Master Thesis

Michał Zaręba
196218

Automated Extraction and Categorization of Product Information from Receipts

Diploma thesis in the field of
Information Science

Thesis under the supervision of
dr hab. inż. Leszek Chmielewski, prof. SGGW
Institute of Information Technology
Department of Artificial Intelligence

Warsaw, year 2017

WARSAW
UNIVERSITY
OF LIFE SCIENCES

Faculty of Applied
Informatics and
Mathematics

Declaration of the Thesis Supervisor

I hereby declare that this thesis has been prepared under my supervision, and I confirm that it meets the conditions for presenting this work in the procedure for the award of a professional title.

Date ..... Supervisor's signature .....

Declaration of the Author of the Thesis

Aware of legal liability, including criminal liability for submitting a false declaration, I hereby declare that this diploma thesis was written by myself and did not contain the content obtained in a manner inconsistent with applicable law, in particular the Act of 4 February 1994 on copyright and related rights (Journal of Laws of 2019, item 1231, as amended).

I declare that the submitted work has not previously been the basis for any procedure related to awarding a diploma or obtaining a professional title.

I certify that this work version is identical to the attached electronic version.

I acknowledge that the diploma thesis will be subject to the procedure of the anti-plagiarism.

Date ..... Author's signature .....

Streszczenie

OCR + BERT

Tematem niniejszej pracy było zaimplementowanie klasy LaTeXowej pozwalającej na formatowanie tekstu zgodnie z wytycznymi nałożonymi przez uczelnię. Praca zawiera dwie główne części. Pierwsza z nich zawiera opis najważniejszych aspektów implementacji klasy. Natomiast druga część skupia się na sposobie użycia klasy przez osoby piszące prace dyplomowe.

Słowa kluczowe – OCR, BERT, Tesseract, thesis, implementation, SGGW, Warsaw University of Life Sciences

Summary

OCR + BERT

The subject of this thesis was to implement a LaTeX class that allows formatting text according to the guidelines imposed by the university. The thesis contains two main

Keywords – LaTeX, class, thesis, implementation, SGGW, Warsaw University of Life Sciences

Contents

1 Introduction

1.1 Motivation

Tracking of expenses and managing personal finances is an important aspect of modern life. With an increasing number of daily transactions and a vast variety of products available, individuals face significant challenges in effectively monitoring their spending and managing their budgets. Although receipts contain valuable details that could help consumers analyze and control their expenses, the majority of consumers either discard receipts shortly after purchase or find it too tedious and time-consuming to analyze them manually. Automating the extraction and categorization of product information from receipts could significantly simplify budget tracking and provide insights into spending habits, enabling consumers to understand precisely where their money goes. Simple and efficient way to track expenses is essential for individuals who wish to maintain a clear overview of their spending habits and make informed financial decisions.

1.2 Problem Statement

Most existing expense-tracking solutions focus primarily on invoices, bank statements, or require manual input. Large corporations and organizations typically possess the necessary budgets and technical resources to implement robust, automated systems for extracting and categorizing expense data from structured documents such as invoices or bank statements. For personal use, however, the most commonly available and practical source of spending information remains paper receipts. Current receipt-based solutions are often limited: many tools available today are either designed exclusively for commercial purposes, lack support for languages other than English, or are inadequately trained to accurately process Polish-language receipts. Thus, there is a clear gap and a significant need for a solution that effectively automates extraction and categorization of product details from receipts, specifically accommodating the complexity and linguistic characteristics of the Polish language.

1.3 Objectives of the Study

This study has two primary objectives, each directly addressing the challenges identified in the problem statement:

1. Develop a robust system capable of automatically extracting structured product information (such as product names, and prices) from Polish-language receipts using Optical Character Recognition (OCR).
2. Implement and evaluate a product categorization module based on the embeddings generated by pre-trained models, specifically BERT (Bidirectional Encoder Representations from Transformers), and Sentence-BERT model (Siamese transformer network) fine-tuned with own data.
3. The extracted embeddings serve as input to an XGBoost classifier responsible for categorizing products into predefined expense-related categories.

