Finding Influential People from a Historical News Repository

Aayushee Gupta, Hamonti Dutta, Lipika Dey, Srikanta Bedathur

Abstract—Historical newspaper archives provide a wealth of information. They are of particular interest to genealogists, historians and scholars for People Search. In this paper, we design a People Gazetteer from the noisy OCR text of historical newspapers and identify "influential" people from it. A People Gazetteer is a dictionary of personal names; each entry of the gazetteer is a tuple containing a person name and a list of articles in which his name occurs along with the corresponding topic associated with each article.
To build the People Gazetteer, we first spell correct the noisy text using an edit distance based algorithm. A novel N-gram based evaluation algorithm is designed for measuring the performance of the spell corrector. Next, a Named Entity Recognizer is run on the text of each article to identify person entities and an LDA-based topic detector to assign categories to articles. To identify influential people across each category of People Gazetteer, we define the notion of an Influential Person Index (IPI) and rank based on it.
Our corpus is a sample of 14020 OCR newspaper articles (roughly two months' data) obtained from "The Sun" newspaper in the Chronicling America project. We present results on the top-K influential people obtained from our algorithm by varying its parameters and verify results using Wikipedia.

Index Terms—Gazetteer, Text Mining, Information Retrieval, OCR, Spelling Correction, Historical data, Influential people detection.

1 Introduction

An important use of historical newspapers is for People Search [1], [2]) – for example, to find important people and track the timelines of news articles related to them. Several websites like Genealogy Bank[1], FamilySearch[2], Newspaper Archives[3], Ancestry[4] provide people search service that include obituaries, birth and death lists, newspaper articles, military records, Revolutionary and Civil War pension requests, census records, land grants and other forms of petitions.

To the best of our knowledge, the problem of finding *influential* people from historic newspaper archives has not been studied before. This exercise, however, opens up a wide range of possibilities – for example, news articles related to the influential person can also be linked to a Wikipedia page entry to find out relevant details or build influential people networks that can learn about entities involved in historical events.

2 Problem Description

The goal of this research is to find and rank influential people in historical newspaper OCR archives.

1. http://www.genealogybank.com/gbnk/
2. https://familysearch.org/
3. http://newspaperarchive.com/
4. http://www.ancestry.com/

An influential person can be defined as "a person whose actions and opinions strongly influence a course of events". This allows us to link an influential person with a list of articles that s/he occurs in. A person may also be considered influential if s/he gets talked about frequently in news articles. The problem can be also be phrased as identifying and ranking *popular* people in the news domain. "Popularity" has been defined in other domains by counting number of votes, tweets, citations and followers [3] but similar measures are not applicable in a newspaper setting where only the newspaper articles mentioning multiple people names are available.

We divide the the problem of finding influential people into the following subproblems:

- **Problem 1:** Spell Correction and Cleaning of OCR text
- **Problem 2:** Development of a People Gazetteer – develop an organized structure in order to ease the process of identification of influential people.
- **Problem 3:** Influential People Identification – define the criteria for identifying and ranking people as "influential".

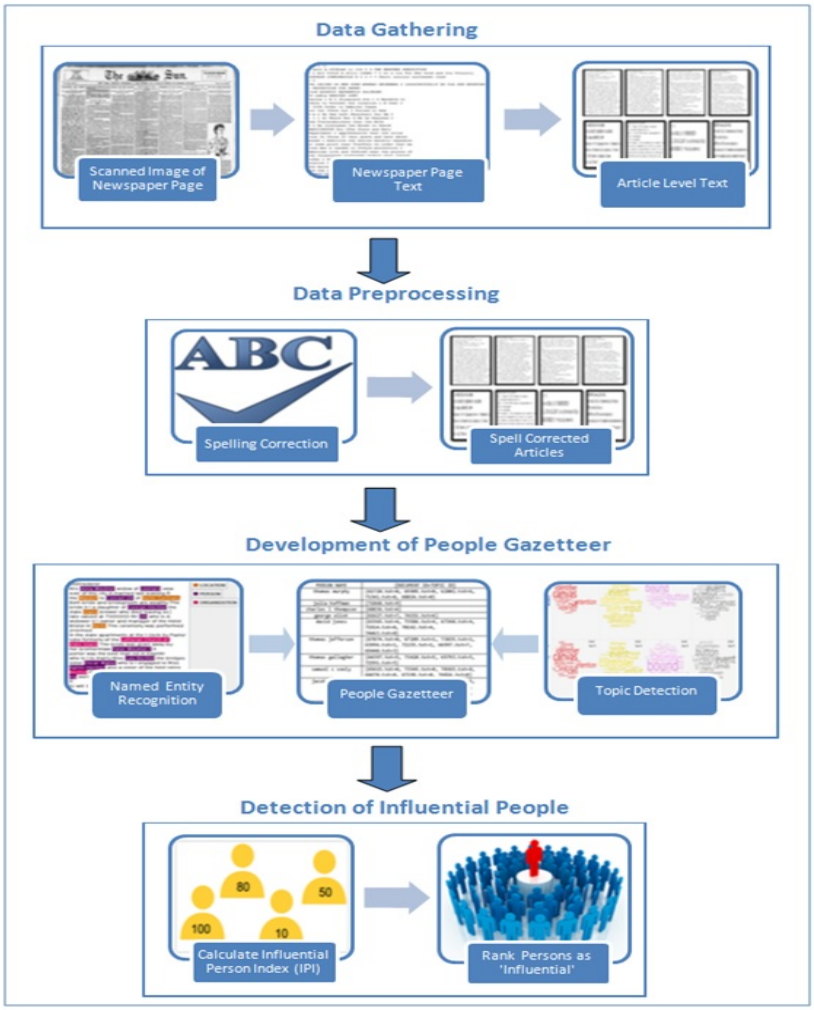


Fig. 1. Research Framework showing components of proposed solution

3 Novel Research Contributions

The paper has the following novel contributions:

1) A new algorithm for *evaluation* of the performance of the spelling correction algorithm is presented.
2) Development of the People Gazetteer – an organized dictionary of people names and a list of articles in which the name occurs along with the corresponding topic of each article to facilitate identification of influential people.
3) Define an Influential Person Index (IPI) and metrics for its calculation in order to identify and rank influential people. Case studies of the top-K influential people detected are also discussed and verified with Wikipedia data.

4 Related Work

Influential people detection has been mostly done in the field of social networks, marketing and diffusion research. [4] work on choosing the most influential set of nodes in a social network in order to maximize user influence in the network. They consider spread of influence from an influential node cascading through a network which further influences other neighborhood nodes but we do not consider the case of a network of connected person entities in our research where influence score of a person entity could be influenced by that of its neighboring person entity nodes. We consider each person entity connected with a list of articles of its occurrence instead and measure the person entity's influence score by finding the effect of influence of each article in that list.

[5] define popularity of a news story in terms

of number of reader votes received by it and predict popularity of a news story over time based on voting history and the probability that a user seeing a story at specific position in a list will vote on it. A more relevant work regarding detection of influential people is presented in [6] where influential bloggers are identified on a blog site. Influence of each blogger is quantified by taking maximum of the influence scores of each blog posted by a blogger. The influence score of each blog is calculated using parameters of importance in a blogsite like number of posts that refer to the blog, number of comments on the blog, number of other posts that the blog refers to and length of the blog. Influential blogger categories are also created based on the temporal patterns of blog posting by bloggers.

