems which allow non-expert end-users to dynamically build extraction templates ad-hoc (Schuster et al., 2013). Similarly, the ITESOFT system (Rusiol et al., 2013) fits a term-position based extraction model from a small sample of human labeled samples which may be updated iteratively over time. More recently, D’Andecy et al. (2018) build upon this approach by incorporating an a-priori model of term-positions to their iterative layout-specific extraction model, significantly boosting performance on difficult fields.

While these approaches deliver high-precision extraction on observed document formats they cannot reliably or automatically generalise to unseen field layouts. Palm et al. (2017) present the closest work to our own with their CloudScan system for zero-shot field extraction from unseen invoice document forms. They train a recurrent neural network (RNN) model on a corpus of over 300K invoices to recognize 8 key fields, observing an aggregate F-score of 0.84 for fields extracted from held-out invoice layouts on their dataset. We consider a similar supervised approach but address the learning problem as one of value ranking in place of sequence tagging. As they note, system comparison is complicated by a lack of a publicly available data for invoice extraction. Given the sensitive nature of invoices and prevalence of personally identifiable information, well-founded privacy concerns constrain open publishing in this domain. We address this limitation in part by rigorously anonymising a diverse set of invoices and submit them for evaluation to publicly available systems — without making public the data itself.

3 Task

We define the extraction task as follows: given a document and set of fields to query, provide the value of each field as it appears in the document. If there is no value for a given field present return `null`. This formulation is purely extractive – we do not consider implicit or inferred field values in our experiments or annotation. For example, while it may be possible to *infer* the value of tax paid with high confidence given the `net` and `gross` amount totals on an invoice, without this value being made explicit in text the correct system output is `null`. We do however consider inference over field names. Regardless of how a value is presented or labeled on a document, if it meets our query field definition systems must extract it. For example, valid `invoice number` values may be labeled as “*Reference*”, “*Document ID*” or even have no explicit label present. This canonicalization of field expression across document types is the core challenge addressed by extraction systems.

To compare system extractions we first normalise the surface form of extracted values by type. For example, dates expressed under a variety of formats are transformed to `yyyy-mm-dd` and numeric strings or reference number types (e.g. `ABN, invoice number`) have spaces and extraneous punctuation removed. We adopt the evaluation scheme common to IE tasks such as Slot Filling (McNamee et al., 2009) and relation extraction (Mintz et al., 2009). For a given field predictions are judged *true-positive* if the predicted value matches the label; *false-positive* if the predicted value does not match the label; *true-negative* if both system and label are `null`; and *false-negative* if the predicted value is `null` and label is not `null`. In each instance we consider the type-specific normalised form for both value and label in comparisons. Standard metrics such as F-score or accuracy may then be applied to assess system performance.

Notably we do not consider the position of output values emitted by a system. In practise it is common to find multiple valid expressions of the same field at different points on a document – in this instance, labeling each value explicitly is both laborious for annotators and generally redundant. This may however incorrectly assign credit to systems for a missed predictions in rare cases, e.g. if both the `net` and `gross` totals normalise to the

same value (i.e. no applicable tax) a system may be marked correct for predicting either token for each field.

3.1 Fields

SYPHT provides extraction on a range of fields. For the scope of this paper and the sake of comparison, we restrict ourselves to the following fields relevant to invoices and bill payments:

Supplier ABN represents the Australian Business Number (ABN) of the invoice or bill supplier. For example, 16 627 246 039.

Document Date the date at which the document was released or printed. Generally distinct from the due date for bills and may be presented in a variety of formats, e.g. 11st December, 2018 or 11-12-2018.

Invoice number a reference generated by the supplier which uniquely identifies a document, e.g. INV-1447. Customer account numbers are not considered invoice references.

Net amount the total amount of new charges for goods and services, before taxes, discounts and other bill adjustments, e.g. \$50.00.

GST the amount of GST charged as it relates to the net amount of goods and services, e.g. \$5.00.

