Research Paper

JIS

Journal of Information Science
1–13
© The Author(s) 2021
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/01655515211000642
journals.sagepub.com/home/jis
SAGE

Searchable Turkish OCRed historical newspaper collection 1928–1942

Houssem Menhour
Department of Computer Engineering, Kocaeli University, Turkey

Hasan Basri Şahin
Department of Information Systems Engineering, Kocaeli University, Turkey

Ramazan Nejdet Sarıkaya
Department of Computer Engineering, Kocaeli University, Turkey

Medine Aktaş
Department of Computer Engineering, Kocaeli University, Turkey

Rümeysa Sağlam
Department of Computer Engineering, Kocaeli University, Turkey

Ekin Ekinci
Department of Computer Engineering, Sakarya University of Applied Sciences, Turkey

Süleyman Eken
Department of Information Systems Engineering, Kocaeli University, Turkey

Abstract

The newspaper emerged as a distinct cultural form in early 17th-century Europe. It is bound up with the early modern period of history. Historical newspapers are of utmost importance to nations and its people, and researchers from different disciplines rely on these papers to improve our understanding of the past. In pursuit of satisfying this need, Istanbul University Head Office of Library and Documentation provides access to a big database of scanned historical newspapers. To take it another step further and make the documents more accessible, we need to run optical character recognition (OCR) and named entity recognition (NER) tasks on the whole database and index the results to allow for full-text search mechanism. We design and implement a system encompassing the whole pipeline starting from scrapping the dataset from the original website to providing a graphical user interface to run search queries, and it manages to do that successfully. Proposed system provides to search people, culture and security-related keywords and to visualise them.

Keywords

Data journalism; full-text search; historical newspapers; microservices; named entity recognition; optical character recognition; visualisation

Corresponding author:
Süleyman Eken, Department of Information Systems Engineering, Faculty of Technology, Kocaeli University, Umuttepe Campus, İzmit 41001, Kocaeli, Turkey.
Email: suleyman.eken@kocaeli.edu.tr

1. Introduction

With the digitalising world, the information produced is increasing day by day, so it is challenging to automatically organise, analyse, summarise, understand and obtain the desired information from the large scale of document collections. More difficulties arise when it comes to historical documents and text-based searches on them. To accomplish these challenging tasks, terms that describe the documents are needed. Terms describing the collections of documents are called keywords. Moreover, the named entities in the document also inform us about the document. As a result, in order to search these historical documents, we need to be mining on them.

Document mining constitutes an important step in many natural language processing (NLP) tasks such as information extraction and retrieval, machine translation, sentiment analysis, syntactic segmentation, multimedia indexing and question-answering systems. Named entity recognition (NER) is the process of finding entities in a text and assigning them to one of the predefined classes such as person, location, date, formula, percentage, organisation and money [1]. With all these, NER is not limited to these data types, but is also used to identify and mark entities specific to the relevant area in studies in different fields. E-mail addresses [2], phone numbers, book titles, project names, gene/protein names in bioinformatics and chemistry texts [3], RNA, DNA, cell information, drug names [4] and chemical names [5] as entity names are the subjects studied. In this article, we recognise named entities in historical newspapers and then do full-text search.

NER operation was performed in different languages, mainly in English, Arabic [6], Russian [7], Chinese [8], Finnish and Japanese [9]. In recent years, many studies have been done on Turkish NER. Özger and Diri [10] performed the identification and labelling of the names of organisations, locations, people and date, money, hour for Turkish documents by means of a subject-independent and a rule-based approach. Dalkılıç et al. [11] developed a method for Turkish based on some grammatical rules of the language and tried it on different documents. Özkaya and Diri [12] used conditional random fields (CRFs) to realise the recognition of three different entity names namely person, organisation and location by deducting some rules on unofficial Turkish e-mails. Küçük et al. [13] presented a dataset of news texts marked with entity names for Turkish. Güngör et al. [14] devised a method that uses recursive neural networks for the Turkish NER task. Güneş and Tantuğ [15] developed bidirectional long short-term memory (BLSTM) and deep bidirectional long short-term memory (DBLSTM) models for Turkish NER operations. Eken [16] proposed an NER system for Turkish tweets (in short texts) in her thesis. Sarı and Aktaş [17] developed a rule-based NER model for Turkish history and geography course texts. Arslan et al. [18] used Apache Lucene, a big data indexing and search software, to detect the presence of email, web addresses and emoji in half a billion web pages. Çekinel et al. [19] recognised named entities in news texts using the CRF method. Akpınar et al. [20] proposed a tool that enables semi-automatic tagging of entities and their relationships, based on bank transactions in image format. All these works are related to Turkish NER tasks on modern documents.

Contributions to the literature with the paper can be listed as follows:

1. NER and optical character recognition (OCR) tasks are done on Turkish historical newspapers.
2. A full-text-based mechanism for allowing searchable historical newspapers is developed.
3. Location-based entities are visualised on the maps.
4. Historical newspapers are serviced for the history, culture and security-related researches to understand organisational and national issues.

The rest of this article is organised as follows: Section 2 focuses on NER-based searchable historical newspapers for different languages. Section 3 describes the presented system architecture with its submodules. Section 4 deals with the realised experiments. In the last section, we discuss the research results and we propose some future research directions.

2. Related works

Historical texts such as newspapers, letters, diaries, medical documents, court reports and so on are important documents that shed light on the past, and are being digitised, OCRed, annotated, stored and published on the web in order to be reachable by anyone who wants. As a result, these texts especially have attracted the attention of researchers and they have searched for methods to extract information from them. Therefore, many studies have been done on this subject from past to present and studies are still ongoing.

When the literature is evaluated, it is seen that users search information from newspapers mostly by using names of people, locations, organisations and so forth [21]. Considering this situation, it can be easily said that NER has an important function in information search. Crane and Jones extracted NERs from newspapers from the 19th century to make collections suitable for searching [21]. The NERs extracted by authors within the scope of the study were personal names, locations, dates, products, organisations, streets, newspapers, ships, regiments and railroads. Because the success in extracting some NERs was low, they suggested using well-defined rule sets, complete knowledge sources and lists prepared by experts as training data. They just focused on 10 different NERs and evaluated them manually. Here, we extract three main NERs, visualise the location-dependent ones, and indexing and searching mechanisms are operated. To enhance searching of Litteraturbanken, Swedish literary classics, Borin et al. [22] applied a generic NER system. The generic NER system worked as an inclusion of word similarity to classical rule-based NER and had capability of obtaining eight named entities with an *F*-measure of 92.8%. There are no search and browsing interfaces for extracted NERs in their system. Kettunen et al. [23] used Finnish rule-based NER tool FiNER in a historical Finnish OCRed newspaper collection between 1771 and 1910 to perform better searching of the information from historical newspapers. With FiNER, they extracted person, location (general, geographical, political), address, educational organisation, company and time expressions but OCR errors caused poor performance in accurate extraction of these named entities. Although it is the closest article to the work we do, no connection of some type of functional NER to their user interface of historical newspaper collection. In the study by Kettunen and Ruokolainen [24], the same Finnish dataset was used to select the best NER tool among FiNER, FST, Connexor and Polyglot. They concentrated on locations, person names, organisation and dates/time and realised that results were extremely low because of bad quality of historical texts, which caused OCR to give bad results. Similarly, we use Enamex style annotation for named entity boundaries and types. Ruokolainen and Kettunen [25] wanted to improve NER results caused by the OCR errors. Apart from NER, OCR errors also negatively affected the search results and hence OCR quality was improved. The authors used the historical Finnish OCRed newspaper collection as before to compare the results and it was seen that as the OCR quality improved, the results also increased. Different from other studies, Labusch et al. [26] used deep learning methods and embedded languages for NER extraction for both historical German texts and contemporary ones for use in search and achieved good results. They did not concern the level of noise in the source OCR texts by means of re-processing the digitised documents. In the study by Ruokolainen and Kettunen [27], historical newspapers and journals published in Finnish between the 19th and early 20th century were used to extract NERs to improve searching. NER extraction was realised with Stanford NER and LSTM-CRF models on OCRed data and manually corrected ground truth data. The Stanford NER and LSTM-CRF models had quite similar success for both datasets, while the corrected dataset was the most successful in extracting NERs with both methods. There are no search and browsing interfaces for extracted NERs in their system. They believed that the La Stampa style of usage of names in the graphical user interface (GUI) of a newspaper collection was more informative and useful for users.