These objectives will be thoroughly addressed and analyzed in subsequent chapters. Given the complexity of Polish-language receipts and limited availability of labeled datasets, achieving optimal results will require careful integration and fine-tuning of multiple technologies. Critical aspects will include the effective integration of OCR and NLP components, as well as the development of a robust classification model capable of accurately categorizing products based on their textual descriptions that might often be ambiguous, multiple-worded, or contain spelling errors.

1.4 Scope and Limitations

The scope of this study is limited to the development of a system capable of automatically extracting product information from receipts and categorizing these products into predefined categories. The system specifically targets Polish-language receipts and will be evaluated primarily on its ability to accurately extract product costs and perform correct product categorization.

The limitations of this study include the following:

- The developed system will not include additional functionalities such as expense tracking over time, financial report generation, or integration with external personal financial management tools.
- The scarcity of comprehensive, labeled Polish-language receipt datasets restricts the potential accuracy and generalization capabilities of the models developed. Consequently, results may vary when encountering receipt formats or text variations not present in the training data.
- The OCR will be employed without utilizing spatial information or context regarding the positioning of text on receipts. Therefore, preprocessing steps such as image cropping, alignment, and noise reduction are necessary to ensure the OCR engine receives properly formatted and isolated textual inputs.

2 Literature Review

2.1 Optical Character Recognition (OCR) Technologies

Optical Character Recognition (OCR) refers to the process of converting textual information from scanned or photographed images into machine-readable formats.

Traditional OCR techniques primarily relied on template matching, statistical classification, and structural analysis, with limited adaptability to varying fonts and noisy inputs.[5]

Modern OCR systems use deep learning models that typically combine convolutional neural networks (CNNs) for feature extraction with recurrent or attention-based architectures (e.g., LSTM, GRU) for sequence modeling. This architecture enables the system to learn complex patterns and recognize characters across varying fonts, sizes, and orientations [9].

Recent OCR models further incorporate attention mechanisms, allowing the network to dynamically focus on relevant regions of the input image. Unlike standard OCR models, LayoutLM introduces explicit spatial awareness by incorporating positional embeddings of text regions, which is particularly effective for structured documents like receipts [12]. While attention-based models with spatial awareness offer notable improvements in accuracy and robustness, their effectiveness remains constrained by the availability of large, annotated datasets. As a result, the OCR systems evaluated in this study—Tesseract and PaddleOCR—do not model document structure explicitly and operate at the text-line level.

Tesseract is an open-source OCR engine maintained by Google, widely adopted across various applications. It supports multiple languages, including Polish, and can be fine-tuned on domain-specific datasets for improved accuracy. Since version 4.0, it incorporates an LSTM-based recognition engine, enhancing its performance on noisy or multilingual documents [10, 11]. Tesseract is highly configurable, offering control over segmentation modes, character whitelists, and language models. Fine-tuning involves training on a targeted dataset, which can significantly enhance accuracy for specific formats such as Polish receipts.

PaddleOCR is a deep learning-based OCR framework developed by Baidu, supporting over 80 languages. It features an end-to-end pipeline using models like DBNet for detection and CRNN or SRN for recognition, making it well-suited for multilingual documents and complex layouts [4]. Its modular and extensible architecture enables users to adapt the pipeline for specific use cases.

2.2 Semantic Text Embeddings for Receipt Item Categorization

To enable machine learning models to analyze text, the text must first be converted into a numerical format—a process known as vectorization or word embedding which is a crucial step in Natural Language Processing (NLP). Word embeddings are fixed-length vector representations that encode the semantic meaning of words and the relationships between them. The words with similar meanings are represented by similar vectors in the embedding space.[1]

There are several approaches to generating these embeddings, and this section outlines the most common methods, explaining how they work and highlighting their key differences. Before the word embedding process, the text must be preprocessed.

