Decision Support Systems 152 (2022) 113662

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

PNRank: Unsupervised ranking of person name entities from noisy OCR text

Haimonti Dutta [a,*], Aayushee Gupta [b]

[a] Department of Management Science and Systems, 160 Jacobs Management Center, State University of New York at Buffalo, NY 14260, USA
[b] Department of Computer Science, IIIT Bangalore, Karnataka, India

ARTICLE INFO

Keywords:
Unsupervised ranking
Kernel density estimation
OCR noise
Named entity recognition

ABSTRACT

Text databases have grown tremendously in number, size, and volume over the last few decades. Optical Character Recognition (OCR) software is used to scan the text and make them available in online repositories. The OCR transcription process is often not accurate resulting in large volumes of garbled text in the repositories. Spell correction and other post-processing of OCR text often prove to be very expensive and time-consuming. While it is possible to rely on the OCR model to assess the quality of text in a corpus, many natural language processing and information retrieval tasks prefer the extrinsic evaluation of the effect of noise on the task at hand. This paper examines the effect of noise on the unsupervised ranking of person name entities by first populating a list of person names using an out-of-the-box Named Entity Recognition (NER) software, extracting content-based features for the identified entities, and ranking them using a novel unsupervised Kernel Density Estimation (KDE) based ranking algorithm. This generative model has the ability to learn rankings using the data distribution and therefore requires limited manual intervention. Empirical results are presented on a carefully curated parallel corpus of OCR and clean text and "in the wild" using a large real-world corpus. Experiments on the parallel corpus reveals that even with a reasonable degree of noise in the dataset, it is possible to generate ranked lists using the KDE algorithm with a high degree of precision and recall. Furthermore, since the KDE algorithm has comparable performance to state-of-the-art unsupervised rankers, using it on real-world corpora is feasible. The paper concludes by reflecting on other methods for enhancing the performance of the unsupervised algorithm on OCR text such as cleaning entity names, disambiguating names concatenated to one another and correcting OCR errors that are statistically significant in the corpus.

1. Introduction

Large-scale text repositories have now been in existence for many decades. The text in these repositories is usually made available in a machine-readable format by using Optical Character Recognition (OCR) software. They are used for a large number of tasks in information retrieval and text analysis such as the search for entities – person, places, art museums, and organizations. Person search [1,2] is one of the most popular search types in document repositories. Most conventional person search methods (developed for clean text) focus on mapping person names to specific people (referents) and help disambiguate among namesakes [3,4]. Extraction of person names and disambiguating them in OCR text is much more complex due to noise inherent in the data. Furthermore, in many applications, a ranked list of entities identified from OCR text is more desirable than simply trying to disambiguate them. Consider, the following two examples:

- Newspaper articles (available as OCR text) collected over a specific period of time in a country. They provide information about political meetings, elections, discussions in parliaments, and congress and other related issues. A student interested in the political history of the country would like to study the corpus of historical newspapers and learn who are the critical political persona of the time based on descriptions found in the newspaper articles. A ranked list of key political players would therefore be of interest.
- An online sports magazine has published articles (in OCR) covering games played by leagues, international competitions, rising stars in certain sports and so on. A fan of basketball would like to find out from the archived volumes of this magazine, who are the key players in the game over a certain period.

In both scenarios, a randomly sorted list of people's names occurring in the repository does not provide adequate information on identifying the individuals.

* Corresponding author.
E-mail addresses: haimonti@buffalo.edu (H. Dutta), aayushee1230@iiitd.ac.in (A. Gupta).

https://doi.org/10.1016/j.dss.2021.113662
Received 4 December 2020; Received in revised form 14 August 2021; Accepted 16 August 2021
Available online 21 August 2021

Unfortunately, the process of text extraction from OCR generates garbled text when transcription is inaccurate. These errors can range from single characters being incorrectly recognized to entire words (e.g. insertions, deletions, and substitutions when compared to the original text.), sentences, and even paragraphs can become illegible. Very often, discussions on OCR quality begin and end with frustrations over its imperfections. When confronted with garbled OCR, one common reaction scholars have is that it is too noisy to use [5]. It is often common practice to avoid such noisy corpora or to select them based on error rates.

An obvious question that comes to mind is: can the noise in the large-scale text repositories then be spell-corrected? Typical OCR errors from documents include: (1) Real word errors: words which are spelled correctly in the OCR text but still incorrect when compared to the original article; for e.g., a word spelled "coil" when it should really have been "and" (2) Non-real word errors: words that have been misspelled due to insertion, deletion, substitution or transposition of characters; for e.g. "tnenty" instead of "twenty" (3) Non-word errors: words that have been spelled incorrectly and are a combination of alphabets and numerical characters; for e.g. "4anrliteii" which should have been "confident" (4) New Line errors: words that are separated by hyphens where part of a word is written on one text line and remaining part in the next line; for e.g. "ex-ceptionally" where "ex" occurs on one line and "ceptionally" in another due to lack of punctuation (5) Word Split and Join errors: words that either get split or joined with other characters to make a new word; for e.g. "Thernndldntesnra" which is a garbled combination of three words "The candidates are" and "v Icrory" which should have been "victory". While several large scale spell correction algorithms exist in the literature [6–9] and have been studied for over 30 years, including those custom-designed for OCR text [10–17], the results of spell correction are still mostly imperfect [5,18]. Even if the image from which the OCR was produced is unavailable or inaccessible, a statistical model trained on the clean text and a likely error distribution can be used for transcription [19–21]. Many spell correction algorithms have focused on improving the correction model but even these either do not give a detailed performance evaluation of the algorithm post spell correction or the evaluation measures used are not able to completely analyze the performance of such algorithms on new datasets.

Given the above issues with OCR post-correction, researchers have suggested the need to "consider *what can be done* with the OCR output available to scholars today." [5,22]. Quoting Smith and Cordell,

> "Even with these advances in OCR post-correction, errors will remain…the most common use for OCR is full-text search. In addition, researchers may wish to apply the same panoply of analysis methods to OCR transcripts that they deploy on other texts: text classification, word clustering and embedding, topic models, part-of-speech taggers, syntactic parsers, text-reuse analysis, and so on. Since scholars naturally are disinclined to devote time to data that turns out not to be useful, it is difficult to assess 'how dirty is too dirty'…"

Some researchers have claimed that as little as 20% OCR correctness provides enough signal to achieve better than random results. However, texts that are just 80% clean are not noticeably worse than their completely clean counterparts [23,24]. Consequently, a three-fold approach is suggested [5]: (a) Inference to produce cleaner texts i.e. OCR post-correction (b) inference about the impact of OCR errors on downstream tasks, from information retrieval for individual passages to part-of-speech tagging to aggregate quantities such as topic models and word embeddings and (c) Statistical communication about the impact of OCR errors. This second approach – that of evaluating the impact of OCR errors on downstream tasks provides much of the motivation for this research.

Many OCR text repositories are made available with an estimation of the overall "quality" obtained from the OCR software. In this case, it is possible to rely on the OCR model to assess text quality. Such an evaluation is often referred to as "intrinsic evaluation". Such assessments may prove to be unsatisfactory because of the reliance on the underlying software/provider so that when the process of OCR creation undergoes changes, there remains no indication of how the OCR quality influences other tasks. In contrast, some scholars [22,25] explicitly study how the noise in the OCR impacts natural language processing and information retrieval tasks such as topic modeling, authorship attribution, named entity recognition, sentence segmentation, dependency parsing, ranking [26] and others. This process is referred to as "extrinsic evaluation". The current work extends the literature on extrinsic evaluation of OCR by examining the effects of noise on an unsupervised entity ranking task.