Gross amount the total gross cost of new charges for goods and services, including GST or any adjustments, e.g. \$55.00.

4 SYPHT

In this section we describe our end-to-end system for key-field extraction from business documents. We introduce a pipeline for field extraction at a high level and describe the prediction model and field annotation components in detail.

Although our system facilitates human-in-the-loop prediction validation, we do not utilise human-assisted predictions in our evaluation of system performance in Section 5.

Preprocessing documents are uploaded in a variety of formats (e.g. PDF or image files) and normalised to a common form of one-JPEG image per page. In development experiments we observe faster performance without degrading prediction accuracy by capping the rendered page resolution (~8MP) and limiting document colour channels to black and white.

OCR each page is independently parsed by an Optical Character Recognition (OCR) system in parallel which extracts textual tokens and their corresponding in-document positions.

Filtering for each query field we filter a subset of tokens as candidates in prediction based on the target field type. For example, we do not consider currency denominated values as candidate fills for a date field.

Prediction OCRed tokens and page images make up the input to our prediction model. For each field we rank the most likely value from the document for that field. If the most likely prediction falls below a tuned likelihood threshold, we emit `null` to indicate no field value is predicted. We describe our model implementation and training in Section 4.1.

Validation (optional) — consumers of the SYPHT API may specify a confidence threshold at which uncertain predictions are human validated before finalisation. We briefly describe our prediction assisted annotation and verification work-flow system in Section 4.2.

Output a JSON formatted object containing the extracted field-value pairs, model confidence and bounding-box information for each prediction is returned via an API call.

4.1 Model and training

Given an image and OCRed content as input, our model predicts the most likely value for a given query field. We use Spacy³ to tokenise the OCR output. Each token is then represented through a wide range of features which describe the token's syntactic, semantic, positional and visual content and context. We utilise part-of-speech tags, word-shape and other lexical features in conjunction with a sparse representation of the textual neighbourhood around a token to capture local textual context. In addition we capture a broad set of positional features including the x and y coordinates, in-document page offset and relative position of a token in relation to other predictions in the document. Our model additionally includes a range of proprietary engineered features tailored to field and document types of interest.

Field type information is incorporated into the model through token-level filtering. Examples of

³spacy.io/models/en#en_core_web_sm

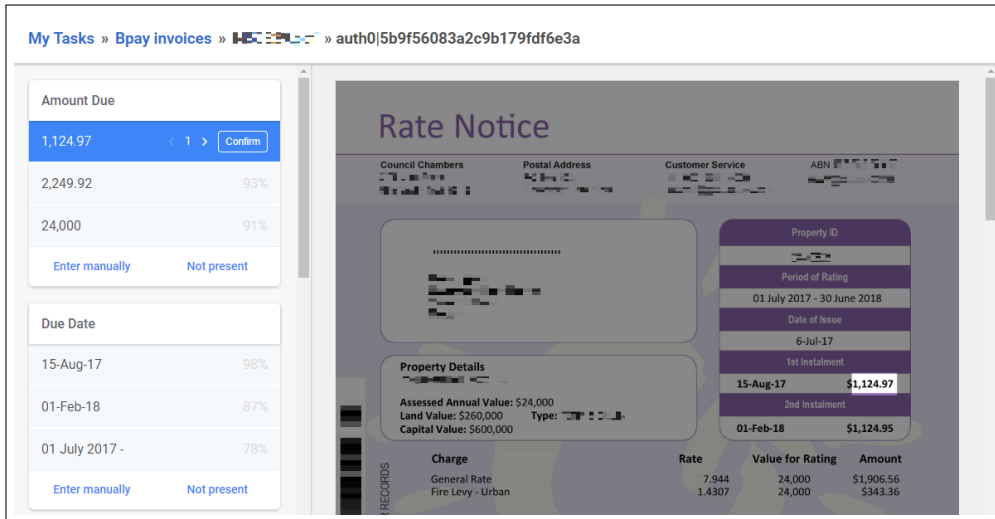


Figure 2: Our annotation and prediction verification tool — SYPHT VALIDATE. Tasks are presented with fields to annotate on the left and the source document for extraction on the right. We display the top predictions for each target field as suggestions for the user. In this example the most likely `Amount due` has been selected and the position of this prediction in the source document has been highlighted for confirmation.