A European Commission–funded project Europeana Newspapers was developed to access historical newspapers from 23 European libraries with digitised 10 million historic newspaper pages [28]. The explicit aim of the project was to make newspaper pages fully searchable. To achieve this, the authors first digitised pages with OCR and then by using OLR they obtained each article separately. The project was also a pioneer for many studies. Neudecker and Antonacopoulos [29] wanted to improve the full search result quality of the available approaches for the Europeana Newspapers project. In the first step, newspapers were organised into files and directories; then binarisation and format change were applied to page images; at the end, different OCR and OLR software were used for good results. Apart from these, NER was also applied to three languages for better refinement. In another study conducted on three languages namely Dutch, French and German in Europeana Newspaper, NER extraction was proposed to provide full-text search with people, locations and organisations [30]. Similarly, we do indexing and searching mechanisms for the same NERs and also visualisation for them. NERs were extracted with high precision using Stanford Named Entity Recognizer over 100 pages for each language.

In Apperley et al. [31], the authors used the Greenstone system for full-text search in The Niupepa collection of historic Maori newspapers. First, by using OCR, newspapers were converted into electronic format; and by using page-level index, page-by-page searching was provided. The Niupepa collection incorporates a page-level index, with text for each page held in a separate file. Here, we use the Elasticsearch engine for indexing data. Gatos et al. [32] devised a segmentation-free algorithm by combining image preprocessing, segmentation, synthetic data creation, word spotting and user's feedback technologies to extract keywords from historical type-written documents. Ahonen and Hyvonen [33] composed a semantic search system for historical texts of the National Library of Finland using a common ontology where cultural objects were annotated semantically. In their work, a multi-faceted semantic search interface was applied to the annotated material. We search on the extracted NERs. Jerele et al. [34] used OCR and a lexicon to simplify full-text search in Slovenian books and newspapers from the 18th and 19th centuries. In this study, while OCR digitised texts, lexicon connected historical words to their contemporary ones. Terasawa et al. [35] claimed that the image quality of historical newspapers was low and that OCR use would not be successful on these images. Hence, the authors customised appearance-based algorithms for full-text search. The algorithm reduced the problem of full-text search to the sequence matching based on character to character matching with its shapes. Also, for decreasing computational burden, pseudo-code expression LSPC was adopted to algorithm. As a result, better results were obtained in terms of search and computational cost. They evaluated the method in terms of the searching accuracy regarding different search keyword length. We do stress tests for different search queries. Thompson et al. [36] devised a web-based History of Medicine tool by benefitting from text mining methods to provide its user an efficient search from historical texts, which were British Medical Journal and London Medical Officer of Health reports. The tool presented its user term, bibliographic metadata, entity, event and named entity–based search. In another study on historical medical text, it was aimed to extract concepts, relations and NERs to make ready these texts for searching purpose [37]. To accomplish this task, the authors took advantage of NER tools and medical ontologies. Wilkinson at al. [38] benefitted from deep learning methods for semantic searching in three historical text datasets. For this aim, a segmentation-free query-by-string word-spotting model called as Ctrl-F-Net was devised. The working of the model was as follows: first region proposals were obtained in the page and then these proposals embedded into word embedding space where the search was taken place. Kettunen et al. [39] used the PIVAJ machine learning–based platform to obtain articles on the same page separately and made an experiment on Finnish journal Uusi Suometar 1869–1898. The purpose of doing this was to provide article-by-article search for refining search quality. In the study by Bogaard et al. [40], it was aimed to understand user behaviour by using metadata categorisation to achieve functional search in historical newspaper collection. It simultaneously produced region proposals and embedded them into a word-embedding space in which searches were performed. Atay et al. [41] developed an architecture that makes content-based figure searches possible on these scanned documents in large quantities. The user can search with some keywords and display related figures in digital documents with their captions. Although there are some studies in the literature on searchable historical papers/newspapers, to our knowledge, no previous study has been performed on searchable OCRed Turkish newspapers.

3. System architecture

The first step in this system is to build the data infrastructure and pipeline from scratch. This infrastructure needs to be robust, scalable and reliable. The architecture we chose to use can be described as a collection of microservices, each of which is dedicated to one small task or part of the system. The whole system, as shown in Figure 1, consists of:

- CouchDB instance;
- Elasticsearch instance;
- Application programming interface (API);
- GUI;
- Data collector and processor.

3.1. Data collection

3.1.1. Scraping the dataset website and downloading the data. Istanbul University Head Office of Library and Documentation has approximately 18,422 volumes of newspapers in the collection of the Central Library. The newspapers had been published in various languages such as Ottoman, Greek, Bulgarian, French, German, English and Turkish. To preserve this valuable newspaper archive, which will shed light on the history of the period, and to put it into the service of researchers, the ‘Gazeteden Tarihe Bakış Projesi’ [42] was carried out. Within the scope of the project, approximately 688 volumes, 55 headings, 581,106 pages of images from national and local newspapers covering the years 1928–1942 were scanned by OCR method and transferred to electronic media, for a total of 511 GB of data. These scanned newspapers are accessible daily. Figure 2 shows some parts of scanned newspapers from the dataset. In this article, we use only Turkish OCRed historical newspapers.

To download the dataset, we write our own script using the Python libraries BeautifulSoup [43] and Requests [44]. Together, they allow us to query the website of the dataset and parse through newspaper names, years, months and the issue number. Once we reach a newspaper release, we download its PDF and store its metadata such as the newspaper name, date and original URL. Then, we run a parallel downloading job to accelerate the download process. However, this results in the connection getting closed by the server, so we had to resort to reducing the number of parallel jobs to a maximum of 4, as well as adding a 100 ms delay between requests.

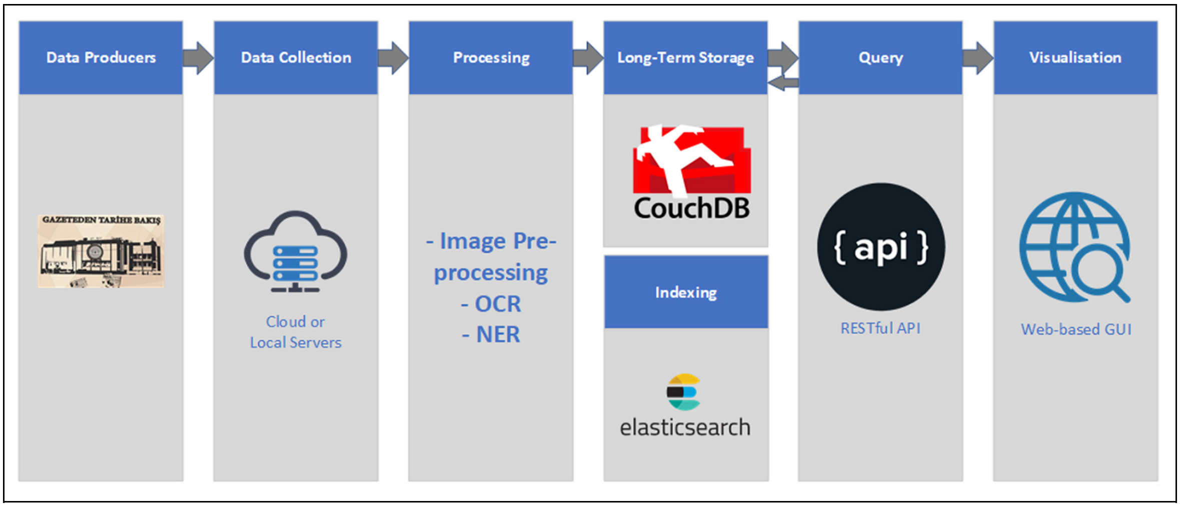


Figure 1. Proposed system architecture.

