FINDING INFLUENTIAL PEOPLE FROM A HISTORICAL NEWS REPOSITORY

Student Name: Aayushee Gupta

IIIT-D-MTech-CS-DE-12-030

Indraprastha Institute of Information Technology
New Delhi

Thesis Committee
Dr. Haimonti Dutta (Chair)
Dr. Srikanta Bedathur
Dr. Lipika Dey

Submitted in partial fulfillment of the requirements
for the Degree of M.Tech. in Computer Science,
with specialization in Data Engineering

Keywords: Gazetteer, Text Mining, Information Retrieval, OCR, Spelling Correction, Historical data, Influential people detection

Certificate

This is to certify that the thesis titled **"Finding Influential People from a Historical News Repository"** submitted by **Aayushee Gupta** for the partial fulfillment of the requirements for the degree of *Master of Technology* in *Computer Science & Engineering* is a record of the bonafide work carried out by her under our guidance and supervision in the Data Engineering group at Indraprastha Institute of Information Technology, Delhi. This work has not been submitted anywhere else for the reward of any other degree.

Dr. Haimonti Dutta
Indraprastha Institute of Information Technology, New Delhi

Abstract

Historical newspaper archives provide a wealth of information. They are of particular interest to genealogists, historians and scholars for People Search.

In this thesis, 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.

Acknowledgments

Any accomplishment requires the effort of many people and this thesis is no different. It gives me immense pleasure in recording my appreciation and a deep sense of gratitude to all the individuals who extended their unstrained cooperation in completing my thesis.

I express my sincere thanks and gratitude towards my advisor Dr. Haimonti Dutta for her continuous support and guidance extended to me in making this thesis possible and providing me an opportunity to work on an interesting research problem. I am highly obliged for the patience, motivation, enthusiasm and knowledge provided by her throughout the research and writing of this thesis.

Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Srikanta Bedathur and Dr. Lipika Dey, for their encouragement and insightful comments.

Last but not the least, I thank my fellow colleagues who were always there to discuss my problems and provide me moral boost- up in developing this thesis.

Aayushee Gupta

Contents

List of Figures

List of Tables

Chapter 1

Introduction

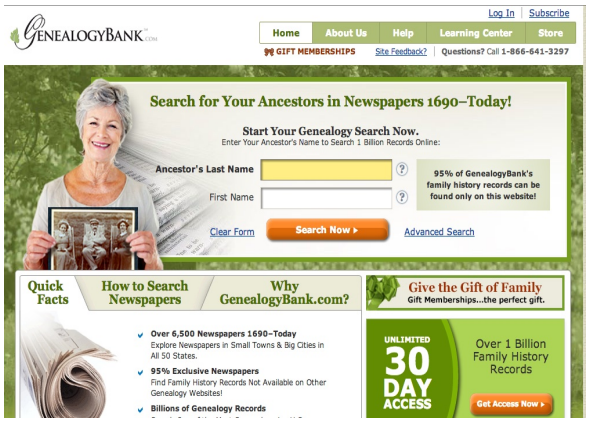
1.1 Motivation

Newspapers are rich sources of history and millions of pages of historical newspapers have been digitized [4] in recent years. A national program to develop an Internet-based, searchable database of U.S. newspapers called the National Digital Newspaper Program (NDNP) was setup in 2004 as a partnership between the National Endowment of Humanities (NEH) and the Library of Congress. Since then, several public and for-profit sectors have also digitized newspapers at a rapid pace making text from historical records available at a staggering rate. To deal with this wealth of information, scholars have mainly focused on text mining and information retrieval techniques followed by visualization of patterns extracted from the records [6,18,26,33,34,37,43].

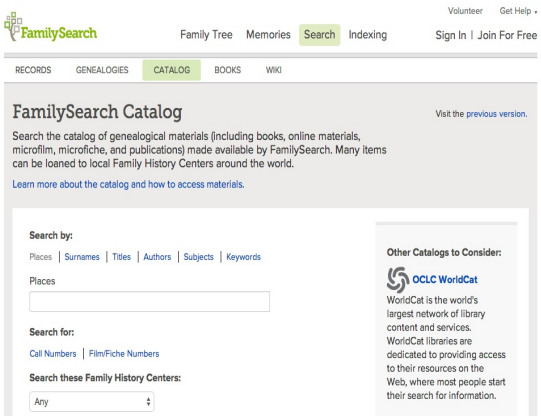
An important use of historical newspapers is for People Search [7, 20]) – for example, to find important people and track the timelines of news articles related to them. Several websites like Genealogy Bank[1], FamilySearch[2], Newspaper Archives[3], Ancestry[4] provide people search service that include obituaries, birth and death lists, newspaper articles, military records, Revolutionary and Civil War pension requests, census records, land grants and other forms of petitions. Figure 1.1 shows two such tools available online.

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.

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



(a) GenealogyBank



(b) FamilySearch

Figure 1.1: Two examples of online People Search Tools which utilize historic newspaper archives, military records, petitions, obituaries, census records and other forms of petitions to match proper nouns.

1.2 Problem Description

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

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 [14] 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".

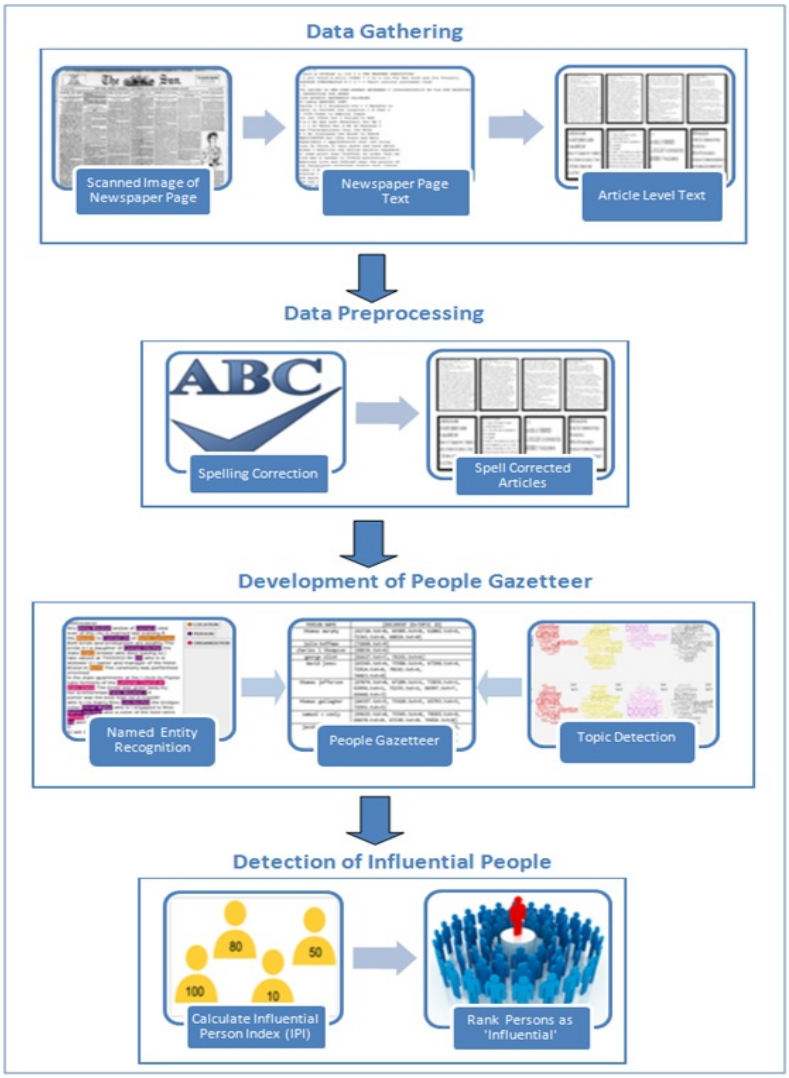


Figure 1.2: Research Framework showing components of proposed solution

1.3 Research Framework

We propose the solution framework in Figure 1.2 for finding influential people from a historical news repository. Each component of the framework is briefly described as follows:

1. **Data Gathering**: This component describes the source of data along with relevant statistics. It also includes descriptions of how the page level newspaper images are converted to text through OCR followed by article level segmentation. The different types of OCR

errors encountered in the data are also described. This component is further discussed in Chapter 3.

2. **Data Preprocessing**: This component describes the preprocessing applied on the news articles. It describes relevant spelling correction algorithms, and then presents a novel algorithm for evaluation of the results. This component is further discussed in detail in Chapter 4.

3. **Development of People Gazetteer**: This component describes the process of development of people gazetteer which involves Named Entity Recognition in order to find person entities. This is followed by topic detection using LDA to assign topics to news articles and link both to obtain an organized structure. This component is discussed in detail in Chapter 5.

4. **Influential Person Identification**: This component defines an "Influential Person Index" (IPI) that incorporates several criteria for identifying and ranking of "influential people". Details about IPI, ranking and final results with some case studies are discussed in Chapter 6.

1.4 Research Contributions

This thesis 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.

