AIDE - Automatic Invoice Data Extraction technique using transformers and contextual analysis

Prem Suresh Pipada
dept. name of organization (of Aff.)
name of organization (of Aff.)
Chennai, India
premsureshpipada@gmail.com

Sayeeshwar Kumar
Dept. Of IT
SSN College Of Engineering
Chennai, India
sayeeshwark@gmail.com

Tushar Nair
Dept. Of CSE
SSN College Of Engineering
Chennai, India
tusharnair02@gmail.com

***Abstract*—AIDE is a system designed using state-of-the-art techniques and embeddings. The model strives to minimize the resources spent on information extraction from invoices in companies. It is able to adapt and process unseen invoice formats and doesn't follow a single rigid template making it ideal for a global scenario.**

We present a Hierarchical model of three layers, each with a discrete function, to classify and output the information on 15 logical fields that were defined taking any global invoice document into consideration. AIDE is compared with to highlight the difference in performance.

We now have two short paragraphs where the results are cited in brief. Filling up the space again so that there is some text in this paragraph as well.

The following paragraphs would explain the conclusions that were drawn from the same. Here lies the conclusions that is currently just garbage text to make the abstract seem like it has content.

I. Introduction

Invoices are documents that bear information regarding transactions between two parties. Extraction of this data in a desirable format is integral to the management of the company. This usually demands an inefficient expenditure of time, energy, money as it is achieved by having one parse the documents manually.

Eliminating the need to expend human resource in order to procure the relevant data is essential. AIDE boasts avant-garde methods to generate the necessary information with minimal effort.

Systems like [1], [2] aimed at creating an environment where data extraction can be automated by configuring a definite template. The documents would be annotated and then scanned by the system and the data is obtained either by means of location information as in [3] or by matching the actual works with a predefined set of keywords [4]. Improving on these methods, CloudScan [5] implemented a system which did away with the need to have any template or configuration to work effectively. By making use of a hierarchical model(3 sub models) which each use Bidirectional Encoder Representations from Transformers, the fields are selectively and accurately classified into the output framework.

AIDE paired with state of the art techniques looks to update the existing methods available for the same. It can work on unseen templates and uses Bert embedding to increase performance metrics like accuracy and F1 score.

Existing practises use basic embedding, with the inclusion of Bert embedding in AIDE, the task is improved in a much better way. Bert embedding makes use of a transformer for the embedding. This helps as it makes use of ”context” to understand words better.

Passing of context as input has not been implemented to this concept before. This improves the accuracy understanding the text by our model and classifying it appropriately.

II. Related Work

CloudScan [5] by TradeShift is the most closely related cloud based software that is used for invoice analysis. The model was designed to work with zero configuration and no annotation on the invoice documents. This allowed for a much broader scope when it came to analysis and extraction of the data. Text is extracted from documents and classified into 33 fields of interest by defining each of these outputs to be N-grams of words. This on further processing defines which N-grams are used in the final document. Finally an invoice in the XML file format is generated with the fields of interest being mapped to the respective N-grams.

Other systems such as Intellix [1] by DocuWare, SmartFIX [2] by DFKI spin-off INSIDERS,and others require some configuration and a definite template in order to extract the information from various invoices accurately. The fields have to be labelled beforehand and are primarily designed and deployed for companies that follow the same pattern in their invoices. They have to rely on these rigid templates and so try to minimize the number of examples required to handle an unseen template.

There a few different ways in which the layout of a template for data extraction can be defined. Cesarini et al. [4] has a global set of keywords defined. It further trains on a new database of keywords for each template. Esser et al. [3] focuses on the absolute positions of the relevant fields on the invoice document to effectively determine and extract the data. Some models like [2] has a set of configuration rules that are laid down for each template.

All these models have worked with datasets that are private and so we are unable to form a direct comparison with the same. These systems however, owning to the exception of [5], all require the datasets to be labelled for a particular template beforehand. While this allows for a wider range of fields of interests to be extracted and produced in the output, it involves a lot of manual labour to define it accordingly for each template. Just like CloudScan [5], AIDE also defines a fixed set of fields that would be represented in the output and does not leave it to the user's choice.

CloudScan categorises text among 8 fields. AIDE is trained and tested to classify text among 16 fields. The inclusion of state of the art techniques like transformers helps with better overall understanding of text and hence improve the results drastically.

