



Catalysing AI tools and talent.

AI Singapore accelerates deep learning models with Adobe PDF Extract API.



Established
2017

Employees: 90
Singapore

<https://aisingapore.org>

40%

Reduction in number of sprints needed to deliver deployable machine learning (ML) model

Products:

[Adobe Acrobat Services >](#)
[Adobe PDF Extract API >](#)

Objectives

Improve quality of data ingested by natural language processing (NLP) classification pipeline

Provide only relevant content needed for corporate sustainability reporting objectives

Develop minimum viable machine learning model within 10 sprints and seven months

Results

Extracted PDF files with **high accuracy** to prove concept for 500 disparate documents

Implemented PDF Extract API in **just two weeks**

Provided data with **better context and structure** for superior model results

Delivered deployable model **40% faster** than planned

When companies in Singapore face a tough challenge and want to explore using artificial intelligence (AI) to achieve their organisation's goals, they turn to AI Singapore (AISG). Launched to catalyse Singapore's capabilities to power its future digital economy with AI, the national programme brings together all Singapore-based research institutions and the vibrant ecosystem of AI start-ups and companies developing AI products to perform use-inspired research, grow the knowledge, create the tools, and develop the talent to power Singapore's AI efforts.

Siavash Sakhavi is Assistant Head of the flagship 100 Experiments (100E) programme, which helps solve organisations' artificial intelligence (AI) problem statements and assists them with building their own AI teams. An organisation may propose 100E problem statements where no commercial off-the-shelf (COTS) AI solution exists, which can potentially be solved by Singapore's ecosystem of researchers and AI Singapore's engineering team within 9 to 18 months.

Sakhavi's team was challenged recently in a corporate sustainability reporting project with a large, multinational financial services client. The client had difficulties extracting text from disparate reports and brochures from various sources in the form of PDF documents. The project team, consisting of several AI, data, and platform engineers, plus AISG apprentices, wanted to feed the extracted information into a natural language processing (NLP) classification pipeline. However, the project team noticed the pipeline was not performing as expected because of large volumes of unstructured, gibberish text being returned by the PDF extractor tools they were using.

For typical projects, the team is able to develop models within one to two months. In this situation, it was already on sprint six of ten, so pressure was building to produce results. Fortunately, they heard about [Adobe PDF Extract API](#), which at the time was emerging from beta to general availability. A new web service from Adobe, PDF Extract API, parses data and context from native and scanned PDF files, extracting text, table, and image elements within a structured JSON file.

“Adobe PDF Extract API was a lifesaver. Without it, we could not have used such disparate inputs to make our designed NLP solution in time.”

Siavash Sakhavi
Assistant Head, 100E, AI Singapore

Shifting gears with better structure and content extraction

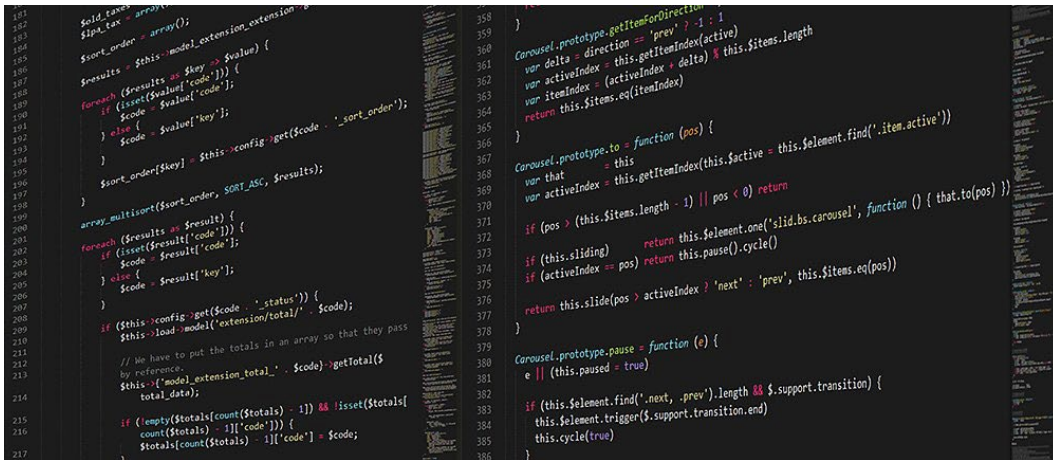
“One demo later, we decided to completely switch to Adobe PDF Extract API, because it looked extremely promising,” says Sakhavi. By the end of the sprint, the team was making great progress. It was soon able to rapidly scale ingestion to its NLP pipeline, ultimately delivering the promised work to the project sponsor ahead of schedule.