field types which benefit from filtering are date, currency and integer fields; and fields with checksum rules. To handle multi-token field outputs, we utilise a combination of heuristic token merging (e.g. pattern based string combination for `Supplier` ABNs) and greedy token aggregation under a minimum sequence likelihood threshold from token level predictions (e.g. name and address fields).

We train our model by sampling instances at the token level. Matcher functions perform normalisation and comparison to annotated document labels for both for single and multi-token fields. All tokens which match the normalised form of the human-agreed value for a field are used to generate positive instances in a process analogous to distant supervision (Mintz et al., 2009). Other tokens in a document which match the field-type filter are randomly sampled as negative training instances. Instances of labels and sparse features are then used to train a gradient boosting decision tree model (LightGBM)⁴. To handle `null` predictions, we fit a threshold on token-level confidence which optimises a given performance metric; i.e. F-score for the models considered in this work. If the maximum likelihood value for a predicted token-sequence falls below the threshold for that field, a `null` prediction is returned instead.

4.2 Validation

An ongoing human annotation effort is often central to the training and evaluation of real-world machine learning systems. Well designed user-experiences for a given annotation task can significantly reduce the rate of manual-entry errors and speed up data collection (e.g. Prodigy⁵). We designed a predication-assisted annotation and validation tool for field extraction – SYPHT VALIDATE. Figure 2 shows a task during annotation.

Our tool is used to both supplement the training set and optionally – where field-level confidence does not meet a configurable threshold; provide human-in-the-loop prediction verification in real time. Suggestions are pre-populated through SYPHT predictions, transforming an otherwise tedious manual entry task into a relatively simple decision confirmation problem. Usability features such as token-highlighting and keyboard navigation greatly decrease the time it takes to annotate a given document.

We utilise continuous active learning by prioritising the annotation of new documents from our unlabeled corpus where the model is least confident. Conversely we observe high-confidence predictions which disagree with past human annotations are good candidates for re-annotation; often indicating the presence of annotation errors.

⁴github.com/Microsoft/LightGBM

⁵<https://prodi.gy/>

4.3 Service architecture

SYPHY has been developed with performance at scale as a primary requirement. We use a micro-service architecture to ensure our system is both robust to stochastic outages and that we can scale up individual pipeline components to meet demand. Services interact via a dedicated message queue which increases fault-tolerance and ensure consistent throughput. Our system is capable of scaling to service a throughput of hundreds of requests per second at low latency to support mobile and other near real-time prediction use-cases. We consider latency a core metric for real-world system performance and include it in our evaluation of comparable systems in Section 5.

5 Evaluation

In this section we describe our methodology for creating the experimental dataset and system evaluation. We aim to understand how a variety of alternative extraction systems deals with various invoice formats. As a coarse representation of visual document structure, we compute a perceptual hash (Niu and Jiao, 2008) from the first-page of each document in a sample of Australian invoices. Personally identifiable information (PII) was then manually removed from each invoice by a human reviewer. SYPHY VALIDATE was used to generate the labels for the task, with between two and four annotators per field dependent on inter-annotator agreement. Annotators worked closely to ensure consistency between their labels and the data definitions listed in Section 3.1, with all fields having a sampled Cohen’s kappa greater than 0.8, and all fields except `net amount` having a kappa greater than 0.9. During the annotation procedure four documents were flagged as low quality and excluded from the evaluation set, resulting in a final count of 129. In each of these cases annotators could not reliably determine field values due to poor image quality. We evaluated against our deployed system after ensuring that all documents in the evaluation set were excluded from the model’s training set.