[7] describe another set of measures for detection of top influential users on Twitter using number of retweets, mentions and followers for an individual. They perform ranking based on each measure separately and use Spearman's rank correlation coefficient to find correlation among ranks and effect of each measure contributing to a person's influence. The influence ranks of topmost influential users on Twitter are presented across various topics as well as time.

In the above mentioned works, although the problem description matches with our research problem but the parameters defined to measure influence or popularity cannot be directly used in the newspaper environment.

5 Dataset Description

The dataset has been taken from Chronicling America. [5] is an initiative of the National Endowment for Humanities (NEH) and the Library of Congress (LC) whose goal is to develop an online, searchable database of historically significant newspapers between 1836 and 1922. In order to make a newspaper available for searching on the Internet, the following processes used in [8] 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 offers a wealth of data and opinion for researchers and historians. 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 contains a large amount of garbled text.

5. http://chroniclingamerica.loc.gov/

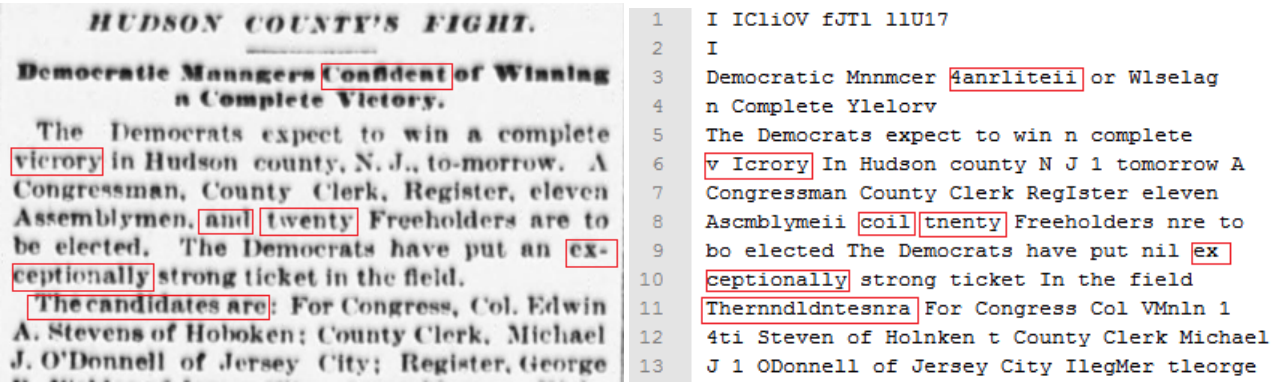
5.1 Data Characteristics

An individual OCR text article has at least one or more of the following types of spelling errors:

- **Real word errors** include words that are spelled correctly in the OCR text but still incorrect when compared to the original newspaper article image. For example: In Figure 2, the word "coil" has been correctly spelled in the OCR text but should have been "and" according to the original newspaper article.
- **Non-real word** errors include words that have been misspelled due to some insertion, deletion, substitution or transposition of characters from a word. For eg. In Figure 2, the word "tnenty" in the OCR text has a substitution error ('n' should have been 'w') which is actually "twenty" according to the original newspaper article.
- **Non-word errors** include words that have been spelled incorrectly and are a combination of alphabets and numerical characters. For example: In Figure 2, the word "4anrliteii" which is a combination of alphabets and number and should have been "confident" as per the original newspaper article.
- **New Line errors** include 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 example: In Figure 2, the word "ex-ceptionally" where "ex" occurs on one line while "ceptionally" in the next and due to no punctuation in the text, they are treated as separate words in OCR text.
- **Word Split and Join errors** include words that either get split into one of more parts or some words in a sentence get joined to a make a single word. For example: In Figure 2, the word "Thernndldntesnra" in the OCR text is actually a combination of three words "The candidates are" while the words "v Icrory" are actually equivalent to a single word "victory" when compared with the original news article.

5.2 Data Statistics

The OCR text available from Chronicling America website is on a page by page level and no article level segmentation is provided. OCR text dataset is therefore, taken from a PostgreSQL database where article level segmentation of page-level OCR text from Chronicling America is available for two months of



HUDSON COUNTY'S FIGHT.

Democratic Managers Confident of Winning a Complete Victory.

The Democrats expect to win a complete vicrory in Hudson county, N. J., to-morrow. A Congressman, County Clerk, Register, eleven Assemblymen, and twenty Freeholders are to be elected. The Democrats have put an exceptionally strong ticket in the field.

The candidates are: For Congress, Col. Edwin A. Stevens of Hoboken; County Clerk, Michael J. O'Donnell of Jersey City; Register, George

```
I ICliOV fJTl llU17
I
Democratic Mnnmcer 4anrliteii or Wlselag
n Complete Ylelorv
The Democrats expect to win n complete
v Icrory In Hudson county N J 1 tomorrow A
Congressman County Clerk RegIster eleven
Ascmblymeii coil tnenty Freeholders nre to
bo elected The Democrats have put nil ex
ceptionally strong ticket In the field
Thernndldntesnra For Congress Col VMnln 1
4ti Steven of Holnken t County Clerk Michael
J 1 ODonnell of Jersey City IlegMer tleorge
```

Fig. 2. Scanned Image of a Newspaper article (left) and its OCR raw text (right)

articles of "The Sun" newspaper from November-December 1894 consisting of 14020 news articles with a total of 8,403,844 tokens. The newspaper database ER diagram [6] is used to extract the required articles text from the database by dumping complete dataset and extracting individual articles linetext based on their unique ID. The individual text articles generated from the database do not have any punctuation and contain a large amount of garbled text containing above mentioned OCR errors.

5.3 Data Preprocessing

The garbled OCR text makes data preprocessing mandatory before application of any text mining algorithms. We,therefore, use edit distance algorithm based on Levenshtein distance to perform spelling correction on the OCR text articles. The algorithm is chosen because of its speed and ability to correct OCR errors compared to the n-gram approach [9]. Our edit distance algorithm also uses an enhanced dictionary for look up to give significance to personal names spelling correction in the dataset.

6 Development of People Gazetteer

People Gazetteer as defined in Section 1 consists of tuples of person names along with list of documents in which they occur and their corresponding topics. It is developed as an organized structure that can facilitate the process of detection of influential persons from the dataset in an efficient and easy way. This section describes the 2-step process of construction of the People Gazetteer by a) Extraction of person names from the news articles dataset using Named Entity Recognition in Section 6.1 and b) Assignment of topics to news articles using LDA topic detection in Section 6.2. Output of People gazetteer developed using these steps is presented in Section 6.3 followed by discussion in Section **??**

6.1 Person Named Entity Recognition (PNER)

6.1.1 Definition

NER (Named Entity Recognition) refers to classification of elements in text into pre-defined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc. Person Named Entity Recognition (PNER) can be defined as the process of NER that marks up only person names that occur in the text.

PNER is required in this research so as to extract all person name entities occurring in the complete dataset and identify influential person entities among them through development of the People Gazetteer. PNER aids in the development of People Gazetteer by first extracting all person names occurring in the dataset followed by reverse linking of a person with the articles in which he/she occurs.

