Unstructured Document Recognition on Business Invoice

Stanford University

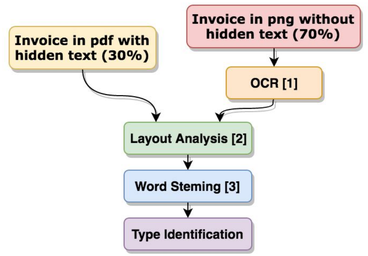
Yaqi Zhang (yaqiz), Billy Wan (xwan), Wenshun Liu (wl88)

Abstract

Invoice recognition has the unique challenge of extracting structured information from unstructured document, with unpredictable templates and choices of keywords. For this project, "bags of potential features" are generated to capture aspects of invoice layout, and then evaluated in multiple models to reveal the key properties that identify specific fields of interest. 8 fields of interest (including a negative class) are trained from 97 different templates, with 4.58 % training error and 19.40 test error.

Data Preparation

Training data is made of the list of text tokens and their coordinates, and are labelled to match the fields of interest (or 0 otherwise).



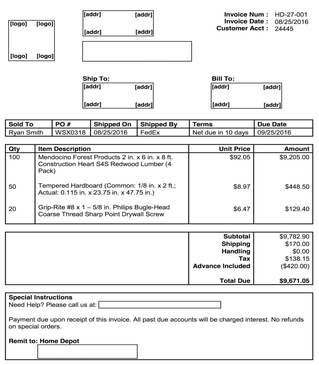


Fig. 1 Sample Invoice

Feature Selection

"Bags of raw features" are generated from the input data that are indicative of various nature of the token, such as its alignment and distance with other tokens, as well as its own type. The ~2000 features generated are then further evaluated based on KL divergence, and selected via filter feature selection.

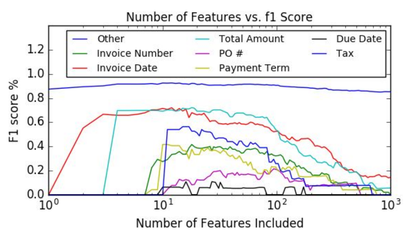


Fig. 2 Feature Selection for Naive Bayes (F1-Score on Training Set)

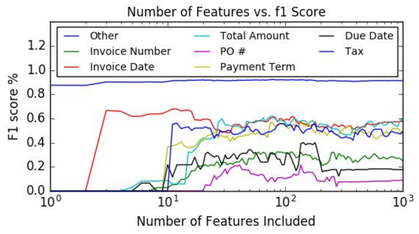


Fig. 3 Feature Selection for SVM (F1-Score on Training Set)

Models

Three models (Naive Bayes, logistic regression, and SVM) are evaluated against the feature set:

Naive Bayes: $\phi_{j,y=k} = \frac{\Sigma_{i=1}^{m} 1(x_j^{(i)} = 1, y^{(i)} = k) + 1}{m + K}$ $\phi_{y=k} = \frac{\Sigma_{i=1}^{m} 1(y^{(i)} = k) + 1}{m + K}$ where $K = \#$ classes

Logistic Regression: $h_\theta(x) = \frac{1}{1 + \exp(-\theta^T x)}$

SVM Loss Function: $L(z, y) = [1 - yz]_+ = \max\{1 - yz, 0\}$ where $z = \theta^T x$

In the input matrix to the model, the (i,j)-entry of this matrix represents whether the j^{th} distinct feature appears for the i^{th} training data. Each text token has the potential to be included in the input matrix multiple times, to represent different feature selection rules (for instance, "invoice_nearby" and "invoice_halign" are two distinct features for text token "invoice", indicating whether another training data is close by or horizontally aligned with it).

Results and Discussions

	Best Training Error (%, ~1490 tokens)	Best Test Error (%, ~638 tokens)
Naive Bayes	16.67	25.44
SVM	4.58	19.40
Logistic Regression	8.06	19.40

Overall, naive Bayes performs noticeably poorer than the other two models, which is likely due to the fact that naive Bayes assumes independence among features, which is incorrect for this problem (e.g. a token that is vertically aligned is likely to show up in the "nearby" set as well). Feature selection results indicate that naive Bayes is also extremely sensitive to overfitting, while logistic regression and SVM are robust enough even with too many non-indicative features.

Fig. 4 on the right indicates that performance can vary significantly for different classes even under the same set of feature selection rules. Classes that perform poorly tend to have inconsistent positioning and less overall presence in training data (such as PO#), or has overly similar features to other classes (such as Invoice Date and Due Date, that are both date type and usually positioned next to each other).

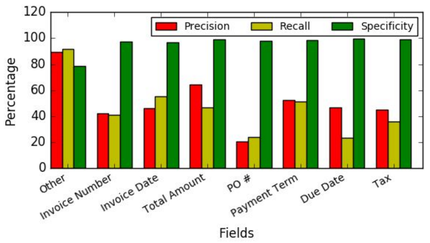


Fig. 4 Precision, recall, and specificity for each field, run on the test set using SVM.

Regardless of the model used, performance is inevitably affected by limitation in image quality and OCR and hidden text extraction results.

Future Works

More work is needed to handle overfitting, which includes regularization, training data size expansion, and "Wrapper" model feature selection uptake for more precise results. Further analysis might be worthwhile on the effectiveness of the current set of feature selection rules in capturing the inherent nature of invoice layout, and perform rule-level feature selection on a larger pool of potential feature generation rules. Additionally, current results are largely affected by poor OCR and pdf text extraction performance, which suggests investigation on more optimized OCR and text extraction solutions or collection of higher quality invoice images.

Reference

1. R. Smith, "An overview of the tesseract ocr engine," 2007.
2. Y. Shinyama, "Pdfminer: Python pdf parser and analyzer (2010)."
3. Python Implementation of Porter2 Stemming Algorithm: https://pypi.python.org/pypi/stemming/1.0
4. E. Loper and S. Bird, "Nltk: The natural language toolkit. 2002," URL http://arxiv. org/abs/cs/0205028.
5. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vander- plas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.