5.1 Compared systems

ABBYY⁶ We ran ABBYY FlexiCapture 12 in batch mode on a modern quad-core desktop computer. While ABBYY software provides tools for creating extraction templates by hand, we utilised

⁶www.abbyy.com/en-au/flexicapture/

the generic invoice extraction model for parity with other comparison systems. By contrast with other systems which provided seamless API access, we operated the user interface manually and were unable to reliably record the distribution of prediction time per document. As such we only note the average extraction time aggregated over all test documents in Table 2

EzzyBills⁷ automate data entry of invoice and account-payable in business accounting systems. We utilised the EzzyBills REST API.

Rossum⁸ advertise a deep-learning driven data extraction API platform. We utilised their Python API⁹ in our experiments.

6 Results

Table 1 presents accuracy results by field for each comparison system. SYPHY delivers the highest performance across measures fields with a macro averaged accuracy exceeding our comparable results by 23.7%, 22.8% and 20.2% (for Ezzy, ABBYY, Rossum respectively). Interestingly we observe low scores across the board on the `net amount` field with every systems performing significantly worse than the closely related `gross amount`. This field also obtained the lowest level of annotator agreement and was notoriously difficult to reliably assess – for example, the inclusion or exclusion of discounts, delivery costs and other adjustments to various sub totals on an invoice often complicates extraction.

The next best system Rossum performed surprising well considering their coverage of the the European market; excluding support for Australian-specific invoice fields such as `ABN`. Still, even after excluding `ABN`, `net amount` and `GST` which may align to different field definitions, SYPHY maintains an 8 point accuracy advantage and more than 14 times lower median prediction latency.

Table 2 summarises the average prediction latency in seconds for each system alongside the times for documents at the 25th, 50th and 75th percentile of the response time distribution. Under the constraint of batch processing within the desktop ABBYY extraction environment we were unable to reliable record per-document prediction

⁷www.ezzybills.com/api/

⁸www.rossum.ai

⁹pypi.org/project/rossum

Field	Ezzy	ABBYY	Rossum	Ours
Supplier ABN	76.7	80.6	-	99.2
Invoice Number	72.1	82.2	86.8	94.6
Document Date	67.4	45.0	90.7	96.1
Net Amount	53.5	51.2	55.8	80.6
GST Amount	69.8	72.1	45.0	90.7
Gross Amount	75.2	89.1	84.5	95.3
Avg.	69.1	70.0	72.6	92.8

Table 1: Prediction accuracy by field.

	Avg.	25th	50th	75th
Rossum	67.06	47.7	54.4	91.0
Ezzy	27.9	20.6	26.9	34.5
ABBYY	5.6	-	-	-
Ours	4.2	3.3	3.8	4.8

Table 2: Prediction latency in seconds.

times and thus do not indicate their prediction response percentiles. SYPHT was faster than all comparison systems, and significantly faster relative to the other SaaS based API services. Even with the lack of network overhead inherent to ABBYY’s local extraction software, SYPHT maintains a 25% lower average prediction latency. In a direct comparison with other API based products we demonstrate stronger results still, with EzzyBills and Rossum being slower than SYPHT by a factor of 6.6 and 15.9 respectively in terms of mean prediction time per document.

7 Discussion and future work

While it is not a primary component of our current system, we have developed and continue to develop a number of solutions based on neural network models. Models for sequence labelling, such as LSTM (Gers et al., 1999) or Transformer (Vaswani et al., 2017) networks can be directly ensembled into the current system. We are also exploring the use of object classification and detection models to make use of the visual component of document data. Highly performant models such as YOLO (Redmon and Farhadi, 2018), are particularly interesting due to their ability to be used in real-time. We expect sub-5 second response times to constitute a rough threshold for realistic deployment of extraction systems in real time applications, making SYPHT the best system in contrast to either of the other two API-based services.

We also see an exciting opportunity to provide self-service model development – the ability for a customer to use their own documents to generate a model tailored to their set of fields. This would allow us to offer SYPHT for use cases where either we cannot or would not collect the prerequisite data. SYPHT VALIDATE provides a straightforward method for bootstrapping extraction models by providing rapid data annotation and efficient use of annotator time through active learning.