6.1.2 Methodology

The Stanford CRF-NER[7] is used for PNER in this research. It can perform NER for 3 classes: Person, Organization and Location and is based on linear chain CRF (Conditional Random Field) sequence models. It is trained across several corpora and is fairly robust across multiple domains and even better when compared to some other open source NER systems as illustrated in [10]. According to their results, Stanford NER gave overall the best performance across 2 OCR datasets, and was most effective for PNER when compared with 3 other open source NER systems.

6. https://power.ldeo.columbia.edu/twiki/pub/Incubator/BodhiDBDesign/FinalERD.pdf

7. http://nlp.stanford.edu/software/CRF-NER.shtml

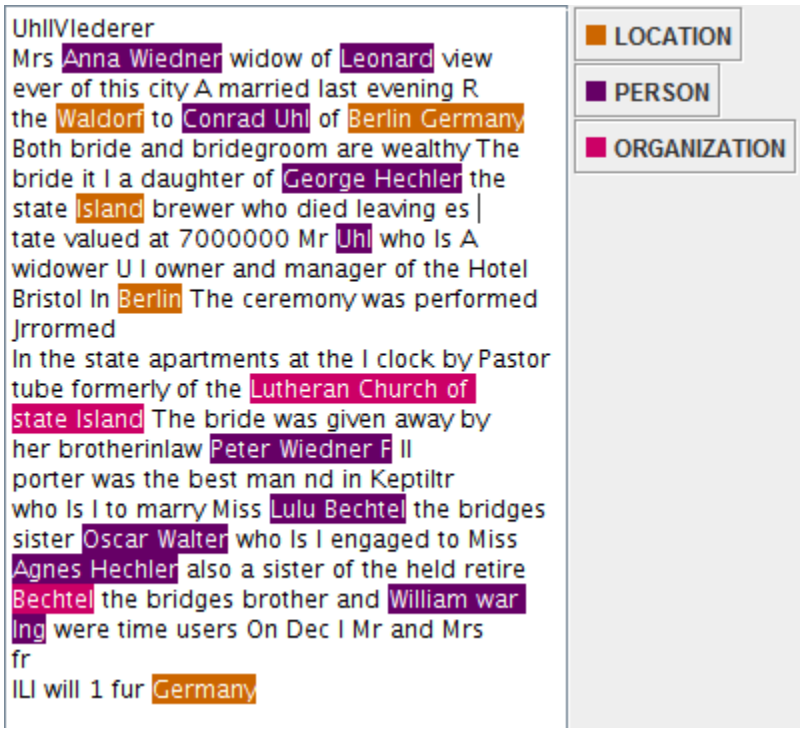


Fig. 3. NER on a sample news article

6.1.3 PNER Results

NER on a sample news article from the dataset can be seen in Figure 3. Stanford NER recognizes a person's full name as separate names by default which is rectified by combining these multi-term entities into single person entities. For example, the person name "John Smith" is recognized as two separate person entities which we combine to form a single multi-term person entity. Person names tagged with "PERSON" category are stored while running NER on the dataset. Whenever a multi-term person name (number of terms in the person name must be greater than 1) occurs in a document, the person entity's name along with the document name is stored to obtain tuples of person names with their document lists. The Stanford NER takes 25 minutes to run on the complete news dataset of 14020 articles extracting a total of 38426 person entities. The output obtained can be seen in Table 1 which shows the number of person entities with the corresponding number of documents in which they occur.

We divide the people entities extracted into following categories so that separate analysis can be done for each category:

- **Marginally Influential**: This category includes all person entities with occurrence in less than 4 news articles. (38066 person entities as calculated from Table 1)
- **Medium Influential**: This category includes all person entities with occurrence from 4 to 15 news articles. (344 person entities)
- **Highly Influential** : This category includes all person entities with occurrence in 16 or more news articles. (16 person entities)

These categories have been created manually simply based on the number of articles of occurrence of a person entity and do not directly lead to the conclusion of a person entity with large number of articles being influential.

6.2 Topic Detection

Topic models are algorithms for discovering the main topics that occur across a large and otherwise unstructured collection of documents and can organize the collection according to the discovered topics. Here, a topic refers to a set of words which describe what any document is about. A topic model examines the set of documents and discovers based on the statistics of the words in each, what the topics might be and what each document's balance of topics is. Documents are considered as a mixture of topics and each topic a probability distribution over words. Topic detection

No. of Person Entities	No. of articles
36615	1
1122	2
329	3
123	4
87	5
48	6
29	7
19	8
16	9
5	10
4	11
6	12
4	14
3	15
2	16
1	17
1	18
3	19
1	20
1	21
1	22
1	23
1	27
1	29
1	31
1	34
1	35

TABLE 1
Table showing output of PNER on 14020 articles

is the process of identifying topics in a document collection using a topic model. A simple example of topic model illustrated by [11] can be seen in Figure 4.

Topic detection is essential to this research in order to determine the topics of individual news articles that a person entity occurs in so that the person entity can be linked to the documents in which he/she occurs along with their respective topics.

6.2.1 Topic Detection Model

6.2.1.1 **:** Latent Dirichlet Allocation (LDA) Model

LDA is a generative probabilistic model in which each document is modeled as a finite mixture over an underlying set of topics and each topic, in turn, is modeled as an infinite mixture over an underlying set of topic probabilities [12]. In other words, documents exhibit multiple topics and each topic is a distribution over a fixed vocabulary. The LDA model can be briefly reviewed as follows:

Given an input corpus of D documents with K topics, each topic being a multinomial distribution over a vocabulary of W words, the documents are modeled by fitting parameters 'Φ' and 'Θ'. 'Φ' is a matrix of size $D \times K$ in which each row is a multinomial distribution of document d indicating the relative importance of words in topics. Θ is the matrix of size $W \times K$ with each column a multinomial distribution of topic j and corresponds to the relative importance of topics in documents.

Given the observed words x = x_{ij}, LDA inference is done by computing the posterior distribution over the latent topic assignments z = z_{ij}, the mixing proportions Θ_j and the topics Φ_k. The inferencing is either done using variational bayesian methods or Gibbs sampling which involves integration and sampling of latent variables. However, the simple LDA approach can take several days to run over a large corpora.

6.2.1.2 **:** Distributed LDA Model

The simple LDA method takes a long time for topic modeling which is why the distributed version suits large datasets such as ours. The data is partitioned across separate processors and inference is done in a parallel, distributed fashion.