Chapter 2

Related Work

2.1 Learning from newspapers

Crowdsourcing has been used extensively in historical newspaper archives in recent years to digitize, create, clean and process content and provide editorial or processing interventions. For example, the Australian Newspapers Digitization Program [1] allows communities to explore their rich newspaper heritage by enabling free online public access to over 830,000 newspaper pages containing 8.4 million articles. The public enhanced the data by correcting over 7 million lines of text and adding 200,000 tags and 4600 comments [21–24]. The California Digital Newspaper Collection (CDNC)[1], which contains 61,412 issues comprising 545,955 pages and 6,364,529 articles from newspapers published in California between 1846-1922 has also crowdsourced text correction. The National Library of Finland embraced the idea of making crowdsourced text correction a game – users corrected their digitized newspapers by playing the game "Hunt the Mole!" [16]. The program consists of two games featuring adventures of a mole. In Mole Hunt the players are shown two different words and they must determine as quickly as possible if the words are the same. In Mole Bridge players try to write the word that appears in the screen correctly.

Several digital humanities projects that have used machine learning and natural language processing techniques to learn from historic newspaper archives are relevant to this work – the libraries of Richmond and Tufts have examined the Richmond Times Dispatch during the civil war years for more than two decades and their work focuses on automatic identification and analysis of full OCR text in newspapers to provide advanced searching, browsing and visualization [17]. The focus of this work was on named entity extraction and ten categories prominent in these newspapers were studied including ship names, railroads, streets and organizations. In an earlier project at the universities, the Perseus project [44–46], a general system to extract dates and names from text was developed in order to detect significant events in document collections.

[36] use a combination of Statistical Topic Modeling and Named Entity Recognition techniques

[1] http://cdnc.ucr.edu/cgi-bin/cdnc

for analyzing the entities, topics trends and topics that relate entities mentioned in a news articles dataset. They also create networks based on the topic model based relationships among the entities. [30] discuss their approach for designing a news analysis system [2] where information about several types of entities can be searched. They allow searching over all entities found in the news sources, present juxtaposition for each entity, i.e., other entities mentioned in context, temporal and spatial analysis, popularity time series graph in terms of number of number of references and co-reference names for the entity. Both these research works stress on person entities in a newspaper environment but do not focus on finding influential entities in which respect our research is different from their work.

2.2 Developing Gazetteers

Different types of gazetteers are discussed in [3]. They define gazetteers as set of lists containing names of entities such as cities, organizations, days of the week, etc. along with their types. They use gazetteer either as set of entity list or as a processing resource which is used to find occurrences of the entity names in text, e.g. for the task of named entity recognition. We use this definition to develop our People Gazetteer as a processing resource that finds person name entities from the news articles repository, associates each unique person entity from news articles with a list of articles of its occurrence and their respective topic.

Gazetteer lists are also discussed in [11] where they are used for learning name entity tagger using partial perceptron and aid in performing better NER compared to CRF based entity taggers. [53] discuss automatic generation of gazetteer list by finding entities with similar type labels from Wikipedia articles which can further be used for the purpose of NER. The evaluation is done over scientific domain of Archeology considering subject, temporal terms and location as named entities but no evaluation is presented for person entities. There is also no relevant work that builds or uses historical person names gazetteer list for data mining that we know of.

We discuss more related work regarding data preprocessing using Spelling correction algorithms in Section 4.1.1 and finding influential people in Section 6.1.

[2] http://www.textmap.com
[3] http://gate.ac.uk/sale/tao/splitch13.html

Chapter 3

Data Description

This chapter describes the dataset used for developing the People Gazetteer. Following sections provide details of data source, characteristics and some data statistics.

3.1 Data Source

The dataset has been taken from Chronicling America. *Chronicling America*[1] 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. The New York Public Library (NYPL) is part of this 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 used in [18] 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.

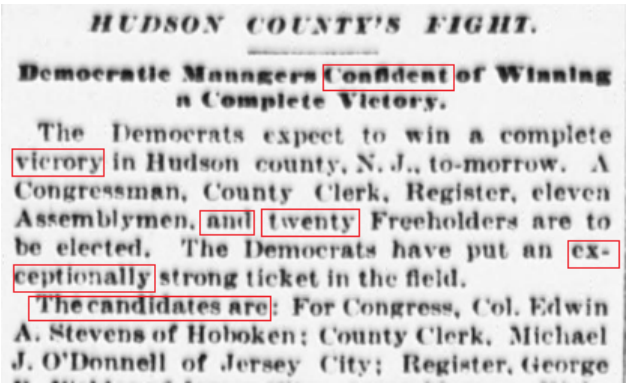
The newspaper titles and digitized pages available through the Chronicling America website can be searched using the OpenSearch protocol[2]. Unfortunately, the current search facilities are rudimentary and irrelevant documents are often more highly ranked than relevant ones. The newspapers are scanned 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 contains a large amount of garbled text. In a bid to serve its patrons better, the New York Public Library employed human annotators to clean headlines of articles and text, but the process of manually reading all the old newspapers article-by-article and cleaning them soon became very

[1] `http://chroniclingamerica.loc.gov/`
[2] `http://www.opensearch.org/Home`

expensive.

3.2 Data Characteristics

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



HUDSON COUNTY'S FIGHT.

Democratic Managers Confident of Winning a Complete Victory.