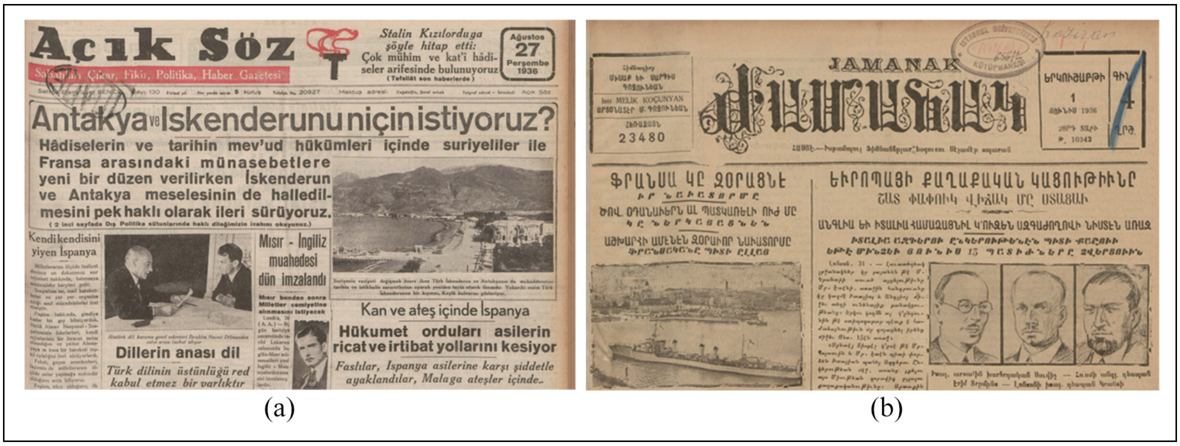
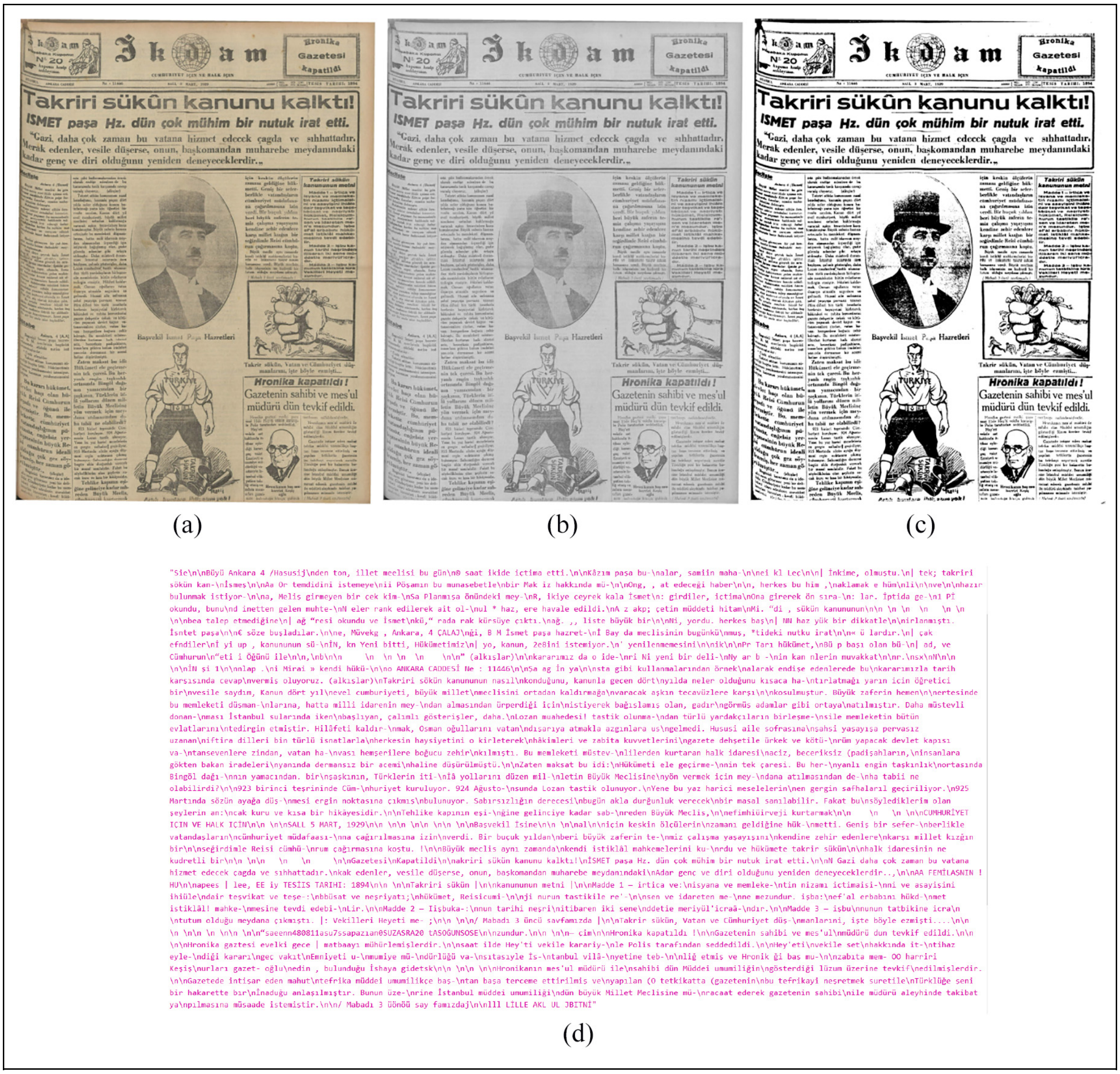


Figure 2. Some parts of scanned newspapers from dataset: (a) Açık Söz, 27 August 1936 and (b) Jamanak, 1 June 1936.

3.2. Data preprocessing steps

3.2.1. Processes the PDFs to extract and clean single pages. Document understanding is usually done on scanned documents/images. Lovegrove and Braisford [45] made the first study to identify combined lines from scanned PDF images. With the popular PDF format, Anjewierden [46] extracted text and graphic objects from PDFs with xpdf. Later, Hadjar et al. [47] also compared the performance results of object extraction libraries from the PDF. Chao and Fan [48] tried to extract text and figure objects directly from PDF code (content analysis) and vector objects such as lines and tables (layout-design analysis) from bitmap images. Hassan [49] proposed a bottom-up clustering algorithm according to visual principles.

It is possible to miss-recognise some characters with OCR. Therefore, it is inevitable to perform an NLP-supported correction with OCR. There are different approaches to OCR post-correction such as machine learning, combining multiple system outputs or high-frequency words based. Niklas [50] combined several methods to ensure the best correction of misspelled words: general spelling (Anagram Hash), OCR adaptive character form and bigram. This approach was conducted manually on the London Times Newspaper Archive, which was published in the English newspaper between 1785 and 1985 and the error reduction rate was increased to 75%. In 2007, Hauser [51] developed an edit distance for OCR post-correction. Généreux and Spano developed the SimString library [52], which uses the CPMerge [53] algorithm to provide fast approximate sequence matching.