III. Automatic Invoice Data Extraction

A. Overview

We present AIDE, a software designed to relieve any difficulties that processing an invoice by traditional methods would pose. It effectively extracts information from any invoice like document. AIDE works well even with unseen data and does not required a rigid template making it flexible as well as portable. AIDE functions in accordance with the following 4 steps:

1) **Text Extractor.** Invoices are passed as the input. The relevant data along with details of it's position in the document is extracted. The location information of a block of data is obtained using the Google Vision API.This OCR assembles the information in a JSON format.
2) **Elmo Embedding.** We use ELMo embedding for our model to achieve state of the art results. ELMo represents words as vectors or embeddings. Traditional word embeddings fail to perform as well as our ELMo embedded model.
3) **Bert Embedding.** We use BERT embedding for our model to achieve state of the art results. BERT is a bidirectional transformer-based model that makes use of 12 encoders. Traditional word embeddings fail to perform as well as our Hierarchical model that implements this embedding in the first layer.
4) **n-Closest Blocks.** For any block we define the coordinates at the top-left, top-right and bottom-right positions respectively in order to compute the midpoint of the block. The x coordinate is calculated as in (1) and the y coordinate of the midpoint is calculated as in (2):

$$(top\_left\, x + top\_right\, x)/2 \tag{1}$$

$$(top\_right\, y + bottom\_right\, y)/2 \tag{2}$$

On having the midpoints of all the blocks we then compute the distance between the incoming block and every other block in the dataframe, using the formula for shortest distance between two points (3), in order to arrive at the 'n' number of blocks closest to the former.

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{3}$$

This allows us to add an extra dimension in understanding the context and position of the incoming block.
5) **Classifiers.** In the first layer of AIDE, the incoming block is classified on 12 labels that are common to any invoice scheme. On passing this block to the second layer, it is now fed with the location information of 'n' labelled blocks closest to the former. The accuracy of the label assigned in the first layer is checked and the block is moved to the final layer of the model accordingly. Here, the labelled block is classified once again in order to enhance the significance of the block. For e.g. Value is split into Tax Amount, Total, Quantity and Amount.
6) **Post Processing.** The block of data scanned by the OCR need not contain only the required output field exactly within the dimensions. In the post processing layer, by making use of regular expressions the extra elements from the block are eliminated, leaving behind only the essential information.

IV. Methods

We put our model to the test by feeding it 1) invoice of a seen template and 2) invoice of an unseen template. This gives us two performance metrics to consider and evaluate the competence of the model.

The data set used by our model consists of x invoices. We split our invoices into a training and testing set, consisting of 85% and 15% respectively. This split is used for both the tests.

Our model extracts 15 fields and classifies data into one of these 15 fields. The initial 12 fields are address, value, date, description, email, GST, sac, tax_percent, tot_in_words, company_name, phone_number and none. Then we fine tune the label 'value' to add more lucidity by labelling it as either tax, total, amount or quantity in the final phase of the model.

Comparing the generated and validated data we can measure our model's performance. We check the performance on both seen and unseen invoice templates in the following two models in order to compare and contrast the results. 1) Baseline classification model 2) Hierarchical Model.

A. GloVe Model

GloVe, standing for Global Vectors, is a word vector technique that surfaced after the introduction of Latent Semantic Analysis (LSA) [6] and Word2Vec [7]. Word vectors are basically a mathematical representation of words that preserves the semantic relationship existing between them. This allows for the building for an algorithm that could analyse and choose the most apt word for a scenario simply based off the word vectors.

GloVe embedding makes use of a matrix factorization technique. A large matrix of data is created and this allows a word to easily look for context among the corpus of data. For each word, the words falling under a defined area is checked if they are under the same context. Only words that come under a closely bound region are considered. This allows the system to easily filter words that wouldn't fit the scenario according to the co-occurrence of words.