Users of the noisy text repositories often need to search for entities (person, location, organization, and others) in them. In this setting, a significant problem that surfaces is that the text has to be verified against the ground truth to estimate its performance, which may not exist. When keyword searches for entities are performed on original or partially corrected OCR text, they can produce irrelevant or no matches. In addition, the categorization of noisy documents is non-trivial. In this paper, we study the impact of OCR errors on a downstream NLP task – *automatic* extraction of a ranked list of names of people from noisy document repositories. This can be achieved by first detecting person name entities from large proportions of garbled text; using context-sensitive, but noisy phrases (concatenated to form pseudo documents) to extract *profiles* [27–29] and extracting document-based features that preserve properties of the corpus. An *unsupervised* ranking algorithm can then be designed which can learn from the underlying features without requiring human annotations. The novel algorithm discussed in this paper is based on Kernel Density Estimation (KDE) and has been shown to have comparable performance to state-of-the-art rankers on a historic newspaper archive. We then make claims about the effectiveness of learning complex search and retrieval tasks on noisy repositories by comparing them to parallel clean corpora, curated specifically for this study. The use of the noisy text to directly perform complex search and retrieval operations has innumerable benefits – such as saving time for manual annotation and cleaning and facilitating the repository's use for research and scholarly work.

This paper is organized as follows: Section 2 discusses related work; Section 3 summarizes the data used for experiments. The ranking system for person-name entities is discussed in Section 4. Empirical results are presented in Section 5 and Section 6 concludes the paper and discusses future work.

2. Related work

Daniel et al. [22] performed a series of extrinsic assessment tasks for the impact of OCR error on sentence segmentation, named entity recognition, dependency parsing, information retrieval, topic modeling, and fine-tuning of neural language models using popular out-of-the-box tools. Their findings revealed that for both information retrieval tasks (such as ranking) and language models (word2vec), decreasing OCR quality had an adverse effect on the generated output - i.e. rankings diverged when compared to human corrected text with the number of hits declining and increasing number of false positives; for language models – it was possible to obtain robust word to vector representations, but more rigorous tests were required leading to promising paths for future investigation. Hill and Hengchen [25] conducted a series of experiments on topic modeling, authorship attribution, collocation analysis and vector space modeling. They were able to demonstrate that at the character level the long-s and ligatures were statistically more likely to result in erroneously recognized words. Furthermore, topic modeling and vector space models were deemed to be less problematic than collocation analysis with eighteenth-century OCR texts. Traub et al. [30] studied how OCR errors bias results and suggested that tool makers and data providers should be aware of such limitations. The effect of OCR errors on ranking and feedback has been explored in [31]. Their results

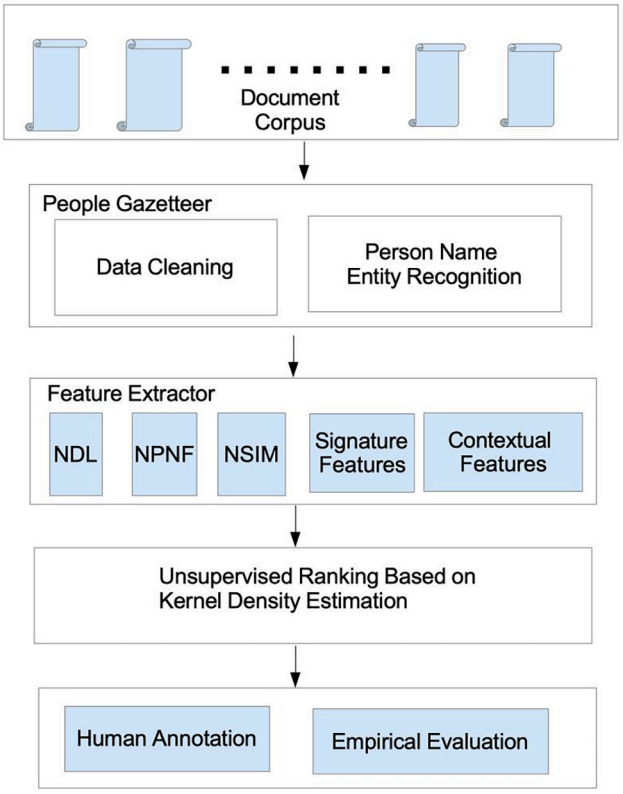

Fig. 1. Ranking System for Person Name Entities. The Feature Extractor generates the following features: NDL – Normalized Document Length, NPNF – Normalized Person Name Frequency, NDSIM - Number of Dissimilar Articles, Signature and Contextual Features. An Unsupervised Ranking algorithm based on Kernel Density Estimation is used to rank person name entities. Empirical Evaluation is guided by human annotation.

revealed that the average precision and recall is not affected when the OCR version is compared to its corresponding corrected set. However, search and retrieval exhibited different properties for seriously degraded documents.

Several researchers have studied the effect of OCR noise on Named Entity Recognition (NER). Miller et al. [32] trained their named entity recognition model – a Hidden Markov Model on OCR data with recognition at the character level. Their results revealed that the model was quite robust in the face of erroneous input and the performance degraded linearly as a function of word error. Dinarelli et al. [33] studied named entities with complex tree structures annotated on them and they proposed a three-step cleaning procedure which involved re-segmentation of sentences that ended in a dash character, re-tokenization of words, and OCR correction made by exploiting the reference correction provided for entity surface and manual correction. Packer et al. [34] applied four extraction algorithms - dictionary-based, regular expression-based, Maximum Entropy Markov Model (MEMM), and Conditional Random Field (CRF) and improved upon the performance of individual models by making an ensemble. Hamdi et al. [35] simulated different kinds of OCR errors by adding noise to the CONLL-03 NER corpus. They found that NER accuracy dropped from 90% to 60% when the word and character error rates increase from 1% to 7% and 8% to 20% respectively. Ruokolainen et al. [36] study NER extraction on Finnish historical newspaper archive using the Stanford NER and an LSTM-CRF NER and reported that with ground truth data an F-score of 0.84 was achieved on person name extraction. Boros et al. [37,38] proposed a model based on a hierarchical stack of transformers for the NER task and showed its effectiveness on German and French text.

To the best of our knowledge, the impact of OCR noise on the unsupervised ranking of named entities has never been studied. The

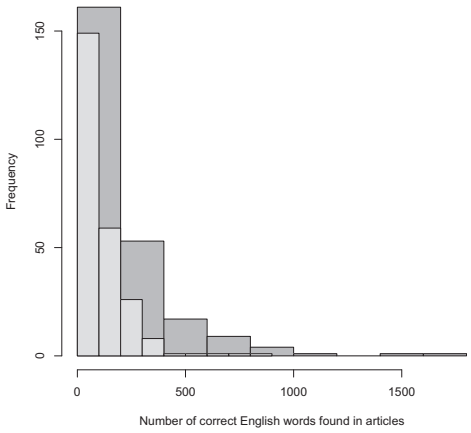

Fig. 2. Distribution of the number of correct English words found in 248 articles of the Parallel Corpus – clean (Dark Gray) and OCR (Light Gray).

following sections present a brief description of our data and the framework used to rank person name entities.

3. Data description

Chronicling America (http://chroniclingamerica.loc.gov) is an initiative of the National Endowment for Humanities (NEH) and the Library of Congress (LC). The goal of the initiative is to develop an online, searchable database of historical newspapers between 1836 and 1922. The New York Public Library (NYPL) is part of the Chronicling America (http://chroniclingamerica.loc.gov) initiative and has scanned 200,000 newspaper pages published between 1890 and 1920 from microfilm. In order to make a newspaper available for searching on the Internet, the following processes must take place: (1) the microfilm copy or paper original is scanned; (2) master and Web image files are generated; (3) metadata is assigned for each page to improve the search capability of the newspaper; (4) OCR software is run over high-resolution images to create searchable full text and (5) OCR text, images, and metadata are imported into a digital library software program. The scanned newspaper holdings of the NYPL can be searched using the OpenSearch protocol (http://www.opensearch.org/Home). Unfortunately, the current search facilities are rudimentary and irrelevant documents are often more highly ranked than relevant ones. The newspapers are scanned on a page-by-page basis and article-level segmentation is poor or non-existent; the OCR scanning process is far from perfect and the documents generated from it contain a large amount of garbled text. Article level segmentation of text is available for only two months – since this is a laborious process requiring human intervention. Articles of "The Sun" newspaper from Nov.-Dec. 1894 consisting of 14,020 news articles are used in our study.

