

CAPITAL UNIVERSITY OF SCIENCE AND
TECHNOLOGY, ISLAMABAD



Extracting Named Entities and Relations From Text for Populating Knowledge Graph

by

Raabia Mumtaz

A dissertation submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the

Faculty of Computing

Department of Computer Science

2023

Extracting Named Entities and Relations From Text for Populating Knowledge Graph

By

Raabia Mumtaz

(DCS151011)

Dr. Rehan Akbar, Associate Professor
Universiti Teknologi Petronas, Malaysia
(Foreign Evaluator 1)

Dr. Donghong Ji, Professor
Wuhan University, Wuhan, Hubei, China
(Foreign Evaluator 2)

Dr. Muhammad Abdul Qadir
(Research Supervisor)

Dr. Abdul Basit Siddiqui
(Head, Department of Computer Science)

Dr. Muhammad Abdul Qadir
(Dean, Faculty of Computing)

DEPARTMENT OF COMPUTER SCIENCE
CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY
ISLAMABAD

2023

Copyright © 2023 by Raabia Mumtaz

All rights reserved. No part of this dissertation may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, by any information storage and retrieval system without the prior written permission of the author.

Dedicated to my parents.



**CAPITAL UNIVERSITY OF SCIENCE & TECHNOLOGY
ISLAMABAD**

Expressway, Kahuta Road, Zone-V, Islamabad
Phone: +92-51-111-555-666 Fax: +92-51-4486705
Email: info@cust.edu.pk Website: <https://www.cust.edu.pk>

CERTIFICATE OF APPROVAL

This is to certify that the research work presented in the thesis, entitled “**Extracting Named Entities and Relations from Text for Populating Knowledge Graph**” was conducted under the supervision of **Dr. Muhammad Abdul Qadir**. No part of this thesis has been submitted anywhere else for any other degree. This thesis is submitted to the **Department of Computer Science, Capital University of Science and Technology** in partial fulfillment of the requirements for the degree of Doctor in Philosophy in the field of **Computer Science**. The open defence of the thesis was conducted on **October 04, 2023**.

Student Name :

Raabia Mumtaz (DCS151011)

Raabia

The Examination Committee unanimously agrees to award PhD degree in the mentioned field.

Examination Committee :

(a) External Examiner 1: Dr. Zahid Halim
Professor
GIKI, Topi, Swabi

Zahid Halim

(b) External Examiner 2: Dr. Seemab Latif
Associate Professor
SEECS, NUST, Islamabad

Seemab Latif

(c) Internal Examiner : Dr. Aamer Nadeem
Professor
CUST, Islamabad

Aamer Nadeem

Supervisor Name :

Dr. Muhammad Abdul Qadir
Professor
CUST, Islamabad

Muhammad Abdul Qadir

Name of HoD :

Dr. Abdul Basit Siddiqui
Associate Professor
CUST, Islamabad

Abdul Basit Siddiqui

Name of Dean :

Dr. Muhammad Abdul Qadir
Professor
CUST, Islamabad

Muhammad Abdul Qadir

AUTHOR'S DECLARATION

I, **Raabia Mumtaz (Registration No. DCS151011)**, hereby state that my PhD thesis titled, '**Extracting Named Entities and Relations from Text for Populating Knowledge Graph**' is my own work and has not been submitted previously by me for taking any degree from Capital University of Science and Technology, Islamabad or anywhere else in the country/ world.

At any time, if my statement is found to be incorrect even after my graduation, the University has the right to withdraw my PhD Degree.



(Raabia Mumtaz)

Dated:

4th October, 2023

Registration No: DCS151011

PLAGIARISM UNDERTAKING

I solemnly declare that research work presented in the thesis titled “**Extracting Named Entities and Relations from Text for Populating Knowledge Graph**” is solely my research work with no significant contribution from any other person. Small contribution/ help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and Capital University of Science and Technology towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/ cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of PhD Degree, the University reserves the right to withdraw/ revoke my PhD degree and that HEC and the University have the right to publish my name on the HEC/ University Website on which names of students are placed who submitted plagiarized thesis.

Dated:

4th October, 2023

Rabia
(Raabia Mumtaz)

Registration No: DCS151011

List of Publications

It is certified that following publication(s) have been made out of the research work that has been carried out for this dissertation:-

1. **Raabia Mumtaz**, and Muhammad Abdul Qadir. “CustRE: a rule based system for family relations extraction from english text,” *Knowledge and Information Systems*, vol. 64, pp. 1817–1844 (2022), doi: 10.1007/s10115-022-01687-4, url: <https://link.springer.com/article/10.1007/s10115-022-01687-4>
2. **Raabia Mumtaz**, and Muhammad Abdul Qadir. “CustNER: A Rule-Based Named-Entity Recognizer With Improved Recall,” *International Journal on Semantic Web and Information Systems (IJSWIS)* vol. 16, no. 3, pp. 110-127, 2020, doi: 10.4018/IJSWIS.2020070107, url: <https://www.igi-global.com/article/custner/256549>
3. **Raabia Mumtaz**, Muhammad Abdul Qadir and Asif Saeed. “CustFRE: An annotated dataset for extraction of family relations from English text,” *Data in brief* vol. 41, pp. 107980, 2020, doi: <https://doi.org/10.1016/j.dib.2022.107980>, url: <https://www.sciencedirect.com/science/article/pii/S352340922001913>

(Raabia Mumtaz)

Registration No: DCS151011

Acknowledgement

First of all I thank Almighty Allah, the most Gracious, the All-Wise, the Illuminator, for guiding me to this day. I want to thank all those who helped and supported me in reaching this goal. Dr. M. Abdul Qadir, thank you for shaping my research work, for showing me direction when I got lost during this journey, for giving me hope in those times when I was feeling down. I also thank other faculty members at CUST who provided valuable input to my research work and guided me, specially Dr. Azhar Mahmood, and those who said kind words to me: Dr. Nayyer Masood, Dr. Abdul Basit, Dr. M. Tanvir Afzal, and Dr. Qamar Mehmood.

I am extremely grateful to my family; my mother for looking after my home and kids and letting me concentrate on my work; my father for managing without my mother when she was with me; my husband for providing me pick and drops and all administrative and emotional support; my kids for bearing with me when I could not give you enough time; my siblings for understanding my unavailability; my extended family and friends, specially Salma Begum, for tolerating my long absence in gatherings and events; it would not have been possible without support from all of you.

I want to thank my students, lab-mattes, friends and ex-colleagues at CUST who assisted me with my work; Ali Jaffer and Qamar for helping me with the very first manual working/analysis of dataset; Qaisar Manzoor for doing the dataset annotations on very short notice, for providing snacks in lab when I did not have time to visit cafe; Muhammad Saqib for bringing me tea in lab when I was having cold; Farhana for providing company for cafe visits; Samreen Ayaz, for your help with last minute dataset annotations, for providing good company for tea/lunch, for lighting up the routine dull time at university; Muhammad Asif, for your useful suggestions; Maimoona Qudsia for easing my faculty-to-student transition at CUST and offering me tea in your office.

Many thanks to my boss at work, Madam Zubaida Iqbal, for accepting my frequent leave requests and letting me continue my research work. I also thank my other colleagues who provided me support; Zoya Saman, Rabia Basri, Shumaila, Sumaira, Anam; you were always willing to manage my duties when I was unavailable.

Special thanks to Dr. Shabana Amanullah, for her understanding and support at my new workplace.

Lastly, I want to thank all those who provided administrative support in university; the library staff, for quick responses to my frequent requests of research literature; Farooq sb., for arranging research meetings with supervisor, Khalid sb. for providing guidance with administrative issues; your behind the scenes help did not go unobserved.

Each one of you has a part in the completion of this dissertation. Thank you, all of you.

(Raabia Mumtaz)

Abstract

In this era of information explosion, quickly and accurately obtaining meaningful information from the massive data available on the web, has become an urgent problem to be solved. Mostly text data on the web is not structured, and can not be meaningfully queried. Low precision and high recall is a known problem for the web data. Converting this unstructured text to a structured Knowledge Graph (KG) would enable making meaningful and precise queries on the data. Constructing KG involves two main tasks; first is named entity recognition, NER, i.e. recognizing and typing named entities, NEs, contained in text (these form nodes of the KG), and second is relation extraction, RE, i.e. identifying and classifying semantic relationships between entities appearing in the text (these form edges of the KG). The main hindrance in converting textual data on the web to KG is the in-adequate performance of automated information extraction (IE) systems for NER and RE. This dissertation improves state-of-the-art for these two fundamental problems for KG population, that is NER and RE.

A deep understanding of the problem has been developed by reviewing existing relevant literature. Existing systems for NER and RE have been identified, evaluation datasets have been selected and prepared. An information extraction system has been proposed for improved extraction of named entities and family relations from text. The system has been designed and implemented using Python. Evaluation results show that on NER task, the proposed system outperforms existing systems on OKE dataset by making an F1 score of 81.03% (which is 2.53 points better than previous best), and gives results comparable to existing systems on CoNLL03 dataset (for which the system is not trained) by making an F1 score 94.67%. Proposed system's performance for gender classification is also satisfactory (F1 score 89.48%). On family relation extraction (FRE) task, the proposed approach makes a great improvement over existing methods by achieving an F1 score of 70.4% on TACRED family relations dataset, which is 7.3 points higher than the best score reporter on TACRED. Further, on another dataset too, the

CustFRE dataset, the system performs better than all existing systems, with F1 score 76.6%, which is 18.5 points higher than the previous best performing system.

Clarification of learning has been achieved, by identifying factors which have positively contributed to the research success, as well as, by identifying cases where the proposed system fails. Conclusions have been drawn. It has been demonstrated that the proposed system is not dataset specific; it recognizes person, location and organization NEs for general English text and extracts family relations from general English sentences. The methods adopted, the system proposed and the evaluation results obtained, have been communicated for comparison and further improvement via journal publications and this dissertation.

The proposed system is available at the following link:

<https://github.com/Raabia-Asif/CustKnowledgeExtractor>

Contents

Author’s Declaration	v
Plagiarism Undertaking	vi
List of Publications	vii
Acknowledgement	viii
Abstract	x
List of Figures	xv
List of Tables	xvi
Abbreviations	xviii
Symbols	xx
1 Introduction	1
1.1 Background	1
1.2 Knowledge Graph	6
1.3 Named Entity Recognition	7
1.4 Relation Extraction	11
1.5 Problem Statement	16
1.6 Research Questions	17
1.7 Research Objectives	17
1.8 Research Scope	17
1.9 Research Methodology	18
1.10 Dissertation Contributions	20
1.10.1 NER Contributions	20
1.10.2 RE Contributions	20
1.11 Dissertation Organization	21
2 Literature Review	22
2.1 Named Entity Recognition	24

2.1.1	Rule Based Systems	24
2.1.2	Machine Learning Based Systems	26
2.1.3	Hybrid Systems	33
2.2	Relation Extraction	37
2.2.1	Rule based methods	37
2.2.2	Learning based methods	39
2.2.3	Hybrid methods	43
2.3	Findings of Literature Review	46
2.3.1	Selection of Baselines	48
2.3.2	Gap Analysis	49
3	Datasets Selection and Preparation	51
3.1	Introduction	51
3.2	Datasets for NER	52
3.2.1	The CoNLL Dataset	53
3.2.2	The OKE Dataset	54
3.3	Datasets for Relation Extraction	56
3.3.1	The TACRED-F Dataset	56
3.3.2	The CustFRE Dataset	58
3.4	Need for an Explicit Method to Assess Datasets	61
3.5	Criteria to Assess NER and FRE Evaluation Datasets	62
3.5.1	Accuracy	64
3.5.2	Completeness	65
3.5.3	Appropriate Size	67
3.6	Assessment of Datasets	69
3.6.1	Accuracy	70
3.6.2	Completeness	71
3.6.3	Size	72
3.7	Improving the Datasets	72
3.7.1	The Improved OKE Dataset	72
3.7.2	The Improved TACRED-F Dataset	75
3.7.3	Evaluating the Improved Datasets	76
3.8	Summary	77
4	CustNER - An Improved System for Named Entity Recognition	78
4.1	Introduction	78
4.2	The System CustNER	79
4.2.1	Pre-Processor	80
4.2.2	Rule Engine	80
4.2.3	Identifying Gender of Person NEs	88
4.2.4	Querying DBpedia	88
4.3	Experimental Setup	89
4.4	Evaluation	92

4.4.1	Evaluation Measure	92
4.4.2	Results	94
4.4.3	Analysis of Errors	98
4.4.3.1	Errors from NERs Used	98
4.4.3.2	Errors from Part of Speech Tags	98
4.4.3.3	Incorrect Type from DBpedia	99
4.4.3.4	DBpedia Disambiguation Pages	99
4.4.3.5	Ambiguous Cases	100
4.5	Summary	101
5	CustRE - An Improved System for Relation Extraction	103
5.1	Introduction	103
5.2	The System CustRE	104
5.3	System Implementation	108
5.3.1	Relation Words Lists L	108
5.3.2	Pattern Extractor	110
5.3.3	Regex Base	111
5.3.4	Explicit Triples Generator	111
5.3.5	Implicit Triples Generator	114
5.4	Evaluation Results and Analysis	114
5.5	Analysis of Errors	124
5.5.1	Coref Errors	124
5.5.2	Overlapped Triple Errors	125
5.5.3	Other Errors	125
5.6	Summary	126
6	Conclusion and Future Work	127
	Bibliography	131
	Appendix A	
	The Relation Words Lists	147
	Appendix B	
	Search Strategy for Literature Review	149