The Approximate Distributed LDA (AD–LDA) model as proposed by [13] uses distributed computation where total dataset D is distributed equally among multiple P processors. Initialization involves data and parameters distribution to each processor and random assignment of topics so that each processor has its own copy of words x_p, topics z_p, word topic counts N_{wk_p} and topic counts N_{kj_p}. The topic model inferencing then uses simultaneous local Gibbs sampling approach on each processor for a pre-decided

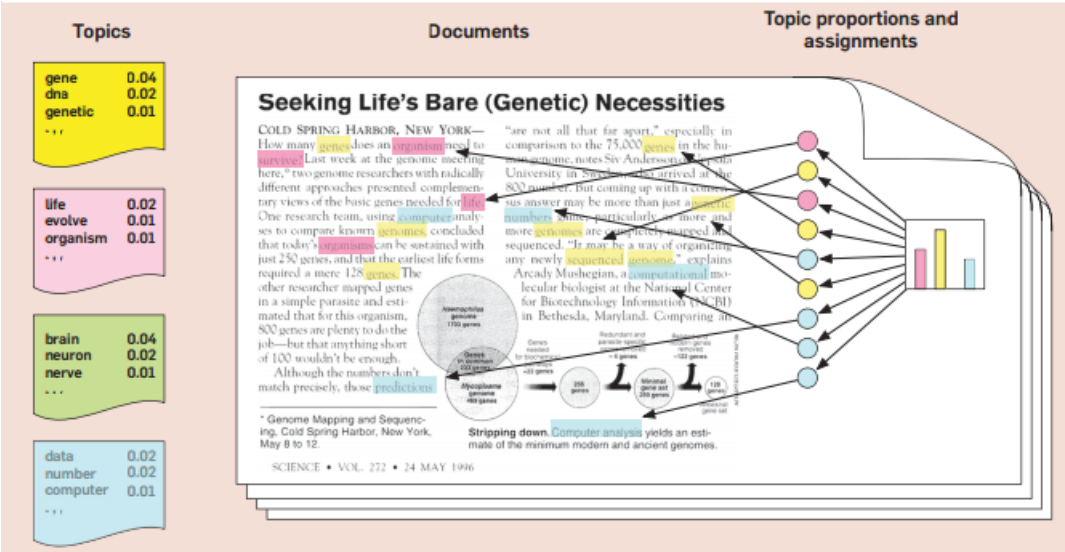


Fig. 4. Simple topic modelling approach for a single article [11]

.

number of iterations to reassign topic probabilities z_p, word topic N_{wkp} and topic counts N_{kj_p}. Global update is performed after each pass by using a reduce-scatter operation on word topic count N_{wkp} to get a single set of counts and obtain final topic assignments. The model requires user set parameters before inferencing such as number of processors/threads for parallel sampling of data, number of iterations of Gibbs sampling, number of topics and Dirichlet parameters.

6.2.2 Topic Models Evaluation

Different topic models can be evaluated using the metric of "Perplexity" which can be defined as how surprised a trained model is when given a held out test data. It has been used in [13] and [12] for evaluating the topic detection models under different parameter settings. Perplexity can be calculated using the following formula:

$$Perplexity = \exp(-\frac{\text{Log Likelihood of held-out test set}}{\text{Number of tokens in held-out test set}})$$

Here, held-out test set refers to the fact that complete dataset is split into two parts: one for training and the other for testing. The test set is taken as the held-out set for which perplexity is calculated. The document mixture is learned using the training data and log probability of the test data containing unseen documents is computed using the model developed.

Perplexity is a decreasing function of the log likelihood of the unseen documents as can be seen from its formula and lower the perplexity, better is the topic model.

6.2.3 Results

The AD-LDA model as described in [13] and implemented in the Mallet [14] toolkit (known as PLDA model) is used for topic detection over the complete dataset of 14020 news articles. Several topic models are first evaluated with different parameter settings in order to pre-decide the number of iterations, processors and topics for the final topic model to be used.

Perplexity is calculated by splitting the data into 90% for training and rest 10% for testing. Figure 5 shows the variation of the test perplexity versus the number of topics for one random $90 - 10$ split of the data[8]. The maximum number of words in each topic is set to 20, number of iterations 500 and the number of processors 4 for this experiment. It exhibits a decreasing perplexity with increase in number of topics. Typically, the number of topics should be chosen as high as possible in order to consider a better model with low perplexity but the model with high number of topics also takes longer to run on a large dataset. The number of topics is set to a value from where further increase in number of topics does not lead to a large decrease in perplexity. We choose the number of

8. We also vary the number of iterations from 100 to 500 and number of processors from 1 to 8 to study their effect on perplexity. However the number of topics is most influenced by perplexity and hence the other results are not presented here.

Number of Topics	Perplexity
10	17227
20	16431
30	15940
40	15634
50	15480
60	15390
70	15316
80	15208
90	15031
100	14988
200	14355

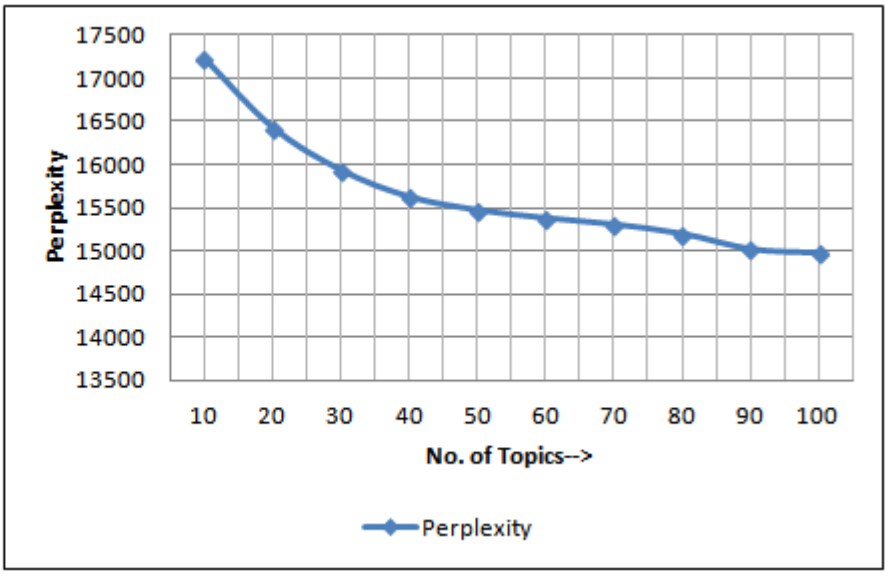


Fig. 5. Test Set Perplexity versus Number of Topics for a random $90 - 10$ split of the data. The maximum number of words in each topic is 20, number of iterations 500 and the number of processors 4 for this experiment.

topics as 30 and 100 and demonstrate their effect on the influential people detection.

From the various topic models and parameter settings, the variability in perplexity with respect to the number of topics has been found to be much greater than the variability due to the number of processors or number of iterations. This is why two values of number of topics are experimented further while number of processors and number of iterations are kept fixed. The number of iterations of Gibbs sampling still need to be above the typical burn-in period of 200 which is why 500 is chosen as the parameter value for number of iterations. Number of threads/processors is similarly taken as 4 as least training time is obtained with this parameter value.

The two models from topic detection are thus used with following parameters:

1) **30 Topics LDA Model** : Number of topics = 30, Number of iterations = 500, Number of threads=4
2) **100 Topics LDA Model** : Number of topics = 100, Number of iterations = 500, Number of threads=4

The first model takes 7.5 minutes for training while the second one takes 8.6 minutes. The set of 30 topics obtained through the first model are illustrated in Table **??** and the other model with 100 topics in Appendix Table **??**. Some of the topics words can be easily identified to belong to the following topics: music performance, court events, elections and government and shipping.

Topic modeling gives as output, for each article in the dataset, a set of topics with their probability distribution score for the article. The topic with highest topic probability score is associated with each article in the dataset.