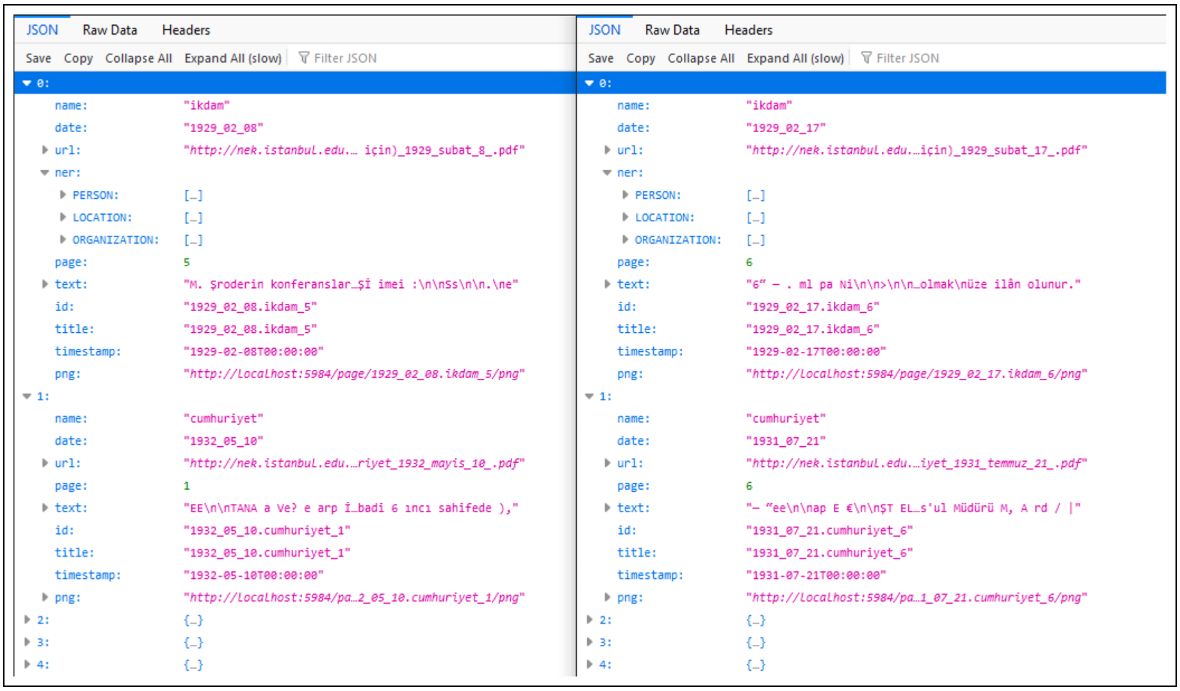
```
"Sie\n\nBüyü Ankara 4 /Hasusij\nden ton, illet meelisi bu gün\n® saat ikide içtima etti.\n\nKâzım paşa bu-\nalar, samiin maha-\n\nei kl Leç\n\n| İnkime, olmuştu.\n| tek; takriri
sökün kan-\nİsmeş\n\nAa Or temdidini istemeye\nii Pöşamın bu munasebetle\nbir Mak iz hakkında mü-\n\nOng, , at edeceği haber\n\n, herkes bu him ,\naklamak e hüm\nli\n\nve\n\nhazır
bulunmak istiyor-\n\na, Meliş girmeyen bir çek kim-\nSa Planmışa önündeki mey-\nR, ikiye çeyrek kala İsmet\n: girdiler, içtima\nOna girerek ön sıra-\n: lar. İptida ge-\n1 Pİ
okundu, bunu\nd inetten gelen muhte-\nN eler rank edilerek ait ol-\nul * haz, ere havale edildi.\nA z akp; çetin müddeti hitam\nMi. "di , sükün kanununun\n\n \n \n  \n   \n \n
\n\nbea talep etmediğine\n| ağ "resi okundu ve İsmet\nkü," rada rak kürsüye çıktı.\nağ. ,, liste büyük bir\n\nNi, yordu. herkes baş\n| NN haz yük bir dikkatle\n\nirlanmıştı.
İsntet paşa\n\n€ söze buşladılar.\n\ne, Müvekg , Ankara, 4 ÇALAJ\nği, B M İsmet paşa hazret-\nİ Bay da meclisinin bugünkü\nmuş, *tideki nutku irat\n\n« ü lardır.\n| çak
efndiler\nİ yi up , kanununun sü-\nİN, kn Yeni bitti, Hükümetimiz\n| yo, kanun, 2e8ini istemiyor.\n' yenilenmemesini\n\nik\n\nPr Tarı hükümet,\nBü p başı olan bü-\n| ad, ve
Cümhurun\n"eti i Öğünü ile\n\n,\nb\n\n     \n  \n \n  \n    \n\n" (alkışlar)\n\nkararımız da o ide-\nri Ni yeni bir deli-\nNy ar b -\nin kan nlerin muvakkat\n\nr.\nsx\nN\n\n
\n\nİN şi 1\n\nnlap .\ni Mirai » kendi hükü-\n\no ANKARA CADDESİ Ne : 11446\n\nŞa ag İn ya\n\nsta gibi kullanmalarından örnek\nalarak endişe edenlerede bu\nkararımızla tarih
karşısında cevap\nvermiş oluyoruz. (alkışlar)\nTakriri sökün kanununun nasıl\nkonduğunu, kanunla geçen dört\nyılda neler olduğunu kısaca ha-\ntırlatmağı yarın için öğretici
bir\nvesile saydım, Kanun dört yıl\nevel cumburiyeti, büyük millet\nmeclisini ortadan kaldırmağa\nvaracak aşkın tecavüzlere karşı\n\nkosulmuştur. Büyük zaferin hemen\n\nertesinde
bu memleketi düşman-\nlarına, hatta milli idarenin mey-\ndan almasından ürperdiği için\nistiyerek bağıslamıs olan, gadır\ngörmüş adamlar gibi ortaya\natılmıştır. Daha müstevli
donan-\nması İstanbul sularında iken\nbaşlıyan, çalımlı gösterişler, daha.\nLozan muahedesi! tastik olunma-\ndan türlü yardakçıların birleşme-\nsile memleketin bütün
evlatlarını\ntedirgin etmiştir. Hilâfeti kaldır-\nmak, Osman oğullarını vatan\ndışarıya atmakla azgınlara us\ngelmedi. Hususi aile sofrasına\nşahsi yaşayışa pervasız
uzanan\niftira dilleri bin türlü isnatlarla\nherkesin haysiyetini o kirleterek\nhâkimleri ve zabita kuvvetlerini\ngazete dehşetile ürkek ve kötü-\nrüm yapacak devlet kapısı
va-\ntansevenlere zindan, vatan ha-\nvası hemşerilere boğucu zehir\nkılmıştı. Bu memleketi müstev-\nlilerden kurtaran halk idaresi\naciz, beceriksiz (padişahların,\ninsanlara
gökten bakan iradeleri\nyanında dermansız bir acemi\nhaline düşürülmüştü.\n\nZaten maksat bu idi:\nHükümeti ele geçirme-\nnin tek çaresi. Bu her-\nyanlı engin taşkınlık\nortasında
Bingöl dağı-\nnın yamacından. bir\nşaşkının, Türklerin iti-\nİâ yollarını düzen mil-\nletin Büyük Meclisine\nyön vermek için mey-\ndana atılmasından de-\nha tabii ne
olabilirdi?\n\n923 birinci teşrininde Cüm-\nhuriyet kuruluyor. 924 Ağusto-\nsunda Lozan tastik olunuyor.\nYene bu yaz harici meselelerin\nen gergin safhalarıl geçiriliyor.\n925
Martında sözün ayağa düş-\nmesi ergin noktasına çıkmış\nbulunuyor. Sabırsızlığın derecesi\nbugün akla durğunluk verecek\nbir masal sanılabilir. Fakat bu\nsöylediklerim olan
şeylerin an:\ncak kuru ve kısa bir hikâyesidir.\n\nTehlike kapının eşi-\ngine gelinciye kadar sab-\nreden Büyük Meclis,\n\nefimhiüirveji kurtarmak\n\n    \n   \n \n\nCUMHURİYET
IÇIN VE HALK IÇİN\n\n \n\nSALL 5 MART, 1929\n\n \n\n \n\n \n\n \n\nBaşvekil İsine\n\n \n\nal\n\niçin keskin ölçülerin\nzamanı geldiğine hük-\nmetti. Geniş bir sefer-\nberlikle
vatandaşların\ncümhuriyet müdafaası-\nna çağırılmasına izin\nverdi. Bir buçuk yıldan\nberi büyük zaferin te-\nmiz çalışma yaşayışını\nkendine zehir edenlere\nkarşı millet kızğın
bir\n\nseğirdimle Reisi cümhü-\nrum çağırmasına koştu. !\n\nBüyük meclis aynı zamanda\nkendi istiklâl mahkemelerini ku-\nrdu ve hükümete takrir sükün\n\nhalk idaresinin ne
kudretli bir\n\n \n\n   \n   \n     \n\nGazetesi\nKapatildi\n\nakriri sükün kanunu kalktı!\nİSMET paşa Hz. dün çok mühim bir nutuk irat etti.\n\nN Gazi daha çok zaman bu vatana
hizmet edecek çagda ve sıhhattadır.\nkak edenler, vesile düşerse, onun, başkomandan muharebe meydanındaki\nAdar genç ve diri olduğunu yeniden deneyeceklerdir..,\n\nAA FEMİLAŞNIN !
HU\n\napees | lee, EE iy TESİIS TARIHI: 1894\n\n \n\nTakriri sükün |\n\nkanununun metni |\n\nMadde 1 — irtica ve:\nisyana ve memleke-\ntin nizamı içtimaisi-\nni ve asayişini
ihiüle\ndair teşvikat ve teşe-:\nbbüsat ve neşriyatı;\nhükümet, Reisicumi-\n\nji nurun tastikile re'-\n\nsen ve idareten me-\nne mezundur. işba:\nef'al erbabını hükd-\nmet
istiklâl! mahke-\nmesine tevdi edebi-\nLir.\n\nMadde 2 — Iişbuka-:\nnun tarihi neşri\nitibaren iki sene\nddetie meriyül'icraâ-\ndır.\n\nMadde 3 — işbu\nnunun tatbikine icra\n
\ntutum olduğu meydana çıkmıştı. |: Vekilleri Heyeti me- ;\n\n \n\n/ Mabadı 3 üncü savfamızda |\n\nTakrir sükün, Vatan ve Cümhuriyet düş-\nmanlarıni, işte böyle ezmişti....\n\n
\n \n\n \n \n\n \n\n"saeenn480811asu7ssapazıan0SUZASRA20 tASOĞUNSOSE\n\nzundur.\n\n \n\n— çim\n\nHronika kapatıldı !\n\nGazetenin sahibi ve mes'ul\nmüdürü dun tevkif edildi.\n\n
\n\nHronika gaztesi evelki gece | matbaayı mühürlemişlerdir.\n\nsaat ilde Hey'ti vekile karariy-\nle Polis tarafından seddedildi.\n\nHey'eti\nvekile set\nhakkında it-\ntihaz
eyle-\ndiği kararı\ngeç vakıt\nEmniyeti u-\nmumiye mü-\ndürlüğü va-\nsıtasıyle İs-\ntanbul vilâ-\nyetine teb-\n\nliğ etmiş ve Hronik ği bas mu-\n\nzabıta mem- OO harriri
Keşiş\nurları gazet- oğlu\nedin , bulunduğu İshaya gidetsk\n\n \n\n \n\nHronikanın mes'ul müdürü ile\nsahibi dün Müddei umumiliğin\ngösterdiği lüzum üzerine tevkif\nedilmişlerdir.
\n\nGazetede intişar eden mahut\ntefrika müddei umumilikçe baş-\ntan başa terceme ettirilmiş ve\nyapılan (O tetkikatta (gazetenin\nbu tefrikayi neşretmek suretile\nTürklüğe şeni
bir hakarette bır\nİnaduğu anlaşılmıştır. Bunun üze-\nrine İstanbul müddei umumiliği\ndün büyük Millet Meclisine mü-\nracaat ederek gazetenin sahibi\nile müdürü aleyhinde takibat
ya\npılmasına müsaade istemistir.\n\n/ Mabadı 3 üönöü say famızdaj\n\nlll LİLLE AKL UL JBITNİ"
```