4. Ranking system for person name entities

This section describes the framework (illustrated in Fig. 1) for ranking person name entities from noisy OCR text. It comprises the following components:

4.1. Document ingestion layer

This layer is responsible for maintaining the document corpus. It is comprised of two parts – (a) A large OCR repository and (b) a sampled parallel corpus. The large repository contains 14,020 articles that appeared in two months Nov.-Dec., 1894 of "The Sun" newspaper published from NY. The sampled parallel corpus comprises 248 OCRed articles and their manually cleaned counterparts. In order to generate clean text from the OCRed articles, annotators were provided the

Table 1
The performance of the Stanford CRF-NER software on clean (left) and garbled (right) text.

Original Text	OCR text
False Alarm of Fire for Carnegie Hall The glare of a calcium light through the small windows on the twelvth story addition of the Carnegie Music Hall where men were at work last night and a thick cloud of steam hanging heavily in the moist atmosphere around the cornice made Morris Reno President of the Music Hall Company think there might be a fire in the building He got Superintendent Johnson of the Hotel Grenoble across the avenue to tune in an alarm The firemen found out what was the matter and then explanations were made The Creation was being given as the first performance this season by the Oratorio Society in the Music Hall but the people in the hall knew nothing of the alarm	False Alarm of Fire for Carnegie Hall The glare of a calcium light through the small windows on the twelfth story addition of the Carnegie Music Hall where men were at work hast niuhit and a thick cloud of steam hanging heavily In the moist atmosphere around the cornice made MorrU Heno President of the Music Hall Company think thiro might bau tire in bite building Ho got bupcrintwident Johnson of the Hotel Urenoblr across the avetiue to turn in an alarm The firemen Sound out what was the matter anti then explanations were made The Creation was being given as the first Mrlormance this season by the Oratorio Society in the Music Hal but the people in the hall knew nothing of the alarm

4.2. Pre-processing

Stop words are eliminated from the parallel corpus and the large repository using standard English lists provided in the MAchine Learning for LanguagE Toolkit (MALLET) [39] toolkit. Punctuations (if present) are removed and the text is converted to lower case.

4.3. People Gazetteer

The People Gazetteer consists of tuples comprised of person names along with the list of documents in which they occur. To build the gazetteer, a Named Entity Recognizer (NER, Stanford CRF-NER [40–42]) is run on every document in the large OCR repository and on the parallel corpus to mark up people names occurring in the text. This process involves *chunking* or segmentation of the text, followed by classifying the name by entity type (for e.g. person, organization, and location). Only multi-term entities of type "person" are retained. Repetitions of entity names are not allowed – for example, if the person names "John", "John Smith", "Smith" are recognized, we only consider "John Smith" as an entry for the gazetteer. For the large OCR repository, a total of 36,364 person entities are extracted from 14,020 news articles.

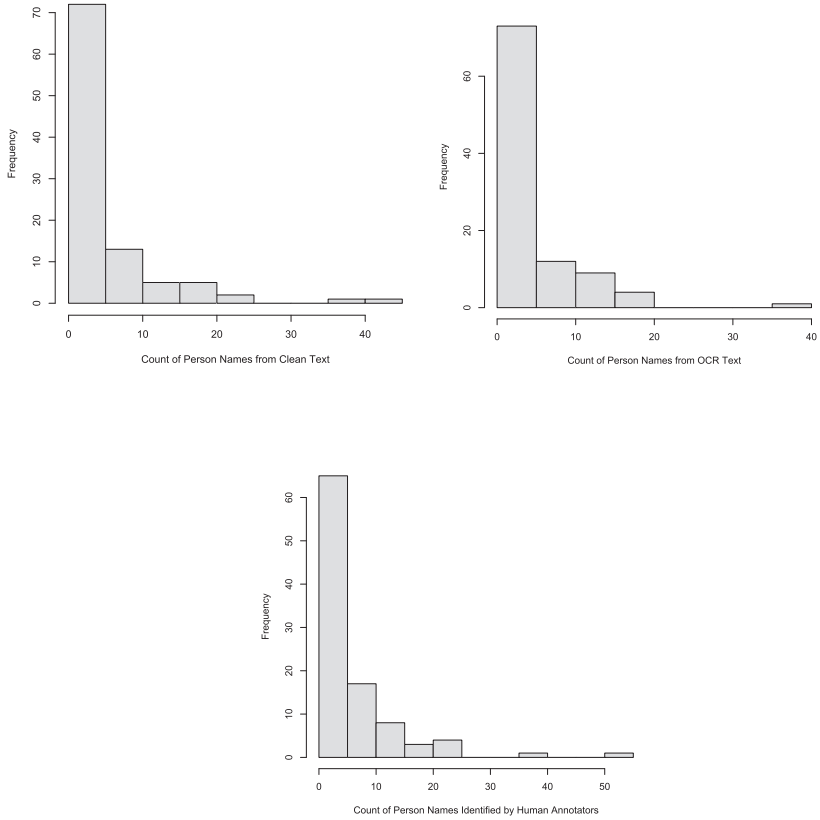

Fig. 3. Distribution of the number of people names found in 248 articles of the parallel corpus along with those identified by human annotators from clean text.

original image of the article and were asked to ensure that each line of the original text matched the corrected text and merged articles that were spread over multiple images.[1] The quality of the corrected text was subjected to rigorous tests by another annotator hired for the task. Fig. 2 shows the distribution of the number of correct English words found in the clean and OCR corpus articles.

The effectiveness of the NER procedure in the presence of noise was tested on the 248 sampled articles in the parallel corpus. The Stanford CRF-NER was run on the clean and OCR corpora independently and a count of the number of people's names identified in each case were kept (Table 1:). In addition, human annotators were asked to identify the number of people's names that were found in these 248 articles with clean text. There were 501 person names identified by humans, 406 from the clean text, and 303 from the OCR text. Fig. 3 shows the distribution of the number of people names found in the sample from the clean and garbled text along with what a human annotator would have identified. It was found that there was no significant difference in the

[1] Detailed instructions of the task are available from https://github.com/aayushee/Annotation/blob/master/Instructions.txt

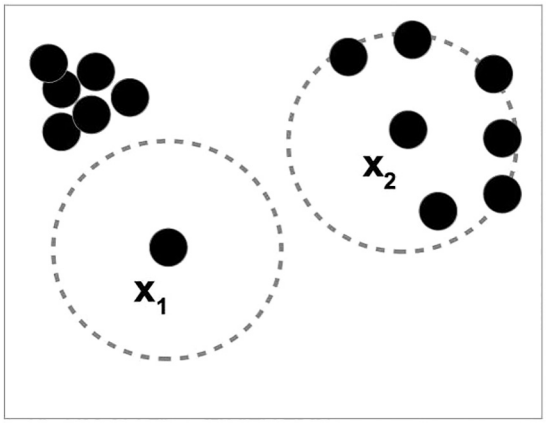

Fig. 4. An illustration of Kernel Density Estimation (KDE) for two points x_1 and x_2 in two dimensions.

distributions except that the human eye would find several more people names that the NER approach failed to recognize. Furthermore, errors from the garbled text produced noisy results in the NER phase. Once the gazetteer is built, features (Fig 5) are extracted and person name entities are ranked.

4.4. Feature extractor

The feature extractor has the following design features: (a) *Signature Features:* For every person name identified, we examine the words before it to identify occurrences of specific titles. Titles are assumed to signify veneration, official position, and academic or professional qualifications. We use this as an indicator of whether a person should be ranked high or not. Specifically, we use (i) the categorization of titles from the Wikipedia (https://en.wikipedia.org/wiki/Title) page but only restrict it to the following categories: legislative and executive, aristocratic (both genders are considered), judicial, ecclesiastical, academic, military, and ranks of other organizations. We also use a hand-curated list with titles collected from several online sources. For each category from both lists, we examine the occurrence of the title before the person's name. Two features are designed corresponding to each list – in both cases, the feature value is set to 1 if it does occur and 0 otherwise. (b) The *Contextual Features* module examines the text surrounding a person's name in an article and extracts contextual features. Neural network-based language models [43–45] are developed which rely on the fact that similar words tend to be close to one another in high dimensional vector space representations. A two-layer neural network is used for experiments reported in this paper and the output is a 100-dimensional representation of words; these vectors, called *neural word embeddings*, are obtained using a continuous bag of word representation (CBOW). (c) The *Features from the Corpora:* include three main document features – (1) Normalized Person Name Frequency (NPNF): Person Name Frequency (PNF) accounts for the number of occurrences of a person's name in a document. It is normalized by the maximum PNF in the corpus [46] and is given by: $NPNF = \frac{PNF}{max(PNF)}$ (2) Normalized Document Length (NDL): Document Length is defined as the number of tokens contained in a document. It is normalized by the maximum length of a document in the corpus [46]. Thus, $NDL = \frac{\text{Document Length}}{\text{Maximum Document Length in the corpus}}$ (3) Number of dissimilar articles (NDSIM): This captures how a person name occurring in a document is associated with a mixture of topics [47] and is influenced by topics of other documents in the corpus. It is calculated by estimating the KL divergence between documents. These features crafted from the data, enable us to rank person name entities using unsupervised ranking algorithms.