6.3 People Gazetteer Output

The list of articles obtained for each person entity after application of PNER and highest scoring topic assigned to each article during Topic Detection are combined to obtain People Gazetteer. In each tuple of the gazetteer, a person entity gets associated with its list of articles where each article is further associated with its corresponding highest scoring topic.

Two people gazetteers are obtained, each corresponding to the two model settings of 30 Topics LDA Model and 100 Topics LDA Model, respectively. A snapshot of the people gazetteer using 30 Topics LDA Model can be seen in Figure 6 where each person entity is followed by a document list consisting of a Document ID and its corresponding Topic ID. A similar people gazetteer is also obtained using 100 Topics LDA Model. Both People Gazetteers are further used in Chapter **??** for detecting and ranking influential person entities from them.

PERSON ENTITY NAME	DOCUMENT LIST {DOCUMENT ID→DOCUMENT TOPIC}
Thomas Murphy	{61720.txt→16, 62002.txt→11, 65905.txt→19, 71341.txt→28, 68024.txt→16}
George Eliot	{74151.txt→5, 61627.txt→15}
Charles L Thompson	{68836.txt→9}
Thomas Jefferson	{67874.txt→19, 67209.txt→28, 63996.txt→6, 73835.txt→6, 71155.txt→6, 65440.txt→5, 66997.txt→20}
Jacob Schaefer	{70205.txt→21, 63936.txt→22, 68554.txt→21, 73420.txt→21, 74550.txt→21, 74922.txt→21, 64577.txt→21, 74759.txt→21, 67340.txt→0, 67924.txt→21}
Queen Victoria	{68231.txt→5, 74775.txt→5, 75097.txt→5, 72221.txt→2, 62731.txt→5, 62616.txt→17, 68368.txt→17}
Thomas Gallagher	{64397.txt→28, 65793.txt→21, 72591.txt→0, 73420.txt→21}
Samuel S Seely	{70365.txt→2, 64670.txt→23, 65615.txt→23, 67198.txt→19, 73545.txt→23, 74816.txt→16}
Matthew Parker	{64363.txt→11}
Daniel Frohman	{63704.txt→5, 66992.txt→25, 69668.txt→4, 68743.txt→5, 67554.txt→25, 67450.txt→5, 72274.txt→24, 69444.txt→4}

Fig. 6. Snapshot of People Gazetteer with Person names, Document list of occurrence and their corresponding Topic ID

7 Influential People Detection

To measure influence in the newspaper environment and to compare and rank people as influential, we define an influence score measure called *"Influential Person Index" (IPI)* corresponding to each person entity in the people gazetteer. To calculate IPI for each person entity, we first define the *"Document Index" (DI)* to measure how each document in the person entity's associated list of documents affects his influence score. Following subsections describe the parameters for calculation of DI and IPI of a person entity followed by the complete algorithm for detection of influential persons:

7.1 Document Index (DI)

The Document Index (DI) of an article in the people gazetteer helps to measure a person's influence score. Following parameters are considered for the calculation of this index:

1) **Normalized Document Length** (NDL)
Document Length affects the influence score in the sense that a longer news article in which a person entity occurs is deemed to be more important than a shorter one. It is defined as the number of tokens contained in a news article. Document Length is further normalized by dividing it with the maximum news article length (of 14020 articles in the dataset) to get Normalized Document Length as follows:

$$NDL = \frac{\text{Document Length}}{\text{Maximum Document Length in the dataset}}$$

2) **Normalized Term Frequency** (NTF)
Term Frequency (TF) accounts for the number of occurrences of a person's name in a news article. The TF of the person name affects a document's influence score as a higher number of occurrences in the document makes it more important. TF is further normalized and calculated as follows:

$$NTF = 1 + \log(\text{TF of person entity in current article})$$

3) **Number of similar articles** (NSIM)
This parameter is used in calculation of the

DI by finding articles of similar topic in the document list. Two documents are considered similar if they belong to the same topics. For a document d whose DI is to be calculated, we consider

SIM= Number of articles with the same topic as that of d in the document list of person entity.

This measure is normalized by dividing it with the number of total articles in the document list of the person entity as follows:

$$NSIM = \frac{\text{SIM}}{\text{Total number of articles in the person's document list}}$$

NSIM can be said to be equivalent to the proportion of topic similar articles that any document d has.

This parameter takes into account the effect of a document's score on a person's IPI when there exist several other documents of the same topic in the person's list.

DI for each document is a function of the above mentioned parameters and is calculated using the following formula :

$$DI = w_a.NDL + w_b.NSIM + w_c.NTF$$

where, w_a,w_b and w_c are the weights associated with each of the parameters NDL, NSIM and NTF respectively.

DI is actually a heuristic measure of these three parameters where each of the parameters can be weighted as per dataset characteristics and user requirements. For example, a higher value to w_a and lower to w_b and w_c indicates documents with longer lengths are considered more important for influencing a person's IPI. On the other hand, a higher value to w_b and lower to w_a and w_c indicates a document with larger proportion of topic similar articles influences the person's IPI more suggesting assignment of high influence score to a person entity occurring repeatedly in a specific news topic.

7.2 Influential Person Index (IPI)

Once DI is calculated for each document in a person's list, an index is calculated for the person entity in order to measure its influence in the news dataset and calculate its influential score. The "Influential Person Index" defined for this purpose is calculated as follows:

$$IPI = maxDI(d_1, d_2, ..., d_n) + UniqT$$

where , $maxDI(d_1, d_2, ..., d_n)$ = Maximum Document Index of a document d_i in a person entity's list of n articles, and

$$UniqT = \frac{\text{Number of Unique Article Topics in a person entity's document list}}{\text{Total Number of Topics in the corpus}}$$

The parameter $UniqT$ is used to account for the fact that a single person entity can be talked about multiple news topics in the news articles and to include its effect on the person entity's influence score. It is normalized by dividing it with the total number of topics as obtained during topic detection on all 14020 articles.

Ranking is done across each person category of the people gazetteer to obtain top most influential persons. For this, IPI for each person entity across the person categories are sorted in decreasing order to obtain the most influential person entities with highest IPI at the top.

7.3 Procedure for finding influential persons

Algorithm 1 depicts the procedure for measuring influence and ranking of influential people from the gazetteer. It starts with calculation of DI for each news article in a person's document list by calculating the required parameters of NDL, NSIM and NTF which are assigned 0 values initially. The respective weights w_a,w_b,w_c are taken as inputs and multiplied with each parameter to get final DI score which is added to the list of DI scores $DIScoreList$. The list is sorted to find the maximum DI value among all news articles in the person's document list. The maximum DI score is then added to the UniqT parameter to get the final IPI for each person entity which are again stored and sorted to obtain a ranked list of influential person entities.