Figure 3. An example of data preprocessing and OCR result on Ikdam's frontpage from 5 March 1929: (a) raw image, (b) grey image, (c) binary image and (d) OCR result.

Within the scope of this study, each page of the PDF format historical newspapers is first converted into an image and then images are converted into grey ones. After that, grey images are converted to binary forms using Otsu thresholding technique [54]. Finally, OCR task is realised on all binary newspaper pages using pytesseract [55] tool. Python-tesseract is an OCRtool for python. Python-tesseract is a wrapper for Google's Tesseract-OCR Engine. All these preprocessing steps are implemented in multithreaded programming. Figure 3 shows outputs of all these preprocessing steps.

In order to minimise OCR errors, these words are normalised by using Zemberek-NLP [56], an NLP library developed with Java, which is specific to the Turkish language. Spelling errors are corrected with Turkish sentence normalisation method according to Turkish morphology of Zemberek-NLP.

3.2.2. Performing NER. Zemberek-NLP provides a simple NER module for Turkish. This module does not have any model in it; therefore, we need to train our own model. For creating an NER model, we first need a training set, a file that contains correct examples. Training data can be prepared using three annotation methods: bracket style annotation, OpenNLP style annotation and Enamex style annotation.



```
▼ 0:
    name:               "ikdam"
    date:               "1929_02_08"
  ▶ url:                "http://nek.istanbul.edu.… için)_1929_subat_8_.pdf"
  ▼ ner:
    ▶ PERSON:           […]
    ▶ LOCATION:         […]
    ▶ ORGANIZATION:     […]
    page:               5
  ▶ text:               "M. Şroderin konferanslar…Şİ imei :\n\nSs\n\n.\ne"
    id:                 "1929_02_08.ikdam_5"
    title:              "1929_02_08.ikdam_5"
    timestamp:          "1929-02-08T00:00:00"
    png:                "http://localhost:5984/page/1929_02_08.ikdam_5/png"
▼ 1:
    name:               "cumhuriyet"
    date:               "1932_05_10"
  ▶ url:                "http://nek.istanbul.edu.…riyet_1932_mayis_10_.pdf"
    page:               1
  ▶ text:               "EE\n\nTANA a Ve? e arp İ…badi 6 ıncı sahifede ),"
    id:                 "1932_05_10.cumhuriyet_1"
    title:              "1932_05_10.cumhuriyet_1"
    timestamp:          "1932-05-10T00:00:00"
  ▶ png:                "http://localhost:5984/pa…2_05_10.cumhuriyet_1/png"
▶ 2:                    {…}
▶ 3:                    {…}
▶ 4:                    {…}
```

```
▼ 0:
    name:               "ikdam"
    date:               "1929_02_17"
  ▶ url:                "http://nek.istanbul.edu.…için)_1929_subat_17_.pdf"
  ▼ ner:
    ▶ PERSON:           […]
    ▶ LOCATION:         […]
    ▶ ORGANIZATION:     […]
    page:               6
  ▶ text:               "6“ — . ml pa Ni\n\n>\n\n…olmak\nüze ilân olunur."
    id:                 "1929_02_17.ikdam_6"
    title:              "1929_02_17.ikdam_6"
    timestamp:          "1929-02-17T00:00:00"
    png:                "http://localhost:5984/page/1929_02_17.ikdam_6/png"
▼ 1:
    name:               "cumhuriyet"
    date:               "1931_07_21"
  ▶ url:                "http://nek.istanbul.edu.…iyet_1931_temmuz_21_.pdf"
    page:               6
  ▶ text:               "— “ee\n\nap E €\n\nŞT EL…s'ul Müdürü M, A rd / |"
    id:                 "1931_07_21.cumhuriyet_6"
    title:              "1931_07_21.cumhuriyet_6"
    timestamp:          "1931-07-21T00:00:00"
  ▶ png:                "http://localhost:5984/pa…1_07_21.cumhuriyet_6/png"
▶ 2:                    {…}
▶ 3:                    {…}
▶ 4:                    {…}
```

Figure 4. API response showing first results for the keywords ‘sansür’ (censor) and ‘sahte’ (counterfeit) in JSON format.

PER (Person), ORG (Organisation) and LOC (Location) named entity types are used in this article. The training process is made using the ENAMEX data (enamex_train.txt) on the official Github page. Then, with the help of PerceptronNer, the named entities are determined in three classes as person, location and organisation. Performance results are presented in Section 4.2.1.

3.3. Dockerised services

Microservice architecture is an architecture style where a complex system is split into small loosely coupled parts, called services, which can be deployed independently. This architecture allows for fast, frequent and reliable delivery of large, complex applications.

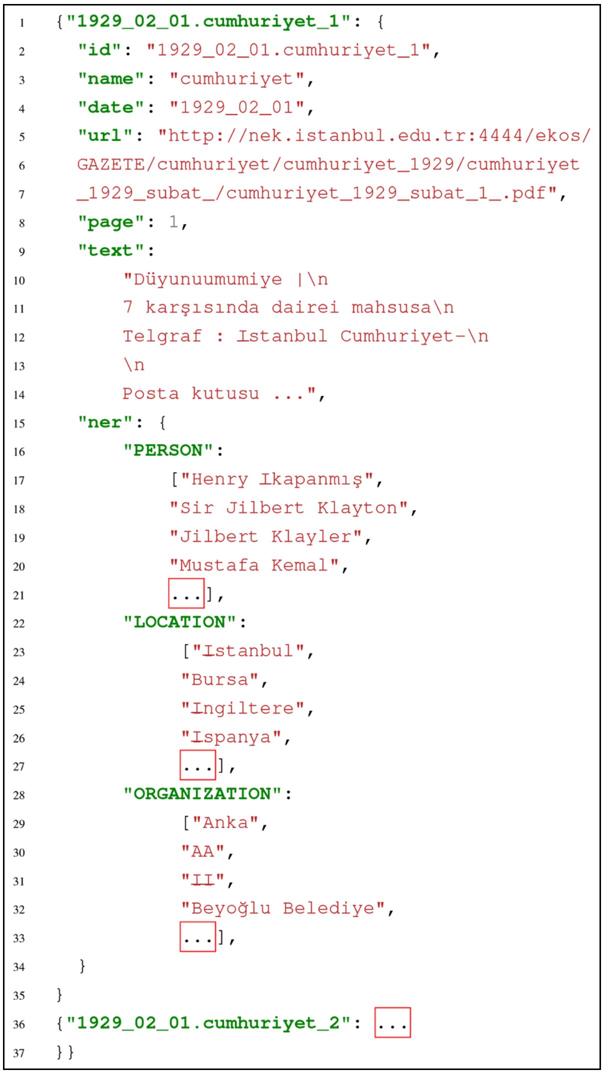
3.3.1. Customised RESTful API.