Prior to embedding, text must be preprocessed. This includes normalization steps such as lemmatization or stemming, and tokenization, which segments text into individual words or tokens [6].

Then there are several methods of generating word embeddings, which can be broadly categorized into two groups: count-based and prediction-based methods.

2.2.1 Count-based Methods

Count-based methods rely on the idea that the meaning of a word can be inferred from its co-occurrence with other words in a given context. Those methods typically involve creating a sparse matrix, where each row and column represents a word in the vocabulary, and the values in the matrix represent the frequency of co-occurrence between pairs of words. The two most common approaches are the Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF)[6].

Bag of words represents a document as a vector of word counts, ignoring the order of words and their grammatical relationships.

TF-IDF model, on the other hand, assigns weights to words based on their frequency in a document relative to their frequency in the entire corpus. This helps to highlight important words that are more informative for the specific document.

2.2.2 Prediction-based Methods

Prediction-based methods are word representation techniques that learn embeddings by predicting words from context (or vice versa), unlike count-based methods which rely on raw frequency statistics. The two most common prediction-based methods are Word2Vec and BERT which are based on neural networks and their representation of words is contained in dense matrix different from the sparse matrix used in count-based methods. The dense matrixes turns out to be more efficient and effective for representing the meaning of words in a lower-dimensional space.[6]

Word2vec is a framework used for calculating a static embeddings, that mean there is a certain numerical representation for each word in vocabulary. There are two architectures for training Word2Vec models: Continuous Bag of Words (CBOW) and Skip-Gram.

1. **Continuous Bag of Words (CBOW)**: predicts the target word from surrounding context words.
2. **Skip-Gram** Skip-Gram: predicts context words from a single target word.

Skip-Gram is particularly effective for representing rare words, as it captures word–context associations more accurately, whereas CBOW offers greater computational efficiency and faster training times [7].

BERT (Bidirectional Encoder Representations from Transformers) is a deep learning model introduced by Google that generates contextualized word embeddings by processing text in both directions (left-to-right and right-to-left) simultaneously. Unlike earlier models that produce static embeddings, BERT captures the meaning of a word based on its entire sentence context. Transformer architecture, the backbone of BERT, employs self-attention mechanisms to weight the importance of words in relation to each other, allowing the model to derivate complex semantic relationships between them. To achieve this, BERT is pretrained using a two-step self-supervised training process:

1. **Masked Language Model (MLM)**: Randomly masks a percentage of input tokens and trains the model to predict the masked words based on context.

2. **Next Sentence Prediction (NSP)**: Trains the model to predict whether two sentences are follow each other or not, improving its understanding of sentences coherence.

These context-aware embeddings have demonstrated strong performance across various NLP tasks, including classification.[3]. In this study, embeddings generated by pre-trained BERT models serve input features for classification task which will categorize products into predefined expense-related categories.

2.3 XGBoost Classifier

XGBoost stands for eXtreme Gradient Boosting and is an open-source machine learning library that implements the gradient boosting framework for decision trees. It is an state-of-the-art algorith for supervised learning that achives best performance among many machine learning challanges.[2] Gradient boosting is an ensemble learning technique that instead of training a single model, build an initial model and then iteratively fits new models through loss function minimization.[8] XGBoost model use set of rules in tree generation.

1. Regularization which is used to prevent too complex trees and tends to choose the most predictive ones while also minimizing overfitting.
2. Gradient Tree Boosting introduces a second-order Taylor approximation to the loss function, incorporating both the first (gradient) and second (Hessian) derivatives. This allows accurate estimation while ensuring high efficiency of the algorithm.
3. Shrinkage is a technique that after each boosting step, the prediction scores are multiplied by a learning rate parameter. This prevent overfitting and allows for more robust models that learn more slowy and carefully.
4. Column Subsampling: parameter that derives from random forest algorithm, randomly selects part of features for each tree.