List of Figures

1.1	Example text from wikipedia webpage of Javed Iqbal	2
1.2	Source of the example text available on the web	2
1.3	Google results for “place of birth of children of Allama Iqbal”	3
1.4	Domain specific tagging of the example text	4
1.5	Knowledge Graph of the example text	5
1.6	The Design Science Research Methodology Process for this work	19
3.1	An entity annotation entry from OKE data set	55
3.2	A sample example from TACRED dataset	58
3.3	The focus group introduction script	64
3.4	The questions for stimulating focus group discussion	65
3.5	Definition of Valid Named Entity Annotation	66
3.6	Definition of Valid Family Relation Annotation	67
4.1	The System CustNER	79
4.2	Example of DBpedia disambiguation page	100
5.1	Architecture of the proposed system, CustRE	109
5.2	An example run of CustRE for the input “Kollek is survived by his widow Tamar, son Amos and daughter Osnat.”	115
5.3	F1 Scores Comparison of FRE task on TACRED-F test set	118
5.4	F1 Comparison with respect to number of persons in sentence	121
5.5	KG for text “Kollek is survived by his widow Tamar, son Amos and daughter Osnat.”	124
6.1	Google result for query <i>007</i>	130

List of Tables

1.1	Example NEs not annotated by existing NER systems	10
1.2	F1 scores comparison of systems on RE and FRE tasks	13
1.3	Output of RE systems for the example sentence	15
2.1	Year wise distribution of papers reviewed	23
2.2	Summary of Rule-based NER systems	27
2.3	Summary of ML NER systems	31
2.4	Summary of hybrid NER systems	36
2.5	Summary of Rule-based FRE systems	40
2.6	Summary of learning based FRE systems	44
2.7	Summary of hybrid FRE systems	47
3.1	Some statistics of the CoNLL dataset	53
3.2	An example sentence from CoNLL03 dataset	54
3.3	Some statistics of the OKE dataset	55
3.4	Annotations in OKE dataset for example sentence	56
3.5	Relation distribution of TACRED-F dataset	59
3.6	Relation distribution of CustFRE evaluation dataset	61
3.7	Expert Profiles for Focus Group	63
3.8	Summary of evaluation datasets assessment against devised criteria	70
3.9	Examples of corrections made to the OKE dataset	74
3.10	Examples of Corrections made to TACRED-F Dataset	76
4.1	List of notations used in Algorithm 1	83
4.2	Examples of Rules	86
4.3	Tools used by the system module CustNER	91
4.4	Datasets used for evaluation of system module CustNER	91
4.5	NERC Results Comparison on OKE test set	95
4.6	NERC Results Comparison on CoNLL03 test set	96
4.7	Comparison of empirical run time for NER task	98
4.8	Examples of incorrect PoS tags	99
5.1	Mapping of Wikidata Properties to family relations	110
5.2	Features extracted from t and the symbols used to represent them	110
5.3	List of regular expressions	112
5.4	Extraction Rules for matched regular expressions	112
5.5	Examples of triples extraction	113

5.6	Results Comparison for FRE task on TACRED-F test set	116
5.7	Results Comparison for FRE task on CustFRE evaluation dataset	119
5.8	Output analysis of three top performing systems	121
5.9	Annotations by three top performing systems for input sentence “ She had a daughter, Maureen in 1941 and adopted a son, Michael in 1945.”	122
5.10	Comparison of empirical run time for FRE task	123

Abbreviations

ANN	Artificial Neural Network
BERT	Bidirectional Encoder Representations from Transformers
BiLSTM	Bidirectional Long Short-Term Memory
BiTransformer	Bidirectional Transformer
CLL	Conditional Log Likelihood
CNN	Conditional Random Field
CRF	Conditional Random Fields
CUST	Capital University of Science & Technology
CustFRE	CUST’s Family Relation Extraction Dataset
CustNER	CUST’s Named Entity Recognition System
CustRE	CUST’s Relation Extraction System
CWR	Contextualized Word Representation
DL	Deep Learning
EL	Entity Linking
FRE	Family Relation Extraction
GCN	Graph Convolution Network
HTML	HyperText Markup Language
HMM	Hidden Markov Models
IE	Information Extraction
KB	Knowledge Base
KG	Knowledge Graph
k NN	k Nearest Neighbour
LOC	Location class

LSTM	Long Short-Term Memory
LSVM	Linear Support Vector Machine
MaxEnt	Maximum Entropy Classifier
ML	Machine Learning
MLP	Multilayer Perceptron
NE	Named Entity
NED	Named Entity Disambiguation
NEN	Named Entity Normalization
NER	Named Entity Recognition
NERD	Named Entity Recognition and Disambiguation
NERL	Named Entity Recognition and Linking
NLP	Natural Language Processing
NN	Neural Network
nonPLO	NE Types which cannot be PER, LOC or ORG
NYT	New York Times
ORG	Organization class
PER	Person class
PLO	PER, LOC, ORG
PoS	Part of Speech
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
RE	Relation Extraction
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
SOTA	State of the Art
SVM	Support Vector Machine
TBL	Transformation Based Learning
URI	Universal Resource Identifier
URL	Uniform Resource Locator

Symbols

A_D	The accuracy of dataset D
C_D	The completeness percentage of dataset D , that is, from the total number of possible annotations on D , what percentage of annotations is actually found in D .
e	The desired level of precision or the margin of error
n_o	The necessary sample size
$N_{correct}$	The number of correct annotations in the dataset D
N_{total}	The total or actual number of annotations in the dataset D
$N_{possible}$	The number of possible annotations on D
p	The (estimated) proportion of the population which has the attribute in question
S_{FRE_D}	The size of an FRE dataset D
S_{NER_D}	The size of an NER dataset D
Z	The z - value found in a Z table

Chapter 1

Introduction

1.1 Background

We are drowning in information
but starved for knowledge

John Naisbitt

Data on the web is usually in unstructured form, rendered through HTML (HyperText Markup Language). HTML has tags for headings, links, paragraphs, etc. which tell web browsers how to display the content, so that it is readable for humans. But it is difficult for machines to interpret this data, because the tags are not domain specific. In order to search web data, user provides keywords to search engine, which in turn applies text matching techniques to retrieve a list of relevant documents from the web. These search results are highly sensitive to vocabulary. Many times the user has to repeat the search by trying different semantically similar keywords in order to get the desired results. So user basically has to guess the terminology used in the document which he/she is trying to search and provide that terminology to search engine. Meaningful queries can not be performed on the web, resulting in high recall and low precision. A lot of results are returned, most of which are not very relevant, and the user has to manually search from

them the most relevant result. High recall, low precision is a reported problem for the web. There is a need to structure the web data in a form that is more machine-processable, so that meaningful queries can be performed on it and precise results can be obtained. This can be done by structuring data in conceptual spaces according to its meaning.

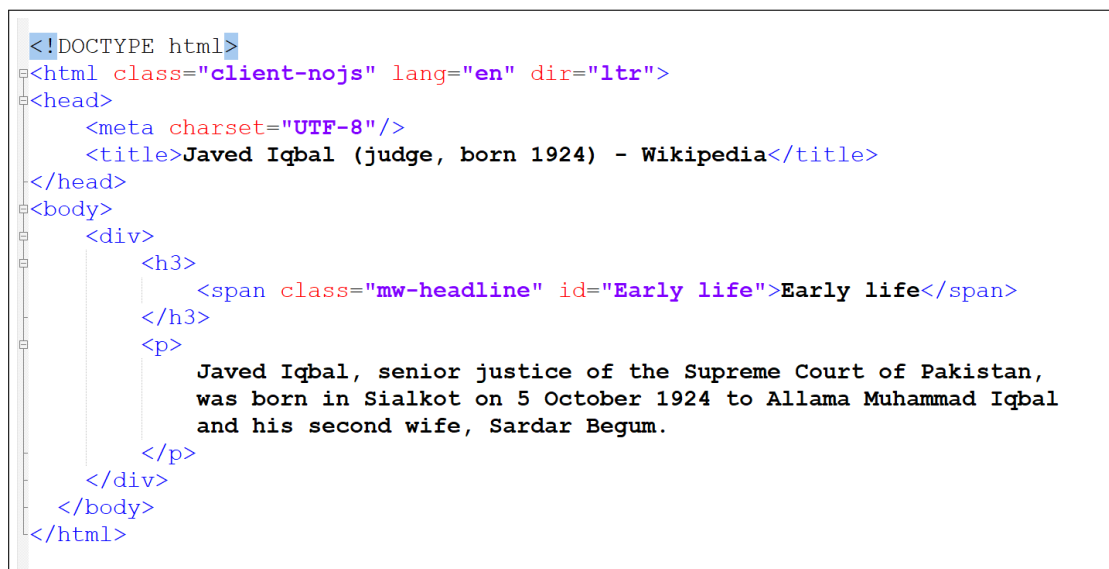
Consider some example text from the web, given in Fig 1.1, taken from [https://en.wikipedia.org/wiki/Javed_Iqbal_\(judge,_born_1924\)](https://en.wikipedia.org/wiki/Javed_Iqbal_(judge,_born_1924)).

Early life

Javed Iqbal, senior justice of the Supreme Court of Pakistan, was born in Sialkot on 5 October 1924 to Allama Muhammad Iqbal and his second wife, Sardar Begum.

FIGURE 1.1: Example text from wikipedia webpage of Javed Iqbal

The source of this example text available on the web, is given in Fig 1.2. The



```
<!DOCTYPE html>
<html class="client-nojs" lang="en" dir="ltr">
<head>
  <meta charset="UTF-8"/>
  <title>Javed Iqbal (judge, born 1924) - Wikipedia</title>
</head>
<body>
  <div>
    <h3>
      <span class="mw-headline" id="Early life">Early life</span>
    </h3>
    <p>
      Javed Iqbal, senior justice of the Supreme Court of Pakistan,
      was born in Sialkot on 5 October 1924 to Allama Muhammad Iqbal
      and his second wife, Sardar Begum.
    </p>
  </div>
</body>
</html>
```

FIGURE 1.2: Source of the example text available on the web

HTML tags in the figure, like `< h3 >`, `< p >` are about how the content should be displayed, and do not provide any information about the domain concepts. Now let us say a person wants to know about “place of birth of children of Allama Iqbal”, so he gives these keywords to google for search. The first page of the search results returned¹ is given in Fig 1.3. Around 2.7 million results are returned by

¹Accessed on 22nd June 2022

place of birth of children of allama iqbal

About 2,710,000 results (0.64 seconds)

https://en.wikipedia.org/wiki/Muhammad_Iqbal

Muhammad Iqbal - Wikipedia

Iqbal was born on 9 November 1877 in an ethnic - Iqbal's father, Sheikh Noor Muhammad (died 1930), was a tailor, not formally educated, but a religious man.

[Personal life](#) · [Efforts and influences](#) · [Literary work](#) · [Modern reputation](#)

<https://www.aa.com.tr/asia-pacific/pakistani-literary-...>

Pakistani literary giant Allama Iqbal's family recalls his legacy

09-Nov-2021 — Epitomizing grace and poise, Munira Salahuddin (91), the last living **child** of Pakistan's national poet, **Allama** Sir Muhammad Iqbal has vivid ...

<https://lahorecafe.pk/positive-pakistan/biography-all...>

Biography Of Allama Iqbal For Kids In Pakistan - Lahore

Here we have **biography** of **Allama Iqbal** for kids. **Allama Iqbal** is our national poet. He was born on 9th Nov 1877 and he died on 21st April 1938.

<https://www.thefamouspeople.com/profiles/muham...>

Sir Muhammad Iqbal Biography - TheFamousPeople

Iqbal married three times in his life: his first marriage (1895) was with Karim Bibi and he had two children with her - Miraj Begum and Aftab Iqbal. His second ...

https://familypedia.fandom.com/wiki/Muhammad_I...

Muhammad Iqbal (1877-1938) - Familypedia - Fandom

Children ; Offspring of Muhammad Iqbal and Mukhtar Begum ; Name, **Birth**, Death ; Javed Iqbal ; Munazza (from Sardar Begum) ...

<https://tribune.com.pk/story/allama-iqbals-family-re...>

Allama Iqbal's family recalls his legacy - The Express Tribune

09-Nov-2021 — LAHORE: Epitomising grace and poise, Munira Salahuddin (91), the last living **child** of Pakistan's national poet, **Allama** Sir Muhammad Iqbal ...

<https://sekho.com.pk/college-essays/allama-iqbal-bl...>

Allama Iqbal Biography For Kids - Sekho.com.pk

06-Nov-2013 — **Allama Iqbal Biography for Kids** ... Allama Mohammad Iqbal the great poet and scholar of Muslims were born on 9th November 1877 at Sialkot a city ...

<https://www.pakpedia.pk/Personality>

Allama Iqbal - Pakpedia

Allama Iqbal Biography. The family of Iqbal was Kashmiri Pandit who accepted Islam as their religion in the fifteenth century. His father sent Allama ...

<http://www.allamaiqbal.com/review/oct64/2.htm>

DATE OF IQBAL'S BIRTH - Allama Iqbal

1. The **birth** certificate mentions that it relates to a male **child** of Shaikh Nathoo (which is the pet-name of Shaikh Noor Muhammad, father of Iqbal). · 2. The ...

<https://www.hilal.gov.pk/eng-article/detail/ODkx>

Munira Salahuddin - the Last Living Child of Allama Iqbal

Munira Salahuddin was born on August 30, 1931. She was 7 years old on April 21, 1938 when her father, the great poet **Allama** Dr. Muhammad Iqbal passed away. For ...