8 Results

Two ranked influential person lists, namely L1 and L2 are obtained after calculation of IPI from the people gazetteer (developed in Chapter 6) using 30 Topics and 100 Topics LDA Model respectively. The weights w_a, w_b and w_c are all set to 1 to give equal importance to each of the parameters during calculation of DI and IPI. The statistics obtained from both lists with respect to each person category of the people gazetteer are shown in Table 3. It can be clearly observed from the table that Highly Influential Persons occur in most number of news articles on an average and with highest average term frequency followed by Medium Influential and Marginal Influential Persons. Document Length need not always be too high for a person to be ranked higher as can be observed from the fact that average document length obtained for Marginally Influential People is high in spite of their Average IPI being low indicating that the varying number of similar articles for each document as well as

Function Name	Description
GetPersonTF(doc)	Calculates TF of the person entity in document doc
GetDocLength(doc)	Calculates number of tokens in doc.
GetMaxDocLength()	Calculates maximum number of tokens in any document.
GetTopicSimilarArticles(doc,DocList)	Calculates normalized number of topic similar articles for doc in the $DocList$.
Sort(DIScoreList)	Sorts the $DIScoreList$
Max(DIScoreList)	Finds the maximum score from $DIScoreList$.
GetUniqueTopics(Person,TopicList)	Calculates normalized unique topics for $Person$ in its $TopicList$.
Sort(IPIScores)	Sorts the $IPIScores$ by IPI values.
PrintPersonNameandMaxIPI(IPIScores)	Prints $Person$ name with its IPI in decreasing order of IPI value.

TABLE 2
Description of the functions used in Algorithm 1

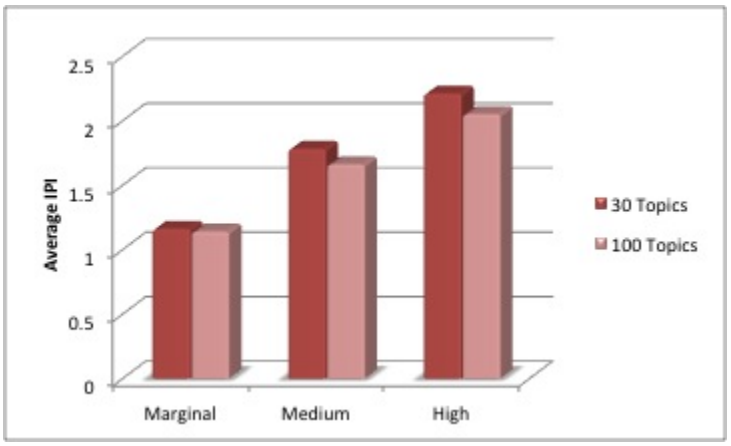


Fig. 7. Comparison of the Average IPI for two ranked lists L_1 and L_2 using 30 and 100 topics respectively.

its Term Frequency share also play an important part in measuring influence. Figure 7 shows the average IPI from the two ranked lists – it appears that the average IPI for highly influential people is more susceptible to changes in number of topics.

The following sections present comparison between the ranked influential person lists L1 and L2, some case studies and evaluation results:

8.1 Comparison Across Ranked Influential Person Lists

The top 10 influential persons from List L1 and L2 detected from each of the people gazetteers are presented in Table 4 and 5 respectively. It can be clearly seen from both the tables that the person category labels assigned during development of people gazetteer do not hold true after detection of influential persons. This suggests that the highly influential category people which were defined as person entities with more than 16 articles in the dataset might not necessarily be the most influential. The top 10 influential persons in both tables are dominated by Medium and Marginal category persons having considerably less number of articles of occurrence. This indicates the fact that number of articles of occurrence has not been given priority while measuring influence of a person entity. The statistics for top 10 influential people from both the tables also suggest that none of the measures of NDL, NTF or NSIM can be alone used to say whether a person entity is influential since these value do not decrease or increase consistently although the NTF measure does contribute most to the IPI of any person.

Person Category	Number of Person Entities	Average Number of Documents	Average Document Length	Average Term Frequency
Marginal	38066	1.04	2119.6	1.07
Medium	344	5.75	1976.3	6.68
High	16	22.8	2971.5	29.870

TABLE 3
Table illustrating average statistics for each Person Category of People Gazetteer across 2 Topic Models

Person Name	IPI	Number of Articles	Person Category	NDL	NTF	NSIM	TOPIC WORDS	UniqT	Rank
capt creeten	3.38	10	Medium	0.56	1.95	0.8	mr court police judge justice case yesterday street district	0.06	1
capt hankey	3.02	6	Medium	0.68	1.6	0.66	club game team play football half ball left college back	0.06	2
capt pinckney	2.93	3	Marginal	0.38	1.84	0.67	man ho men night back wa room left house told bad	0.03	3
john macdonald	2.85	3	Marginal	0.55	2.2	0	great people life man women good country world american part	0.1	4
john martin	2.82	12	Medium	0.56	1.6	0.5	mr court police judge justice case yesterday street district witness	0.17	5
aaron trow	2.81	1	Marginal	0.7	2.07	0	man ho men night back wa room left house told	0.03	6
mrs oakes	2.79	5	Medium	0.08	2.04	0.6	street mrs mr avenue wife house miss yesterday years home	0.06	7
buenos ayres	2.76	6	Medium	0.43	1.6	0.67	white water indian black long found thu big dog time	0.06	8
alexander iii	2.74	31	High	0.24	2.04	0.29	great people life man women good country world american part	0.16	9
mr got	2.73	3	Marginal	0.56	1.47	0.67	mr court police judge justice case yesterday street district witness	0.03	10

TABLE 4
Table showing top 10 influential persons of List L1 detected from People Gazetteer with 30 Topics LDA model. Parameters NDL, NTF,NSIM and Topic Words belong to the maximum scoring DI in the person's document list.

The ranked influential lists L1 and L2 can be contrasted in terms of NSIM, UniqT and Topic Words since they vary across different number of topics and to see the effect of 30 and 100 Topics LDA Models on influential person detection. If NSIM (normalized number of topic similar articles) remains same in L1 and L2 during influential person detection from both the people gazetteers, then the same highest scoring article' DI is selected for calculation of IPI in both of them. This is why the parameters NDL (Normalized Document Length) and NTF (Normalized Term Frequency) remain same across both the lists. This can be seen for person like "capt creeten", "capt hankey", "aaron trow" and "mrs oakes" in Tables 4 and 5. But the value of UniqT for these persons decreases leading to decrease in their final IPI in the second table. This is because LDA model with higher number of topics (100) is used in this case due to which the proportion of unique topics becomes lower when NSIM does not change. However, when the NSIM (normalized number of topic similar articles) value changes because of change in number of topics, a different article with maximum DI score can get selected leading to change in the values of NDL, NTF, UniqT and the final IPI. This causes a shift in the ranking of influential persons across the two lists and can be seen when the rank of "alexander iii" in the first table moves from 9 to 3 in the second table. This indicates the fact that LDA Topic Model used affects the ranking of influential persons when number of topics are varied.