2.4 Integration of OCR and NLP

[...]

3 Methodology

3.1 System Architecture

[...]

3.2 Data Collection and Preprocessing

[...]

3.3 Optical Character Recognition with Tesseract

[...]

3.4 Text Parsing and Information Extraction

[...]

3.5 Product Categorization Using BERT

[...]

3.6 Data Grouping and Organization

[...]

4 Implementation

4.1 Development Environment

[...]

4.2 Integration of Tesseract OCR

[...]

4.3 BERT Model Fine-Tuning

[...]

4.4 System Workflow

[...]

5 Evaluation and Results

5.1 Evaluation Metrics

[...]

5.2 Experimental Setup

[...]

5.3 Results and Analysis

[...]

5.4 Comparison with Existing Methods

[...]

6 Discussion

6.1 Interpretation of Results

[...]

6.2 Challenges and Limitations

[...]

6.3 Recommendations for Future Work

[...]

7 Conclusion

7.1 Summary of Findings

[...]

7.2 Contributions to the Field

[...]

7.3 Final Remarks

[...]

8 References

[...]

9 Appendices

9.1 Sample Receipt Data

[...]

9.2 Code Snippets

[...]

9.3 Additional Figures and Tables

[...]

Bibliography

[1] Felipe Almeida i Geraldo Xexéo. *Word Embeddings: A Survey*. 2023. arXiv: 1901.09069 [cs.CL]. URL: https://arxiv.org/abs/1901.09069.

[2] Tianqi Chen i Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". W: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. ACM, sierp. 2016, s. 785–794. DOI: 10.1145/2939672.2939785. URL: http://dx.doi.org/10.1145/2939672.2939785.

[3] Jacob Devlin i in. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL]. URL: https://arxiv.org/abs/1810.04805.

[4] Yuning Du i in. "PP-OCR: A Practical Ultra Lightweight OCR System". W: *arXiv preprint arXiv:2009.09941* (2020).

[5] Noman Islam, Zeeshan Islam i Nazia Noor. "A Survey on Optical Character Recognition System". W: *ITB Journal of Information and Communication Technology* (grud. 2016). DOI: 10.48550/arXiv.1710.05703.

[6] Daniel Jurafsky i James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. 3rd. Online manuscript released January 12, 2025. 2025. URL: https://web.stanford.edu/~jurafsky/slp3/.

[7] Tomas Mikolov i in. *Efficient Estimation of Word Representations in Vector Space*. 2013. arXiv: 1301.3781 [cs.CL]. URL: https://arxiv.org/abs/1301.3781.

[8] Alexey Natekin i Alois Knoll. "Gradient Boosting Machines, A Tutorial". W: *Frontiers in Neurorobotics* 7 (2013), s. 21. DOI: 10.3389/fnbot.2013.00021.

[9] Baoguang Shi, Xiang Bai i Cong Yao. "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition". W: *arXiv preprint arXiv:1511.04176* (2016). DOI: 10.48550/arXiv.1511.04176.

[10] Ray Smith. "An overview of the Tesseract OCR engine". W: *Ninth International Conference on Document Analysis and Recognition (ICDAR)*. IEEE. 2007, s. 629–633.

[11] Ray Smith. “History and future of the Tesseract OCR engine”. W: *Document Recognition and Retrieval XX*. T. 8658. SPIE. 2013, s. 865802.

[12] Nishant Subramani i in. “A Survey of Deep Learning Approaches for OCR and Document Understanding”. W: *CoRR* abs/2011.13534 (2020). arXiv: 2011.13534. URL: https://arxiv.org/abs/2011.13534.

Wyrażam zgodę na udostępnienie mojej pracy w czytelniach Biblioteki SGGW w tym w Archiwum Prac Dyplomowych SGGW.

.....
(czytelny podpis autora pracy)