The Democrats expect to win a complete victory 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 ex-ceptionally 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
```

Figure 3.1: Scanned Image of a Newspaper article (left) and its OCR raw text (right)

- **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 3.1, 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 3.1, 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 3.1, 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 3.1, 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 3.1, 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.

3.3 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 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 [3] 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.

[3] https://power.ldeo.columbia.edu/twiki/pub/Incubator/BodhiDBDesign/Final ERD.pdf

Chapter 4

Data Preprocessing

This chapter describes the preprocessing steps applied on the historical news articles. The garbled OCR text makes data preprocessing mandatory before application of any text mining algorithms.

4.1 Spelling Correction

Several kinds of spelling errors exist in the data as described in chapter 3. This chapter first provides a brief review of spelling correction algorithms that exist in literature (Section 4.1.1); Section 4.1.2 describes the spelling correction algorithm used in this research and evaluation results on the OCR dataset are presented in Section 4.1.4 followed by discussion in Section 4.1.5.

4.1.1 Related Work

Kukich [27] comprehensively discusses various spelling correction techniques based on Non word, Isolated word and Real word spelling errors. N-gram analysis, Dictionary lookup and Probabilistic techniques are used for correcting isolated and nonword errors while Context-Dependent techniques are used mostly for correcting real word errors including the correction of word split and join errors [19].

N-gram techniques work by examining each n-gram in the text string and comparing against a pre-compiled table of n-gram statistics to retrieve the correct word while Dictionary look up techniques directly check whether the text string appears in the dictionary using string matching algorithms. Both techniques require a dictionary or a large text corpus and take frequency of n-grams or word occurrence into account in order to find the correct spelling [47], [40]. Probabilistic techniques use transition and word confusion probabilities to estimate likelihood of the correction in order to rank and retrieve correct word spelling.

On the other hand, Context-dependent techniques require contextual information and use either extensive NLP techniques or Statistical Language Modeling (SLM) for spelling correction.

Bassil and Alwani [5] use Google 1-5 gram word dataset to gain context information in order to determine the correct words sequence in the text for correction. Tong and Evans [49] use SLM approach involving information from letter n-grams,character confusion and word bi-gram probabilities to perform context sensitive spelling correction obtaining a 60 percent error reduction rate.

All these spelling correction techniques have developed over time and have been used in combination to achieve improved accuracy [10]. [3] use a combination of Google suggestions, LCS and character confusion probabilities for choosing the correct spelling on a small set of historical newspaper data and achieve recall and precision of 51% and 100% respectively.

Edit distance approach, suggested initially by Wagner and Fischer [51], is a dictionary lookup approach commonly used for OCR data correction because of the large number of substitution errors in OCR data [27] [15] which can be corrected using this technique. String edit distance approaches with faster correction are discussed in [31], [42] with variants like Levenshtein automata and normalized edit distance.

Personal name spelling correction has also been studied separately including comparison among various techniques indicating that there is no one single technique that outperforms all others though pattern matching techniques result in better matching quality compared to phonetic encoding techniques [15]. Almost all the personal name spelling correction techniques use personal names directory/dictionary for matching against wrongly spelled names in the dataset or queries in case of People Search [50].

We use edit distance algorithm for spelling correction because of its speed and ability to correct OCR errors compared to the n-gram approach [13]. Context-dependent spelling correction is not used because of unavailability of n-gram words corpus or ground truth dataset containing OCR and true word pairs. Our edit distance algorithm also uses an enhanced dictionary for look up to give significance to personal names spelling correction in the dataset.

4.1.2 Spelling Correction Algorithm

The Edit Distance algorithm based on Levenshtein distance [29] has been used for spelling correction. It is an isolated word correction technique that uses dictionary based-look up method and distance between strings for matching the text and correcting it. An "edit distance" corresponds to the minimum number of insertions, deletions, and substitutions required to transform one string into another. The algorithm corrects Non-Real Word spelling errors up to an edit distance of 2 , i.e. , it corrects words which have spelling errors that can be corrected by making at most 2 operations of insertion, deletion and substitution of letters in the word. The choice of 2 is governed by the trade off between algorithm runtime and quality of spelling correction. A bigger value improves the spelling correction accuracy but increases the runtime also while a smaller value decreases accuracy and the algorithm runtime. The spelling corrector has been designed as suggested by Peter Norvig[1]. The algorithm requires a dictionary which is used to

[1] http://norvig.com/spell-correct.html

check if each word of the text exists in it or not. If the word already exists in the dictionary then no change is made to the word and if not, then a candidate list of words is created from the word to be corrected by inserting, substituting or deleting up to 2 letters from it. This list of words is again checked for in the dictionary and returned as suggestions for the word to be corrected. The correction is made with the word formed from lowest edit distance and occurring with more frequency in the dictionary. This makes the edit distance algorithm dependent on the type of dictionary chosen for correction which means the dictionary must be well chosen for spelling correction of a specific document collection. The algorithm runs faster by reading the dictionary only once and keeping a data structure in memory for its word counts which can be referred to whenever a word comes up for correction.

4.1.3 Spelling Correction Algorithm Evaluation

There has not been much related work regarding automatic evaluation of word-by-word post spelling correction on OCR dataset consisting of Word Split and Join errors. Semi-automatic spelling correction system is discussed in [48] that corrects these errors but requires user interaction in order to perform complete correction and system evaluation. Rice [39] discusses OCR errors similar to the ones in our dataset. Their algorithm evaluates edit distance spelling correction by estimating word accuracy defined as the percentage of correctly recognized words; the length of LCS between correct and incorrect strings on a page by page level is used as the relevant metric. The evaluation strategy works correctly but the definition of accuracy does not give a complete coverage of the spell correction as it does not provide any information on the errors missed by the spelling corrector.

For evaluation on our dataset, the raw OCR text and OCR text after application of spelling correction algorithm needs to be compared with the original newspaper text. The OCR text is extremely garbled with Word Split and Join errors due to which word-to-word alignment with the original newspaper text is impossible. For this purpose, a novel algorithm called SCE (Spelling Correction Evaluation) based on N-gram approach is proposed for automatic evaluation of the corrected text word-by-word against the manually corrected subset of the news articles dataset.

Following sections describe the evaluation parameters for estimating the performance of Spelling Corrector on the OCR dataset used along with the SCE algorithm:

Evaluation Parameters

- **Accuracy** The evaluation metric used for measuring the performance of Spelling correction algorithm is Accuracy which requires calculation of number of OCR errors that got corrected when compared to the original scanned newspaper text. The measure has been chosen so as to include the complete text coverage and not just check for words that got corrected after spell correction as in the latter case, the number of FP and TN get missed which won't give the correct measure of how well the spell corrector works. The formula used

for calculating Accuracy is defined by Manning and Schutze,1999 (p.268-269) as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

where,

TP=Number of True Positives,

TN=Number of True Negatives,

FP=Number of False Positives,

FN=Number of False Negatives.

The aim of the SCE algorithm is to make a word-to-word correspondence between the OCR corrected text and the original OCR text and to mark each token in the OCR text as a TP, FP, TN or FN. Reynaert and Martin [38] suggest a way to define these terms by distinguishing between correct words and incorrect words in the text through the set of non-target, target and selected words and use Precision and Recall evaluation measures for measuring performance of spelling correction.

According to our spelling corrector, a "true positive" is said to occur when a word from the OCR text gets corrected and the corrected spelling matches the one in original article text while a "false positive" occurs if the corrected spelling does not match the corresponding word in the original article text. A "true negative" occurs when a word does not get corrected by the algorithm as it is already correct and matches the correct word in the original text also. On the other hand, a "false negative" occurs when the algorithm is unable to correct the word (there is no change in spelling of the word) and it does not match the corresponding word in the original text but should have been corrected.

•**Time taken for Spelling Correction** Time is also an essential parameter while measuring the performance of spelling correction. Since the dataset is quite large, it is important that the algorithm does not take too long to correct an article and needs to be parallelized in case it takes more time for correction.

•**Person Names Detection Rate(PNDR)** The spelling correction algorithm is evaluated on the basis of another parameter which is used to consider the special case of person entity names spelling correction, as the main goal of research is to detect these names with correct spellings. Person Names Detection Rate can be defined as the ratio of person names recognized through Named Entity Recognition (NER) before spelling correction process and the total number of person names recognized in the original newspaper articles.

$$PersonNamesDetectionRate = \frac{\text{Person Names recognized before/after spelling correction}}{\text{Person Names recognized in original newspaper articles}}$$

SCE (Spelling Correction Evaluation) Algorithm

The SCE algorithm is based on an N-word grams approach. To make the correspondence between corrected and original OCR text, a window of n-word grams in the scanned image text article is considered (Original.txt) which can be seen in a diagrammatic representation in Figure 4.1.

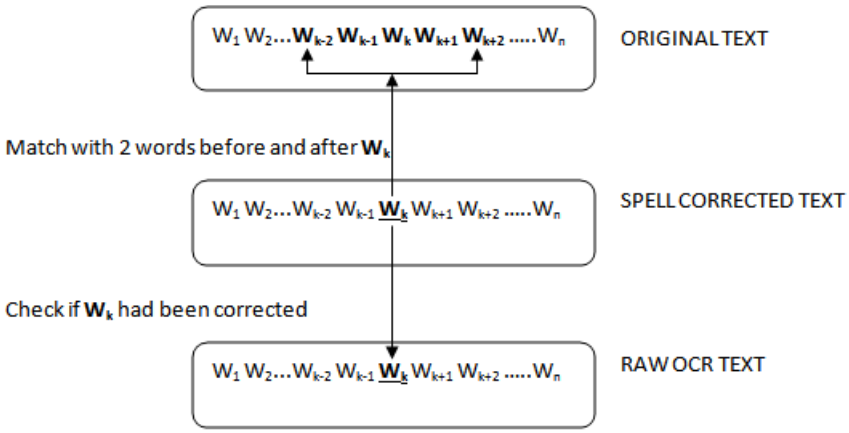


Figure 4.1: Schematic diagram for alignment of spell corrected article text with original article text for a word W_k

For each token in the spell corrected text (Corrected.txt), the corresponding token in the scanned text article along with 2 tokens before and 2 tokens after it are considered for alignment[2]. If the token being considered matches with any of these words in the scanned text article words window and its spelling has been corrected when compared to the corresponding token in raw OCR text (OCR.txt), then it is marked as a "True Positive" which is actually rewarding the Spell corrector for making the correct spelling change. A "False Positive" is marked if it does not match any of the words despite its spelling being corrected. If the token being considered matches any of the words in the words window but no spelling correction has been made for it, then it is marked as a "True Negative" and if it does not match any word in the window and the spelling corrector also did not correct it, then it is marked as a "False Negative" as the word got missed by the corrector.

Several cases could occur like difference in the lengths of linetext between OCR and Original text or while considering the first, second or the last tokens from the Corrected text for which the corresponding word window in Original text needs to be smaller. All such cases have been covered in SCE Algorithm 1 which calls function 'MatchWordGrams' (Algorithm2) for these different cases.

A limitation of the SCE algorithm is that it requires all 3 versions of a newspaper article (Original, Corrected and OCR) to have the same number of lines. In case of difference in the number of lines of text due to some Word Split and Join errors, the words window needs to be extended so as to cover previous and next line texts also for alignment.

[2] The choice of 2 is based on the Word Split and Join errors in the dataset. Choosing n=2 allows a window of

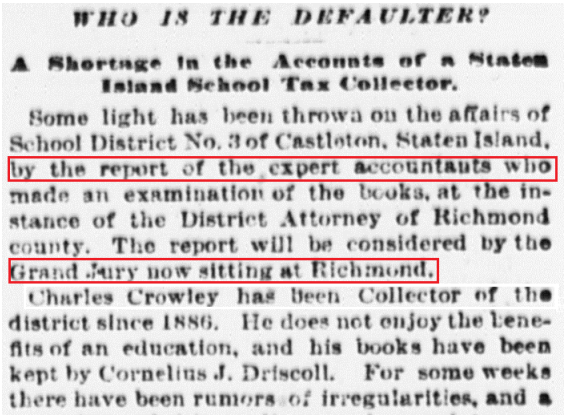
```
Input: Ocr.txt,Corrected.txt,Original.txt
Output: Spell Corrector Accuracy
OcrLine:=a line of text from Ocr.txt;
CorrectedLine:=a line of text from Corrected.txt;
OriginalLine:=a line of text from Original.txt;
tp ←0 fp ←0 tn ←0 fn ←0;
for (int i=0; i < CorrectedLine.length ; i++) do
    if (CorrectedLine.length<4 || OriginalLine.length<4) then
        MatchWordGrams((OcrLine,CorrectedLine,OriginalLine,0,OriginalLine.length,i));
    end
    else
        if (i==0) then
            MatchWordGrams((OcrLine,CorrectedLine,OriginalLine,0,3,0));
        end
        else if (i==1) then
            MatchWordGrams((OcrLine,CorrectedLine,OriginalLine,0,4,1));
        end
        else if (i==(CorrectedLine.length-2) || (CorrectedLine.length-1) ||
        (CorrectedLine.length) || (CorrectedLine.length+1)) then
            MatchWordGrams((OcrLine,CorrectedLine,OriginalLine,i-
            2,OriginalLine.length,i));

        end
        else if (i >= CorrectedLine.length+2) then
            MatchWordGrams((OcrLine,CorrectedLine,OriginalLine,OriginalLine.length-
            3,OriginalLine.length,i));

        end
        else
            MatchWordGrams((OcrLine,CorrectedLine,OriginalLine,i-2,i+2,i));
        end
    end
end
Accuracy = (tp + tn)/(tp + tn + fp + fn);
```

Algorithm 1: SCE Algorithm for Spell Correction

words to be considered to make up for the alignment lost because of Word Split and Join errors.