Person Name	IPI	Number of Articles	Person Category	NDL	NTF	NSIM	Topic Words	UniqT	Rank
capt creeten	3.33	10	Medium	0.56	1.95	0.8	mr police witness committee capt asked captain money inspector paid	0.02	1
mrs martin	3.23	8	Medium	0.20	2.38	0.5	mrs mr years wife home house ago woman city died	0.02	2
alexander iii	3.09	31	High	0.49	2.04	0.48	emperor prince french alexander czar london nov government imperial russian	0.07	3
capt hankey	2.97	6	Medium	0.68	1.6	0.66	game team football play half line ball back yale eleven	0.02	4
aaron trow	2.79	1	Marginal	0.70	2.07	0	day place long great water time feet found good men	0.01	5
john macdonald	2.77	3	Marginal	0.55	2.2	0	people american man great country men world life good english	0.02	6
mrs oakes	2.74	5	Medium	0.08	2.04	0.6	mrs mr years wife home house ago woman city died	0.02	7
john martin	2.71	12	Medium	0.56	1.6	0.5	mr police witness committee capt asked captain money inspector paid	0.05	8
ed kearney	2.63	7	Medium	0.16	1.6	0.85	won time race ran mile furlough half lo track fourth	0.01	9
caleb morton	2.61	1	Marginal	0.70	1.9	0	day place long great water time feet found good men	0.01	10

TABLE 5
Table showing top 10 influential persons of List L2 detected from People Gazetteer with 100 Topics LDA model. Parameters NDL, NTF,NSIM and Topic Words belong to the maximum scoring DI in the person's document list.

8.2 Case Studies

Some of the topmost 10 influential person entities of lists L1 and L2 (Table 4 and 5) identified from each person category of the 2 people gazetteers are discussed below:

1) Highly Influential Category- This category as defined in Section 6.1.3 includes person entities influencing number of news articles greater than 16. However, only one person entity ("alexander iii") from this category occurs in the top 10 influential persons. The entry for "alexander iii" has an IPI of 2.94 and 3.09 respectively in list L1 and L2 . The person entity occurs in 31 news articles with 5 and 7 different topics in each of the lists. The most common topic words associated with this person entity are: "emperor prince french alexander czar london nov government imperial russian" indicating the importance of this entity in government related news topics. The 100 Topic LDA model increases the IPI value of this entity because the NSIM value increases (more number of similar topic articles talk about this person) and a longer article gets maximum DI score resulting in a high IPI value and improvement in the ranking from rank 9 in the first table to rank 3 in the second.
2) Medium Influential Category- The top 10 influential entities from Tables 4 and 5 contain the most number of person entities from this person category. The person entity"capt creeten" has been ranked as highest influential (Rank 1) across both the tables. It occurs in 10 news articles with 9 of them belonging to the same topic indicating the person influencing news articles of high topic similarity. Some of the most common topic words for this entity include " mr police witness committee capt asked captain money inspector paid" indicating the importance of this entity in a judicial or police related news topic. Several persons from this category like "mrs martin" , "mrs oakes" although identified among the top 10

function CALCULATEIPI
Input: $PeopleGazetter(Persons, (DocList,$
Input: $TopicList))$, w_a,w_b,w_c
Result: Ranked list of Person Name and IPI
$NTF \leftarrow 0$, $NDL \leftarrow 0$, $NSIM \leftarrow 0$, $DI \leftarrow 0$, $UniqT \leftarrow 0$, $IPI \leftarrow 0$;
for *(String PersonName : Persons)* **do**
  for *(String doc : DocList)* **do**
    $NTF = 1 + \log(GetPersonTF(doc))$;
    $NDL = GetDocLength(doc)/GetMaxDocLength()$;
    $NSIM = GetTopicSimilarArticles(doc, DocList)$;
    $DI = w_a.NDL + w_b.NSIM + w_c.NTF$;
    $DIScoreList.add(DI)$;
  end
  $Sort(DIScoreList)$;
  $UniqT = GetUniqueTopics(Person, TopicList)$;
  $IPI = Max(DIScoreList) + UniqT$;
  $IPIScores.put(PersonName, IPI)$;
end
$Sort(IPIScores)$;
$PrintPersonNameandMaxIPI(IPIScores)$;
end function

Algorithm 1: Procedure to calculate IPI and rank person entities based on it

influential persons but suffer from the problem of co-referred person names and named entity disambiguation as it is hard to identify which exact person they refer to due to lack of first names.

3) Marginally Influential Category- Person entities belonging to this category have extremely low occurrence in news articles although the IPI of topmost influential entities belonging to this category are comparable to those in the other 2 categories. Several person entities with low occurrences in news articles like "aaron trow", "caleb morton", "john macdonald" belong to this category. These entities in spite of occurring in very few articles (1 to 3) occur a large number of times in those articles with comparatively longer article length indicating the importance of these entities with respect to the articles they occur in. Since each of the features has been given equal weight during the calculation of IPI, these person entities with high NDL and NTF have been identified among the top 10 influential persons. The person entity "mr got" ranked as a high influential person belonging to this category has actually been falsely detected as influential as the PNER seems to have misrecognized this entity as a person entity.

8.3 Evaluation

Due to the unavailability of ground truth consisting of influential people in the newspaper archives from November-December 1894, there is no way to validate our results. To broadly evaluate our results, a simple web search query with the person entity's name in the context of 19th century was done on the Wikipedia website for the top 30 influential persons of Lists L1 and L2 detected from the people gazetteer with 30 Topics LDA and 100 Topics LDA Model respectively.

Among the top 30, 16 person entities from List L1 and 14 from List L2 were found to be influential and popular in the 19th century across topic categories like theatre, politics, government, shipping, etc. Some of these influential persons from Lists L1 and L2 found in Wikipedia are shown in Figure 8.

Most of the false positives although influential in other respects but were not influential 'person' entities which can attributed to the incorrect PNER (Person Named Entity Recognition) on noisy OCR data. False positives are obtained for person entities such as "mr got" which is not a person entity at all and for entities such as "ann arbor" and "van cortlandt" which are in fact locations but got recognized as highly influential person entities.

The ranked list of the top 30 influential persons with their IPI from Lists L1 and L2 can be seen in the Appendix (Table 6. 7) where evaluation result for each person entity is also presented.

9 DISCUSSION

- We used a linear combination of each of the parameters in calculation of DI and IPI and assigned equal values to the weights associated with each of them by not favoring any specific parameter. This is evident from the results which do not consistently favor any specific parameter. The parameters defined are based on heuristics and can be re-weighted according to user requirements or new parameters can be defined to do so.
- The parameters for calculation of DI and IPI can also be learned by performing regression analysis using a manually developed sample of topmost influential people and obtaining the complete list of ranked

Fig. 8. Some of the top 30 influential persons obtained from the dataset and also found on Wikipedia during evaluation

influential people based on the learned parameters.

- The NDL(Normalized Document Length) parameter defined for calculation of DI is normalized using the maximum length of any document in the dataset. However, there might exist other ways of normalization of Document Length like using total number of tokens in a person entity's document list or total number of tokens in the complete dataset which can be experimented with according to the dataset.
- The topmost influential people contain several false positives also which occur not due to the influence measures defined but due to other factors discussed in Section **??**. Several location and organization names have been misrecognized as person entities after performing Spelling correction and PNER resulting in false detection of some highly influential entities like "van cortlandt", "ann arbor" , "sandy hook", etc. There is also the problem of resolution of person name co-references in cases where persons like "mrs martins" , "mrs oakes", etc. have been recognized as influential.
- The choice of parameters for topic detection also affects the detection of influential people which is evident from the fact that we get different ranking of influential people for the two different LDA Topic model settings used.