FIGURE 1.3: Google results for “place of birth of children of Allama Iqbal”

the search engine (very high recall). First 3 pages (containing 29 results) were checked without opening, none directly gave any information about “place of birth of Allama Iqba’s children” (low precision). Whereas a human can easily tell from the example text in Fig. 1.1 that Allama Iqba’s son Javed Iqbal is born in Sialkot. But since data on the web is not semantically structured, and is mostly presented using HTML, which does not have any domain specific tags, so the search engine searches the query words in HTML web pages, and the pages in which these words appear are returned. Many of the pages returned are about Allama Iqbal’s place of birth, or about Allama Iqbal’s father. Now the user will open the returned links one by one and search himself/herself for the desired information.

If, instead, the data is conceptually organized according to its meaning, by giving domain specific tags, like in Fig 1.4, which tell that the entities Allama Iqbal and Javed Iqbal are persons, the relation between persons Allama Iqbal and Javed Iqbal is child relationship, and that Javed Iqbal is born in Sialkot, it could be easily inferred that Allama Iqbal’s child’s place of birth is Sialkot. This can be

```

<rdf:Description rdf:about="https://dbpedia.org/resource/Muhammad_Iqbal">
  <rdf:type rdf:resource="https://dbpedia.org/ontology/Person"/>
  <dbo:name>Allama Muhammad Iqbal</dbo:name>
  <foaf:gender>male</foaf:gender>
  <dbo:spouse rdf:resource="http://www.example.org/Sardar_Begum"/>
  <dbo:child rdf:resource="https://dbpedia.org/resource/Javed_Iqbal_(judge,_born_1924)"/>
</rdf:Description>

<rdf:Description rdf:about="http://www.example.org/Sardar_Begum">
  <rdf:type rdf:resource="https://dbpedia.org/ontology/Person"/>
  <dbo:name>Sardar Begum</dbo:name>
  <foaf:gender>female</foaf:gender>
</rdf:Description>

<rdf:Description rdf:about="https://dbpedia.org/resource/Javed_Iqbal_(judge,_born_1924)">
  <rdf:type rdf:resource="https://dbpedia.org/ontology/Person"/>
  <dbo:name>Javed Iqbal</dbo:name>
  <foaf:gender>male</foaf:gender>
  <dbo:birthPlace rdf:resource="https://dbpedia.org/resource/Sialkot"/>
  <dbo:employer rdf:resource="https://dbpedia.org/resource/Supreme_Court_of_Pakistan"/>
</rdf:Description>

```

FIGURE 1.4: Domain specific tagging of the example text

achieved by using RDF and RDFS. The domain’s concepts can be specified in RDFS, by defining classes of entities and relations, along with their hierarchies. Data can then be organized in RDF according to the schema defined in RDFS, and can be visualized as a labeled directed graph (known as knowledge graph, KG) in

which nodes represent entities and edges represent relations between entities, or properties of entities. Once data is converted to KG, precise queries can be made on it by using SPARQL. A KG for the example text is given in Fig 1.5, with the relevant path shown in red color. In a KG, every resource has a globally unique

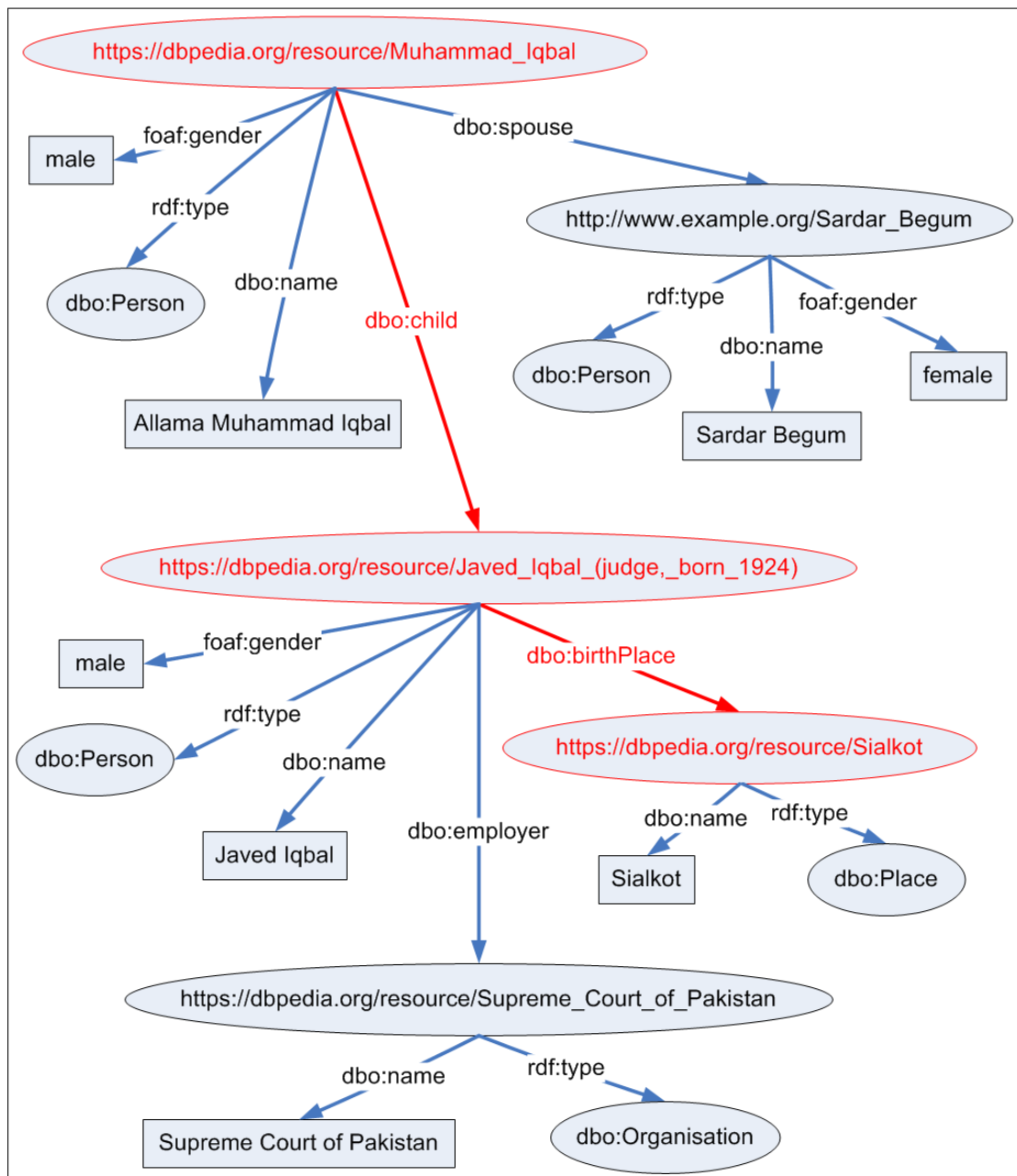


FIGURE 1.5: Knowledge Graph of the example text

URI (universal resource identifier). Let *person1*, having URI *u1*, be a resource in a KG, say *KG1*. When some other KG, say *KG2*, is created about *person1*,

since *person1* has a globally unique identifier *u1*, so *KG1* and *KG2* get linked at *person1* and the KG grows bigger.

1.2 Knowledge Graph

A KG is a network of entities, their semantic types, properties, and relationships between these entities [1]. Presently web data is mostly in un-structured and semi-structured formats. Because of this it is nearly impossible to extract precise information from huge dumps of data available on the web. When queries are made on the web, too many results are returned, most of which are not accurate and one needs to search oneself among these results for the required information. If the web data is instead populated into a KG, this would enable the user to make meaningful queries and in return get precise and accurate results. For instance, when the example text in Fig. 1.1 is structured into a KG in Fig. 1.5, now meaningful queries can be performed on this data, like “Where was Allama Iqbal’s son born?” and precise answers can be obtained, like “Sialkot”.

In order to construct a KG, two main problems need to be solved, named entity recognition (NER) and relation extraction (RE). For instance, given the text “*Salma is supported by her father, Salman*”, identifying the named entities, or equivalently, KG nodes, *Salma* and *Salman*, and extracting the triple, or equivalently, KG edge (*Salma, father, Salman*). Combining such extractions, it is then possible to produce a KG of relational facts between persons, organizations, and locations in the text [2].

Development of NER and RE systems has been encouraged to address the challenges in Information Extraction (IE) domain, by several editions of evaluation events such as the message understanding conference, MUC [3, 4], the conference on natural language learning, CoNLL [5], the automatic content extraction program, ACE [6], the Text Analysis Conference, TAC [7–12], and the open knowledge extraction challenge, OKE [13]. The subsequent editions of these conferences,

MUC, CoNLL, ACE, TAC and OKE have contributed significantly in the domain of entity and relation detection and recognition.

NER and RE are considered important components of information retrieval and knowledge extraction applications, for instance, narrative extraction from news articles [14], text summarization [15], question answering systems [16], document theme extraction [17], rumour detection on social media [18], prediction and recommendation tasks in academic citation networks [19], document indexing [20], text mining in Genetics and Biomedical Sciences [21, 22], automatic spell checking [23], and making intelligent virtual assistant (IVAs) [24], to name a few.

1.3 Named Entity Recognition

The first problem that needs to be solved for KG construction is NER. The term “named entity” (NE) means names of persons, organizations and geographical locations [3]. NER is the process of identifying named instances of pre-defined classes in running text [25]. Consider the following sample text: *Salma lives in Rawalpindi and is studying Computer Science at Capital University of Science and Technology. She is a part time worker at a call center in Islamabad.* If the pre-defined classes are person (PER), location (LOC) and organization (ORG), then the output of NER task on this text is the annotated text as given below:

*[person] **Salma** lives in [location] **Rawalpindi** and is studying Computer Science at [organization] **Capital University of Science and Technology**. She is a part time worker at a call center in [location] **Islamabad**.*

Typically, NER demands optimally combining a variety of clues including, orthographic features, parts of speech, similarity with existing database of entities, presence of specific signature words and so on. This makes NER a non-trivial modelling challenge, not solved yet with an acceptable high precision and recall, despite over three decades of research in the field [26]. To better understand the complexity of NER task, consider another example:

Florence May Harding studied at a school in Sydney, and with Douglas Robert Dundas, but in effect had no formal training in either botany or art.

The output of NER task on this example is:

*[person] **Florence May Harding** studied at a school in [location] **Sydney**, and with [person] **Douglas Robert Dundas**, but in effect had no formal training in either botany or art.*

In this example, *Florence May Harding* is name of a person, but *Florence* is also name of a city in Italy, and *May* is also name of month, so identifying that *Florence May Harding* is actually name of a person in this sentence is a difficult task. Moreover, the problem might seem trivial at first glance by assuming that all proper nouns are named entities, but this is not necessary. There are proper nouns that do not belong to any of the concerned NE classes, for example in the following excerpts from the Open Knowledge Extraction (OKE) dataset, the bold words are marked proper nouns by Stanford Part of Speech (PoS) tagger but are not PER, LOC or ORG typed NEs:

- The **Health** Survey for England
- The 2022 **World Cup**
- The **FIFA Confederations Cup**
- On **Tuesday** afternoon

Likewise there are NEs that are not identified as proper nouns by PoS taggers. In the following excerpts from the OKE dataset, the bold words are PER, LOC or ORG NEs, but are not identified as proper nouns by Stanford PoS tagger.

- The **Italian parliament**
- The **Scottish government**
- **RBI**
- NATO, the **G8**, the **G20**, and the OECD.

Moreover, deciding the boundaries of entities is also very difficult, as many times it is unclear whether a word is part of the NE or not. So solving the NER problem is not a trivial task. Sarawagi [26] observes that correctly recognizing all the entities available in a document is a big challenge because without extensive labeled data it is not possible to even detect what was missed in the large mass of unstructured information.

As English grammar is generally not changing, so most of the times it is possible to identify language patterns for NEs in text. This key observation leads to the following hypothesis:

Hypothesis: “Regular patterns for NEs (PER, LOC, ORG) exist in natural language texts.”

If regular patterns exist, then rules can be created against those patterns to recognize NEs at very low cost in terms of computation time because rules are deterministic.

In order to identify the regular patterns, natural language texts need to be analyzed. Comprehensive sets of natural language texts are available in the form of published datasets. So we studied the OKE training dataset for NER [13], and performed experiment on the dataset with state-of-the-art NER systems. The NE annotations made by four NER systems have been analyzed; the Stanford NER [27, 28], the Illinois NER [29, 30], the Federated knOwledge eXtraction framework, FOX [31] and the adaptive entity recognition and linking framework, ADEL [32, 33]. Many NEs available in the dataset are not annotated by these NERs. Table 1.1 highlights some example NEs which were not annotated by these NERs (i.e. the false negatives). “NA” in the table cell represents “not annotated”. Carefully inspecting such NEs, following regular patterns have been identified in the missed NEs (the false negatives) of existing systems:

1. NEs contain nationalities

TABLE 1.1: Example NEs not annotated by existing NER systems

Named Entity	Annotation by			
	Stanford NER	Illinois NER	FOX	ADEL
Biblis	NA	org:Biblis	NA	NA
Italian parliament	NA	NA	NA	NA
High Court	NA	NA	NA	NA
RTÉ	NA	org:RTÉ	NA	NA
Scottish government	NA	NA	NA	NA
UK government	NA	NA	NA	NA
South Korean police	NA	NA	NA	NA
Fast Company	NA	NA	NA	NA
Nintendo	NA	org:Nintendo	NA	NA
Twitter	NA	NA	NA	NA
RBI	NA	org:RBI	NA	NA

2. NEs have corresponding resources in DBpedia²
3. NEs are acronyms
4. NEs are recurrences of already identified NEs

Consider the false negative NEs from Table 1.1. The entities *Italian parliament*, *Scottish government*, *UK government* and *South Korean police* belong to organizations augmented with nationalities; the entities *High Court*, *Fast Company*, *Nintendo*, *Twitter* and *Biblis* have corresponding resources in DBpedia; while *RTÉ* and *RBI* are acronyms of organizations.