```
function MatchWordGrams(OcrLine, CorrectedLine, OriginalLine, jstart, jend, i)
    for (int j=jstart; j<jend; j++) do
        if
    ((CorrectedLine[i].equals(OriginalLine[j]))&&(!(OcrLine[i].equals(CorrectedLine[i]))))
    then
            tp = tp + 1;
            flag0=false;
            return tp;
        end
        else if
    ((CorrectedLine[i].equals(OriginalLine[j]))&&(OcrLine[i].equals(CorrectedLine[i])))
    then
            tn = tn + 1;
            flag1=false;
            return tn;
        end
    end
    if (!(OcrLine[i].equals(CorrectedLine[i]))&&flag0==true) then
        fp = fp + 1;
        return fp;
    end
    else if ((OcrLine[i].equals(CorrectedLine[i])) && flag1==true) then
        fn = fn + 1;
        return fn;
    end
end function
```

Algorithm 2: MatchWordGrams Function called by Algorithm 1



WHO IS THE DEFAULTER?

A Shortage in the Accounts of a Staten Island School Tax Collector.

Some light has been throwa on the affairs of School District No. 3 of Castleton, Staten Island, by the report of the expert accountants who made an examination of the books, at the instance of the District Attorney of Richmond county. The report will be considered by the Grand Jury now sitting at Richmond.

Charles Crowley has been Collector of the district since 1886. He does not enjoy the benefits of an education, and his books have been kept by Cornelius J. Driscoll. For some weeks there have been rumors of irregularities, and a

```
1  WHO IS THE DEFAULTER
2  A Shortage in the accounts of a States
3  Island School Tax Collector
4  some light has been thrown on the affairs of
5  School District No 3 of Castlaton Staten Island
6  by the report of the expert accountants who
7  made an examination of the books at the in
8  stance of the District Attorney of Richmond
9  county The report will be considered by the
10 Grand jury now sitting at Richmond
11 Charles Crowley has Been Collector of the
12 district since 1886 He does not enjoy the bene
13 fits of an education and his books have been
14 kept by Cornelins J Driscoll For some weeks
15 there have been rumors of irregularities and a
```

Figure 4.2: Scanned image of a newspaper article (left) along with its original text (right)

```
1  into Ji Jin tntfAtnTKiit
2  A Hhortim In he Aeeonnt nf HtntMS
3  lalaad Heknol Tux 4ollrelor
4  tome llRlil Imi h cn tlimwa on tho affairs of
5  SVhool District Ko iof Castlolon Staten Island
6  by tltn rejmrt of th cxpert aocountauts who
7  msdn an examination of tho bcoka nt the In
8  nUnco of tho District Attorney of ltlchmon1
9  cntmty Tho report will bo considered by the
10 Irnniluttry iiownllllnu at tilchmond
11 Chnrles Crowley ha llecli Collector nf the
12 district since 1NNO lie iloe nnt onjoy the lene
13 fltnof nn education nnd his books have been
14 kept by C rnlln J Drlicoll Kor sorao weeks
15 there have been rumors of Irrecuturltles and a
```