4.5. Ranking models

The ranking process is crucial to developing a hierarchy of person names in the corpus based on the extracted features. We use a novel Kernel Density Estimation (KDE) based unsupervised ranking algorithm for this task.

A motivating example: In order to illustrate the ranking approach we present an example. Suppose there are two person names extracted from the text and for each of them all the features discussed above are extracted. The data points x_1 and x_2 represent the feature space of these two people and the dots surrounding them are the feature representation of other people names in their vicinity. Both x_1 and x_2 have low densities (See Fig. 4). The density at x_1 is quite different from its neighbors, whereas the density of neighbors at x_2 is close to x_2. x_1 is more likely to stand out than x_2 due to its relatively low density in contrast to those at its neighbors. This property helps to differentiate x_1 from its neighbors [48] and serves as the key property that will be used to distinguish highly ranked people from those who are not.

Kernel Density Estimation (KDE) based Ranking: KDE is a non-parametric method to estimate the probability density function of a random variable. Given univariate, independently and identically distributed data samples drawn from an underlying distribution, with an unknown density f, the goal is to estimate the shape of this density function f. We use this density estimation method for the task of ranking. Specifically, we assume that the rank of a person can be modeled by identifying the set of documents in which the person's name occurs and

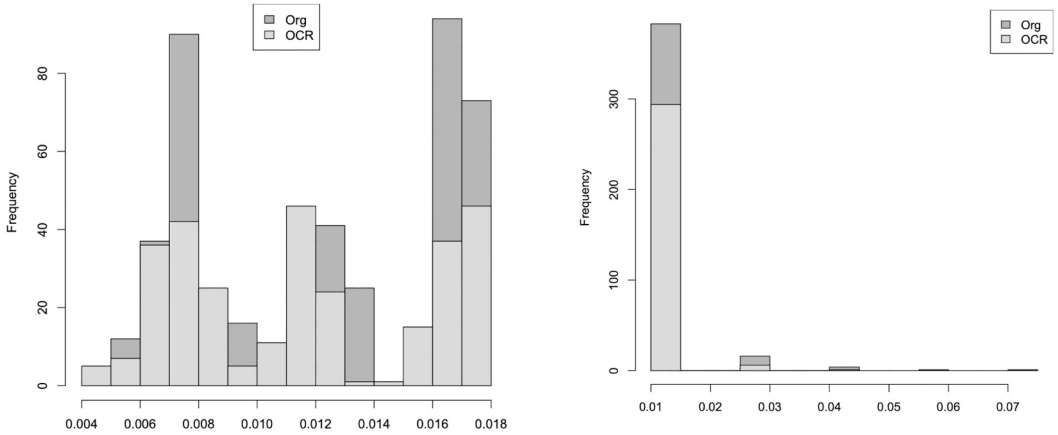

Fig. 5. Distribution of NDSIM (Left) and NPNF (Right) found in 248 articles of the parallel corpus. In both the features, while the OCR mimics the distribution of the clean text, the proportion of such occurrences is less due to the inherent noise in the data.

ALGORITHM 1: KDE-Based Ranking Algorithm

Input: $D : m \times n$;

K : Kernel function;

h : bandwidth estimator

1. Compute $\hat{f}_D(x) = \frac{1}{|D|} \sum_{s \in D} \prod_{i=1}^{n} \frac{1}{h_i} K(\frac{x_i - s_i}{h_i})$
2. Score$(x) = \frac{1}{\hat{f}_D(x)}$
3. Sort Score$(x), \forall x \in D$ in descending order to obtain a ranked list

then modeling the characteristics of these documents using probabilistic methods and ranking the scores from the probabilistic model. Since documents are represented by a vectorized feature space, we independently find the KDE estimate for each feature.

The probability of a person being reputed is then modeled as

$$P(P_j = reputed) = P(f_1, f_2 \cdots, f_n) \tag{1}$$

i.e. we adapt the *product* kernel which is typically used for density estimation from multi-dimensional data. We learn the probabilities of our model by estimating the non-parametric densities [49,50] of all features learnt from the corpus. Specifically, let $X \subseteq R^n$ be the domain of interest with n-features and $f : X \rightarrow R_{\geq 0}$ be an unknown probability density function. Let $D \subseteq X$ be a set of samples drawn from f. For any $x \in X$ and $s \in D$ we estimate $f(x)$ by using the function:

$$\widehat{f}_D(x) = \frac{1}{|D|} \sum_{s \in D} \prod_{i=1}^{n} \frac{1}{h_i} K\left(\frac{x_i - s_i}{h_i}\right) \tag{2}$$

where x_i and s_i are the i^{th} components of x and s respectively and h_i is the i^{th} component of the bandwidth vector h and K is the kernel function. Thus, given a n dimensional target point x_i, for each dimension i the product kernel (Eq. (2)) first computes the density contribution of points in set s to x_i based on the distance on dimension i. The final density contribution by s to x_i is the product of the density contributions by s on all dimensions, so called *product* kernel. Several variants of the kernel function are studied in literature including uniform, triangular, Epanechnikov, normal, and others; in this work, the normal (Gaussian) kernel is used. The normal plug-in method is used to choose the bandwidth, which is expressed as: $h_i = \left(\frac{4}{|D|(n+2)}\right)^{\frac{1}{n+4}} \widehat{\sigma_i}$.

Finally, it must be noted that the rank of a person is inversely proportional to the density which is computed using the KDE method. Recall that points with low-density values stand out from their neighbors and are therefore assigned higher scores. The scores are sorted in descending order to obtain the ranked list. Algorithm 1 presents the main steps.

Other Unsupervised Ranking Methods: We compare the performance of our KDE based ranking algorithm with several state-of-the-art rankers including two rank aggregation methods – Borda [51] and Combination of Multiple Strategies (CombXXX [52], where XXX is replaced with MIN, MAX, SUM, ANZ, and MNZ respectively) and an Unsupervised Algorithm for Rank Aggregation (ULARA) [53].

4.6. Evaluation of models

We use the notion of top-K lists, ubiquitously used in the information retrieval community, for comparing ranked lists. Several metrics, such as precision and recall at various values of K, are used to assess the quality of the top-K lists. These metrics are computed by comparing them against a "ground truth". We use two metrics – Kendall Tau [54] and Spearman's Rank Order Correlation Coefficient for comparing top-K lists and define them here for the sake of completeness.

Definition: A permutation σ is a bijection from the set $D = D_\sigma$ (also called domain or universe) onto a set $[n] = \{1, 2, \cdots, n\}$ where $[n]$ is the size of $|D|$. Let S_D be the set of all permutations of D. For a given permutation σ, $\sigma(i)$ refers to the rank of element i, $1 \leq i \leq n$. If $\sigma(i) < \sigma(j)$, then i is ranked ahead of j. Let $\mathscr{P} = \mathscr{P}_D = \{(i,j) | i,j \in D\}$ be the set of unordered pairs of objects in D and $\sigma_1, \sigma_2 \in S_D$.