3.3.1. Customised RESTful API. Representational State Transfer (REST) [57] is an architectural style used to design distributed systems. It was introduced and defined by Roy Fielding in his doctoral dissertation in 2000. REST architecture works over the HTTP protocol and allows for communication and data transfer in client–server architectures by carrying Extensible Markup Language (XML) and JavaScript Object Notation (JSON) data.

We use a RESTful API to sit between our GUI and the elastic search and CouchDB instances, it gives us a simple and standardised way to query our processed dataset by just sending a GET request in the following format: http://localhost:4000/query?keyword= < > start= < > end= < > . The response then comes in the form of a JSON document as shown in Figure 4.

3.3.2. Storing data in CouchDB. Traditionally, SQL databases have been the dominant choice for any data storage option, but explosive growth of our generated data and the rise of new applications brought with it have shown the need for a new approach instead of SQL. NoSQL replaces the rigid data schema in SQL with a more flexible one, allowing us to work with large sets of non-structured data, in distributed and real-time applications [58]. Some of the main qualities of such databases include a simpler and more flexible design, ability to scale horizontally and better performance [59].

Considering that we are dealing with documents in our dataset, a document-oriented database would be the optimal choice for it, hence our choice of CouchDB. A document-oriented database stores each object in a dedicated instance (as seen in Figure 5), instead of spreading it across multiple tables as is the case with SQL.



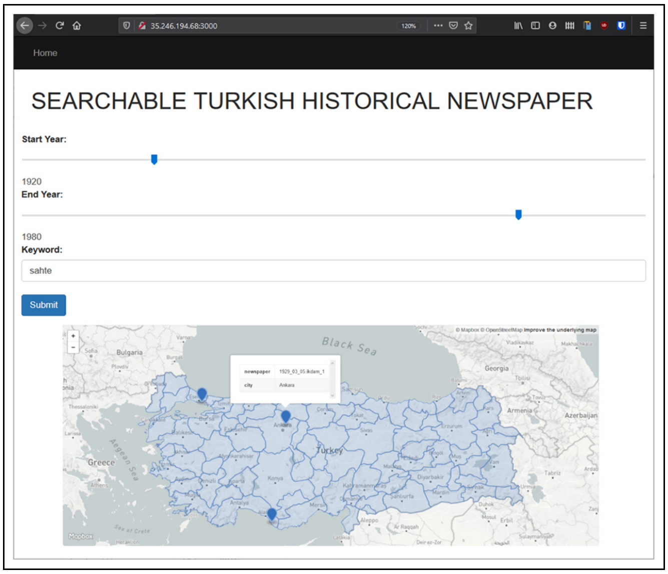
```
{"1929_02_01.cumhuriyet_1": {
  "id": "1929_02_01.cumhuriyet_1",
  "name": "cumhuriyet",
  "date": "1929_02_01",
  "url": "http://nek.istanbul.edu.tr:4444/ekos/
  GAZETE/cumhuriyet/cumhuriyet_1929/cumhuriyet
  _1929_subat_/cumhuriyet_1929_subat_1_.pdf",
  "page": 1,
  "text":
      "Düyunuumumiye |\n
      7 karşısında dairei mahsusa\n
      Telgraf : Istanbul Cumhuriyet-\n
      \n
      Posta kutusu ...",
  "ner": {
      "PERSON":
          ["Henry Ikapanmış",
          "Sir Jilbert Klayton",
          "Jilbert Klayler",
          "Mustafa Kemal",
          ...],
      "LOCATION":
           ["Istanbul",
           "Bursa",
           "Ingiltere",
           "Ispanya",
           ...],
      "ORGANIZATION":
           ["Anka",
           "AA",
           "II",
           "Beyoğlu Belediye",
           ...],
  }
}
{"1929_02_01.cumhuriyet_2": ...
}}
```

Figure 5. An example of CouchDB JSON document from our proposed system.

CouchDB, implemented in Erlang, handles concurrency with a form of multi-version concurrency control (MVCC). It offers an intuitive RESTful HTTP/JSON API and uses MapReduce of JavaScript functions as the query method.

Other features include document-level ACID (atomicity, consistency, isolation, durability) semantics and multi-master replication, which allows it to scale across machines to build high-performance systems. A database handler is created to facilitate running any operation across the whole proposed system.

3.3.3. Indexing data in Elasticsearch. Due to its popularity and ease of use, we decided to use Elasticsearch [60] to implement our indexing and search engine part of the system, Elasticsearch is a full-text search and analysis engine based on the Apache Lucene project [61]. Together with Kibana and Logstash, they form the ELK stack, an open source solution for logging, indexing textual data and full-text search. There are many reasons to choose Elasticsearch such as scalability, fast performance, multilingual support, document oriented (JSON), auto-completion and instance search, and schema free. It will run perfectly fine on any machine or in a cluster containing hundreds of nodes, and the experience is almost identical. By using distributed inverted indices, Elasticsearch quickly finds the best matches. Elasticsearch uses JSON as the serialisation format for documents. JSON has become the standard format used by the NoSQL movement. Auto-completion is a navigational feature to guide users to relevant results as they are typing, improving search precision. Elasticsearch does not require some definitions such as index type and field type before the indexing process.



35.246.194.68:3000

Home

SEARCHABLE TURKISH HISTORICAL NEWSPAPER

Start Year:

1920

End Year:

1980

Keyword:

sahte

Submit

newspaper	1929_03_05.ikdam_1
city	Ankara

Figure 6. GUI for submitting search queries.

3.3.4. A web-based GUI and visualisation. The GUI is a simple flask application with a search page that allows the user to run queries and filter them based on a keyword and a slider for the date range as shown in Figure 6. The results are then displayed in a table containing the OCRed text, the date, newspaper name, as well as a link to download the original PDF.

The location NER data are also used to provide a visualisation in the form of a map with pins of mentioned locations, this could help researchers explore the data if they do not know what to look for well enough.

4. Experimental results

4.1. Experimental setup

This project was implemented in Python 3 and tested on a local machine with the following specifications: CPU: Intel Core i7-7700HQ, RAM: 12 GB DDR4-2400, GPU: Nvidia GeForce GTX 950M 2GB GDDR5, OS: Ubuntu 18.04 LTS.

4.2. Performance evaluation and discussion

4.2.1. NER results. In general, NER results with only a few noticeable mistakes are shown below in as an example. Correct versions are given within parenthesis.

Cumhuriyet front page from 4 February 1929:

- PERSON: YUNUS NADİ Marehanesi (İdarehanesi), Haydarpaşa, Jermen La, Şozüttin, Emanullah Hazretlerin (Hazretleri), Habibulla (Habibullah), Ali Ahmet, Jermen La- (Jermen Labort), LawRens;
- LOCATION: İstanbul, Fırtına, Avrupa, Amerika, Amerikada, Türkiye, Berlin, Antalya, Karabiğa (Karabağ), Konya, Mersin, İzmir, Trabzonda, Londra, Moskova, Efganistanda, Atinaya;
- ORGANISATION: A Baş Muharriri, Liman Şirketi, Rasathine (Rasathane).

4.2.2. Indexing results. To index the OCRed data, we run eight parallel jobs to grab the stored data in CouchDB, and index in Elasticsearch, we measured a throughput of around 80 docs/s, this was bottle-necked by CouchDB queries and network latency rather than the actual throughput limit of Elasticsearch.

We do a few more tests to get a better idea about the performance of Elasticsearch indexing and search as shown in Table 1 and Figure 7.

We should note that those results are without the use of *n*-gram tokenizer in Elasticsearch, and we plan to explore this feature in the future.

Table 1. Elastic search results.

Query	Result count	Time taken (ms)
q1	512	900
q2	350	371
q3	202	249
q4	41	175

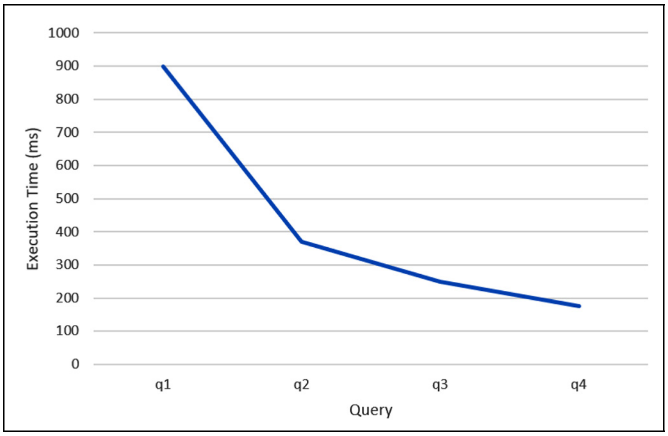


Figure 7. Execution time of search queries.