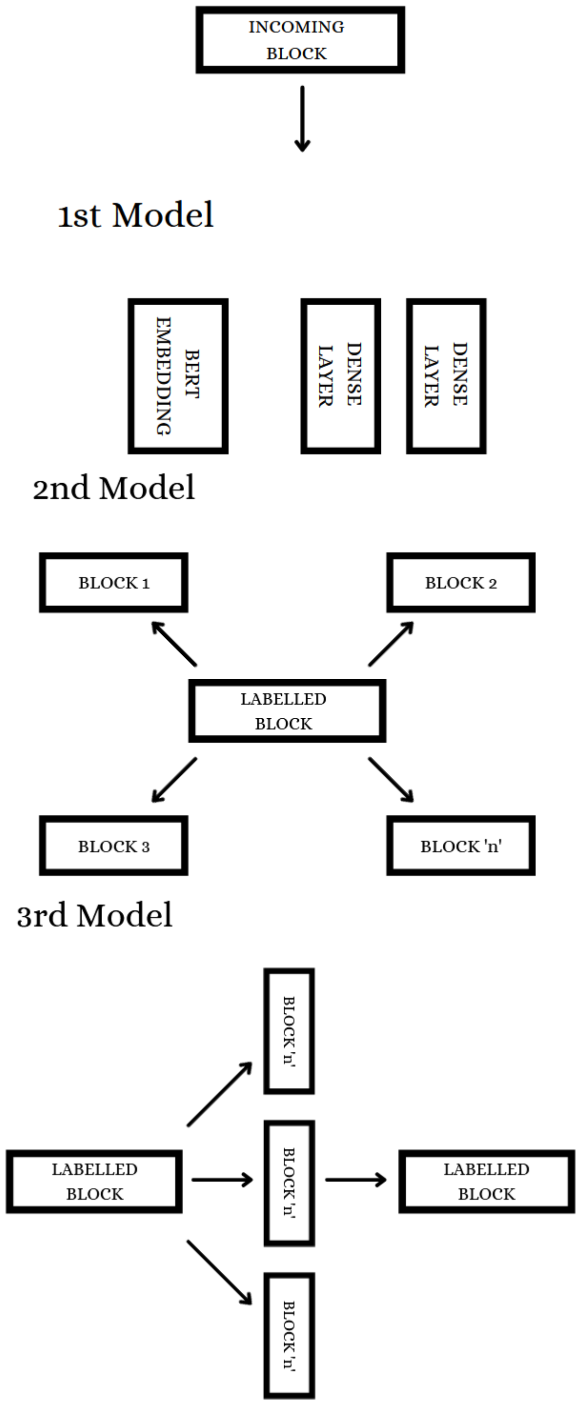

Fig. 1. The Hierarchical Model Architecture

B. *Elmo Model*

Embeddings from Language Models, or ELMo, is a state-of-the-art model that introduces a new type of deep contextualized word representation [8]. Breaking it down, ELMo is able to easily perform tasks like sentiment analysis which requires the system to have some knowledge with respect to the context. This builds on the results obtained from previous language models such as Word2Vec [7], Glove [9], BagofWords [10]. The latter systems extract every word and are compared on the face value with a global set of words without checking the semantics.

The ELMo model model has a two-layer bidirectional LSTM base. There is a residual connection [11] between the two layers to allow a more efficient training of the model. The slight difference in the working of this system [12] gives it a significant edge over the previous works. ELMo converts each token passed into the input layer into a character-embedding representation rather than the word-embedding representation that was followed in [7], [9]or [10]. This then gets passed through a convolutional layer and then the two-layer network before finally being fed into the LSTM. Character embedding allows the model to pick up on morphological features as well as out-of-vocabulary words. But even this had been done in Rafal et al. [13].

To see how the ELMo model really changes the game in the field of language models we must understand the mathematical details behind it.

$$ELMo_k^{task} = \gamma_k \cdot (s_0^{task} \cdot x_k + s_1^{task} \cdot h_{1,k} + s_2^{task} \cdot h_{2,k}) \quad (4)$$

Where k is any word of input, s_i represents the softmax-normalized weights and γ_k is a scaling factor that is task-specific. x_k is the word representation and $h_{1,k}$,$h_{2,k}$ are the bidirectional hidden layer representations.

C. Bert Embedding

Bidirectional Encoder Representations from Transformer, or BERT, is a state-of-the-art model which makes use of separate layers of embeddings with a functionality unique to each.

Token Embeddings are used to transform a word into a vector representation of some fixed dimension of798-dimensional vector. The input text gets tokenized by a method called WordPiece Tokenization which is done before passing it into the embedding layer. So each wordpiece token is then converted to the 768-dimensional matrix. [14] [15] Segment Embeddings, the second layer, is used to help the BERT model semantically distinguish between tokens in any pair of input. Position Embeddings, with a stack of Transformers [16], process the sequence of input and assign a vector representation for each position in the input.

All of these are then summed up to produce a single layer which would be passed into the BERT's Encoder Layer.

D. Hierarchical Model Architecture

1) **1st Model.** In the first layer of our Hierarchical Model, we pass the input block which was extracted from the OCR. This layer consists of a Bert Embedding layer or frozen Bert layers to be more precise as we aim to reduce the number of trainable parameters and the training time. The input block is then passed through two dense layers and finally classified on one of the twelve output fields. This labelled block then passed over to the next layer in the model architecture.
2) **2nd Model.** In the second layer of the model we pass the information of 'n' blocks who's labels have already been predicted. In the light of this new location context information we then predict the label for the incoming block on the 12 output labels once again. The closest blocks are defined by deriving the coordinate information and testing the distance between the midpoints of every single block with the incoming block. This predicted label is of maximum accuracy but is still passed into one more layer of the model, the final layer, where the block's label is confirmed.
3) **3rd Model.** In the final layer, the current label for the block is examined for the last time. Based on the classification it is scrutinized one more time in order to give the block a more detailed and specific identity. This layer performs the function of fine tuning the output.