Kendall Tau ($KT(\sigma_1, \sigma_2)$): For each pair $(i,j) \in \mathscr{P}$, if i, j are in the same order in σ_1, σ_2, then $K_{i,j}(\sigma_1, \sigma_2) = 0$, else if i, j are in reverse order i.e. i is before j in σ_1, but j is before i in σ_2, then $K_{i,j}(\sigma_1, \sigma_2) = 1$. Thus, Kendall Tau $= KT(\sigma_1, \sigma_2) = \sum_{ij \in \mathscr{P}} K_{ij}(\sigma_1, \sigma_2)$. This metric is traditionally used for comparing ranked lists and is further adapted to work with top-K lists. Given two top-K lists τ_1 and τ_2, we define $\mathscr{P}(\tau_1, \tau_2) = \mathscr{P}_{D_1 \cup D_2}$ to be the set of all unordered pairs of elements in $D_{\tau_1} \cup D_{\tau_2}$. For top-$K$ lists, the minimizing Kendall Tau metric between two top-K lists $KT_{min}(\tau_1, \tau_2) = \min K_{i,j}(\sigma_1, \sigma_2)$ where σ_1, σ_2 are permutations of $D_{\tau_1} \cup D_{\tau_2}$ and where $\sigma_1 \succcurlyeq \tau_1$ and $\sigma_2 \succcurlyeq \tau_2$.

Spearman's Rank Order Correlation Coefficient (ρ) [55]: This is defined as follows: $\rho = 1 - \frac{\sum d_i^2}{n(n^2-1)}$, where d_i=difference in paired ranks. This metric can take values between -1 and $+1$, with $+1$ indicating perfect association with ranks and -1 indicating a perfect negative association between ranked lists.

4.7. Human annotation

The top-K lists generated from the ranking algorithm are examined manually by human annotators to provide labels corresponding to whether a person's name should be highly ranked or not. A majority vote is computed and used as a ground truth label for estimating the precision of the ranking algorithm. Following standard practice in information retrieval and machine learning, precision is computed as the fraction of relevant instances among the retrieved instances. It can be evaluated at a given cut-off rank, considering the topmost results returned by the system. This measure is called precision at n or P @ n.

5. Empirical results

5.1. Aims

The objective of the empirical study reported in this section is to perform an *extrinsic* evaluation of the effect of OCR noise on the unsupervised ranking algorithm based on Kernel Density Estimation. Specifically, the research question we study is.

R1: What is the effect of OCR errors on the unsupervised ranking of people names?

Towards this end, we first use the parallel corpus to do a statistical analysis of the generated features and the language model and then study the effect of OCR noise on the ranking task. Next, we apply the

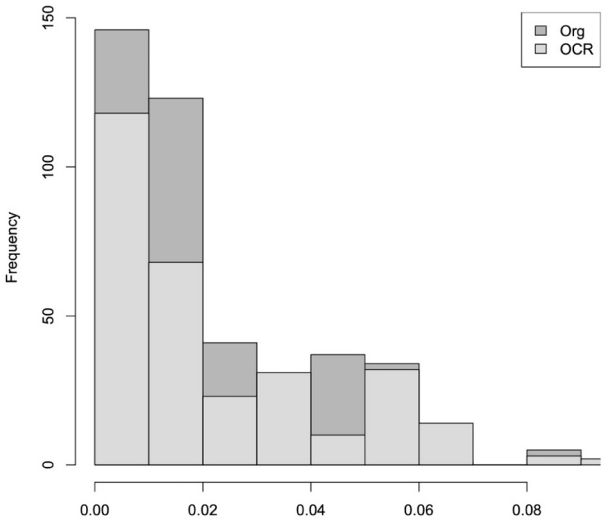

Fig. 6. Distribution of the NDL feature found in 248 articles of the parallel corpus.

unsupervised learning algorithm on the larger corpus for which human annotations do not exist. Consequently, we present results on the effectiveness of the ranking algorithm by testing it against other unsupervised ranking algorithms known in literature and manually corrected human annotation.

5.2. Results

5.2.1. Statistical analysis of the parallel Corpus

The parallel corpus of 248 articles was used to study the distribution of features generated from the OCR and clean text to rank person name entities. Figs. 6 and 7 show the distribution of NDSIM, NPNF and NDL features while Fig. 8 (Left) illustrates the distribution of the signature feature. At a first glance, we notice that the distribution of the OCR data mimics that of the clean text, however, the frequencies are compromised. The average NDL feature on the clean text is 0.02 ± 0.016 while on the OCR it is 0.02 ± 0.02. Similarly, the average NSIM is 0.012 ± 0.004 on the clean text and 0.011 ± 0.004 on OCR while the average NTF is 0.015 ± 0.005 and 0.015 ± 0.003 on clean and OCR respectively. It is observed that errors are predominant when the words are longer in length, although it cannot be conclusively stated that length alone is a factor in erroneous recognition. It is also noted that some characters such as "f", "r", "rr", "ff", "ffl" are more difficult to recognize and words with these character sequences resulted in errors more frequently than others. Fig. 7 (Right) illustrates the effect of OCR errors on Language Models (LMs) particularly word2vec representations used in this work. Though minor OCR errors should not affect the quality of the LMs significantly, our results in Fig. 7 reflect that aberrations do occur. The plot demonstrates the difference in absolute norms of the word2vec representations for the same person ranked at the top of the list from the clean and OCR data versus the percentage of error in the OCR text. Smaller OCR error typically suggests that the individual word2vec representations are not tremendously affected, but larger errors in OCR may produce random behavior.

5.2.2. Experiments on the parallel corpus

The parallel corpus comprising of OCR and clean text produced two data sets with 105 features and 303 and 406 instances respectively. The KDE-based ranking method was executed on each dataset independently, with the same parameters (normal kernel, all data points, and simple plug-in method for bandwidth estimation) to ensure comparable results. Two ranked lists of person names were generated. A blind test set was created by randomly sampling 10% of the names from the two lists which were used for evaluation of the rankers.

Towards this end, we devised a confusion matrix such that True Positive (TP) was when a person ranked by the ranker on OCR text was also ranked by the one designed on the clean text. Similarly, True Negative (TN) was when a person ranked in OCR text was not ranked in clean text; False Positive (FP) was when the person not ranked in OCR text was ranked in clean and False Negative (FN) was when a person not ranked in OCR text was also not identified in the clean text. This helped to define Precision$= \frac{TP}{TP+FP}$ and Recall$= \frac{TP}{TP+FN}$. Our results showed that the system had a *Prec* @ 20 $= 0.64$ and *Recall* @ 20 $= 0.82$. Our results reveal that probabilistic matching of person names plays a crucial role in ensuring that the ranking system has high precision and recall. Table 2 provides examples of names misspelled in the OCR text. The NER also made errors when identifying people names – for example, the following phrases were marked as person name entities "turcoimuan gotemmuir" ("turcoman governor" in clean text), "ixli introdtiiid" ("been introduced" in clean text), "tremendon ran" ("tremendous run" in clean text). Errors in identifying long people names were common – for example, "sir george tyler the lord mayor of London" was identified as "george tyler iha" and "the princess louise the grand duke alexis of russia" as "irani duko alexis". It was also observed that in addition to titles being used as indicators of high ranking, it was not uncommon to add titles at the end of a name to signify reverence – for example, 'sir george tyler the lord mayor of London". This was particularly true for long names, and while our approach would correctly identify the title "sir" in front of the name "george tyler", the title at the end "the lord mayor" would have been missed. A future enhancement of our ranking framework will incorporate this feature.

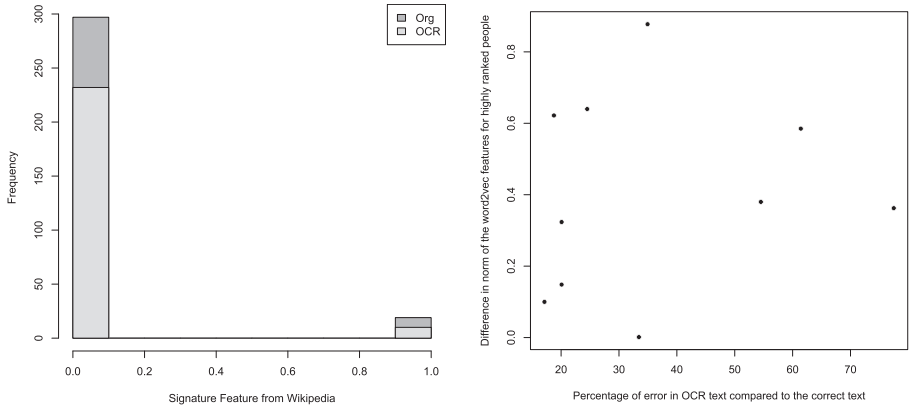

Fig. 7. (Left) Distribution of the signature feature using Wikipedia titles found in 248 articles of the parallel corpus. (Right) Plot of the degree of similarity between word2vec features versus the percentage of OCR errors for people ranked at the top of the list.