So the hypothesis is found true, that is, regular patterns for NEs (PER, LOC, ORG) do exist in natural language texts. Then there must be rules to detect these missing patterns in order to extract the NEs not annotated by existing systems.

²DBpedia is an online resource that allows querying Wikipedia (an online Encyclopedia) like a semantic database, available at <https://wiki.dbpedia.org/>

1.4 Relation Extraction

The second problem that needs to be solved for KG construction is RE, that is, extraction of relations between NEs. Bach and Badaskar define a relation as a tuple $t = (e_1, e_2, \dots, e_n)$ where the e_i are entities in a predefined relation within a document [34]. Most RE systems focus on extracting binary relations. Examples of binary relations include mother-of (Sara Ali, Sana Ali), located-in (Islamabad, Pakistan), place-of-birth (Yahya Sher, Islamabad), etc. Relations represent various types of connections between entities and are at the core of expressing relational facts in most general knowledge bases (KBs) [35, 36].

Relations are generally extracted from text in two ways: at global level and at mention level. Global level RE concerns the identification of any entity pairs from text for which any semantic relation exist. These RE systems generally take a large text corpus as input and produce a list of such entity pairs as output. Open Information Extraction (Open IE) systems extract relations at global level. They extract any relations between entity pairs in text, not requiring mapping to a pre-specified vocabulary [37, 38]. For instance, from the text “*Alian won against Amna*”, an Open IE extractor may extract the triple (*Alian, won against, Amna*). Many effective Open IE extractors have been proposed to extract triples, including Text-Runner [39], ReVerb [37, 38], R2A2 [38], DepOE [40] and Stanford Open IE [41]. Although, Open IE has recently been an active area of research within the IE domain but a major limitation of these systems is that the same semantic relation may be represented by multiple relation phrases, as such extractors only yield relation patterns between entities, without aggregating and clustering the results according to a schema, leading to redundant relations in KBs [42].

On the other hand, mention level RE systems extract relations according to a pre-defined vocabulary from the input text. The work in this dissertation concerns mention level relation extraction. Consider the entity mentions *Ali* and *Karachi* in the sentence: *Ali, born in Karachi, was brought up by her Aunt in Lahore*. Here, the mention level RE system would identify the place-of-birth relation between Ali and Karachi, if place-of-birth is in the pre-defined vocabulary. Consider another

sentence: *Ali likes Karachi*. Here, mention level RE system should identify that no relation exists between Ali and Karachi in this particular sentence. Mention level RE systems concern identifying instances of predefined relations between two given entities in text, or relation classification. For instance, Automatic context Extraction (ACE 2004) Relation Detection and Characterization (RDC) task specified 23 relations for extraction (e.g. located, near, part-whole).

When the relations to be extracted are family relations (like siblings, parents, etc.), the task can be called family relations extraction (FRE). Formally, the FRE task can be defined as:

Given a text t , a person subject s and a person object o , finding the relation r that relates s to o , out of the six relations: the five family relations (parents, children, siblings, spouse, other_family), and not_known if no family relation can be inferred between the two persons from t .

These relation classes (i.e. parents, children, siblings, spouse, other_family) have been taken from the Text Analysis Conference’s (TAC) Knowledge Base Population (KBP) relation classes, as TAC KBP is the most widely known effort to evaluate knowledge base population systems [2]. In fact, if a dataset has two base family relations about persons; spouse and child, and has persons’ genders specified, then any other family relation between persons can be derived by applying queries on the dataset and whole family tree can be built. But at times these base relations are not mentioned in the text, so other family relations like siblings etc. are needed to extract the base relations.

Extracting family relations from text is useful for many purposes. It is an important step in linking persons across different genealogical documents and sources [43, 44]. Extracting many family relationships from unstructured archive documents can help automatically produce family trees, which aids in discovering social patterns, such as typical household structure, family size, etc. [45]. Extracted family relations can assist fiction readers to better understand its content and plot, and get a bird’s eye view on the landscape of the core story [46], can help literary analysis by providing basic facts for further reasoning on the story [47], because a

key step towards story understanding is to understand the relations between the characters that occur in the story [48, 49].

FRE holds special importance in biomedical domain. Identifying family members in electronic health record texts helps build family history (FH) information which is important to assess the risk of inherited medical conditions and to improve patient care and decision making [50, 51]. Patients' FH is a critical risk factor associated with numerous diseases. However, Claims database and electronic health records (EHR) database do not usually capture kinship or family relationship information, but this information is often documented in clinical narratives. In 2019, the National NLP Clinical Challenge (n2c2) organized shared tasks to solicit NLP methods for FH IE [52]. Family relationship information is also imperative for genetic research. He et al. suggest extracting names and family relations from online obituaries as a new data source and supplementing EHR databases with family relations information for genetic research [53]. Genealogical knowledge graphs (GKGs), or family trees, are imperative for biomedical research such as disease heritability and risk prediction [54].

When four SOTA relation extraction systems, Stanford KBP relation extractor [55], TACRED-PA [2], SpanBert [42] and LUKE [56], were evaluated for family relations, it was found that the systems' F1 scores are very low for extraction of family relations. A comparison of F1 scores of systems on RE and FRE is given in Table 1.2. It can be seen from the table that systems good at general relation extraction do not perform well on extraction of family relations. Therefore, a system with improved family relations extraction is needed.

TABLE 1.2: F1 scores comparison of systems on RE and FRE tasks

	Stanford	TACRED-PA	SpanBert	LUKE
F1 score on RE	60.5%	67.20%	70.80%	72.70%
F1 score on FRE	13.9%	41.6%	42.1%	64.70%

As English grammar is generally not changing, so most of the times it is possible to identify language patterns for family relations in text. This key observation

leads to the following hypothesis:

Hypothesis: “Regular patterns for family relations exist in natural language texts.”

If regular patterns exist, then rules can be created against those patterns to extract family relations at very low cost in terms of computation time because rules are deterministic.

To identify the regular patterns, natural language texts need to be analyzed. Comprehensive sets of natural language texts are available as published datasets. So we studied the TACRED dataset for family relations [2], and performed experiment on the dataset with state-of-the-art RE systems. The FRE annotations made by four RE systems have been analyzed. Many family relations are incorrectly annotated by these systems. Consider an example text given below, with all instances of persons shown in bold text.

*Francis Goncalves, a chef, said **he** believed that **his** father, **Basil**, 73, contracted Covid while in hospital and **his** mother, **Charmagne**, 65, and brother **Shaul**, 40, picked it up at a family dinner.*

The family relations that should be extracted from the example text are given in Table 1.3, along with the output of the RE systems on this input text. Incorrect annotations by the systems are shown in bold text. It can be seen from the table that existing RE systems are making many annotation mistakes when relations are of family types. This performance is not satisfactory.

Carefully inspecting family relation annotations examples, following regular patterns have been discovered for family relations or in general any relation between subject and object. In Table 1.3, the type of each relation is also mentioned.

1. **Direct relation** (s, r, o) , when the relation r between subject s and object o is directly mentioned in the text.

TABLE 1.3: Output of RE systems for the example sentence

Sr.	(Subject, Object)	Relation	Annotation by				Relation Type
			Stanford	TACRED-PA	SpanBert	LUKE	
1	(Francis Goncalves,Basil)	per:parents	no_relation	per:parents	per:parents	per:parents	coref
2	(Francis Goncalves,Charmagne)	per:parents	no_relation	per:siblings	per:parents	per:parents	coref
3	(Francis Goncalves, Shaul)	per:siblings	no_relation	per:children	per:siblings	per:siblings	coref
4	(he, Basil)	per:parents	no_relation	per:children	per:parents	per:parents	coref
5	(he, Charmagne)	per:parents	no_relation	per:siblings	per:parents	per:parents	coref
6	(he, Shaul)	per:siblings	no_relation	no_relation	per:siblings	per:siblings	coref
7	(his, Basil)	per:parents	no_relation	no_relation	per:parents	per:parents	direct
8	(his, Charmagne)	per:parents	no_relation	per:siblings	per:parents	per:parents	coref
9	(his, Shaul)	per:siblings	no_relation	per:parents	per:siblings	per:siblings	coref
10	(Basil, Francis Goncalves)	per:children	no_relation	per:parents	per:parents	per:children	reverse
11	(Basil, he)	per:children	no_relation	per:parents	per:parents	per:children	coref
12	(Basil, his)	per:children	no_relation	per:parents	per:parents	no_relation	reverse
13	(Basil, his)	per:children	no_relation	per:parents	per:parents	no_relation	coref
14	(Basil, Charmagne)	per:spouse	no_relation	per:siblings	per:parents	per:spouse	transitive
15	(Basil, Shaul)	per:children	no_relation	per:parents	per:siblings	per:children	transitive
16	(his, Basil)	per:parents	no_relation	per:parents	per:parents	per:parents	coref
17	(his, Charmagne)	per:parents	per:children	per:siblings	per:parents	per:parents	direct
18	(his, Shaul)	per:siblings	no_relation	per:parents	per:siblings	no_relation	direct
19	(Charmagne, Francis Goncalves)	per:children	no_relation	per:parents	per:parents	per:children	transitive
20	(Charmagne, he)	per:children	no_relation	per:parents	per:parents	no_relation	transitive
21	(Charmagne, his)	per:children	no_relation	per:parents	per:parents	no_relation	reverse
22	(Charmagne, Basil)	per:spouse	no_relation	per:parents	per:parents	per:parents	transitive
23	(Charmagne, his)	per:children	per:parents	per:siblings	per:parents	no_relation	reverse
24	(Charmagne, Shaul)	per:children	no_relation	per:siblings	per:siblings	per:siblings	transitive
25	(Shaul, Francis Goncalves)	per:siblings	no_relation	per:siblings	per:siblings	per:siblings	coref
26	(Shaul, he)	per:siblings	no_relation	per:siblings	per:siblings	no_relation	reverse
27	(Shaul, his)	per:siblings	no_relation	per:children	per:siblings	no_relation	reverse
28	(Shaul, Basil)	per:parents	no_relation	per:siblings	per:siblings	per:parents	reverse
29	(Shaul, his)	per:siblings	no_relation	per:siblings	per:siblings	no_relation	reverse
30	(Shaul, Charmagne)	per:parents	no_relation	per:spouse	per:siblings	per:parents	reverse

2. **Reverse relation** (o, r', s) , when (s, r, o) is a relation.
3. **Transitive relation** (s, r, o) , when $(s, r1, x)$ and $(x, r2, o)$ are relations.
4. **Coref relation** (s, r, cO) or (cS, r, o) , when (s, r, o) is a relation and cO is a co-reference of o , or cS is a co-reference of s , respectively.

So the hypothesis is found true, that is, regular patterns for family relations do exist in natural language texts. Then there must be rules to detect these patterns in order to extract family relations.

1.5 Problem Statement

Existing NER systems fail to recognize NEs (see Table 1.1) which:

1. contain nationalities
2. have corresponding resources in DBpedia
3. are acronyms
4. are re-occurrences of NEs

Furthermore, existing RE systems do not perform well on extraction of family relations (see Table 1.2 and Table 1.3), which may be one of the following types:

1. direct relation
2. reverse relation
3. transitive relation
4. coref relation

A technique is therefore needed, which is able to extract above kinds of NEs and family relations from text, which are incorrectly extracted by existing systems.

1.6 Research Questions

To solve the above mentioned problem, the following research questions (RQs) need to be answered:

RQ1: How to formulate rules to recognize NEs which are missed by existing NER systems?

RQ2: How to formulate rules to extract family relations which are incorrectly extracted by existing RE systems?

1.7 Research Objectives

This research achieves the following objectives:

RO1: Devise a technique to recognize named entities from text, specially those instances which are missed by existing NER systems

RO2: Devise a technique for better extraction of family relations from text

1.8 Research Scope

The scope of this research is to develop an information extraction system, to:

1. Recognize from text, NEs belonging to following classes:
 - (a) Person
 - i. Male
 - ii. Female
 - (b) Location
 - (c) Organization
2. Extract family relations between persons as one of the following:

- (a) Spouse
- (b) Children
- (c) Parents
- (d) Siblings
- (e) Other Family
- (f) Not Known

1.9 Research Methodology

The research methodology adopted for carrying out this research work is the Design Science research methodology as proposed by Dresch et al. [57].

Design Science seeks to consolidate knowledge about the design and development of solutions, to improve existing systems, solve problems and create new artifacts.