E. BiLSTM

The inclusion of BiLSTM in our model is to improve the learning on sequential data passed from our inputs. A bidirectional LSTM consists of two LSTM layers which enables better and more efficient contextual learning. BiLSTMs learns information by taking inputs and learning through forward and backward propagation. Therefore the addition of this layer greatly increases learning of the model by passing context of words preceding and succeeding it. Our objective was to pass in the coordinates information of N closest blocks along with the text contained in these blocks as context to each block. We believe this will better capture and learn sequential data extracted from invoices.

V. Results

Etiam in mauris rhoncus, malesuada ante ut, vulputate libero. Donec sodales libero vitae odio hendrerit, ac consequat tortor blandit. Phasellus volutpat in est eget semper. Pellentesque ut feugiat libero. Nulla pulvinar id eros a posuere. Aliquam sagittis sodales neque at dictum. Praesent et risus facilisis, varius dui sed, fringilla lectus. Suspendisse ac luctus sapien, et eleifend felis. Quisque at ullamcorper odio. Curabitur purus massa, rhoncus sed dignissim id, porta at velit. Morbi semper mauris nec lorem venenatis viverra. Sed condimentum efficitur neque et scelerisque. Nulla facilisi.

Fusce non sodales est, vitae rutrum turpis. Nulla odio magna, mollis eu euismod id, vehicula vitae sapien. Vivamus euismod malesuada feugiat. Nulla in vulputate nisl, et pellentesque eros. Aliquam porttitor tellus eu tellus interdum bibendum. Vivamus mollis metus nec turpis venenatis, at pulvinar urna fermentum. Vestibulum fermentum urna non purus hendrerit semper. Duis at magna sit amet massa posuere eleifend.

Etiam volutpat nisi arcu, a convallis elit feugiat eu. Nam quis sapien nibh. Mauris eget nisi erat. Praesent et elit lorem. Fusce mollis nisi id leo ultricies, ac porttitor velit aliquam. Nullam sed ex vitae nulla bibendum convallis eu in magna. Phasellus eleifend neque ligula, vitae viverra elit varius id. Pellentesque iaculis a erat maximus pretium. Phasellus vitae dui tempus, iaculis sem ac, convallis lectus. Cras turpis metus, lobortis id eros non, elementum volutpat metus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Duis ullamcorper a eros a pellentesque. Fusce molestie nisi ac rhoncus condimentum.

TABLE I
ABC ANALYSIS RESULTS.TESTS RUN ON DOCUMENTS XYZ. PERFORMANCE OF MODEL AS EXPECTED.

Field	Precision	Recall	F1
Address	0.84	0.85	0.85
Amount	0.67	0.95	0.78
Bill Period	1	0.67	0.8
Cin	0.67	0.80	0.73
Date	0.99	0.92	0.95
Description	0.78	0.72	0.75
Email	0.86	1	0.92
Org GST	1	0.54	0.70
Vendor GST	0.90	0.97	0.93
None	0.76	0.78	0.77
Pan	0.80	0.80	0.80
Phone Number	0.78	1	0.88
Quantity	1	0.71	0.83
SAC	0.97	0.98	0.98
Tax Amount	1	0.25	0.40
Tax Percent	0.7	0.78	0.74
Total (In Words)	1	1	1
Total Amount	0.80	0.46	0.59
Vendor Name	0.80	0.59	0.68
Macro Avg.	0.86	0.78	0.79
Weighted Avg.	0.85	0.84	0.84

TABLE II
COMPARISON BETWEEN BASELINE AND AIDE MODEL. ANALYSIS PERFORMED ON BASELINE MODEL.

