# PDF Extraction API Evaluation Template

**Team: TEAM 4**

**Team members: Tanvi Satish Inchanalkar, Yuga Santosh Kanse, Sarthak Shailesh Somvanshi**

**Link to your analysis:** [**https://docs.google.com/document/d/1qDShZuGxuYcaMlCq4teq80jhdpqmhUyKspXCs-Ynwdg/edit?tab=t.0**](https://docs.google.com/document/d/1qDShZuGxuYcaMlCq4teq80jhdpqmhUyKspXCs-Ynwdg/edit?tab=t.0)

**Summary:**

This evaluation compares two tools for PDF extraction: PyPDF (an open-source Python library) and Azure Document Intelligence (a cloud-based AI service). PyPDF offers a cost-effective, community-supported solution for basic PDF text extraction but lacks advanced capabilities like handling complex layouts and images. It is best suited for smaller projects or cost-sensitive scenarios. Azure Document Intelligence, on the other hand, provides high accuracy in text and table extraction, supports complex document structures, and includes robust OCR capabilities. It integrates well with enterprise-grade workflows, offers scalability, and is backed by Microsoft's compliance standards. However, it comes with a higher cost, which is justified by its advanced features and performance. The choice between the two depends on project needs: PyPDF is ideal for those seeking a free, script-based solution, while Azure is recommended for scalable, high-accuracy requirements.

## 1. General Information

|  |  |
| --- | --- |
| Attribute | Details |
| API Name | PyPDF, Azure Document Intelligence |
| Vendor | PyPDF: Community-driven, Azure: Microsoft |
| Version/Release Date | |  | | --- | |  |  |  | | --- | | PyPDF: Latest version used, Azure: 3.2.0 | |
| Pricing Model | PyPDF: Free, Azure: S0 Pricing Tier |
| Licensing and Compliance | PyPDF: MIT License, Azure: GDPR, HIPAA compliant |

## 2. Technical Capabilities

|  |  |  |
| --- | --- | --- |
| Feature | PyPDF (Open Source) | Azure Document Intelligence (Enterprise) |
| File Format Support (PDF, DOCX, etc.) | PDF | PDF, JPG, PNG, DOCX, and more |
| OCR (Optical Character Recognition) | Basic OCR when combined with Tesseract | Advanced OCR for printed and handwritten text |
| Table Extraction | Limited accuracy | High accuracy, auto-detects tables |
| Form Extraction | Not supported | Extracts forms, checkboxes, etc. |
| Complex Layout Support | Basic layout handling | Handles complex layouts like columns, embedded images |
| Multi-language Support | Limited | Supports multiple languages |
| Scalability and Performance | Suitable for smaller datasets | |  | | --- | |  |  |  | | --- | | Highly scalable, processes large datasets efficiently | |
| API Integration and Usability | Easy to integrate with Python | Well-documented SDKs for Python and other languages |
| Customization Options | Limited | |  | | --- | |  |  |  | | --- | | Offers customization for extraction rules | |
| Accuracy and Error Handling | Depends on PDF structure | High accuracy with detailed error handling |

## 3. Business and Strategic Considerations

|  |  |  |
| --- | --- | --- |
| Evaluation Metric | PyPDF (Open Source) | Azure Document Intelligence (Enterprise) |
| Cost Efficiency (Pricing vs. Features) | Free but feature-limited | S0 Pricing: $10 per 1,000 pages for prebuilt models, $30 per 1,000 pages for custom extraction. |
| Vendor Reputation and Stability | Community-driven | Backed by Microsoft, highly reliable |
| Customer Support and SLA | Community support | |  | | --- | |  |  |  | | --- | | Professional support with SLA options | |
| Security and Privacy | Data processed locally | Data encryption, complies with GDPR, HIPAA |
| Documentation and Training Resources | Limited online resources | Extensive documentation and training |
| Community and Ecosystem | Small, active community | Large user base, integrations available |
| Roadmap and Innovation | Limited roadmap | Continuous innovation by Microsoft |
| Vendor Lock-in Risk | No vendor lock-in | Possible, but well-documented for migration |

## 4. Performance Metrics

|  |  |  |
| --- | --- | --- |
| Metric | PyPDF (Open Source) | Azure Document Intelligence (Enterprise) |
| Latency | Varies based on PDF size | Generally low latency |
| Throughput | ~10-20 pages/minute | ~50-100 pages/minute |
| Error Rate | Frequency and types of errors during processing. | Low error rate, especially with complex documents |
| Data Loss/Integrity | May miss structured data like tables | |  | | --- | |  |  |  | | --- | | High data integrity | |

## 5. Value-Add Features

|  |  |  |
| --- | --- | --- |
| Feature | PyPDF (Open Source) | Azure Document Intelligence (Enterprise) |
| Advanced AI/ML Capabilities | None | AI-based extraction for better understanding |
| Pre-built Templates for Specific Use Cases | None | Available for finance, healthcare, etc. |
| Document Classification/Tagging | Not available | Available |
| Metadata Extraction | Limited | Extracts detailed metadata |

## 6. Overall Evaluation

|  |  |  |
| --- | --- | --- |
| Attribute | Rating | Comments |
| Technical Fit | PyPDF: 6/10, Azure: 9/10 | Azure provides better support for complex layouts and OCR accuracy. |
| Business Fit | PyPDF: 7/10, Azure: 8/10 | PyPDF is cost-effective; Azure is feature-rich and scalable. |
| Total Cost of Ownership | PyPDF: 9/10, Azure: 6/10 | PyPDF is free, but Azure may become costly with large volumes. |
| Ease of Implementation and Use | PyPDF: 7/10, Azure: 9/10 | Azure is easier to integrate with various tools. |
| Vendor Reliability and Support | PyPDF: 6/10, Azure: 9/10 | Azure offers enterprise-level support and reliability. |

## 7. Recommendations

|  |  |
| --- | --- |
| Recommendation | Details |
| Best Fit for the Use Case | Azure Document Intelligence is preferred for enterprise needs with scalable pricing through the S0 tier. PyPDF is suitable for small-scale or budget-conscious projects. |
| Further Considerations | Consider the S0 pricing if using Azure for a high volume of documents to take advantage of the per-page rates for prebuilt and custom models. |