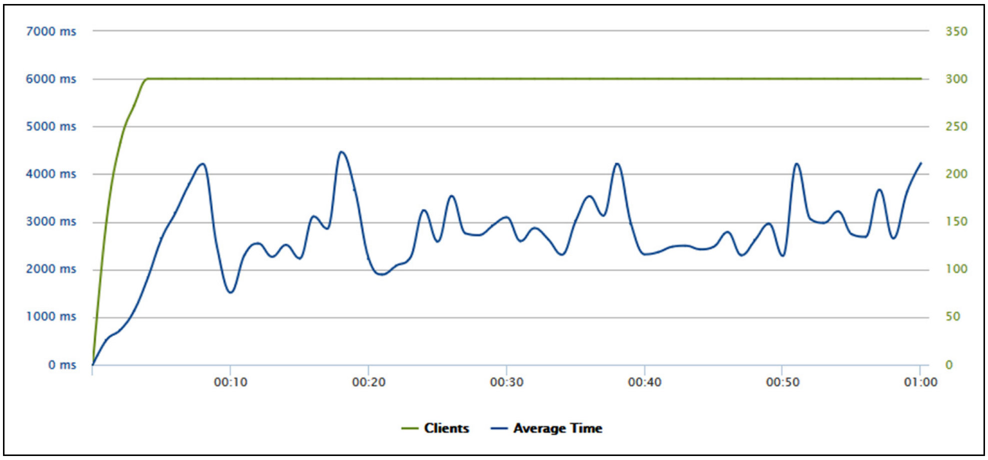


Figure 8. Load test results for search queries.

Using a combination of Loader.io [62] and Postman [63], we stress tested our API, the results show that the response time is around 317 ms per query for one client, and it can sustain a load of 300 clients/ms with a 2500 ms response time (Figure 8). This, however, is the function of the cloud machines we are using and can be scaled down or up as needed.

5. Conclusion and future work

Newspapers contain very diverse information, hence being an excellent source for various scientific disciplines. They are easily readable materials for the broader public – but only these issues are interesting, which are connected to well-known historical events, providing the reader with a historical experience. The OCR-processed text of newspapers is highly usable for text mining the vocabulary of a community given their collective and collaborative character, which makes them very representative for the written language used in the respective time period. The historical press needs to be digitised because of the low quality of its paper and at times acidic ink, which gradually destroys the material. Newspapers are especially important as documents for the history and culture of ethnical minorities.

In this work, we take it upon ourselves to turn the valuable Turkish historical newspaper dataset into a searchable text format and make it available for researchers from all fields to benefit from. So, they can be used for some people, culture and security-oriented works. The codes are available at: https://github.com/fourplusone41/AcikHack2-GazetedenTariheBakis.

However, there is still room for improvements and optimisations. This includes but is not limited to: testing nGram indexing and variable page sizes in ElasticSearch, comparing performance of different machine configurations and cluster sizes, trying alternative tools like Apache Solr [64] or a bare-bone Lucene [61]. In future, we also plan to tackle the problem of semantic search as well, which will improve our system significantly if integrated with Elasticsearch successfully.

Acknowledgements

We are grateful to ACIKHACK for their mentoring at Turkish Natural Language Processing Online Competition Program.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship and/or publication of this article.

ORCID iDs

Houssem Menhour https://orcid.org/0000-0001-8920-7830
Ekin Ekinci https://orcid.org/0000-0003-0658-592X
Süleyman Eken https://orcid.org/0000-0001-9488-908X

References

[1] Nadeau D and Sekine S. A survey of named entity recognition and classification. *Lingvisticae Investigat* 2007; 30(1): 3–26.
[2] Minkov E, Wang RC and Cohen W. Extracting personal names from email: applying named entity recognition to informal text. In: *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, Vancouver, BC, Canada, 6–8 October 2005, pp. 443–450. Stroudsburg, PA: ACL.
[3] Tanabe L, Xie N, Thom LH et al. Genetag: a tagged corpus for gene/protein named entity recognition. *BMC Bioinformat* 2005; 6: S3.
[4] Kim JD, Ohta T, Tsuruoka Y et al. Introduction to the bio-entity recognition task at jnlpba. In: *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications*, pp. 70–75, Citeseer, http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=8A6D0C627F9541E6A72802BE5B13E3D0?doi=10.1.1.153.5837&rep=rep1&type=pdf
[5] Eltyeb S and Salim N. Chemical named entities recognition: a review on approaches and applications. *J Cheminformat* 2014; 6(1): 17.
[6] Shaalan K and Raza H. Arabic named entity recognition from diverse text types. In: *International conference on natural language processing*, Gothenburg, 25–27 August 2008, pp. 440–451. New York: Springer.
[7] Anh LT, Arkhipov MY and Burtsev MS. Application of a hybrid Bi-LSTM-CRF model to the task of Russian named entity recognition, arXiv, 2017, https://arxiv.org/abs/1709.09686
[8] Gao J, Li M, Huang CN et al. Chinese word segmentation and named entity recognition: a pragmatic approach. *Comput Linguist* 2005; 31(4): 531–574.
[9] Isozaki H. Japanese named entity recognition based on a simple rule generator and decision tree learning. In: *Proceedings of the 39th annual meeting of the association for computational linguistics*, Toulouse, 6–11 July 2001, pp. 314–321. New York: ACM.
[10] Özger ZB and Diri B. Türkçe dokümanlar için kural tabanlı varlık ismi tanıma. *Türkiye Bilişim Vakfi Bilgisayar Bilimleri Ve Mühendisliği Dergisi* 2012; 5(2): 91–101.
[11] Dalkılıç FE, Gelişli S and Diri B. Named entity recognition from Turkish texts. In: *2010 IEEE 18th signal processing and communications applications conference*, Diyarbakir, 22–24 April 2010, pp. 918–920. New York: IEEE.

[12] Özkaya S and Diri B. Named entity recognition by conditional random fields from Turkish informal texts. In: *2011 IEEE 19th signal processing and communications applications conference (SIU)*, Diyarbakir, 22–24 April 2010, pp. 662–665. New York: IEEE.

[13] Küçük D, Küçük D and Arıcı N. A named entity recognition dataset for Turkish. In: *2016 24th signal processing and communication application conference (SIU)*, Zonguldak, 16–19 May 2016, pp. 329–332. New York: IEEE.

[14] Güngör O, Üsküdarlıg S and Güngör T. Recurrent neural networks for Turkish named entity recognition. In: *2018 26th signal processing and communications applications conference (SIU)*, Izmir, 2–5 May 2018, pp. 1–4. New York: IEEE.

[15] Güneş A and TantuĞ AC. Turkish named entity recognition with deep learning. In: *2018 26th signal processing and communications applications conference (SIU)*, Izmir, 2–5 May 2018, pp. 1–4. New York: IEEE.

[16] Eken B. *Kısa Metinlerde Varlık İsmi Tanıma*. PhD Thesis, Fen Bilimleri Enstitüsü, Ankara, 2015.

[17] Sarı ÖC and AktaşÖ. Türkçe ders metinleri için özelleştirilmiş bir varlık ismi tanıma yapısı. *Türkiye Bilişim Vakfi Bilgisayar Bilimleri ve Mühendisliği Dergisi* 2018; 11(2): 52–68.

[18] Arslan A, Alkılınç A and Dinçer BT. Büyük veri setlerinde varlık tanıma: En sık geçen e-posta, web adreslerinin ve emojilerin tespit edilmesi. *Academic Perspect Proced* 2018; 1(1): 399–406.

[19] Cekınel RF, Ağriman M, Karagöz P et al. Named entity recognition with conditional random fields on Turkish news dataset: revisiting the features. In: *2019 27th signal processing and communications applications conference (SIU)*, Sivas, 24–26 April 2019, pp. 1–4. New York: IEEE.

[20] Akpınar MY, Oral B, Engin D et al. A semi-automatic annotation interface for named entity and relation annotation on document images. In: *2019 4th international conference on computer science and engineering (UBMK)*, Samsun, 11–15 September 2019, pp. 47–52. New York: IEEE.