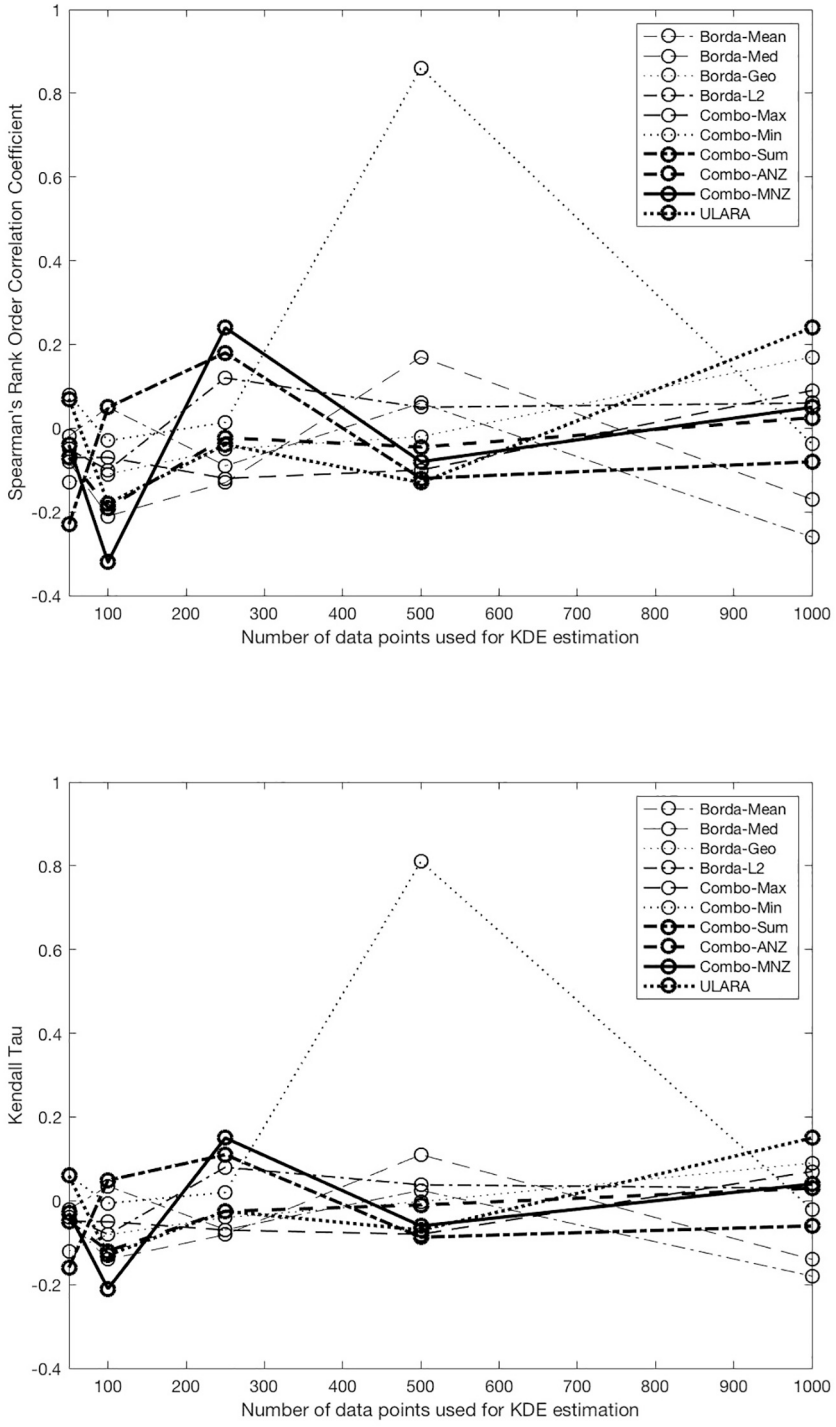

Fig. 8. Performance of KDE based ranking algorithm versus other unsupervised ranking algorithms known in the literature using Spearman's Rank Order Correlation Coefficient and Kendall Tau on the large real-world corpus.

5.2.3. Experiments on large, real-world corpus

Comparison of KDE-based ranking algorithm to other unsupervised ranking algorithms: Since the performance on controlled parallel corpus met expectations, we allowed the ranking algorithm to run on the entire dataset of 14020 articles from which slightly more than 36000 person names were extracted. Due to the relatively large number of people's names, manually checking the results was impractical. We compared the performance of the KDE-based ranking method to other state-of-the-art ranking algorithms – Borda, CombXXX, and ULARA. Fig. 3 compares the performance of the KDE-based ranking technique with these algorithms using the Spearman's Rank Order Correlation Coefficient and Kendall Tau metrics. The number of data points used for KDE is varied from 50 to 1000. We use a Normal Kernel function and the simple plug-in method for bandwidth estimation. For the ULARA method, the number of iterations of the algorithm is fixed at 10, the learning rate = 0.00000005, and the threshold (k_i) is 1000. Not surprisingly, when the number of data points sampled is small (less than 500) there is a large variance in the performance of the KDE-based method, but this noise stabilizes as more and more data points are sampled. However, sampling a large number of points does imply a larger compute time for the algorithm. It is observed

Table 2
Comparison of person names identified from OCR and clean text.

Person Name in OCR	Person name in Clean Text
william kdmunds hay	william edmunds ray
capt james ityan	capt james ryan
katlier ducey	father ducey
james l	james j bevins
peter jvolt	peter wolt
frederick fitzgerald	frederick fitzgrealdof
capt ilalnbrldce kofi	capt bainbridge hoff
dukulitorki 5	duke george

Table 3
Precision@30 for the ranking algorithms on the historic newspaper archive data.

Ranking Algo.	Precision @30	Ranking Algo.	Precision @30
Borda Median	0.43	Borda L2N	0.7
Combo Min	0.63	Combo Max	0.66
ComboANZ	0.46	Combo MNZ	0.7
Combo Sum	0.7	ULARA	0.4
KDE	0.7		

Table 4
Inter-annotator agreement measured by Krippendorff's Alpha for the unsupervised ranking algorithms.

Ranking Algorithm	Krippendorff's alpha
Borda Median	0.40
Borda L2N	0.71
KDE	0.92
Combo Min	0.57
Combo Max	0.70
Combo ANZ	0.64
Combo MNZ	0.76
Combo Sum	0.76
ULARA	0.54

that the strongest correlation is seen between the KDE method and CombMIN while strong correlations are also observed between KDE based method and CombSUM and CombMNZ methods. Intuitively, this implies that the ranked list obtained from KDE aims to minimize the probability that a non-relevant document would be highly ranked. Furthermore, the ULARA based ranking method also has a strong correlation with the KDE method for large sample sizes. Both the Spearman's Rank Order Correlation coefficient and the Kendall Tau metric reflect this behavior for the dataset.

Human annotation and performance of the unsupervised ranking algorithms: Since ground truth labels are absent in the large real-world corpus, we designed an annotation task wherein humans were asked to judge whether names occurring in top-K lists from ranking algorithms could be deemed reputed or not. Guidelines for the task were provided a priori (available from https://docs.google.com/document/d/1ctKSVN0ifIsqgsYcDtR6XKskzciwn6UI05E2_RCVKFY/edit), and annotators had to use their own intuition and other relevant details to make a conclusive judgment. They were advised against using external knowledge bases such as Wikipedia, Google, and online knowledge sources. Each list was annotated by three annotators who were graduate students well versed with natural language processing techniques. For each name, a 1 or 0 was provided depending on whether the person was reputed or not. A majority vote was taken for each name and this was assigned the ground truth label. The inter-annotator agreement was measured using Krippendorff's Alpha [56] and the results are presented in Table 4. We measure performance using the precision in the top-K ($K = 30$) as in [57,58]. The results are presented in Table 3. The KDE-based technique provides good ranking precision and is comparable to state-of-the-art rank aggregation techniques. It performs much better than the ULARA algorithm.