```
1  into Ji Jin tntfAtnTKiit
2  A Portim In he Aeeonnt of hints
3  Leland Henok tax 4ollrelor
4  time Llull Imi h cn Limca on the affairs of
5  school District Ko of CASTILLON state Island
6  by than report of the expert accountants who
7  Sdn an examination of the Boka it the In
8  lunch of the District Attorney of ltlchmon1
9  entity Tho report will bo considered by the
10 Irnniluttry iiownllllnu at Richmond
11 Charles Crowley ha Lesli Collector of the
12 district since into lie Ilon not enjoy the line
13 fltnof an education and his books have been
14 kept by C roll J Driscoll For Sorbo weeks
15 there have been rumors of Irrecuturltles and a
```

Figure 4.3: OCR raw text (left) and Spell corrected text (right) of the article

An Example

Working of the SCE algorithm can be demonstrated with the help of the following example: Consider 3 versions of a scanned image of a newspaper article, the original text of the scanned image in Figure 4.2 and the raw OCR text and the corrected text (after spell correction) in Figure 4.3. As highlighted in the figures, for line 6 the line texts are:

OcrLine= *by tltn rejmrt of th cepert aocountauts who*

CorrectedLine= *by than report of the expert accountants who*

OriginalLine= *by the report of the expert accountants who*

Here, for each token of CorrectedLine, we find its index and call the MatchWordGrams function accordingly. For the first token 'by' at index i=0 in CorrectedLine, we consider the word window to be "by the report" (index j=0 to 2) in OriginalLine by matching iteratively with each token to see if there is a match and also if there has been a spelling correction by comparing with the corresponding token in OcrLine. Here, no change was made to the spelling of 'by' and it matches with a word in words window, so it is marked as a FN. For the second token 'than' at index i=1, we consider the word window to be "by the report of" (index j=0 to 3) for which there

is no match in the window but there has been a spelling correction from 'tltn' to 'than', which implies the correction was wrong and the token is marked as a FP. For the third token 'report' at index i=2, we consider the window as "by the report of the" (index j=0 to 4) in Original Line and find that there is a match in the word window and there has been a spelling correction too from 'rejmrt' to 'report' which makes this token a TP. Similarly, rest of the tokens get marked for each line in the Corrected.txt.

Another example can be considered from Line 10 in Figure 4.2 and Figure 4.3 where the number of tokens is different in CorrectedLine and OriginalLine. In such a case, direct alignment between tokens is not possible because of which the words window becomes useful. Here, when the last token 'Richmond' of CorrectedLine is considered at index i=3, the corresponding words window becomes "Jury now sitting at Richmond" (index j=1 to 5) for which there is a match in the words windows and corresponding spelling has also been changed from 'tilchmond' to 'Richmond' which makes it a TP. Had the word window not been considered, the corresponding token at index j=3 in OriginalLine would have been chosen as 'sitting' which would have resulted in a FP.

4.1.4 Spelling Correction Algorithm Evaluation Results

Aims: The aim of our experiments is to answer the following questions:

- **Question 1:** How good is the spell corrector? The metrics for evaluation are accuracy and time to correct the text.
- **Question 2:** How good is the Person Names Detection Rate? The metric for evaluation is PNDR.

Materials: The spelling correction algorithm is used to correct all the 14020 OCR raw text articles in the dataset. The dictionary used for look-up is a concatenation of several public domain books from Project Gutenberg and lists of most frequent words from Wiktionary and the British National Corpus[3]. This is augmented with a large people names list which is obtained by running Stanford NER-CRF parser on subsets of the ClueWeb12 dataset made available in the TREC 2013 Crowdsourcing Track[4]. This enhanced dictionary has been used to give special consideration to correction of person names in the dataset.

Methods: In order to answer **Question 1** we do the following:

3 versions of each newspaper article are required: OCR raw text, spelling corrector corrected text and the original scanned newspaper article text. Since the dataset is quite large (14020) and it is not possible to get original text of each of these newspaper images, a smaller number of articles are chosen to study the results of spelling correction. 50 scanned newspaper images are taken and an online OCR [5] is run on them followed by some manual correction to get the

[3]http://norvig.com/big.txt
[4]http://boston.lti.cs.cmu.edu/clueweb12/TRECcrowdsourcing2013/
[5]www.onlineocr.net

original articles text. Accuracy can then be calculated for all 3 versions of 50 newspaper articles using the SCE algorithm by marking each word in the OCR text article as a TP, FP, TN or FN.

In order to answer **Question 2** we do the following:

1. Person Name Detection from raw OCR (**Baseline:**) The NER is run on the raw garbled OCR text.

2. Person Name Detection from spell corrected text (**PND+Spell Correction+Simple Dictionary:**)The NER is run on the spell corrected (using the edit distance algorithm) OCR text without the people names list in the dictionary.

3. Person Name Detection from spell corrected text with enhanced dictionary (**PND+Spell Correction+Enhanced Dictionary:**) The NER is run on the spell corrected (using the edit distance algorithm) OCR text with the enhanced people names list in the dictionary.

Results: The spelling corrector shows an Accuracy of 72.7% when corrected text is compared to OCR text and original article text. We believe that the results are less accurate due to the presence of a large number of Non-word, New Line, Word Split and Join errors in the OCR data which can not be corrected by the spelling correction algorithm used for this research.

The spelling corrector takes 9 seconds on an average to correct the newspaper OCR articles. It takes a total of 36 hours to run on 14020 articles.

But the spelling correction is useful in terms of Person Names Detection Rate *(PNDR)*. Following are the statistics obtained for *PNDR*:

PNDR for **Baseline:** : 71.5%

PNDR for **PND+Spell Correction+Simple Dictionary:** : 63.3%

PNDR for **PND+Spell Correction+Enhanced Dictionary:** : 85.5%

These statistics indicate that spelling correction using an extended dictionary for personal names is useful for detecting person names from the garbled newspaper articles and the results are dependent on the type of dictionary being used for spell correction. Table 7.1 shows some of the person names recognized while calculation of PNDR for each case. The reason for PNDR going down after spell correction with simple dictionary is demonstrated from the table as unnecessary spelling corrections are made while using this dictionary. On the other hand, since the enhanced dictionary contains several person names and no unnecessary spelling corrections are made, more persons name are recognized in this case when compared to the original text leading to a much higher PNDR.

4.1.5 Discussion

- We believe a better accuracy of spell correction can be obtained by correcting the New Line errors in the articles. This can be done by checking for if the word at last index of

a text line or the word at first index of the next text line is a word not present in the dictionary and combining the two and checking again in the dictionary for a valid word. The new word, if present in the dictionary can be replaced by the two words from which it is formed thereby removing the New Line error.

• Similar approach can be applied for Word Split and Join errors but would require each word of an article not present in the dictionary to be analyzed along with some window of words before and after it to make a correction.

• Choice of a dictionary for the edit distance algorithm affects the results as indicated by improvement in PNDR while using an enhanced person names dictionary. We believe using a dictionary with historical terms, places and people names can certainly perform spelling correction better and improve the accuracy also.

• Our spell corrector corrects Non-Real Word errors by focusing on isolated words in the dataset. Other spelling correction algorithms like context dependent spelling correction can also used to correct the dataset and measure accuracy using our SCE algorithm along with other evaluation parameters to compare among multiple algorithms and decide which one suits the dataset better and gives best accuracy.

Chapter 5

Development of People Gazetteer

People Gazetteer as defined in Section 1.4 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 chapter 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 5.1 and b) Assignment of topics to news articles using LDA topic detection in Section 5.2. Output of People gazetteer developed using these steps is presented in Section 5.3 followed by discussion in Section 5.4

5.1 Person Named Entity Recognition (PNER)

5.1.1 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.

5.1.2 Methodology

The Stanford CRF-NER[1] 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

[1]http://nlp.stanford.edu/software/CRF-NER.shtml

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 [41]. 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.

5.1.3 PNER Results

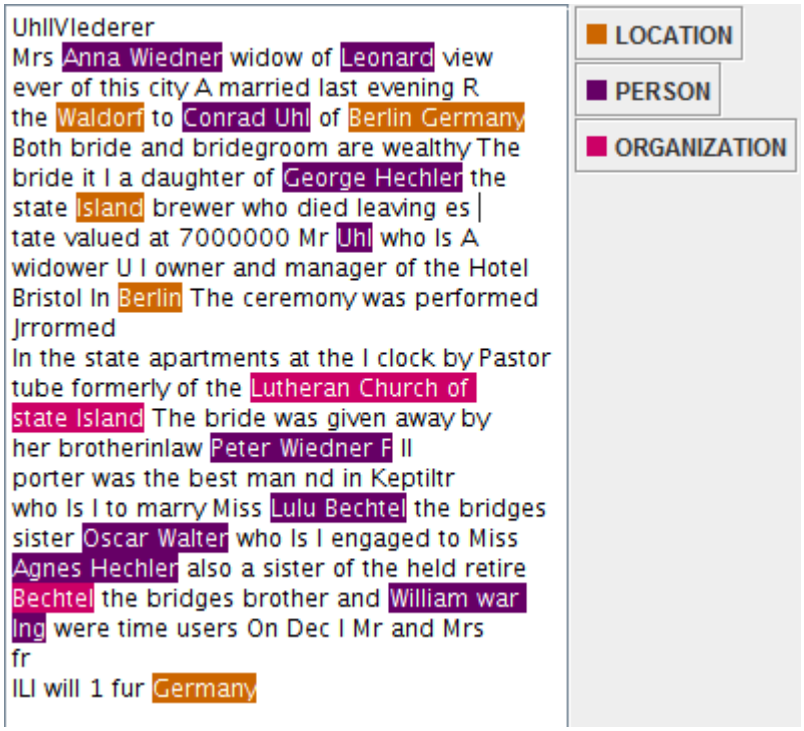


Figure 5.1: NER on a sample news article

NER on a sample news article from the dataset can be seen in Figure 5.1. 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 5.1 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:

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 5.1: Table showing output of PNER on 14020 articles

•**Marginally Influential** : This category includes all person entities with occurrence in less than 4 news articles. (38066 person entities as calculated from Table 5.1)

•**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.

5.2 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 is the process of identifying topics in a document collection using a topic model. A simple example of topic model illustrated by [8] can be seen in Figure 5.2.

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.

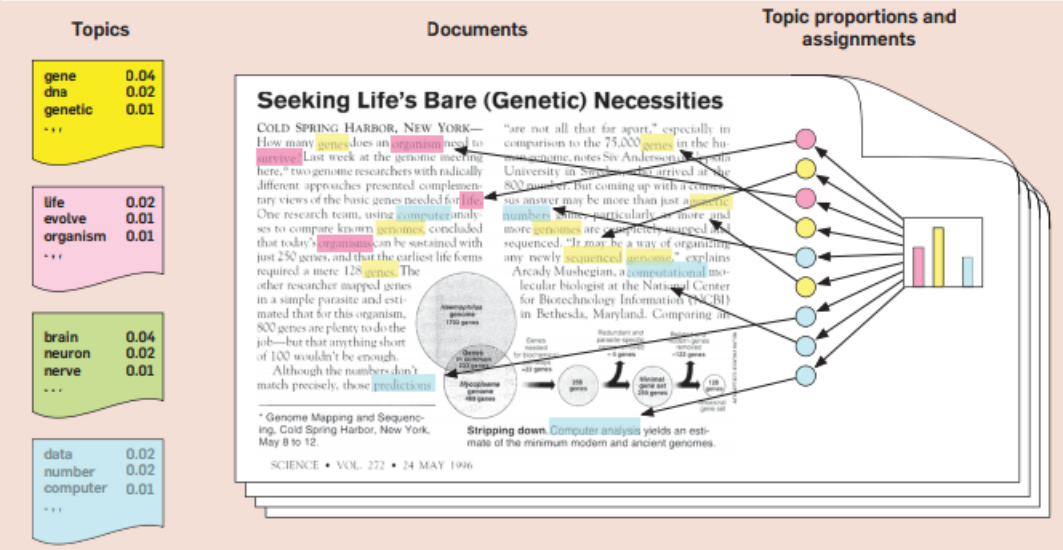


Figure 5.2: Simple topic modelling approach for a single article [8]

.

5.2.1 Topic Detection Model

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 [9]. 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 $\mathrm{x} = x_{ij}$, LDA inference is done by computing the posterior distribution over the latent topic assignments $\mathrm{z} = z_{ij}$, 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.

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 [35] 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_{wkp} and topic counts N_{kj_p}. The topic model inferencing then uses simultaneous local Gibbs sampling approach on each processor for a pre-decided 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.

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 [35] and [9] 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.

5.2.2 Results

The AD-LDA model as described in [35] and implemented in the Mallet [32] 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.

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

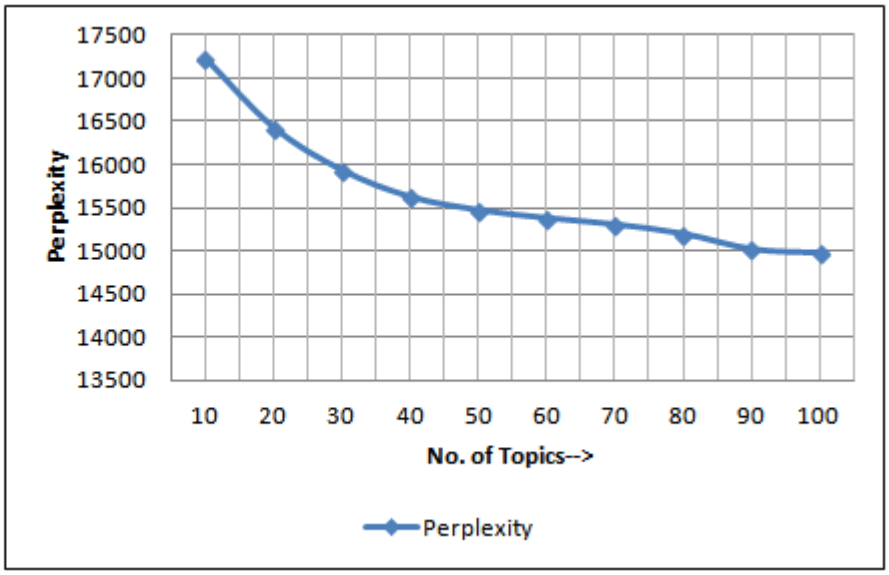


Figure 5.3: Test Set Perplexity versus Number of Topics for a random $90-10$ split of the data. The maximum number of words in each topic is 20, number of iterations 500 and the number of processors 4 for this experiment.

Perplexity is calculated by splitting the data into 90% for training and rest 10% for testing. Figure 5.3 shows the variation of the test perplexity versus the number of topics for one random $90-10$ split of the data[2]. 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 topics as 30 and 100 and demonstrate their effect on the influential people detection.

[2]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.

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 5.2 and the other model with 100 topics in Appendix Table 7.4. 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.

TOPIC ID	TOPIC WORDS
1	total ii won club score night ran furlough alleys tournament time mile fourth rolled curling scores race national game
2	la lu ot lo tu au tb ta ha tea day al aa ut ar uu wa tt te
3	iii lie tin nail tn lit hut ill ii nn thu tu anti thin inn hit lu lo nut
4	line street feet point western easterly northerly feel southerly distance place distant lo fret hue beginning laid early felt
5	opera theatre music company week play stage evening night performance concert mme audience manager season de orchestra house miss
6	great people life man women good country world american part ot ha made la years make long place bad
7	election mr party republican state district vote democratic county senator elected city committee mayor political candidate majority york democrats
8	time ho work tn men city bo lie anti day thin long thu made part ago lot york make
9	st room av sun wife board front lo december rent lot november sunday ht west ar house private si
10	dr book st story books cloth author cure free work york blood illustrated remedy goods medical library health price
11	church dr father funeral school st college sunday year rev catholic pastor services late service held society holy clock
12	horse race class horses won racing years prize record year show ring track mile money jockey trotting trotter ran
13	cent year week pf market total net stock today central st ft lit sales short cotton ohio lot month
14	white water indian black long found thu big dog time ground wild tree killed birds bird day great lake
15	price black silk goods prices ladies worth dress fine white full tea quality style wool made fancy cloth fur
16	street mrs mr avenue wife house miss yesterday years home woman night ago husband found died daughter children mother
17	war american government army chinese japanese china japan foreign united nov emperor states prince minister military french port navy
18	feet north minutes avenue boundary seconds degrees west york minute degree point east south feel city angle county laid
19	man ho men night back wa room left house told bad door found turned place ran lie front morning
20	water feet building boat company car train road fire miles railroad island work line city great river built bridge
21	club game team play football half ball left college back yale played harvard line eleven men match yacht field
22	ii iii ill lit ll si ti il im vi st iv ft mi li till lull lui oil
23	bank money national gold amount notes banks hank business treasury account cent paid bonds note currency company stock estate
24	mr john william york henry charles james club city ii george dec dr thomas smith jr brooklyn van held
25	piano st rooms car york daily chicago city sunday upright parlor furnished broadway hotel av west train brooklyn monthly
26	york daily steamship nov directed letter dec fur orleans al steamer walls letters close australia china japan city london
27	mr court police judge justice case yesterday street district witness jury charge asked attorney trial arrested lawyer told office
28	mr law present made public year state committee president secretary bill report states con tin united number meeting york
29	air ran ur fur ui full tt al tl late mr ant liar art lay told met ti tr
30	company york trust bonds city cent railroad mortgage interest wall bond stock street st central january coupon committee jan

Table 5.2: Table showing Topic ID and words obtained from the 30 Topics LDA model.

5.3 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 5.4 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 6 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}

Figure 5.4: Snapshot of People Gazetteer with Person names, Document list of occurrence and their corresponding Topic ID

5.4 Discussion

1. The Stanford NER, although being one of the best Named Entity Recognizers, is still not able to deal with the dataset of OCR noisy text successfully due to which many of

the person entities have been missed and not recognized from the dataset. This can lead to leaving out several potential influential person entities from the process of influential person detection.

2. The OCR text does not contain any punctuation due to which the PNER gives several false positives leading to a large number of garbled person entities in the People Gazetteer. In many cases, either the noisy words get recognized as person entities or they get recognized along with the person entity name making the person name useless for further analysis since it contains extra words in addition to the actual person name. This is one of the main causes of too many unnecessary person entities in the People Gazetteer. It is difficult to identify such people names and filter them out without having a dictionary which contains every possible historical person entity name. Even if all such cases are filtered out, the person entity names which are attached to the noisy words get removed which might remove some important potential influential person entities from further analysis.

3. The PNER isn't able to recognize individual entities when multiple person entities are mentioned together without punctuation in the OCR text. They are recognized as one long person entity (false positives get increased) making it impossible to separate out the individual person entities from it. This makes them useless for further analysis and again leads to missing out analysis on potential influential person entities. For example, in an article with the original line text as: "They gave money to Ronn, Collector. A. Augustus Healy, Speaker has been appointed..." has the OCR line text as: "They gave money to Ronn Collector A Augustus Healy Speaker has been appointed.." leading to the recognition of "Ronn Collector A Augustus Healy Speaker" as one single person entity.

4. The issues of co-reference resolution of person names (For Example, person entities such as "William Schmittberger","Captain Williams" are same but recognized as separate persons) and named entity disambiguation (Occurrence of different persons with similar name in news articles. For example, the person "John Smith" detected in two different articles might or might not be the same person) also occur in the People Gazetteer which are not taken care of by PNER and need to be addressed separately. While the issue of co-reference can be still addressed by analyzing each news article, it is extremely hard to disambiguate among persons with similar names that can occur in multiple news articles with different topics.

5. The LDA topic detection model is also not geared to be used on OCR dataset directly since it recognizes topics having completely meaningless words. This can be observed from the output of topic detection model in Table 5.2 where topics 2,3 and 22 have completely garbled words.

Chapter 6

Influential People Detection

This chapter describes the process of detection of influential people from the people gazetteer developed in Chapter 5 and its results with some case studies. Section 6.1 discusses some related work in the field of influential people detection, Section 6.2 the measures used to define an influential person in the newspaper environment followed by their ranking to obtain top influential persons across each person category with results in Section 6.3 and some points of discussion in Section 6.4.

6.1 Related Work

Influential people detection has been mostly done in the field of social networks, marketing and diffusion research. [25] 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.

[28] 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 [2] 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.

[12] 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.

6.2 Measuring Influence

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:

6.2.1 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.

6.2.2 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.

6.2.3 Procedure for finding influential persons

Algorithm 3 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.