The design science research process recommended by Dresch et al. consists of 12 main steps. Fig. 1.6 presents the 12 step process tailored for this research. First, the problem has been identified, research questions have been formulated and objectives have been defined. A deeper understanding of the problem has been developed by studying existing literature. Two classes of information extraction problem have been configured for this work: NER and FRE. Existing systems for the task have been identified from literature and datasets have been selected and prepared. Solutions for better solving the NER and FRE problems have been proposed and designed. The system modules CustNER and CustRE have been developed, and evaluated using standard metrics: precision, recall and micro F1 score, successfully improving previous results. Clarification of learning has been achieved by identifying the factors which have positively contributed to the research success, along with identifying the cases where the proposed system failed. Conclusions have been drawn and limitations of the research have been reported for future work. Generalizations of the system have been made for a class of problems:

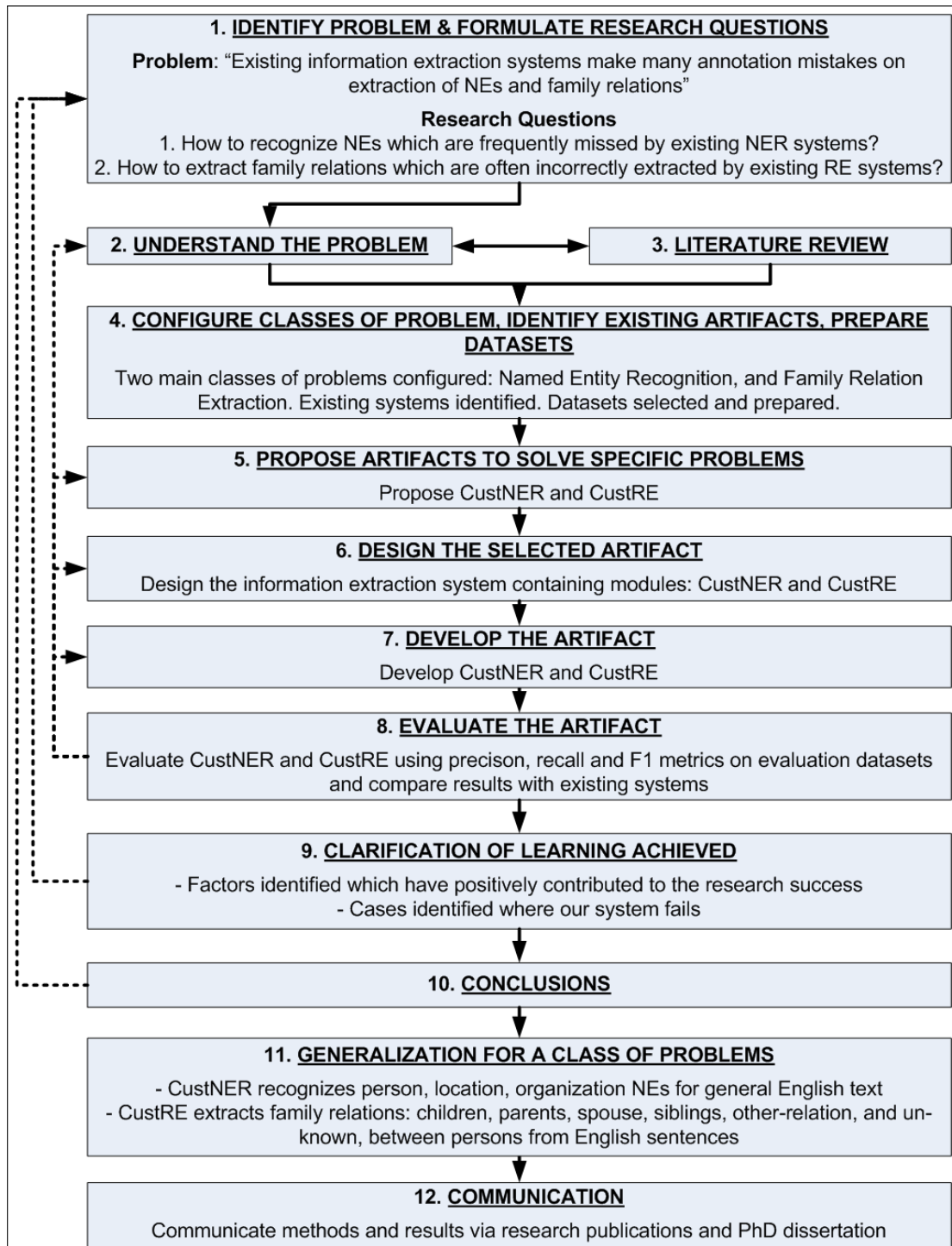


FIGURE 1.6: The Design Science Research Methodology Process for this work

CustNER recognizes person, location and organization NEs for general English texts, along with marking persons as male or female, and CustRE extracts six family relations (children, parents, siblings, spouse, other family and not known) between persons from general English sentences. Finally, the methods and findings

of the research have been communicated to relevant audience through journal publications and this dissertation.

1.10 Dissertation Contributions

The main contribution of this dissertation is an information extraction system that: extracts person, location and organization type named entities, marks person entities as male or female, and extracts any family relations between person entities; from any generic un-structured English text. The specific contributions have been listed in the following two sub-sections.

1.10.1 NER Contributions

1. Type annotation and enhancement of the OKE dataset.
2. Identifying the types of NEs commonly missed by existing NER systems.
3. Formulation of rules to recognize NEs missed by existing NER systems.

1.10.2 RE Contributions

1. Constructing a new comprehensive dataset, the CustFRE dataset, for evaluating family RE systems.
2. Enhancement of the TACRED dataset's family relations subset.
3. Formulation of rules to recognize family relations between person mentions in a sentence.

Through these contributions, the objectives and scope which were defined for this work have been met, the research questions have been answered, and hence the problem which was identified for this research has been solved.

1.11 Dissertation Organization

Rest of this dissertation is organized as follows:

- Chapter 2 describes the relevant literature investigated; the shortcomings identified, the evaluation metrics used, the benchmark datasets available and the systems selected for comparison.
- Chapter 3 details on the selection and assessment of benchmark datasets; the criteria devised for dataset quality assessment, the shortcomings found in available datasets and the preparation and creation of enhanced datasets.
- Chapter 4 explains the first module of the proposed information extraction system, CustNER, for recognition and classification of person (male and female), location and organization type NEs in English text; implementation of CustNER; and its evaluation and comparison with existing systems.
- Chapter 5 describes the second major component of the proposed IE system, CustRE, for extraction and classification of family relations between person entities in text; implementation of CustRE; and its evaluation and comparison with existing systems.
- Chapter 6 concludes the dissertation along with outlining future research directions.

Chapter 2

Literature Review

The main difficulty in building KG is that existing systems are unable to completely and accurately extract entities and relations. The better and accurate these are extracted, the more accurate KG could be build. Initial experiments of this work suggest that results of existing systems for NER and FRE are not very good, see Tables 1.1 to 1.3. To further investigate this, relevant literature has been reviewed, with the following main aims:

1. To study existing NER and RE work to improve our understanding on the topic.
2. To find available datasets for NER and RE.
3. To find existing NER and RE systems whose APIs or implementations are available, in order to prepare a test bed for experimentation and comparison.
4. To find the standard evaluation metrics used for evaluating NER and RE systems.

This chapter mainly describes steps 2, 3 and 4 of the research methodology, see Fig. 1.6, that is, improve understanding of the problem by reviewing literature, identify categories of problem, and identify existing systems and datasets.

TABLE 2.1: Year wise distribution of papers reviewed

	2005-07	2008-10	2011-13	2014-16	2017-19	2020-22	Total
Rule-based NER	1	1	2	0	1	5	10
ML based NER	1	1	0	3	1	4	10
Hybrid NER	0	0	1	1	4	0	6
Rule-based FRE	0	1	1	1	1	1	5
ML based FRE	0	0	0	1	1	3	5
Hybrid FRE	0	0	0	3	0	0	3
Total	2	3	4	9	8	13	39

The search strategy adopted for selecting papers is given in Appendix B. The following inclusion criteria were developed to enhance the likelihood that relevant articles would be included:

1. Literatures published as journal papers, conference papers, books, book chapters and technical reports.
2. Papers describing techniques for recognition of person, location and organization type NEs from English texts.
3. Papers presenting techniques for relation extraction from English texts, that include family types.

Exclusion criteria used to filter the articles include:

1. Papers not written in English.
2. RE papers not explicitly extracting any family relation.
3. Papers describing ML NER systems for languages other than English, or for NE types other than person, location and organization.

A year wise distribution of the papers discussed in this chapter is given in Table 2.1. Next, the literature reviewed has been described, first for the NER field and then for the RE field.

2.1 Named Entity Recognition

The surveyed literature on NER is described under three main headings based on the techniques used: rule-based systems, ML based systems, and hybrid systems. In the next sections, relevant works are described that fall under each of these categories.

2.1.1 Rule Based Systems

These systems are developed by carefully analyzing language texts and making rules that recognize NEs from text. Rule-based systems usually perform better for specific domains and need to be modified for application to another domain. They are transparent and explainable models, that is, the model and the respective predictions made by these systems are easily understandable by humans. These systems are very fast and need less memory and computation power. But rules are static by nature, are domain specific, and designing good rules is time taking for the designer and requires expert knowledge. Several domain-specific rule-based NER systems have been proposed in the literature.

Cucerzan [58] proposes a large scale system for NERD based on regular expressions, gazetteers, and knowledge from Wikipedia and the web, for PER, LOC, ORG and miscellaneous (MISC) classes. F1 0.84 is reported for the NER component of the system, on the CoNLL 2003 evaluation dataset. The CoNLL03 dataset is available online. The system is for general English language, but is described at an abstract level and is not publicly available.

Riaz [59] proposes a rule-based NER algorithm for Urdu language that he demonstrates outperforms the systems which use statistical learning models. The rules are created for 6 classes by analyzing 200 documents of Becker-Riaz corpus, the experiments are run on 2,262 documents and 91.1% F1 is reported. The rules are only described at an abstract level and the system is not openly available. Singh, Goyal, and Lehal [60] later propose an Urdu NER system that tags into the 12 NE

classes used in IJCNLP-08 workshop. System is described at very abstract level, reporting F1 score of 0.6 and 0.88 on two self created test datasets.

The system proposed by Hakimov, Oto and Dogdu [61] identifies NEs, disambiguates and links them to DBpedia, it does not classify NEs. The system parses text using an adjusting sliding window, consecutive words in the window are searched in DBpedia and the window is adjusted if a match in DBpedia is not found. When multiple matches are found, all of them are kept as candidates for later disambiguation. A graph is then constructed where nodes are the spotted entities, and edges are the wikiLinks between them. Graph centrality scores are then used to disambiguate the entities, the node with highest centrality score is kept and the other nodes are removed from the graph. Two gold sets are used for evaluation: DBpedia Spotlight Project dataset and a self-created dataset, and F1 50% and 41% has been reported. Online links for both dataset and system are given in the article, but both the links are not working.

Mesmia, Haddar, Friburger, and Maurel propose CasANER [62], a system for recognizing NEs for Arabic language. The system is based on a deep categorization made using Arabic Wikipedia corpus. This system reports F1 0.91 on a corpus constructed using Arabic Wikipedia articles and 0.67 on ANERcorp.

Sanjaya and others [63–65] have designed rules around morphological and contextual features to detect PER, LOC and time entities from Balinese texts, reporting F1 0.85 on self created dataset. Rules are not described in the article, only one sample rule is presented. Prasad and Sharma [66] develop a rule based system for recognizing entities related to food and health issues from home remedies weblogs in Hindi language, and report accuracy of 0.925 on self created dataset.

Tarmizi and Saad [67] propose a rule base system for extraction of eight type of entities (person, prophet, group, location, afterlife, god, creation, stime) from English translation of the Holy Book of Quran. Output of Spacy NER is used as input to the system. Two types of rules are developed, one that use pre-defined gazetteers to match entities, and other that use regular expressions to

match common patterns of entities. F1 0.94 is reported on the English translation of chapter 21 of the Holy Quran, whereas Spacy performed only 0.61 F1.

Table 2.2 presents a summary of the systems described in this section, in terms of the domain for which system is built, datasets used, results reported, and whether the datasets and the systems are openly available or not. It can be seen from Table 2.2 that existing rule-based NER systems are mostly domain-specific, or are for languages other than English, and are usually tested on self created unpublished datasets. The works by Cucerzan [58] and Hakimov, Oto and Dogdu [61] are for general English language, but [61] only identifies NEs, it does not classify them. Only the system by Cucerzan [58] recognizes NEs from general English text for general classes. This system reports an F1 0.84 on the CoNLL03 dataset, which is low and should be improved further. Moreover, none of these systems is openly available for other researchers to use and test. Usually rules are not described by the authors, and even when described, they are presented at very abstract level that does not enable other researchers to use them. Of the datasets used by these systems, the only dataset for English language which is openly available, is the CoNLL03 dataset.

2.1.2 Machine Learning Based Systems

These systems use machine learning techniques or statistical models for recognizing NEs. ML systems are dynamic in nature, can usually be retrained on different datasets and applied to different domains, but are complex and difficult to comprehend. In ML, NER task is considered a classification task i.e. to classify text into predefined classes e.g. PER, LOC, ORG classes. The Stanford NER and the UIUC (University of Illinois at Urbana Champaign) NER, are well-known and widely used systems [68–71]. Developers of Stanford NER, Finkel et al. emphasize the need to model global structure in extraction models and use Gibbs sampling to incorporate long-distance text dependencies into a CRF-based NER system, train it on a mixture of CoNLL03, MUC-6, MUC-7 and ACE NE corpora, and get improved F1 of 86.86 on CoNLL03 dataset [27].