8 Conclusion

We present SYPHT, a SaaS API for key-field extraction from business documents. Our comparison with alternative extraction systems demonstrate both high accuracy and lower latency across extracted fields – enabling applications in real time for invoices and bill payment.

Acknowledgements

We would like to thank members of the SYPHT team for their contributions to the system, annotation and evaluation effort: Duane Allam, Farzan Maghami, Paarth Arora, Raya Saeed, Saskia Parker, Simon Mittag and Warren Billington.

References

- Francesca Cesarini, Marco Gori, Simone Marinai, and Giovanni Soda. 1998. Informys: A flexible invoice-like form-reader system. *IEEE Trans. Pattern Anal. Mach. Intell.* 20(7):730–745.
- Bertrand Couasnon. 2006. Dmos, a generic document recognition method: application to table structure analysis in a general and in a specific way. *International Journal of Document Analysis and Recognition (IJ DAR)* 8(2):111–122. <https://doi.org/10.1007/s10032-005-0148-5>.

- Vincent Poulain D'Andecy, Emmanuel Hartmann, and Marçal Rusiñol. 2018. [Field extraction by hybrid incremental and a-priori structural templates](#). In *13th IAPR International Workshop on Document Analysis Systems, DAS 2018, Vienna, Austria, April 24-27, 2018*. pages 251–256. <https://doi.org/10.1109/DAS.2018.29>.
- Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. 1999. Learning to forget: Continual prediction with lstm .
- Bertin Klein, Stevan Agne, and Andreas Dengel. 2004. Results of a study on invoice-reading systems in germany. In Simone Marinai and Andreas R. Dengel, editors, *Document Analysis Systems VI*. Springer Berlin Heidelberg, Berlin, Heidelberg, pages 451–462.
- Paul McNamee, Heather Simpson, and Hoa Trang Dang. 2009. Overview of the TAC 2009 Knowledge Base Population Track. In *Proceedings of the 2009 Text Analysis Conference*.
- Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. [Distant supervision for relation extraction without labeled data](#). In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, USA, volume 2, pages 1003–1011. <http://dl.acm.org/citation.cfm?id=1690219.1690287>.
- Xia-mu Niu and Yu-hua Jiao. 2008. An overview of perceptual hashing. *Acta Electronica Sinica* 36(7):1405–1411.
- Rasmus Berg Palm, Ole Winther, and Florian Laws. 2017. [Cloudscan - A configuration-free invoice analysis system using recurrent neural networks](#). In *14th IAPR International Conference on Document Analysis and Recognition, ICDAR 2017, Kyoto, Japan, November 9-15, 2017*. pages 406–413. <https://doi.org/10.1109/ICDAR.2017.74>.
- Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767* .
- Maral Rusiol, Tayeb Benkhelfallah, and Vincent Poulain D'Andecy. 2013. Field extraction from administrative documents by incremental structural templates. In *Proceedings of the 12th International Conference on Document Analysis and Recognition*. IEEE Computer Society, pages 1100–1104.
- Daniel Schuster, Klemens Muthmann, Daniel Esser, Alexander Schill, Michael Berger, Christoph Weidling, Kamil Aliyev, and Andreas Hofmeier. 2013. [Intellix – end-user trained information extraction for document archiving](#). In *Proceedings of the 2013 12th International Conference on Document Analysis and Recognition*. IEEE Computer Society, Washington, DC, USA, ICDAR '13, pages 101–105. <https://doi.org/10.1109/ICDAR.2013.28>.
- Y. Y. Tang, C. Y. Suen, Chang De Yan, and M. Cherié. 1995. Financial document processing based on staff line and description language. *IEEE Transactions on Systems, Man, and Cybernetics* 25(5):738–754. <https://doi.org/10.1109/21.376488>.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. pages 5998–6008.