```
function CalculateIPI
    Input: PeopleGazetter(Persons, (DocList, TopicList)), w_a,w_b,w_c
    Result: Ranked list of Person Name and IPI
    NTF ←0, NDL ←0, NSIM ←0, DI ←0, UniqT ←0, IPI ←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 3: Procedure to calculate IPI and rank person entities based on it

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 6.1: Description of the functions used in Algorithm 3

6.3 Results

Two ranked influential person lists, namely L1 and L2 are obtained after calculation of IPI from the people gazetteer (developed in Chapter 5) 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 6.2. 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 its Term Frequency share also play an important part in measuring influence. Figure 6.1 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:

6.3.1 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 6.3 and 6.4 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

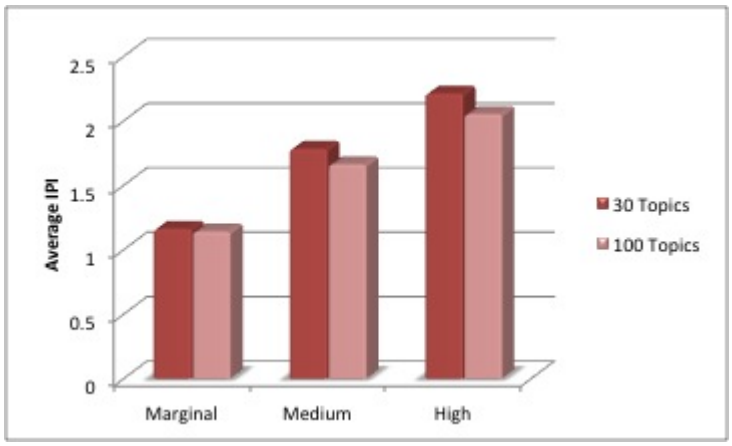


Figure 6.1: Comparison of the Average IPI for two ranked lists L_1 and L_2 using 30 and 100 topics respectively.

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 6.2: Table illustrating average statistics for each Person Category of People Gazetteer across 2 Topic Models

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 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 6.3: 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

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 6.4: 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.

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 6.3 and 6.4. 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.

6.3.2 Case Studies

Some of the topmost 10 influential person entities of lists L1 and L2 (Table 6.3 and 6.4) identified from each person category of the 2 people gazetteers are discussed below:

1. Highly Influential Category- This category as defined in Section 5.1.3 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 6.3 and 6.4 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 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.

6.3.3 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 6.2.

Figure 6.2: Some of the top 30 influential persons obtained from the dataset and also found on Wikipedia during evaluation

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 7.2. 7.3) where evaluation result for each person entity is also presented.

6.4 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 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 5.4. 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.

Chapter 7

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.

Most of the problems faced during this research surfaced due to the noise in OCR dataset used. The problem of finding influential persons from newspaper archives opens up a wide range of research problems as illustrated from the discussion sections 5.4 and 6.4. We believe each of the components of our research framework, namely Spelling correction on OCR news articles, Person Named Entity Recognition, Topic Detection and parameters definition to measure and rank influence scores of persons can be researched further in order to obtain more beneficial results from influential people detection.

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 [52]. 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.

Bibliography

[1] ADNP. Australian newspapers digitization program, 2008.

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

[3] Agarwal, S. Utilizing big data in identification and correction of ocr errors.

[4] Allen, R., Zhu, W., and Sieczkiewicz, R. What to do with a million pages of digitized historical newspapers? In *IConference* (Urbana-Champaign, IL, 2010).