TABLE 2.2: Summary of Rule-based NER systems

Year	Domain	Dataset	Reported F1	Dataset Available?	System Available?
2007 [58]	English Text	CoNLL03 dataset	0.84	✓	×
2010 [59]	Urdu Text	Becker-Riaz corpus	0.91	✓	×
2012 [60]	Urdu Text	2 self-created datasets	0.6 and 0.88	×	×
2012 [61]	English Text	Extraction from DBpedia spotlight, and a self-created dataset	0.5 and 0.4	Link down ¹	Link down ¹
2018 [62]	Arabic Text	Extraction from Arabic Wikipedia, and ANERcorp	0.91 and 0.67	×	×
2021 [63–65]	Balinese Text	Self-created dataset	0.85	×	×
2022 [66]	Hindi home remedies weblogs	Self-created dataset	Accuracy 0.93	×	×
2022 [67]	English Quranic Text	English translation of the 21st chapter of the Holy Quran	0.94	×	×

¹Last accessed 17th June 2022

The developers of the UIUC NER system, Ratinov and Roth [29], emphasize the necessity of using prior knowledge and non-local decisions in NER. The authors have gained improved performance by; using naive greedy left to right inference algorithm; using a regularized averaged perceptron model for sequential inference; using a combination of three techniques for incorporating non-local features: context aggregation, two-stage prediction aggregation and extended prediction history; and using two external knowledge sources (unlabeled text and gazetteers). The system is trained on CoNLL03 shared task data and reports 90.8 F1.

JERL [68] is a probabilistic graphical model that jointly optimizes NER and linking tasks completely together. It is a ML-based method that uses hand-engineered features. JERL extends Semi-CRF to model entity distribution and mutual dependency over segmentations. It uses conditional log likelihood with L2 normalization as the objective function in training and a limited-memory quasi-Newton method [72] to solve the optimization problem. It extends the Viterbi algorithm to exactly infer the best assignment. JERL reports F1 91.4% on CoNLL03 dataset.

Scharolta Katharina [73] uses *word2vec* with continuous skip gram model to extract word vectors, cluster word vectors using k-means clustering, and give cluster of each word as a feature to LSVM for recognizing NEs. CoNLL03 dataset is used and F1 83.8 is reported.

Lample, Ballesteros, Subramanian, Kawakami and Dyer [74] present two NNs for NER. First is a hybrid of biLSTM and CRF. The second constructs and labels segments using a transition based approach inspired by shift-reduce parsers. The models rely on two information sources: character based word representations learned from supervised corpus and unsupervised word representations learned from un-annotated corpora. For both the models, training is done using back propagation algorithm, updating the parameters on every training example, using SGD. CoNLL03 dataset is used. The system is claimed to outperform all previous NER systems for German, English and Spanish languages.

NER frameworks have also been proposed that do not start NER from scratch, but rather use the NER output of existing systems (such as of Stanford NER and/or

UIUC NER) as input to their systems and build on top of that to give improved NER results. René Speck and Ngomo [31] offer a multi-lingual ensemble learning framework, FOX, that takes outputs of several existing NER systems (Stanford, UIUC, the Ottawa Baseline Information Extraction [75] and the Apache OpenNLP Name Finder²) as input and claims to give improved classification results as output for PER, LOC and ORG classes. An F1 of 86.12% is reported on the OKE 2017 dataset for NE identification task [13].

Xiaofeng et al. [76] propose LDDFC, a method for constructing NEs dictionary from labeled data and incorporating dictionary feature in biLSTM-CRF model, for improved recognition of NEs. During the training stage, the dictionary is built from training dataset; but during the testing stage, authors use SENNA³ dictionary (including 171,142 entries categorized into four classes: PER, LOC, ORG and MISC) and token-level matching. Authors report an F1 score of 82.6 on CoNLL03 dataset when the word embeddings are randomly initialized, and 90.8 when GloVe [77] word embeddings are used.

An important decision in constructing ML based NER systems is that of representing entities. Conventional entity representations assign each entity a fixed embedding vector which stores information regarding the entity in a knowledge base [56], for example [78–84]. These approaches cannot represent entities which do not exist in the knowledge base. So recent studies [85–90] have moved towards using contextualized representations of entities computed based on contextualized word representations (CWRs), which provide effective general-purpose word representations trained with unsupervised pre-training tasks based on language modeling. Most recent models are based on transformer trained using a task similar to masked language model. These include BERT [91], RoBERTa [92], XLNet [93], SpanBERT [42], ALBERT [94], BART [95], and T5 [96]. Numerous recent studies have also explored methods to enhance CWRs by injecting them with knowledge from external KBs, for example ERNIE [85], K-Adapter [89], KnowBERT [87] and LUKE [56].

²<https://opennlp.apache.org/>

³<https://ronan.collobert.com/senna/>

LUKE uses transformer with an entity-aware self-attention mechanism which is trained on a large entity-annotated corpus from Wikipedia. LUKE is designed to effectively solve entity-related tasks and achieves superior empirical results to existing methods on many entity related tasks including NER and RE tasks. LUKE is the highest scorer on CoNLL03 NER dataset, reporting an F1 of 94.3. But, these deep learning (DL) models are data-hungry and computationally very expensive. LUKE is pre-trained on Wikipedia dataset containing 3.5 billion words and 11 million entity annotations and trained on CoNLL. LUKE’s pre-training took 30 days on a server with 2 Intel Xeon Platinum 8168 CPUs (each containing 24 cores) and 16 NVIDIA Tesla V100 GPUs. Further, LUKE’s training on CoNLL took 203 minutes on a server with 2 Intel Xeon E5-2968 v4 CPUs (each having 16 cores) and 8 V100 GPUs.

Harrando and Troncy [97] have considered NER as a graph classification task. Authors experimented with 4 different graph representations. The best F1 score is achieved by GCN, i.e. 81.0. Wang et al. suggest k NN-NER [98], a framework for augmenting k nearest neighbours from the training set into the entity labels distribution of existing NER models, at inference stage, for improving performance. Authors experiment with BERT and RoBERTa NER models and demonstrate an improvement of upto +2.24 and +1.59, respectively, in F1 results of the vanilla models. On the English CoNLL03 dataset, highest F score is made by k NN-RoBERTa, 92.39%.

Table 2.3 presents a summary of the systems described in this section, in terms of the main technique used, classes of NEs recognized, dataset used, result reported, and whether the dataset and the system are openly available or not. It can be seen from Table 2.3 that the dataset used by most existing ML NER systems for the purpose of reporting and comparing results is the CoNLL03 dataset. Many of these systems are openly available. Experiments have been performed with some of these openly available well known NERs by giving some test examples. Analysis of the annotations made by these systems on the OKE dataset reveals that the NEs missed by these NERs are mostly similar (refer to Table 1.1). For instance, from the text “The man was arrested upon landing and **South Korean**

TABLE 2.3: Summary of ML NER systems

System/ Year	Technique	Classes	Dataset	Reported F1	Dataset Available?	System Available?
Stanford, 2005 [27]	Conditional random fields with gibbs sampling	PER,LOC,ORG, MISC	CoNLL03	0.86	✓	✓
UIUC, 2009 [29]	Regularized averaged perceptron model for sequential inference	PER,LOC,ORG, MISC	CoNLL03, MUC-7	0.91, 0.86	✓	✓
JERL, 2015 [68]	Extended semi-CRF	PER,LOC,ORG, MISC	CoNLL03	0.91	✓	×
2015 [73]	<i>word2vec</i> , continuous skip gram, LSVM	PER,LOC,ORG, MISC	CoNLL03	0.84	✓	×
2016 [74]	biLSTM-CRF and Stack-LSTM NNs	PER,LOC,ORG, MISC	CoNLL03	0.90	✓	✓

Continued on next page

Table 2.3 – continued from previous page

System/ Year	Technique	Classes	Dataset	Reported F1	Dataset Available?	System Available?
FOX, 2017 [31]	Ensemble Learning of Stanford, UIUC, Balie, OpenNLP NERs	PER,LOC,ORG	OKE	0.86	✓	✓
LDDFC, 2020 [76]	NE dictionary incorporated as feature in biLSTM-CRF	PER,LOC,ORG, MISC	CoNLL03	0.90	✓	×
LUKE, 2020 [56]	BiTransformer with entity-aware self-attention	PER,LOC,ORG, MISC	CoNLL03	0.94	✓	✓
GraphNER, 2021 [97]	Graph classification, GCN	PER,LOC,ORG, MISC	CoNLL03	0.81	✓	✓
k NN-NER, 2022 [98]	Integrate k NN into existing NER models	PER,LOC,ORG, MISC	CoNLL03	0.92	✓	✓

police said he was drunk . . . followers worldwide were checking **Twitter** to find out what . . .”, Stanford, Illinois and FOX NERs fail to recognize the organization NERs South Korean police and Twitter, which belong to missed NERs type 1 (organization NERs augmented with nationalities) and type 2 (NERs having corresponding DBpedia resources) respectively. It has been analyzed that NERs not recognized by ML NER systems are similar, but since ML systems are black box, it could not be interpreted why these systems are missing similar NERs from recognition. DL based systems have generally reported good performances, but this high performance comes at the expense of high computation and memory requirements, for both the training and inference phases of DL. Training a DL model is expensive due to millions of parameters that need to be iteratively refined many times. In DL inference, the input data passes through many layers in sequence, each layer performs matrix multiplications on the data, and the output of one layer is input to the subsequent layer. There are a large number of parameters in the matrix multiplications, resulting in many computations being performed. Thus inference is also very expensive. High accuracy and high resource consumption are defining characteristics of DL methods [99].

2.1.3 Hybrid Systems

Systems have also been proposed in the literature, which are a hybrid of heuristics and ML techniques. Sil and Yates’s [100] propose NEREL, which considers NERL a joint task and solves them together, instead of dealing with them as two separate tasks which are usually performed in a pipeline. Taking a large set of candidate mentions from UIUC NER and heuristics, and a large set of entity links from typical Entity Linking (EL) systems, NEREL ranks the candidate mention-link pairs together making joint predictions. System uses Maximum Entropy model to estimate probabilities, and L2-regularized conditional log likelihood (CLL) as the objective function for training. Three test sets are used: ACE, MSNBC and CoNLL03 datasets. The labeled data for evaluation contains annotations for links of NERs, and not the entity type tags. The system is reported to outperform two

state-of-the-art NER systems: the UIUC NER and the Stanford NER systems, on the NER task, and six state-of-the-art EL systems on the EL task.

Chabchoub et al. [101] propose a system that gets input text annotated using an existing NER system, Stanford NER, and four semantic annotators: Dbpedia Spotlight, Babelfy, AIDA and TagMe. Entities annotated by these five annotators may be different or may have overlapping. If there is overlapping in the entities, then the largest of these overlapping annotations is selected. The system then uses some heuristics to enlarge the mentions further if possible. F score of 78.94% for entity recognition task on OKE16 train dataset and 73.27% on OKE15 evaluation dataset has been reported. This system is the winner of Open Knowledge Extraction (OKE) challenge 2016, task1 (entity recognition, linking and typing).

The system ADEL combines results of several extractors and proposes a flexible system in which user can configure which of these extractors to include for recognition [32, 33, 102]. The extractors are dictionary, POS tagger, off-the-shelf NER systems (Stanford NER, OpenNLP), date, number and co-reference extractors. The system then uses an overlap resolution module, which comprises some heuristics and takes union of annotations from the extractors to finalize the annotations. An F score of 87.21% is reported for ADEL on the OKE 2017 dataset for NE identification task [13].

Marrero and Urbano [103] propose RB-AL (rule-based active learning), a system for generating NER rules. The system semi-automatically learns regular expressions and JAPE (Java Annotations Patterns Engine) patterns without using any annotated corpora, by using some annotated seed examples. The method uses features to describe entities at character and token levels. Experiments have been performed on the Software Jobs Corpus⁴, having annotations for 17 entity types related to jobs postings like postdate, phone, country, city, state, salary, recruiter, etc., 1 more entity type, phone number, is annotated by authors. An F score of 0.73 is reported for annotating entities when 100 seeds are used.

⁴<ftp://ftp.cs.utexas.edu/pub/mooney/ie-data/jobs300data.tar.gz>

Chen et al. [99] demonstrate that prior knowledge (like knowledge graph, syntactic dependency of input text) could be decomposed into a set of logic rules, which could be embedded into RNNs to improve RNNs' performance for NER task. Authors have experimented on CoNLL03 dataset with three methods; RNN, LSTM and bi-LSTM, and reported F scores of methods improving by approximately 2% when logic rules are embedded in them. Best performance has been reported for logic rules embedded biLSTM, an F score of 81.6%.

Han et al. [104] present TANER, which conducts NER and NEN simultaneously and combines bi-LSTM with a rule-based mention-pair extractor and CRF. The NEN module of TANER uses syntactic and lexical rules to identify abbreviated mentions of NEs in text which were missed by Stanford NER. Here is an example rule from TANER: if A is a substring of F and text has pattern $F - A$, then A is an abbreviation of F and both A and F refer to same NE. This way, TANER utilizes general knowledge for recognizing and normalizing new entities from definitions present in the text. On CoNLL03 dataset, an F score of 90.87 has been reported by the NER system. Though, there is a serious problem with TANER's rules. The lexical condition that TANER checks for abbreviations is that it is a substring of the entity. Consider the text: "The University of Lahore-the first to start this program ...". In this text, "The University of Lahore" is the full name F and "the" is its abbreviation A according to the proposed rules. Because it satisfies both a syntactic pattern: " $F - A$ " and a lexical pattern: " A is a substring of F ", hence A is an acronym for F according to TANER, but this is not correct, as "the" is not an acronym of "The University of Lahore". Hence just checking substring is not sufficient for recognizing abbreviated mentions of NEs, thus demonstrating the need for formulation of better rules that can recognize abbreviations/acronyms missed by TANER. Yet, Han et al., [104] have rightly identified one type of NEs which the existing systems fail to recognize, the acronyms.