10 Conclusion

The problem of finding influential people from historical OCR news repository has been studied in this research. In studying this novel problem, our main aim was to develop a complete solution framework for this problem and present insights from the results obtained. We made novel contributions to the problem solution by implementing an evaluation algorithm for measuring accuracy of spell correction on dataset, developing a people gazetteer for facilitating the process of influential people detection and finally defining parameters and measures in the newspaper community to obtain the ranked list of influential people. Spelling correction algorithms with improved accuracy can certainly improve the influential persons results as well as use of a Named Entity Recognizer that can resolve co-referred person name issues in noisy OCR text. Topic detection algorithms also need to be designed to enable them to deal with noisy OCR text in a better manner as some of the topics we obtained using LDA came out to be garbled and were difficult to understand in order to perform human-assigned manual labeling on them and use them further for finding similarity across articles. We didn't consider Named

Entity Disambiguation into account while developing the people gazetteer for detection of influential people which is a difficult problem in itself since it is hard to disambiguate among persons with similar names that can occur in multiple topic related articles in newspapers. The problem still requires research with probably better spelling correction, named entity recognition, topic detection algorithms and stricter measures of calculation of influence score and ranking of influential persons.

The parameters we defined for measuring influence scores of persons in news articles are based on heuristics and can be re-weighted according to user requirements or new parameters can be defined based on the characteristics of an OCR newspaper dataset making it an open research problem.

Non-heuristic based estimation for finding influential persons can also be done using optimization approaches such as unsupervised multiple instance clustering [15]. This approach can be used to cluster person entities into "influential" or "non-influential" by considering each person entity as a bag with articles of their occurrence as the instances for each bag. It is a combination of Constrained Concave-Convex Procedure and Cutting Plane method with faster convergence. Such a method can avoid choosing of parameter weights, biasing of results with respect to any specific parameter and decide which article plays a role in determining whether a person is influential or not.

REFERENCES

[1] M. Bilenko, R. J. Mooney, W. W. Cohen, P. D. Ravikumar, and S. E. Fienberg, "Adaptive name matching in information integration," *IEEE Intelligent Systems*, vol. 18, no. 5, pp. 16–23, 2003.

[2] C. Friedman and R. Sideli, "Tolerating spelling errors during patient validation," *Comput. Biomed. Res.*, vol. 25, no. 5, pp. 486–509, Oct. 1992.

[3] J. Cheng, L. Adamic, P. A. Dow, J. M. Kleinberg, and J. Leskovec, "Can cascades be predicted?" in *Proceedings of the 23rd international conference on World wide web*. International World Wide Web Conferences Steering Committee, 2014, pp. 925–936.

[4] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2003, pp. 137–146.

[5] K. Lerman and T. Hogg, "Using a model of social dynamics to predict popularity of news," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 621–630.

[6] N. Agarwal, H. Liu, L. Tang, and P. S. Yu, "Identifying the influential bloggers in a community," in *Proceedings of the 2008 international conference on web search and data mining*. ACM, 2008, pp. 207–218.

[7] M. Cha, H. Haddadi, F. Benevenuto, and P. K. Gummadi, "Measuring user influence in twitter: The million follower fallacy." *ICWSM*, vol. 10, pp. 10–17, 2010.

[8] H. Dutta, R. J. Passonneau, A. Lee, A. Radeva, B. Xie, D. L. Waltz, and B. Taranto, "Learning parameters of the k-means algorithm from subjective human annotation." in *FLAIRS Conference*, 2011.

[9] I. Chattopadhyaya, K. Sirchabesan, and K. Seal, "A fast generative spell corrector based on edit distance," in *Advances in Information Retrieval*. Springer, 2013, pp. 404–410.

[10] K. J. Rodriquez, M. Bryant, T. Blanke, and M. Luszczynska, "Comparison of named entity recognition tools for raw ocr text," in *Proceedings of KONVENS*, 2012, pp. 410–414.

[11] D. M. Blei, "Probabilistic topic models," *Communications of the ACM*, vol. 55, no. 4, pp. 77–84, 2012.

[12] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[13] D. Newman, A. Asuncion, P. Smyth, and M. Welling, "Distributed algorithms for topic models," *The Journal of Machine Learning Research*, vol. 10, pp. 1801–1828, 2009.

[14] A. K. McCallum, "Mallet: A machine learning for language toolkit," 2002, http://mallet.cs.umass.edu.

[15] D. Zhang, F. Wang, L. Si, and T. Li, "M3ic: Maximum margin multiple instance clustering." in *IJCAI*, vol. 9, 2009, pp. 1339–1344.

Person Name	IPI	Whether found on Wikipedia	Comments
capt creeten	3.380151	no	spelled incorrectly;capt creedon
capt hankey	3.022371	yes	
capt pinckney	2.933288	yes	
john macdonald	2.854389	yes	
john martin	2.827969	yes	
aaron trow	2.814171	yes	fictional character
mrs oakes	2.791536	no	false positive
buenos ayres	2.767399	no	location name
alexander iii	2.742552	yes	
mr got	2.736363	no	false positive
mrs martin	2.719383	no	false positive
ann arbor	2.681657	no	location name
caleb morton	2.63808	no	fictional character
anthony comstock	2.633381	yes	
toledo ann arbor	2.610495	no	location name
john thompson	2.609841	yes	
nat lead	2.594452	no	false positive
ed kearney	2.543152	yes	name of horse
van cortlandt	2.533131	no	location
louis philippe	2.523525	yes	
mrs talboys	2.522888	yes	fictional character
jim hooker	2.500915	yes	false positive
marie clavero	2.497384	no	false positive
father watson	2.450817	no	false positive
james mccutcheon	2.431448	no	part of an organization name
hugh allan	2.4287	yes	
william i	2.4222	yes	
marie antoinette	2.40731	yes	
schmitt berger	2.396639	no	spelled incorrectly;max f schmittberger
jacob schaefer	2.392976	yes	

TABLE 6
Table representing top 30 influential person entities detected from people gazetteer with 30 Topics LDA Model along with evaluation results and comments.

Person Name	IPI	Whether found on Wikipedia	Comments
capt creeten	3.333485	no	spelled incorrectly; capt creedon
mrs martin	3.23105	no	false positive
alexander iii	3.090361	yes	
capt hankey	2.975704	yes	
aaron trow	2.790838	yes	
john macdonald	2.774389	no	
mrs oakes	2.744869	no	false positive
john martin	2.711302	yes	
ed kearney	2.629342	yes	name of horse
caleb morton	2.614746	no	fictional character
john ward	2.57499	yes	
nat lead	2.571118	no	false positive
mrs talboys	2.499555	yes	fictional character
buenos ayres	2.490502	no	location
van cortlandt	2.490169	no	location
john thompson	2.482063	yes	
louis philippe	2.476858	yes	
marie clavero	2.474051	no	false positive
hardy fox	2.449248	no	
mme melba	2.415785	yes	
charles weisman	2.405938	no	false positive
hugh allan	2.405367	yes	
mr got	2.389697	no	false positive
schmitt berger	2.373305	no	spelled incorrectly
phil king	2.363644	yes	
henry a meyer	2.350396	yes	
north orlich	2.348236	no	false positive
james mccutcheon	2.338115	no	part of organization name
gen porter	2.330658	yes	
miller hageman	2.327831	no	

TABLE 7
Table representing top 30 influential person entities detected from people gazetteer with 100 Topics LDA Model along with evaluation results and comments.