[5] Bassil, Y., and Alwani, M. Ocr context-sensitive error correction based on google web 1t 5-gram data set. *arXiv preprint arXiv:1204.0188* (2012).

[6] Berberich, K., Bedathur, S., Neumann, T., and Weikum, G. A time machine for text search. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (2007), SIGIR '07, pp. 519–526.

[7] Bilenko, M., Mooney, R. J., Cohen, W. W., Ravikumar, P. D., and Fienberg, S. E. Adaptive name matching in information integration. *IEEE Intelligent Systems 18*, 5 (2003), 16–23.

[8] Blei, D. M. Probabilistic topic models. *Communications of the ACM 55*, 4 (2012), 77–84.

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

[10] Brill, E., and Moore, R. C. An improved error model for noisy channel spelling correction. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics* (2000), Association for Computational Linguistics, pp. 286–293.

[11] Carlson, A., Gaffney, S., and Vasile, F. Learning a named entity tagger from gazetteers with the partial perceptron. In *AAAI Spring Symposium: Learning by Reading and Learning to Read* (2009), pp. 7–13.

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

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

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

[15] Christen, P. A comparison of personal name matching: Techniques and practical issues. In *Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference on* (2006), IEEE, pp. 290–294.

[16] Chrons, O., and Sundell, S. Digitalkoot: Making Old Archives Accessible Using Crowdsourcing. In *HCOMP 2011: 3rd Human Computation Workshop* (2011).

[17] Crane, G., and Jones, A. The challenge of virginia banks: an evaluation of named entity analysis in a 19th-century newspaper collection. In *Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries* (2006), ACM, pp. 31–40.

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

[19] Elmi, M. A., and Evens, M. Spelling correction using context. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1* (1998), Association for Computational Linguistics, pp. 360–364.

[20] Friedman, C., and Sideli, R. Tolerating spelling errors during patient validation. *Comput. Biomed. Res. 25*, 5 (Oct. 1992), 486–509.

[21] Holley, R. Crowdsourcing and social engagement: potential, power and freedom for libraries and users. Tech. rep., november 2009.

[22] Holley, R. How good can it get? analysing and improving ocr accuracy in large scale historic newspaper digitisation programs. *D-Lib Magazine 15*, 3/4 (2009).

[23] Holley, R. Crowdsourcing: How and why should libraries do it? *D-Lib Magazine 16*, 3/4 (2010).

[24] Holley, R. Tagging full text searchable articles: An overview of social tagging activity in historic australian newspapers august 2008 - august 2009. *D-Lib Magazine 16*, 1/2 (2010).

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

[26] Khurdiya, A., Dey, L., Raj, N., and Haque, S. M. Multi-perspective linking of news articles within a repository. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three* (2011), AAAI Press, pp. 2281–2286.

[27] Kukich, K. Techniques for automatically correcting words in text. *ACM Computing Surveys (CSUR) 24*, 4 (1992), 377–439.

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

[29] Levenshtein, V. I. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet Physics Doklady* (1966), vol. 10, p. 707.

[30] Lloyd, L., Kechagias, D., and Skiena, S. Lydia: A system for large-scale news analysis. In *String Processing and Information Retrieval* (2005), Springer, pp. 161–166.

[31] Marzal, A., and Vidal, E. Computation of normalized edit distance and applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 15*, 9 (1993), 926–932.

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

[33] McKeown, K., and Radev, D. R. Generating summaries of multiple news articles. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (1995), SIGIR '95, pp. 74–82.

[34] McKeown, K. R., Barzilay, R., Evans, D., Hatzivassiloglou, V., Klavans, J. L., Nenkova, A., Sable, C., Schiffman, B., and Sigelman, S. Tracking and summarizing news on a daily basis with columbia's newsblaster. In *Proceedings of the second international conference on Human Language Technology Research* (2002), Morgan Kaufmann Publishers Inc., pp. 280–285.

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

[36] Newman, D., Chemudugunta, C., Smyth, P., and Steyvers, M. Analyzing entities and topics in news articles using statistical topic models. In *Intelligence and Security Informatics*. Springer, 2006, pp. 93–104.

[37] Radev, D. R. Topic shift detection - finding new information in threaded news. Tech. Rep. CUCS-026-99, Columbia University, 1999.

[38] Reynaert, M. All, and only, the errors: more complete and consistent spelling and ocr-error correction evaluation. In *LREC* (2008).

[39] Rice, S. V. *Measuring the accuracy of page-reading systems.* PhD thesis, University of Nevada, 1996.

[40] Ringlstetter, C., Hadersbeck, M., Schulz, K. U., and Mihov, S. Text correction using domain dependent bigram models from web crawls. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-2007) Workshop on Analytics for Noisy Unstructured Text Data* (2007).

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

[42] Schulz, K. U., and Mihov, S. Fast string correction with levenshtein automata. *International Journal on Document Analysis and Recognition 5*, 1 (2002), 67–85.

[43] Shahaf, D., and Guestrin, C. Connecting the dots between news articles. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining* (2010), ACM, pp. 623–632.

[44] Smith, D. A. Detecting and browsing events in unstructured text. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval* (2002), ACM, pp. 73–80.

[45] Smith, D. A. Detecting events with date and place information in unstructured text. In *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries* (2002), ACM, pp. 191–196.

[46] Smith, D. A., and Crane, G. Disambiguating geographic names in a historical digital library. In *Research and Advanced Technology for Digital Libraries*. Springer, 2001, pp. 127–136.

[47] Strohmaier, C. M., Ringlstetter, C., Schulz, K. U., and Mihov, S. Lexical postcorrection of ocr-results: The web as a dynamic secondary dictionary? In *ICDAR* (2003), Citeseer, pp. 1133–1137.

[48] Taghva, K., and Stofsky, E. Ocrspell: an interactive spelling correction system for ocr errors in text. *International Journal on Document Analysis and Recognition 3*, 3 (2001), 125–137.

[49] Tong, X., and Evans, D. A. A statistical approach to automatic ocr error correction in context. In *Proceedings of the fourth workshop on very large corpora* (1996), pp. 88–100.

[50] Udupa, R., and Kumar, S. Hashing-based approaches to spelling correction of personal names. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing* (2010), Association for Computational Linguistics, pp. 1256–1265.

[51] Wagner, R. A., and Fischer, M. J. The string-to-string correction problem. *Journal of the ACM (JACM) 21*, 1 (1974), 168–173.

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

[53] Zhang, Z., and Iria, J. A novel approach to automatic gazetteer generation using wikipedia. In *Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources* (2009), Association for Computational Linguistics, pp. 1–9.

Appendix

Original	**OCR**	**Spell Corrected (Simple Dictionary)**	**Spell Corrected (Enhanced Dictionary)**
James H Grady	James H Grady	Gilman James H Grady	Gilman James H Grady
Eugene Kelly	Eugene Kelly	Kelly −> belly	Eugene Kelly
Stephen S Blake	Stephen S Blake	Blake −> blame	Stephen S Blake
Abraham Gruber	Abraham Gruber	Gruber −> rubber	Abraham Gruber
John Mcbride	John Mcbride	Mcbride −> bride	John Mcbryde
John A Cairns	John	John	John A Cairns
Wilmot Smith	- (recognized as organization)	Wilmot Smith	Wilmot Smith
Samuel Goldfarb	bamucl Ooldfarb	Samuel Ooldfarb	Samuel Goldfarb
Lieut Southerland	Lieut Bouther-land	Sutherland	Lieut Southerland
Mr Justice Brewer Potter	Potter	Potter	Mr Justice Brewer Potter
Samuel Rich	Bamuel Rich	Samuel Rich	Samuel Rich
Frederick Betts	Frederick lletts	lletts −> lets	Frederick Betts

Table 7.1: Table showing some of the person names recognized while calculation of PNDR before spell correction (OCR text) and after spell correction (Spell Corrected text) using both Simple and Enhanced Person Names Dictionary. Correctly recognized names are shown in the colored cells.

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 7.2: 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 7.3: Table representing top 30 influential person entities detected from people gazetteer with 100 Topics LDA Model along with evaluation results and comments.

Table 7.4: Topics ID and words obtained from the 100 Topics LDA model.