A summary of the systems for English language described in this section, is given in Table 2.4. It can be seen from the Table that the only general datasets used by existing hybrid NER systems which are freely openly available, are the OKE and CoNLL03 datasets. From the systems listed, ADEL is the only one openly

TABLE 2.4: Summary of hybrid NER systems

System/ Year	Technique	Classes	Dataset	Reported F1	Dataset Available?	System Available?
NEREL, 2013 [100]	Existing system (UIUC NER) + heuristics + MaxEnt	PER,LOC,ORG,MISC	ACE, CoNLL03	0.92, 0.88	For 4000\$, ✓	×
2016 [101]	Stanford NER + heuristics	PER,LOC,ORG	OKE	0.73	✓	Link down ⁵
ADEL, 2017 [32]	Existing systems (Stanford, OpenNLP) + heuristics	PER,LOC,ORG	OKE	0.87	✓	✓
RB-AL, 2018 [103]	Rules + clustering + active learning	18 classes:postdate, country,recruiter,...	Software Jobs corpus	0.73	✓	×
2019 [99]	Rules + RNN	PER,LOC,ORG,MISC	CoNLL03	0.81	✓	×
TANER, 2019 [104]	Rules + bi-LSTM + CRF	PER,LOC,ORG,MISC	CoNLL03	0.90	✓	×

⁵Last accessed 19th June 2022

available. The annotations made by ADEL on the OKE dataset were analyzed and again it was found that the types of its false negatives is also similar (refer to Table 1.1). The entities, which get missed from recognition by Stanford, Illinois and FOX NERs, get missed from annotation by ADEL as well. For instance ADEL also fails to annotate the NEs South Korean police (missed NEs type 1, entities containing nationalities) and Twitter (missed NEs type 2, entities having corresponding DBpedia resources).

2.2 Relation Extraction

In this section, the existing work related to family relations extraction has been reviewed. There are mainly three categories of methods; rule-based, ML based, and hybrid of both.

2.2.1 Rule based methods

Although family relation type has been part of the Automatic Content Extraction (ACE) editions in one form or the other [6], but specifically extracting family relations started gaining more interest during the last decade, after Santos et al. [105] presented a rule based system to extract family relations such as uncle, parent, sibling, etc. from Portuguese narrative. They extend the rule-based grammar, already implemented in syntactic parser of the Portuguese NLP pipeline developed at their L2F⁶ lab, to identify family relations. The system uses 99 rules including both global and specific rules and mainly utilizes the dependency structure of the sentence. The word features used are related to gender, number, the lemma of a word, person nouns and the feature “relative” that is present in every word related to a family relation. The authors evaluate their system on two small datasets: first is the biographies of Portuguese kings from Wikipedia which they manually annotated (contain 105 family relations) and the second is 21 sentences taken from

⁶Spoken Language Systems Laboratory of the Institute of Systems and Computer Engineering - Research and Development

CETEMPublico⁷ corpus. The datasets are not publicly available. F scores of 35% on first dataset and 45% on second dataset have been reported. These scores are not satisfactory, signifying the need for better techniques for extraction of family relations between persons.

Kokkinakis and Malm [46] propose using two techniques for extracting relations between persons from Swedish prose fiction corpus. RE is mainly based on the context between two relevant person entities, and utilizes three online resources: the Relationship vocabulary and two Swedish lexical semantic sources. Appropriate relationship oriented lexical units as well as relation labels are identified with the help of these resources. If the number of tokens (context) between two person entities is less than 4, then they are labeled by simple pattern matching with the resources. Average F score of 84.7 has been reported for such cases. For longer contexts (number of tokens 4 to 10) between persons, bag of words context vectors are produced. Most frequent words of the clusters are then picked manually and mapped to the Relationship⁸ vocabulary. Average precision of 42.9 has been reported for these cases. The low result suggests that considering only the most frequent words between two persons is not sufficient to determine the relationship between these persons. A technique better than frequent words mapping is needed for effective identification of relations between persons.

Janakiraman [48] reports her experiments of extracting relationships between characters of short stories, based on the hypothesis that words which surround a character pair describe the relationship between the pair. Bag of words approach has been used to associate a set of words to a story character pair, then similar character pairs are grouped based on the correlation coefficient between their associated words. The relation for a group of pairs is then determined by computing a similarity score between the group's associated bag of words and the predefined relations. A corpus of 55 short stories taken from Project Gutenberg⁹ has been manually annotated and used for the experiments and precision of 55% has been reported. The system and dataset are not publicly available.

⁷<http://www.publico.pt/>

⁸<https://vocab.org/relationship/>

⁹<https://www.gutenberg.org/>

Romadhony et al. [106] propose rule-based open information extraction system for Indonesian language, to extract as much relation triples as possible, without restriction on the classes to be extracted. F1 score of 0.64 has been reported for rules based on PoS tags and noun-phrases. As the relations generated are not restricted to a vocabulary, it will cause redundancies in the KG.

Norabid and Fauzi [107] extract relation triples from web texts surrounding images (article title and image caption), to construct knowledge graph that describes the image. A rule based triple extractor is developed by linguistically analyzing a set of web news articles, and by utilizing dependency tree and PoS information. The system does not use a pre-defined relation vocabulary, and generally uses verbs as relations. The system reports precision of 0.9 and recall 0.6 on a manually annotated dataset of 60 web news articles, containing 319 relation triples. The approach is fine for describing images, but as it does not use a predefined vocabulary, it will result in redundancies when the KG merges with other KGs.

Some key observations about these works are presented in Table 2.5, in terms of the year of publication, language for which system is built, the relation classes extracted by system, dataset used, result reported, and whether the dataset and the system are openly available or not. It can be seen from Table 2.5 that existing rule-based systems for extraction of family relations are tested on self created unpublished datasets. The reported results are not very good, specifically, one of the two systems for English language [48] reports a precision of 55% while the other [107] does not map extracted relations to any ontology. These results are not satisfactory and should be improved further. Moreover, none of these systems is openly available for other researchers to use, test or extend.

2.2.2 Learning based methods

Efremova et al. [45] study two ways to extract the family relationships: marriage, parent-child, widow-of, sibling-to, and nephew-of, from a collection of Dutch historical notary acts. Authors designed a web interface to get data annotated from

TABLE 2.5: Summary of Rule-based FRE systems

Year	Language	Classes	Dataset	Reported F1	Dataset Available?	System Available?
2010 [105]	Portuguese	Uncle,parent,sibling, cousin,etc.	1. Manually annotated Wikipedia king biographies, 2. Sentences from CETEMPUBLICO	0.35 and 0.45	×	×
2011 [46]	Swedish	Relations between persons, e.g. sibling,hasMet, grandparent,employer,etc.	Sweden prose fiction corpus	0.42 precision	×	×
2014 [48]	English	Relations between characters, e.g. friend, parent,spouse,nephew	Manually annotated corpus of short stories from Project Gutenberg	0.55 precision	×	×
2018 [106]	Indonesian	Open relations	Self created	0.64	×	×
2022 [107]	English	Open relations	Manually annotated dataset from web news	0.72	×	×

experts. In the first approach, authors generate all potential candidate pairs of person names and then classify them into family relations or no relation using SVM, where the text fragments around and between two names are used as features. In the second approach, a HMM is trained and applied to annotate every word in document with an appropriate tag indicating if it is a name, a specified relationship descriptor, or neither. The names connected to each other via relationship descriptors are the relations. For the family relations having more examples in dataset, the results of HMM are better as compared to the classification technique, while for relations having less examples, classification performs better. In the best configuration, average F score for 5 relation types is 40%, with best result reported for marriage relation, which has highest number of examples (530) in dataset. This result is very low, implying that some technique other than SVM or HMM should be developed for effective FRE.

Zhang et al. [2] realize that existing work on RE has been unable to achieve sufficient recall or precision for the results to be usable. Authors identify two reasons that held back RE research, the models used not being properly adapted for RE task, and un-availability of well annotated dataset. Zhang et al. therefore propose a new model, which integrates entity's position-aware attention into an LSTM sequence model for better suiting to relation extraction problem. The word embeddings fed into the model have been initialized by pre-trained GLoVe vectors [77] and are augmented with PoS and NER tags (using Stanford corenlp). Furthermore, with the aid of crowdsourcing authors build TACRED, a very large RE dataset, containing 106,264 examples of 42 TAC KBP relation types between persons, locations and organizations (includes 5 family relation types). An F score of 67.2% has been reported. This result is better than results of previous systems, but still needs improvement.

In 2019, Researchers at Google AI Language propose a language representation model BERT, Bidirectional Encoder Representations from Transformers [91], which has caused a stir in the ML community by presenting SOTA results in a range of NLP tasks, including natural language inference, question answering, etc. BERT is pre-trained on unlabeled 3.3 billion word corpus (comprised of the BooksCorpus

[108] and the English Wikipedia texts) and can be fine-tuned with one additional task specific output layer per task, for the downstream NLP tasks. The pre-training took 4 days on 16 cloud TPUs (64 TPU chips total). Numerous efforts have since continued and various modifications of BERT have been proposed for several NLP tasks including relation classification.

Joshi et al. present SpanBERT [42], a modification of BERT that represents and predicts spans of tokens, instead of individual tokens. Authors suggest masking contiguous random spans, rather than random tokens and predicting the entire content of the masked span, without relying on the individual token representations within it, and demonstrate the effectiveness of their proposed technique on several tasks including the relation extraction task, and report an F1 score of 70.8% on the TACRED RE benchmark dataset.

LUKE [56] extends BERT by using a new pre-training task that treats words as well as entities in a given text as independent tokens and is trained on a large entity-annotated corpus retrieved from Wikipedia. LUKE achieves strong empirical performance on five entity-related NLP tasks, and reports an F1 score of 72.7% on TACRED, which, to the best of our knowledge, is the highest score reported by an available system on TACRED relation extraction dataset.

Zhang et al. propose REKnow (relation extraction with Knowledge enhancement) [109], a relation extraction framework for various relation settings. REKnow combines input texts with knowledge available in dataset and background external knowledge from a KG (DBpedia), and gives this combined text to an existing generative language model such as BART or T5, which generates relation triples. This way, REKnow can generate relations for different RE datasets. Authors have experimented with 5 datasets. F1 score of 74.6% is reported for TACRED dataset. But this system is not openly available.

These DL systems have generally reported good performances, but this high performance comes at the expense of high computational and memory requirements, for both the training and inference phases of DL. Strubell et al. [110] argue that

the accuracy improvements of DL models depend on the availability of exceptionally large computational resources which necessitate substantial energy consumption. Hence, these models are costly to train and develop, both financially, due to the cost of hardware and electricity or cloud compute time, and environmentally, due to the carbon footprint required to fuel modern tensor processing hardware. These models demand multiple instances of specialized hardware such as GPUs or TPUs, thus limiting access to these models on the basis of finances. Even when these expensive computational resources are available, model training also incurs a substantial cost to the environment due to the energy required to power this hardware for weeks or months at a time.

One characteristic of ML systems is their lack of interpretability. These models are generally opaque, and the international scientific community has labeled them as black-box models because of the complex mathematical functions contained in them, making them un-explainable [111].

Some key observations about these systems are presented in Table 2.6. It can be seen from the Table that the RE dataset TACRED is available for a small fee, the systems by Zhang et al. [2], SpanBERT and LUKE are openly available, and of the openly available systems LUKE reports highest F1 score on TACRED dataset (that is, 0.72). Scores of these systems are generally good on relation extraction task, but not that good on extraction of family relations, see Table 1.2. As ML based systems are black box, it could not be find out why these systems are not good at extracting family relations.

2.2.3 Hybrid methods

Makazhanov et al. [47] use a combination of heuristics and supervised approaches for extracting family relations like father, mother, daughter, son, child, etc. from Jane Austen’s English novel “Pride and Prejudice”. Their method consists of four main steps. First, each utterance (a span of narrative in quotes) is attributed to one of the story characters. Then, those utterances are identified which contain any

TABLE 2.6: Summary of learning based FRE systems

System / Year	Technique	Classes	Dataset	Reported F1	Dataset Available?	System Available?
2015 [45]	SVM, HMM	Marriage,parent-child, widow-of,sibling-to, nephew-of	Dutch historical notary acts	0.40	×	×
2017 [2]	LSTM, Glove word embeddings	Includes sibling,parent, child, spouse, other_family	TACRED	0.67	For 25\$	✓
SpanBert, 2020 [42]	BERT for token spans	Includes sibling,parent, child,spouse,other_family	TACRED	0.70	For 25\$	✓
LUKE, 2020 [56]	BERT for words and entities	Includes sibling,parent, child,spouse,other_family	TACRED	0.72	For 25\$	✓
REKknow 2022 [109]	Generative language model (Bart, T5)	Depends on dataset used	TACRED	0.74	For 25\$	×

of the nominal from a list of 635 nominals compiled by collecting WordNet¹⁰ synonyms and hypernyms of basic family relation words. Next, relations are extracted between the speakers and the characters they address. Finally, new relations are derived from already extracted relations by propagation rules. F score of 61% has been reported in best setting.

Devisree and Raj [49] propose a hybrid scheme that combines supervised and unsupervised learning and rule-based approaches for extracting relations from stories. Input story to the system is first pre-processed: tokenized, PoS tagged, NER tagging and anaphora resolution is also achieved (using Stanford's system). Sentences from story are selected if they include the specified pair of story characters in them and are classified into relation classes using Naïve Bayes Classifier. For sentences that fail to get classified, semantic similarity using UMBC¹¹ service is used to classify them. Dataset has been prepared by collecting sentences related to respective relationship (parent-child and friendship) and scoring sentences (0 to 5) in accordance with the respective relationship. The system is tested on a set of 300 character pair relationships from 100 short stories for kids, the average F1 score is 83.19%. This score seems much better than all other reported scores on FRE, but this system only classifies two relations (and just one family relation), and both these relations are un-directed, whereas systems that classify parent and child relations separately have to care for the direction of relation as well. Moreover, the system is tested on very small size dataset, only 300 examples.