“Adobe PDF Extract API was a lifesaver. Without it, we could not have used such disparate inputs to make our designed NLP solution in time,” Sakhavi says. “Up to that point, we had been building and refining how the machine learning model should work. But the other PDF extractors were giving us terrible results, creating a bottleneck in generating quality text inputs for our model.”

The open source extraction tools the team had been testing were unable to accurately identify paragraphs. Many of the sentences delivered were cut off or otherwise unusable and had to be thrown out. Others were missed entirely. Many times, numbers and labels from charts were improperly extracted as body text. The outputs lacked structure entirely.

“It's not an easy task to go in manually and extract what you want from these texts and eventually classify them. We had to go out and find a better PDF extractor to provide automation and efficiency for our work,” says Sakhavi. “Thankfully we were able to plug in PDF Extract API just in time, right when we needed to start quickly ramping up our NLP processing volume.”

PDF Extract API delivered outputs based on paragraphs instead of just sentences and fragments. “The context data and capabilities with grouping paragraphs proved tremendously valuable. This really boosted our ingestion pipeline and drove better outcomes in our machine learning algorithm as well,” Sakhavi says.



```
181 sold_tax = array();
182 slipa_tax = array();
183
184 $sort_order = array();
185
186 $results = $this->model->extension->extension->
187
188 foreach ($results as $key => $value) {
189     if (isset($value['code'])) {
190         $code = $value['code'];
191     } else {
192         $code = $value['key'];
193     }
194     $sort_order[$key] = $this->config->get($code - '_sort_order');
195 }
196
197 array_multisort($sort_order, SORT_ASC, $results);
198
199
200 foreach ($results as $result) {
201     if (isset($result['code'])) {
202         $code = $result['code'];
203     } else {
204         $code = $result['key'];
205     }
206     if ($this->config->get($code - '_status') == 'active') {
207         $this->load->model('extension/total/' . $code);
208     }
209
210 // We have to put the totals in an array so that they pass
211 // by reference.
212 $this->load->model('extension/total/' . $code)->getTotal($
213     total_data);
214
215 if (empty($totals[count($totals) - 1]) && !isset($totals[
216     count($totals) - 1]['code'])) {
217     $totals[count($totals) - 1]['code'] = $code;
218 }
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Increasing relevancy for superior model results

“Our results were amazing,” Sakhavi says. “PDF Extract API plugged in very nicely. We made an adaptor for it to only extract exactly what we needed and ultimately the API was impressively accurate, enabling a quick acceleration against our project schedule.”

The AISG team planned to develop a Bidirectional Encoder Representations from Transformers (BERT) deep learning model. To feed the model, the project sponsor provided a keyword glossary definition for factors, each belonging to a certain label, that were important to a particular corporate Environmental, Social, and Governance (ESG) initiative. The AISG team's goal was to perform a similarity matching process between those definitions and the documents being analysed, to determine which parts of the text were relevant for the project.

“The context for these texts was really domain-specific related to environmental sustainability, so we couldn't train a model until we had extracted relevant text to match with the glossary information,” says Sakhavi. “PDF Extract API was able to correctly identify important sentence-level information in the context of paragraphs about topics related to sustainability, providing a high quality of data for ingestion.”

“Our results were amazing. PDF Extract API plugged in very nicely. We made an adaptor for it to only extract exactly what we needed and ultimately the API was impressively accurate, enabling a quick acceleration against our project schedule.”

Siavash Sakhavi
Assistant Head, 100E, AI Singapore

The corporate sponsor wanted to analyse 500 disparate documents from different sources, looking to Sakhavi's team to take care of the first 10 while proving the concept, before handing over a full-fledged, deployable model that would allow the company to handle the remaining 490. “The reports and brochures involved were not straightforward by any means. There were a lot of images, in addition to text elements floating around on different parts of pages. The sizes and content arrangements had a lot of variability,” he says.

Sakhavi and his team were able to implement PDF Extract API in just two weeks, for a massive improvement in data quality. "It was a very short learning curve," he says. The team quickly learned how to incorporate lists of data into its pipeline, parsing everything based on path attributes to achieve the desired results.

"At the end of sprint six we were able to present promising results to our corporate client turning the corner to enable a production-worthy solution with a bit more work from our team," Sakhavi says.

Fostering an innovative Singaporean AI ecosystem

The project scope has since been expanded for research probing in a new area. It's been handed over to a university in Singapore, working in tandem with the AI Engineering team of AISG. The good news for the corporate sponsor is that the same people who worked on the AISG project team joined the university research team to continue helping bring this initiative to fruition, ultimately training a new model using the labeled data and producing superior results.

"AISG continues to have many projects that require PDF parsing," Sakhavi says. "My recommendation to all teams is to utilise PDF Extract API and also suggest it to the sponsoring clients for their ongoing use."