Topic ID	Topic Words
0	chinese japanese china japan war government american port united foreign states minister treaty arthur british country despatch admiral army
1	theatre box taught sat tonight mat open academy north marine grand extra matinee bank terms today manager time proprietor
2	boat ship water crew vessel steamer island vessels capt port sea deck captain ships boats nov board american schooner
3	mr committee mayor city meeting york strong col hall dr members night seventy held member yesterday tammany club office
4	day place long great water time feet found good men town make people work days big miles side la
5	round jack match fight pounds champion night rounds contest weight left club bout jackson light pound referee ring hilly
6	white women color black dress picture made blue front woman silk skirt pretty green velvet pink beautiful satin fashion
7	sun sunday december november york saturday monday tuesday wednesday friday daily thursday tile association printing cents publishing ann country
8	time law make made letter public matter state question action ha con opinion bo tin ho put order fact
9	bank money business amount hank estate company account made national check firm paid insurance yesterday property stock real banks
10	dr school college university prof schools students education harvard student medical armenian turkish president teacher sultan study professor columbia
11	emperor prince french alexander czar london nov government imperial russian german lord france foreign today court russia minister von
12	tif fff strict iliff hip mrt hackett fat acting usmc agree minimum trtd incur client fairly found hf minuit
13	court judge justice case jury attorney trial district supreme lawyer charge counsel yesterday law mr cases prison charges defendant
14	kt ii mate chess game ik kit ki kk kill york play match ktr move played pk won white
15	hut girl woman heart girls love eyes wife moment lady child mother miss clarence life voice hand word husband
16	liquor saloon goods law dry beer license wine salons drink stores excise keepers men grocer dealers sell society bottle
17	yacht cup club challenge race york lord committee deed americas squadron america royal racing unravel letter english boat secretary

Continuation of Table 7.4	
Topic ID	Topic Words
18	time great made number years fact ot part large year la form work ago bo con point interest present
19	train car railroad road company city bridge street line track station cable trolley avenue engineer passenger cars elevated men
20	man ho good time told thing day dont hut young make lie bad back asked ha put men give
21	cent week year pf total market ft net stock today central month sales short january exchange national receipts earnings
22	birds bird bear deer wild gun shot killed hunter woods county shooting game hunting yards hunt hawk thee hunters
23	murphy vi marie heat vv de ih martin adolphe oconnell wheeler mcdonald franklin day fuller june casey minute blanche
24	st west broadway christmas furniture york prices holiday carpets av cor goods world presents stores tub ave free machine
25	clark damage clarke lee cab lark darby nitrate town damp pigs clarks holland steed stanley trifling isoa marks beadles
26	price black silk ladies prices goods fine quality tea worth wool full regular special fancy extra yard men long
27	osborn allen failing harder jenkins erases lids oppenheim swain waterbury alms hester klos curtly acid ecb lind younker mia
28	clay city bv head kentucky goodwin kitty driver lay br wilkes halo chief welter prices maud class triple jesse
29	president united today nov cleveland secretary dec washington states press chicago southern morning state service news received house collector
30	piano upright av monthly wanted st private brooklyn price open york broadway bargain free good rent school factory prices
31	bill tax income gold states treasury united mr law government congress national currency tariff house notes bank senate direct
32	canal minnie mackinnon talboys canals mcginley obrien freda condo ollrlen imre elbow ida toxic mingle pt tilde vos champlin
33	company york bonds trust cent mortgage interest city bond stock wall committee railroad st coupon central street bondholders agreement
34	trade country american cotton cent business price state cents market coal states sugar labor population united prices free supply
35	hotel dinner hotels cook restaurant waiter allison money kitchen proprietor guests majestic guest table wine luncheon soup dinners lodging
36	church father catholic archbishop priest st holy bishop altar priests parish dr mass corrigan rector cathedral present rev service

Continuation of Table 7.4	
Topic ID	Topic Words
37	army war men navy military company regiment officers service admiral col naval officer department gen command guard battle capt
38	room rooms st furnished board rent av house west front private bath large city heated ht east small light
39	tn thu thin lie time ho nn anti tin nail hut und bo ns hu tu nut tint tim
40	ii ll la li ti uu al tl ui il tt air rt iv ra iii ft rl tut
41	election party republican vote state democratic district senator elected majority candidate democrats republicans county ticket democrat congress york political
42	iii ii ill lit ll st si vi il iv ti im till ft mi li lull oil ml
43	iii ii lit ill lie tin tu nail ll lu hit mi lo ti fur hut ut im nl
44	lu lui uu ur ilium lulu city fruit au izu ilii ul fur fluid isle loud lou lupus tu
45	sir john canadian thompson canada macdonald montreal whist dominion toronto quebec partner ottawa charles premier halifax lead ontario donald
46	people american man great country men world life good english england war public power history political women spirit wo
47	game team football play half line ball back yale eleven left played harvard end college field men tackle princeton
48	feet line street north point minutes feel boundary easterly western avenue seconds northerly distance degrees southerly west degree fret
49	north fair south weather york western west wind texas states southern carolina rain city average virginia jersey northern yesterday
50	book author story books work cloth published volume illustrated library paper american life history york edition written read art
51	jan january dec annual union payable transfer declared savings bo dee tile ou thereto draw ending point monday semiannual
52	lu la tu tb lo fur wa bad ut ha ou uu les tea late au lb ot ibe
53	daily car york chicago sunday st train pullman parlor point dining west city foot week night company buffalo anti
54	martin taylor wolf lena wallet pp consul andrew hollman jacobus brayman softly schlosser martins towel pile llrauman forgot trauma
55	inches white coral jump ice state black pitch skates vault bar poly heel blade tentacles foot tooth inch stating
56	st york city corner day ave ac court ward park ht rt january water inth terms con tub fewer
57	opera theatre play stage music week performance company night audience evening concert season mme house manager york orchestra musical
58	daily steamship directed letter walls close al fur letters australia direct japan china steamer la supplementary mall ship hawaii

Continuation of Table 7.4	
Topic ID	Topic Words
59	funeral dec clock late year residence st church kelly nov services interest york saturday friday attend cemetery west thursday
60	york nov dec orleans london liverpool city oct antwerp si hamburg sew havana passed dee bs hull st charleston
61	dr health cure remedy physician goods blood cold medical disease liver medicine sick weak syrupy cured nervous pure cures
62	street avenue yesterday police policeman station arrested night west man east court john morning found held justice detective clock
63	won time race ran mile furlough half lo track fourth jockey curling today favorite selling handicap lot run tim
64	time limo lime limit amid tim semi whim miami lmm mitt jv sum html tiers coon strait ime anal
65	building feet schaefer built build run work brick bulliard buildings wall cost ives play shot walls wizard structure corner
66	men killed death body man shot found murder dead dec ho night blood head revolver kill negro wound bullet
67	fish baker seely linked lake trout angel bank leather water fishing frederick frog pond creek belly shoe licked crane
68	jerry ave disbursements bem natel watts mock illegal neuron checked noo elliot reserve britain loop hugo stallion llullltt ila
69	money found office stolen burglar store worth post burglars diamond thieves thief robbery robbed nov open watch robbers robber
70	gun inch guns armor powder feet steel plate topped pounds navy battle shell ships works mortar charge explosion carriage
71	club meeting held association york league clubs men members brooklyn record national annual ii athletic season amateur bicycle race
72	arbor ann toledo holder north lion comptrollers tick depositary wall fractions entire entitle hundred brooklyn expense rr title appearing
73	mrs mr miss john william henry charles daughter george evening james sir dr society van street large present jr
74	av wife st lo john ar yrs tn si mary al prop wm west henry ano lot win jr
75	henry scott miller wright ogden murray finn jones june donaldson walker illicit klan whisper king jasper david bulletin bentley
76	mr police witness committee capt asked captain money inspector paid told called testimony commissioner superintendent charges martin sir force
77	nov hats election prepare oct fain turner casino flank wyman hopper coming carrie hutton columbus contact flay jansen pastor

Continuation of Table 7.4	
Topic ID	Topic Words
78	horse horses class years prize show ring trotting trotter year mare shown york turf owner hands hone record br
79	john county city william james thomas smith charles brooklyn ward johnson george hoard clerk district york yesterday jersey alderman
80	park landscape architect driveway city harlem van work branch central kearney hoard clausen kinds attention carriage dark carriages hell
81	money work bet field morton lady fred trial point strong run ran dwyer good owned domino made kennel dixie
82	time night room wa house clock door morning men back left man front began ran turned started ing floor
83	mr sir smith hale hell week air gentleman russell jar ingram wrote letter aid wilson daniel webster browne friends
84	dam lally attila wick older uer rival denton lofty hold weimar kramer alae hampton jintur iullnian wallklll judith tetanic
85	water feet air inches iron power gas steel machine surface light weight pounds wood tube paper eye inch metal
86	year report present city made time work york department public make number system hoard plan years commission bo money
87	tramp card cards hand poker pot cowboy dealer deal labia pony opened takes flush count deals busy draw xml
88	indian indians territory dog desert sheep white tribes cattle life mexico animals snake tribe dogs animal patagones live panthers
89	part term clear day adjourned calendar court march cases motion november iv called die count trial earl parts case
90	special cigar woolsey iota earls clint cecil lax uku trees linear weal otto status mihir ossium recite rau fiona
91	ran ill tn lo ant ton ll ten al end tl fur nail tie met part nl work er
92	mrs mr years wife home house ago woman city died children ho husband mother time father family left year
93	xx thu guns filth xxlll kvas xvi vill xxi karp izu knox xxhlch xi xvhlch clanton iite tax lax
94	la ot ta aa te day tea wa aad tb bat br ia au ar sad tt today ran
95	total ii club score night alleys tournament rolled scores game empire iso columbia national brooklyn team jersey games alley
96	olcott mh iiiiii madman kkhti lilly sots lah warm kai murri billet ieiri mutual jlyn sis orin concu mell
97	work women men labor strike union strikers trade wages employees day denver factory cloak hall shop sums unions workers

Continuation of Table 7.4	
Topic ID	Topic Words
98	walters mount tarbell addicus orbit chips pri cartwright dillon coxe walden trudell kov lssp pencil chip villain adduct allude
99	church dr pastor rev sunday churches minister congregation methods choir baptist presbyterians pulpit episcopal sermon meeting christian services service