Field	Precision	Recall	F1
Address	0.84	0.85	0.85
Amount	0.67	0.95	0.78
Bill Period	1	0.67	0.8
Cin	0.67	0.80	0.73
Date	0.99	0.92	0.95
Description	0.78	0.72	0.75
Email	0.86	1	0.92
Org GST	1	0.54	0.70
Vendor GST	0.90	0.97	0.93
None	0.76	0.78	0.77
Pan	0.80	0.80	0.80
Phone Number	0.78	1	0.88
Quantity	1	0.71	0.83
SAC	0.97	0.98	0.98
Tax Amount	1	0.25	0.40
Tax Percent	0.7	0.78	0.74
Total (In Words)	1	1	1
Total Amount	0.80	0.46	0.59
Vendor Name	0.80	0.59	0.68
Macro Avg.	0.86	0.78	0.79
Weighted Avg.	0.85	0.84	0.84

VI. Discussion

Etiam in mauris rhoncus, malesuada ante ut, vulputate libero. Donec sodales libero vitae odio hendrerit, ac consequat tortor blandit. Phasellus volutpat in est eget semper. Pellentesque ut feugiat libero. Nulla pulvinar id eros a posuere. Aliquam sagittis sodales neque at dictum. Praesent et risus facilisis, varius dui sed, fringilla lectus. Suspendisse ac luctus sapien, et eleifend felis. Quisque at ullamcorper odio. Curabitur purus massa, rhoncus sed dignissim id, porta at velit. Morbi semper mauris nec lorem venenatis viverra. Sed condimentum efficitur neque et scelerisque. Nulla facilisi.

Fusce non sodales est, vitae rutrum turpis. Nulla odio magna, mollis eu euismod id, vehicula vitae sapien. Vivamus euismod malesuada feugiat. Nulla in vulputate nisl, et pellentesque eros. Aliquam porttitor tellus eu tellus interdum bibendum. Vivamus mollis metus nec turpis venenatis, at pulvinar urna fermentum. Vestibulum fermentum urna non purus hendrerit semper. Duis at magna sit amet massa posuere eleifend.

Etiam volutpat nisi arcu, a convallis elit feugiat eu. Nam quis sapien nibh. Mauris eget nisi erat. Praesent et elit lorem. Fusce mollis nisi id leo ultricies, ac porttitor velit aliquam. Nullam sed ex vitae nulla bibendum convallis eu in magna. Phasellus eleifend neque ligula, vitae viverra elit varius id. Pellentesque iaculis a erat maximus pretium. Phasellus vitae dui tempus, iaculis sem ac, convallis lectus. Cras turpis metus, lobortis id eros non, elementum volutpat metus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Duis ullamcorper a eros a pellentesque. Fusce molestie nisi ac rhoncus condimentum.

Acknowledgment

Etiam in mauris rhoncus, malesuada ante ut, vulputate libero. Donec sodales libero vitae odio hendrerit, ac consequat tortor blandit. Phasellus volutpat in est eget semper. Pellentesque ut feugiat libero. Nulla pulvinar id eros a posuere. Aliquam sagittis sodales neque at dictum. Praesent et risus facilisis, varius dui sed, fringilla lectus. Suspendisse ac luctus sapien, et eleifend felis. Quisque at ullamcorper odio. Curabitur purus massa, rhoncus sed dignissim id, porta at velit. Morbi semper mauris nec lorem venenatis viverra. Sed condimentum efficitur neque et scelerisque. Nulla facilisi.

References

[1] Daniel Schuster, Klemens Muthmann, Daniel Esser, Alexander Schill, Michael Berger, Christoph Weidling, Kamil Aliyev, and Andreas Hofmeier. Intellix – end-user trained information extraction for document archiving. In *2013 12th International Conference on Document Analysis and Recognition*, pages 101–105, 2013.

[2] Andreas Dengel and Bertin Klein. smartfix: A requirements-driven system for document analysis and understanding. volume 2423, pages 433–444, Aug 2002.

[3] Daniel Esser, Daniel Schuster, Klemens Muthmann, and Alexander Schill. Automatic indexing of scanned documents - a layout-based approach. volume 8297, Jan 2012.

[4] F. Cesarini, E. Francesconi, M. Gori, and G. Soda. Analysis and understanding of multi-class invoices. *Document Analysis and Recognition*, 6(2):102–114, Oct 2003.

[5] Rasmus Berg Palm, Ole Winther, and Florian Laws. Cloudscan - a configuration-free invoice analysis system using recurrent neural networks. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 406–413, 2017.

[6] Wikipedia contributors. Latent semantic analysis — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Latent_semantic_analysis&oldid=1037887976, 2021. [Online; accessed 10-August-2021].

[7] KENNETH WARD CHURCH. Word2vec. *Natural Language Engineering*, 23(1):155–162, 2017.

[8] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

[9] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[10] Yin Zhang, Rong Jin, and Zhi-Hua Zhou. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, 1(1):43–52, Dec 2010.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[12] Yoon Kim, Yacine Jernite, David A. Sontag, and Alexander M. Rush. Character-aware neural language models. *CoRR*, abs/1508.06615, 2015.

[13] Rafal Józefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. *CoRR*, abs/1602.02410, 2016.

[14] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016.

[15] Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152, 2012.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017.