Which people are highly ranked? Fig. 9 shows some of the people highly ranked by the KDE based algorithm including Grand Duke George Alexandrovich of Russia (1871–1899), Captain William Bainbridge-Hoff (1774–1833), Fanny Gordon a reputed woman of the civil rights era and Chauncey Mitchell Depew (1834–1928) a Republican politician.

6. Conclusion

Large-scale text repositories are now being made available by using OCR software. If the transcription process is not accurate, it results in large volumes of garbled text. Search, information retrieval, and natural language processing tasks must use this garbled text and render them usable. This raises the natural question – "How dirty is too dirty?" and what can be done with the garbled text without resorting to extensive and costly cleaning processes? This paper presents an extrinsic method of evaluation of a downstream natural language processing task – ranking person name entities – using garbled OCR text. A system for ranking person name entities is presented using a Kernel Density Estimation based unsupervised ranking algorithm. Person names are extracted from a repository of newspaper articles containing garbled OCR and added to a people gazetteer. Features that help determine the rank of a person are extracted from the text and used to generate a top-K ranked list of person names. Careful experiments with a parallel corpus reveal that even though there is a loss in the person name detection

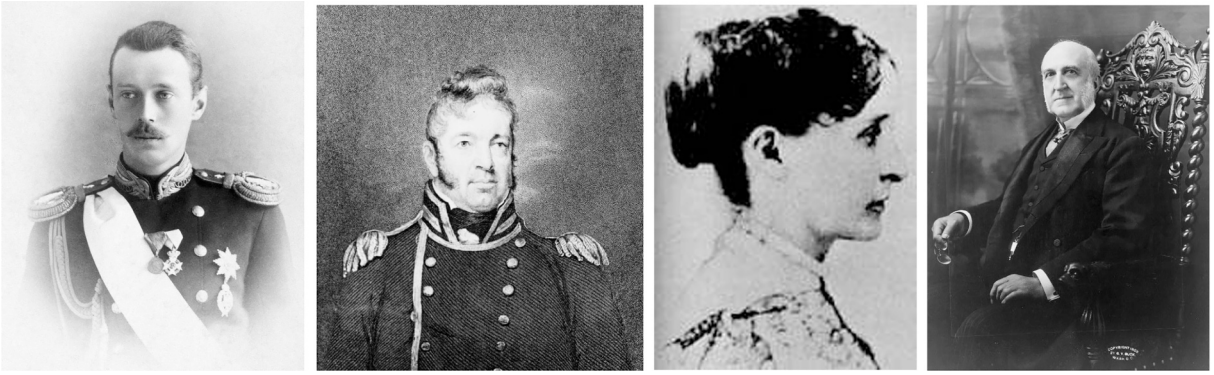

Fig. 9. People highly ranked by the KDE based algorithm from the large real-world corpus: (Left to Right) Grand Duke George Alexandrovich of Russia (1871–1899) was the third son of Emperor Alexander III and Empress Maria of Russia; Captain William Bainbridge-Hoff (1774–1833) was a Commodore in the United States Navy who served under six presidents and won several wars at sea; Fanny Gordon (1837–) was a civil war woman and Wife Of Confederate General John Brown Gordon; Chauncey Mitchell Depew (1834–1928) was an American attorney, businessman, and Republican politician best remembered for his two terms as the United States Senator from New York and for his work for Cornelius Vanderbilt, as an attorney and president of the New York Central Railroad System. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

process from noisy text, the ranking algorithm is able to sort the person names with a high degree of precision and recall, especially at the top of the list. The choice of a generative ranking algorithm based on KDE is justified by the fact that comparison to other state-of-the-art rank aggregation schemes such as Borda, a combination of multiple search strategies and unsupervised learning algorithms for rank aggregation reveals comparable performance. Finally, human annotators are asked to examine the list of highly ranked people, and the results are compared to the KDE algorithm, precision at the top of the list was comparable to state-of-the-art rankers. Several directions for future work exist including rigorous spell correction of people names, devising methods to identify and disambiguate many people names grouped together, and correcting statistically problematic errors in the corpus.

Credit author statement

Haimonti Dutta: Conceptualization, Methodology, Software, Validation, Formal Analysis, Investigation, Resources, Data Curation, Writing, Visualization, Supervision, Project Administration. Aayushee Gupta: Software, Validation, Formal Analysis, Data Curation, Writing.

Acknowledgements

The authors would like to thank Jayashree Chandrasekaran, Jiajia Qu, and Kaushik Panneerselvam for their help with coding and annotation in different phases of the project. Computational resources were provided by the Center for Computational Research at the State University of New York, Buffalo.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.dss.2021.113662.