[21] Jones A and Crane G. The challenge of virginia banks: an evaluation of named entity analysis in a 19th-century newspaper collection. In: *Proceedings of the 6th ACM/IEEECS joint conference on digital libraries (JCDL '06)*, Chapel Hill, NC, 11–15 June 2006, pp. 31–40. New York: IEEE.

[22] Borin L, Kokkinakis D and Olsson LJ. Naming the past: named entity and animacy recognition in 19th century Swedish literature. In: *Proceedings of the workshop on language technology for cultural heritage data (LaTeCH 2007)*, Prague, June 2007, pp. 1–8. Stroudsburg, PA: ACL.

[23] Kettunen K, Mäkelä E, Ruokolainen T et al. Old content and modern tools – searching named entities in a Finnish OCRed historical newspaper collection 1771–1910, 2016, https://arxiv.org/abs/1611.02839

[24] Kettunen K and Ruokolainen T. Names right or wrong: named entities in an OCRed historical Finnish newspaper collection. In: *Proceedings of the 2nd international conference on digital access to textual cultural heritage*, Göttingen, 1–2 June 2017, pp. 181–186. New York: ACM.

[25] Ruokolainen T and Kettunen K. À la recherche du nom perdu – searching for named entities with Stanford NER in a Finnish historical newspaper and journal collection. In: *13th IAPR international workshop on document analysis systems*, Vienna, 24–27 April 2020.

[26] Labusch K, Kulturbesitz P, Neudecker C et al. Bert for named entity recognition in contemporary and historical German, 2019, https://konvens.org/proceedings/2019/papers/KONVENS2019_paper_4.pdf

[27] Ruokolainen T and Kettunen K. Name the name-named entity recognition in OCRed 19th and early 20th century Finnish newspaper and journal collection data, pp. 137–156, http://ceur-ws.org/Vol-2612/paper10.pdf

[28] Willems M and Atanassova R. Europeana newspapers: searching digitized historical newspapers from 23 European countries. *Insights* 2015; 28(1): 51–56.

[29] Neudecker C and Antonacopoulos A. Making Europe's historical newspapers searchable. In: *2016 12th IAPR workshop on document analysis systems (DAS)*, Santorini, 11–14 April 2016, pp. 405–410. New York: IEEE.

[30] Neudecker C. An open corpus for named entity recognition in historic newspapers. In: *Proceedings of the tenth international conference on language resources and evaluation (LREC'16)*, Portorož, 23–28 May 2016, pp. 4348–4352, https://www.aclweb.org/anthology/L16-1689

[31] Apperley M, Cunningham SJ, Keegan TT et al. Niupepa: a historical newspaper collection. *Commun ACM* 2001; 44(5): 86–87.

[32] Gatos B, Konidaris T, Ntzios K et al. A segmentation-free approach for keyword search in historical typewritten documents. In: *Eighth international conference on document analysis and recognition (ICDAR'05)*, Seoul, South Korea, 31 August–1 September 2005, pp. 54–58. New York: IEEE.

[33] Ahonen E and Hyvonen E. Publishing historical texts on the semantic web-a case study. In: *2009 IEEE international conference on semantic computing*, Berkeley, CA, 14–16 September 2009, pp. 167–173. New York: IEEE.

[34] Jerele I, Erjavec T, Pokorn D et al. Optical character recognition of historical texts: end-user focused research for Slovenian books and newspapers from the 18th and 19th century. In: *SEEDI conference: proceedings*, Zagreb, Croatia, 16–20 May 2011.

[35] Terasawa K, Shima T and Kawashima T. A fast appearance-based full-text search method for historical newspaper images. In: *2011 international conference on document analysis and recognition*, Beijing, China, 18–21 September 2011, pp. 1379–1383. New York: IEEE.

[36] Thompson P, Carter J, McNaught J et al. Semantically enhanced search system for historical medical archives. In: *2015 digital heritage*, Granada, 28 September–2 October 2015, pp. 387–390. New York: IEEE.

[37] Thompson P, Batista-Navarro RT, Kontonatsios G et al. Text mining the history of medicine. *PLoS ONE* 2016; 11(1): e0144717.

[38] Wilkinson T, Lindström J and Brun A. Neural word search in historical manuscript collections, arXiv, 2018, https://arxiv.org/abs/1812.02771#:~:text=We%20address%20the%20problem%20of,to%20as%20%22word%20spotting%22

[39] Kettunen K, Ruokolainen T, Liukkonen E et al. Detecting articles in a digitized Finnish historical newspaper collection 1771–1929: early results using the pivaj software. In: *Proceedings of the 3rd international conference on digital access to textual cultural heritage*, Brussels, 8–10 May 2019, pp. 59–64. New York: ACM.

[40] Bogaard T, Hollink L, Wielemaker J et al. Metadata categorization for identifying search patterns in a digital library. *J Document* 2019; 75: 087.

[41] Atay B, Sönmez BC, Eken S et al. Docdig: Dijitalleştirilmiş dokümanlarda içerik tabanlı figür arama. *Düzce Üniversitesi Bilim Ve Teknoloji Dergisi* 2018; 6(1): 68–78.

[42] Gazeteden Tarihe Bakış Projesi, http://nek.istanbul.edu.tr:4444/ekos/GAZETE/index.php (accessed 8 January 2020).

[43] Beautifulsoup, https://www.crummy.com/software/BeautifulSoup/ (accessed 13 July 2020).

[44] Requests, https://requests.readthedocs.io/ (accessed 13 July 2020).

[45] Lovegrove WS and Brailsford DF. Document analysis of pdf files: methods, results and implications. *Electron Publish Originat Dissem Design* 1995; 8(3): 207–220.

[46] Anjewierden A. AIDAS: incremental logical structure discovery in pdf documents. In: *Proceedings of sixth international conference on document analysis and recognition*, Seattle, WA, 13 September 2001, pp. 374–378. New York: IEEE.

[47] Hadjar K, Rigamonti M, Lalanne D et al. Xed: a new tool for extracting hidden structures from electronic documents. In: *First international workshop on document image analysis for libraries*, Palo Alto, CA, 23–24 January 2004, pp. 212–224. New York: IEEE.

[48] Chao H and Fan J. Layout and content extraction for pdf documents. In: *International workshop on document analysis systems*, Florence, 8–10 September 2004, pp. 213–224. New York: Springer.

[49] Hassan T. Object-level document analysis of pdf files. In: *Proceedings of the 9th ACM symposium on document engineering*, Munich, 15–18 September 2009, pp. 47–55. New York: ACM.

[50] Niklas K. *Unsupervised post-correction of OCR errors*. Master's thesis, Leibniz Universitat Hannover, Hannover, 2010.

[51] Hauser AW. *OCR-postcorrection of historical texts*. Master's thesis, Ludwig-Maximilians-Universität, München, 2007.

[52] Généreux M and Spano D. NLP challenges in dealing with OCR-ed documents of derogated quality, https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnxhZGFwdGl2ZW5scDIwMTV8Z3g6Mzc1ZmNjNDg5ZDUzYTY5MA

[53] Okazaki N and Tsujii J. Simple and efficient algorithm for approximate dictionary matching. In: *Proceedings of the 23rd international conference on computational linguistics*, Beijing, China, 23–27 August 2010, pp. 851–859. Stroudsburg, PA: ACL.

[54] Otsu N. A threshold selection method from gray-level histograms. *IEEE T Syst Man Cybernet* 1979; 9(1): 62–66.

[55] pytesseract, https://pypi.org/project/ pytesseract/ (accessed 11 January 2020).

[56] Zemberek-nlp, https://github.com/ahmetaa/zemberek-nlp (accessed 08 January 2020).

[57] Fielding R. *Architectural styles and the design of network-based software architecture*, PhD Dissertation, University of California, Irvine, Irvine, CA, 2000, pp. 76–85.

[58] RDBMS dominate the database market but NoSQL systems are catching up, https://db-engines.com/en/blog_post/23 (accessed 30 September 2018).

[59] Leavitt N. Will NoSQL databases live up to their promise? *IEEE Comput* 2010; 43(2): 12–14.

[60] Elastic stack, https://www.elastic.co/ elastic-stack (accessed 10 June 2020).

[61] Lucene, https://lucene.apache.org/ (accessed 10 June 2020).

[62] Loaderio, https://loader.io (accessed 10 October 2020).

[63] Postman, https://www.postman.com/ (accessed 15 September 2020).

[64] solr, https://lucene.apache.org/solr/ (accessed 11 January 2020).