References

[1] Q. Ma, M. Yoshikawa, Ranking people based on metadata analysis of search results, Web Information Systems Engineering Workshops (2008) 48–60.
[2] D. Kalashnikov, Z. Chen, S. Mehrotra, R. Nuray-Turan, Web people search by connection analysis, IEEE Trans. Knowl. Data Eng. 20 (2008) 1550–1565.
[3] D. Kalashnikov, S. Mehrotra, Z. Chen, R. Nuray-Turan, N. Ashish, Disambiguation algorithm for people search on the web, Proc. IEEE Int. Confer. Data Eng. (2007) 1258–1260.
[4] D. Kalashnikov, S. Mehrotra, Domain-independent data cleaning via analysis of entity-relationship graph, ACM Trans. Database Syst. 31 (2006) 716–767.
[5] D.A. Smith, R. Cordell, A research agenda for historical and multilingual optical character recognition, in: Project Report For the Andrew Mellon Foundation, Coalition for Networked Information, 2018, pp. 1–36.
[6] K. Kukich, Techniques for automatically correcting words in text, ACM Comput. Surv. 24 (1992) 377–439.
[7] D. Jurafsky, J.H. Martin, Speech and Language Processing, Prentice Hall, 2000, pp. 1–940.
[8] LingPipe, 3.9.3 Spelling Tutorial. http://www.alias-i.com/lingpipe/demos/tutorial/querySpellChecker/read-me.html, 2008.
[9] P. Norvig, How to Write a Spell Corrector. http://norvig.com/spell-correct.html, 2007.
[10] R. Dong, D. Smith, Multi-input attention for unsupervised OCR correction, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2018, pp. 2363–2372.
[11] D. Hladek, J. Stas, M. Pleva, Survey of automatic spelling correction, Electronics 9 (2020) 1670–1675.
[12] S. Xu, D. Smith, Retrieving and combining repeated passages to improve OCR, ACM/IEEE Joint Confer. Digit. Libr. (2017) 1–4.
[13] G. Khirbat, OCR post-processing text correction using simulated annealing, in: Proceedings of the Australasian Language Technology Association Workshop, 2017, pp. 119–123.
[14] M. Raynaert, Character confusion versus focus word-based correction of spelling and OCR variants in corpora, Int. J. Doc. Anal. Recognit. 14 (2011) 173–187.
[15] S. Drobac, K. Linden, Optical character recognition with neural networks and post-correction with finite state methods, Int. J. Doc. Anal. Recognit. 23 (2020) 279–295.
[16] Y. Bassil, M. Alwani, OCR context-sensitive error correction based on Google web 1T 5-gram data set, Am. J. Sci. Res. 50 (2012) 1588–1593.
[17] P. Thompson, J. McNaught, S. Ananiadou, Customised OCR correction for historical medical text, Digit. Herit. 1 (2015) 35–42.
[18] C. Rigaud, A. Doucet, M. Coustaty, J.P. Moreux, Competition on post OCR text correction, in: International Conference on Document Analysis and Recognition, 2019, pp. 1588–1593.
[19] R. Schaefer, C. Neudecker, A two-step approach for automatic OCR post-correction, in: Proceedings of the Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature, 2020, pp. 52–57.
[20] W.B. Lund, D.D. Walker, E.K. Ringger, Progressive alignment and discriminative error correction for multiple OCR engines, in: International Conference on Document Analysis and Recognition, 2011, pp. 764–768.
[21] F. Boschetti, M. Romanello, A. Babeu, D. Bamman, G. Crane, Improving OCR accuracy for classical critical editions, in: Proceedings of the European Conference on Research and Advanced Technology for Digital Libraries, 2009, pp. 156–167.
[22] D. van Strien, K. Beelen, M. Ardanuy, K. Hosseini, B. McGillivray, G. Colavizza, Assessing the impact of OCR quality on downstream NLP tasks, in: Proceedings of the International Conference on Agents and Artificial Intelligence 1, 2020, pp. 484–496.
[23] C. Strange, D. McNamara, J. Wodak, I. Wood, Mining for the meaning of a murder: the impact of OCR quality on the use of digitized historical newspapers, Digit. Human. Quarter. 8 (2014) 1–16.
[24] G. Franzini, M. Kestemont, G. Rotari, M. Jander, J.K. Ochab, E. Franzini, J. Byszuk, J. Rybicki, Attributing authorship in the noisy digitized correspondence of Jacob and Wilhelm Grimm, Front. Digit. Human. 5 (2018) 1–15.
[25] M.J. Hill, S. Hengchen, Quantifying the impact of dirty OCR on historical text analysis: eighteenth century collections online as a case study, Digit. Scholar. Human. 34 (2019) 825–843.
[26] G.A. Wang, J. Jiao, A.S. Abrahams, W. Fan, Z. Zhang, ExpertRank: a topic-aware expert finding algorithm for online knowledge communities, Decis. Support. Syst. 54 (2013) 1442–1451.
[27] J.G. Conrad, M.H. Utt, A system for discovering relationships by feature extraction from text databases, in: Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1994, pp. 260–270.
[28] G. Zacharia, A. Moukas, P. Maes, Collaborative reputation mechanisms for electronic marketplaces, Decis. Support. Syst. 29 (2000) 371–388.
[29] L. Feng, C.D. Timon, Who is talking? An ontology-based opinion leader identification framework for word-of-mouth marketing in online social blogs, Decis. Support. Syst. 51 (2011) 190–197.
[30] M.C. Traub, J. van Ossenbruggen, L. Hardman, Impact analysis of OCR quality on research tasks in digital archives, Res. Adv. Technol. Digit. Libr. (2015) 252–263.
[31] K. Taghva, J. Borsack, A. Condit, Effects of OCR errors on ranking and feedback using the vector space model, Inf. Process. Manag. 32 (1996) 317–327.
[32] D. Miller, S. Boisen, R. Schwartz, R. Stone, R. Weischedel, Named entity extraction from noisy input: Speech and OCR, in: Proceedings of the Applied Natural Language Processing Conference, 2000, pp. 316–324.
[33] M. Dinarelli, S. Rosset, Tree-structured named entity recognition on OCR data: Analysis, processing and results, in: Proceedings of the International Conference on Language Resources and Evaluation, 2012, pp. 1266–1272.
[34] T.L. Packer, J.F. Lutes, A.P. Stewart, D.W. Embley, E.K. Ringger, K.D. Seppi, L. S. Jensen, Extracting person names from diverse and noisy OCR text, in: Proceedings of the Workshop on Analytics for Noisy Unstructured Text Data, 2010, pp. 19–26.
[35] A. Hamdi, A. Jean-Caurant, N. Sidere, M. Coustaty, A. Doucet, An analysis of the performance of named entity recognition over OCRed documents, in: Proceedings of the ACM/IEEE Joint Conference on Digital Libraries, 2019, pp. 333–334.
[36] T. Ruokolainen, K. Kettunen, Name the name - named entity recognition in OCRed 19th and early 20th century Finnish newspaper and journal collection data, in: Proceedings of the Conference on Digital Humanities in the Nordic Countries 2612, 2020, pp. 136–157.
[37] E. Boros, A. Hamdi, P.E. Linhares, L.A. Cabrera-Diego, J.G. Moreno, N. Sidere, A. Doucet, Alleviating digitization errors in named entity recognition for historical documents, in: Proceedings of the Conference on Computational Natural Language Learning, 2020, pp. 431–441.
[38] E. Boros, E. Linhares, L.A. Cabrera-Diego, A. Hamdi, J.G. Moreno, N. Sidère, A. Doucet, Robust named entity recognition and linking on historical multilingual documents, in: Proceedings of the Conference and Labs of the Evaluation Forum 2696, 2020, pp. 1–17.
[39] A. McCallum, MALLET: A MAchine Learning for LanguagE Toolkit. http://mallet.cs.umass.edu, 2002.
[40] A. McCallum, W. Li, Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons, in: Proceedings of the Conference on Natural Language Learning, 2003, pp. 188–191.
[41] J.R. Finkel, T. Grenager, C. Manning, Incorporating non-local information into information extraction systems by Gibbs sampling, in: Proceedings of the Annual Meeting on Association for Computational Linguistics, 2005, pp. 363–370.
[42] C. Sutton, A. McCallum, An introduction to conditional random fields, Mach. Learn. 4 (2011) 267–373.
[43] Y. Bengio, R. Ducharme, P. Vincent, C. Janvin, A neural probabilistic language model, J. Mach. Learn. Res. 3 (2003) 1137–1155.
[44] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, in: Proceedings of the International Conference on Learning Representations Workshop, 2013, pp. 1–12.
[45] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Proceedings of the International Conference on Neural Information Processing Systems 2, 2013, pp. 3111–3119.

[46] C.D. Manning, P. Raghavan, H. Schtze, Introduction to Information Retrieval, Cambridge University Press, 2008, pp. 1–581.
[47] D.M. Blei, Probabilistic topic models, Commun. ACM 55 (2012) 77–84.
[48] X. Qin, L. Cao, E.A. Rundensteiner, S. Madden, Scalable kernel density estimation-based local outlier detection over large data streams, in: Proceedings of the International Conference on Extending Database Technology, 2019, pp. 421–432.
[49] D.W. Scott, Multivariate Density Estimation: Theory, Practice, and Visualization, John Wiley & Sons, Inc, 2015, pp. 1–27.
[50] B.W. Silverman, Density Estimation for Statistics and Data Analysis, Chapman & Hall, 1986, pp. 1–176.
[51] J.C. Borda, Mémoire sur les élections au scrutin, Histoire de l'Académie Royale des Sci. (1781) 1–86.
[52] E.A. Fox, J.A. Shaw, Combination of multiple searches, in: Proceedings of the Text Retrieval Conference, 1994, pp. 243–249.
[53] A. Klementiev, D. Roth, K. Small, An unsupervised learning algorithm for rank aggregation, in: Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, 2007, pp. 616–623.
[54] R. Fagin, R. Kumar, D. Sivakumar, Comparing top k lists, in: Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms, 2003, pp. 28–36.
[55] M.G. Kendall, Rank Correlation Methods, Griffin, 1948, pp. 146–163.
[56] K. Krippendorff, Agreement and information in the reliability of coding, Commun. Methods Meas. 5 (2011) 93–112.
[57] A. Klementiev, D. Roth, K. Small, Unsupervised rank aggregation with distance-based models, in: Proceedings of the International Conference on Machine Learning, 2008, pp. 472–479.
[58] A. Klementiev, D. Roth, K. Small, I. Titov, Unsupervised rank aggregation with domain-specific expertise, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2009, pp. 1101–1106.

Haimonti Dutta is an Assistant Professor in the Department of Management Science and Systems (MSS), School of Management, State University of New York (SUNY) at Buffalo, NY. She is also a core faculty member of the Computational and Data Enabled Science and Engineering (CDSE) Program at UB. Her research broadly focuses on machine learning, distributed optimization and large scale distributed and parallel mining.

Aayushee Gupta is a PhD student at IIIT, Bangalore, India. She works in the area of text analytics and data mining.