The Stanford corenlp's KBP relation annotator [55], which is built on top of TAC KBP 2015 slot filling task's winning system [112], is a hybrid system consisting of eight relation extractors; five rule-based relation extractors, an Open IE extractor, a self-trained supervised statistical extractor and a supervised NN extractor. Before performing relation extraction, basic NLP tagging (PoS, lemmas, NEs, dependency parses, corefs and Wikipedia links) is obtained using Stanford CoreNLP [28], the Illinois Wikifier [113] and a set of gazetteers. The rule based extractors use 169 dependency patterns, 4528 syntactic patterns, coref chains, edit distance

¹⁰<https://wordnet.princeton.edu/>

¹¹<http://swoogle.umbc.edu/SimService/>

between organization names and URLs, and a gazetteer of GPE entities. The output from rule based extractors and from an Open IE system along with knowledge from several KBs (Freebase, KBP, Google) is used to train the logistic regression (LR) classifier based statistical model. Some tricks are also applied to avoid over-fitting and class skew. An LSTM based NN is trained on a fully supervised dataset that is constructed from previous years' KBP slot filling assessment files and is labeled by online crowd sourcing. Output triples from the relation extractors are then fed into a series of post-processors. The post-processors generate inverse relations from all predicted forward relations, perform model ensembling by training an SVM over multiple relation extractors, and remove some of the salient errors inevitably generated by the extractors. The system extracts the 41 TAC KBP relations (includes 5 family relations) from text and is openly available online¹². The system reports F1 22% and 31% on the KBP 2016 and 2015 English slot filling datasets, respectively.

Table 2.7 presents key findings about these systems. It can be seen from the Table that none of the datasets used by these systems is freely openly available. The only system available is Stanford's KBP system, which reports a very low F1 result (that is 0.31). The score of Devisree and Raj's system [49] is highest, but this system extracts only two relations and is tested on only 300 examples.

2.3 Findings of Literature Review

Reviewing the relevant literature, it has been found that many of the systems and datasets are not openly available for other researchers to use and test. The evaluation metrics used in the literature for reporting results is almost always precision, recall and micro F1 score. For the general NER classes; PER, LOC, and ORG, two openly available benchmark datasets have been found and hence used for this work:

1. The OKE17 dataset

¹²Available at <http://corenlp.run/>

TABLE 2.7: Summary of hybrid FRE systems

System / Year	Technique	Classes	Dataset	Reported F1 Result	Dataset Available?	System Available?
2014 [47]	Rules + supervised learning	Father, mother, daughter, son, child, etc.	Manually annotated Jane Austen’s novel “Pride and Prejudice”	0.61	×	×
2016 [49]	Rules + supervised and un-supervised learning	Parent-child, friendship, no-relation	Manually annotated short stories	0.83	×	×
Stanford, 2016 [55]	Rules + supervised learning	Relations between PER, LOC, and ORG, includes sibling, parent, child, spouse, other family	TAC KBP slot filling data	0.31	For 1,000\$	✓

2. The CoNLL03 dataset

The OKE dataset contains annotations for PER, LOC, and ORG classes, whereas the CoNLL dataset contains annotations for one more class (i.e. MISC). The MISC tags in CoNLL dataset have been ignored for computing results. For FRE, only one relevant benchmark dataset has been found (available online for 25\$), and hence used for this work:

1. The TACRED dataset

2.3.1 Selection of Baselines

Following criteria has been used for selecting relevant existing systems as baselines for results comparison, as it is inline with the scope defined for this research.

1. NER systems which annotate NEs of general types, and RE systems which annotate relations of family types
2. The systems are publicly available
3. Systems reporting high F1 scores are selected

For NER, following baselines have been selected as they satisfy the criteria.

- LUKE [56]
- The UIUC (or Illinois) NER [29]
- The Stanford NER [27]
- FOX [31]
- ADEL[32]

From among the ML systems, LUKE has the highest reported F1, followed by UIUC, and then Stanford and FOX (see Table 2.3). From the hybrid systems, ADEL is the only one which is publicly available (see Table 2.4), while no system

from rule-based NERs is available (see Table 2.2). The baselines selected for RE using the criteria have been listed below.

- LUKE [56]
- SpanBERT [42]
- The System by Zhang et al. [2]
- The Stanford KBP annotator [55]

From among the ML systems, LUKE has the highest reported F1, followed by SpanBERT, and then Zhang et al.’s (see Table 2.6). From the hybrid systems, Stanford is the only one which is publicly available (see Table 2.7), while no system from rule-based systems is available (see Table 2.5).

2.3.2 Gap Analysis

Some available NER systems from the literature have been tested by giving some input texts, and it has been found that, despite having reported good F1 results, they are making many common mistakes, as mentioned in Table 1.1. Studying the relevant existing literature, it has been further realized that there is much need for improvement in FRE domain. The best results reported for English language for a reasonable vocabulary of family relations are by LUKE and SpanBERT systems, F1 score 72.7% and 70.8% respectively, on TACRED relation extraction dataset. But these RE systems make many in-correct annotations for family relations, as is evident from their decreased F1 scores on extraction of family relations (64.7% and 42.1% respectively).

Best results among existing systems have been reported by DL systems which are very time expensive. In order to extract NEs and relations to build a knowledge graph which can answer precise queries, a deterministic information extraction system for NER and FRE tasks is needed, having improved F1 score and less computation time.

Rule based systems are very light-weight and fast, need less memory and computation power. But designing good rules requires a lot of time, effort and expert knowledge on part of the designer. F1 results reported by rule based systems for general NER and RE are not very good. DL systems have reported best results among existing systems, but are very expensive in terms of computation time and memory usage. There is a need for devising a deterministic system which combines the good of both techniques, that is, it extracts entities and relations with good F1 and in less time.

As English grammar is generally not changing, so most of the times language patterns for entities and family relations can be identified in text. This key observation leads to the following hypothesis.

Hypothesis: Regular patterns for entities (persons, locations, organizations) and relations (family) exist in natural language texts.

By analyzing natural language texts from NER and FRE datasets, regular patterns for NEs and family relations have been identified (listed in Chapter 1) and the hypothesis is found true. So rules can be created against those patterns to recognize NEs and family relations, to get improved F1 and with less expense in terms of computation time. Moreover, as comprehensive datasets have been selected for identifying regular patterns, therefore rules formulated against the patterns should be dynamic and be applicable to any general English text.

Chapter 3

Datasets Selection and Preparation

3.1 Introduction

Correct extraction of information from text is a big challenge. This research aims to develop an IE system for improved extraction of NEs and family relations by focusing on correcting the annotation mistakes made by existing systems. The methodology adopted for developing the proposed system is Design Science research methodology as proposed by Dresch et al. [57], because Design Science seeks to consolidate knowledge about the design and development of solutions, to improve existing systems, solve problems and create new artifacts.

For the development of proposed system, selection and preparation of datasets is an important step of the methodology. The datasets (that is, corpus of running texts with marked instances of NEs and relations) are usually required for training the systems, and are essential for evaluating systems' performance. The datasets used for training should be correct and comprehensive, so that the rules formulated by identifying language patterns in the dataset are dynamic and applicable to any generic text. The datasets used for evaluation must also be correct and comprehensive so that systems are fairly and correctly evaluated and compared.

So before moving on to developing the proposed system, the datasets to be used for training and testing need to be prepared. This chapter describes in detail the corpora selection for this work, for NER and RE. The need for a criteria to assess dataset quality is argued, and the proposed criteria is explained. It further details how the proposed criteria has been applied to assess two existing NER datasets, the OKE and CoNLL03 datasets, and two RE datasets, the TACRED-F and Cust-FRE datasets; and the enhancements made to OKE and TACRED-F datasets. So this chapter mainly describes step 4 of the research methodology (see Fig. 1.6), that is, prepare datasets.

3.2 Datasets for NER

Some of the widely used NER gold standards include the CoNLL03 dataset [5], the MUC06 dataset [3] and more recently the OKE17 dataset [13]. The first two of the mentioned datasets, although more widely used, are homogeneous in the sense that they are collected from single newswire sources: CoNLL is constructed from Reuters Corpus [5] while MUC contains articles from Wall Street Journal [3]. The participants of MUC generally performed well on the dataset, many scoring above 90% F score [114]. CoNLL also has reported results over 90% F score, for example [29].

In contrast, the recent OKE dataset is heterogeneous in the sense that it has been collected from various Web sources like blogs, webpages, news and micro-posts etc., as the goal of the Open Knowledge Extraction (OKE) Challenge was to test the performance of knowledge extraction systems with respect to the Semantic Web. If a system performs well on the OKE dataset, it can be hoped that it will perform well on common domains like web, news, micro-posts etc ¹. The OKE dataset has been used by some recent NER systems as a gold standard benchmark to report their evaluation results, for instance [32]. The reported result of OKE challenge winner system on the task is 49% F1 score [13]. So OKE seems a more

¹<https://project-hobbit.eu/challenges/oke2017-challenge-eswc-2017/>

realistic and difficult to achieve benchmark dataset and if a system performs well on it, it can be hoped that the system will perform well on any general domain.

The NER datasets selected for this work are the CoNLL03 and OKE17 datasets as they are freely available online.

3.2.1 The CoNLL Dataset

The CoNLL dataset here refers to the English dataset used for CoNLL-2003 shared task. It contains newswire from Reuters Corpus: news articles between August 1996 and August 1997. The training and development data comprises news stories from August 1996, while the test set consists of stories from December 1996 [5]. The dataset contains texts from different domains, from Sports, from education, from politics, etc., as news can be about anything. The dataset is annotated with four types of NEs, PER, LOC, ORG and MISC. MISC class represents NEs that do not belong to any of the three class PER, LOC or ORG. Table 3.1 provides some statistics of the dataset.

TABLE 3.1: Some statistics of the CoNLL dataset

Dataset	Documents	Number of Named Entities				
		PER	LOC	ORG	MISC	Total
Train	946	6600	7140	6321	3438	23,499
Dev	216	1842	1837	1341	922	5,942
Test	231	1617	1668	1661	702	5,648
Total	1393	10059	10645	9323	5062	35089

The dataset contains one word per line. Each line has four fields separated by single spaces: the word, its part of speech tag, its chunk tag and its named entity tag. Words tagged with O are outside of named entities and the I-XXX tag is used for words inside a named entity of type XXX. An example sentence from CoNLL train dataset is given in Table 3.2. This sentence contains three NEs, a person Jimmy Thomson, a location Scotland and an organization Raith Rovers.

TABLE 3.2: An example sentence from CoNLL03 dataset

Word	PoS	Chunk	NER
Jimmy	NNP	I-NP	I-PER
Thomson	NNP	I-NP	I-PER
became	VBD	I-VP	O
Scotland	NNP	I-NP	I-LOC
's	POS	B-NP	O
first	JJ	I-NP	O
managerial	JJ	I-NP	O
casualty	NN	I-NP	O
of	IN	I-PP	O
the	DT	I-NP	O
season	NN	I-NP	O
on	IN	I-PP	O
Tuesday	NNP	I-NP	O
when	WRB	I-ADVP	O
he	PRP	I-NP	O
quit	VBD	I-VP	O
Raith	NNP	I-NP	I-ORG
Rovers	NNP	I-NP	I-ORG
,	,	O	O
bottom	NN	I-NP	O
of	IN	I-PP	O
the	DT	I-NP	O
premier	JJ	I-NP	O
division	NN	I-NP	O
.	.	O	O

3.2.2 The OKE Dataset

The OKE dataset here refers to the dataset used for Task1 of Open Knowledge Extraction (OKE) Challenge at the Extended Semantic Web Conference (ESWC) 2017. Table 3.3 provides some statistics of the dataset.

The OKE dataset consists of short passages collected from different, public Web pages (Wikipedia, News, and Blogs) about different topics. The data set is in turtle format and contains annotations for entities that belong to one of the three classes: person, location, and organization. But the entity type tags are not present in the data set, i.e. the information that from these three classes which class an entity belongs to, is not available in the dataset.

TABLE 3.3: Some statistics of the OKE dataset

Dataset	Documents	Number of Named Entities			
		PER	LOC	ORG	Total
Train	60	183	120	91	394
Eval	58	178	129	72	379
Total	118	361	249	163	773

The goal of the OKE Challenge was to test the performance of Knowledge Extraction Systems with respect to the Semantic Web. As most of the content on the Web consists of natural language text, hence a main challenge is to extract relevant knowledge from this content, and publish it as triples on the Semantic Web. The dataset therefore consists of texts collected from various public Web sources like Wikipedia, news, and blogs, etc. An example entity annotation entry from the OKE train data set is given in Figure 3.1.

<code><http://www.ontologydesignpatterns.org/data/oke-challenge-2017/task-1/sentence-59#char=74,78></code>	
<code>a</code>	<code>nif:RFC5147String , nif:String , nif:Phrase ;</code>
<code>nif:anchorOf</code>	<code>"OECD"^^xsd:string ;</code>
<code>nif:beginIndex</code>	<code>"74"^^xsd:nonNegativeInteger ;</code>
<code>nif:endIndex</code>	<code>"78"^^xsd:nonNegativeInteger ;</code>
<code>nif:referenceContext</code>	<code><http://www.ontologydesignpatterns.org/data/oke-challenge-2017/task-1/sentence-59#char=0,80> ;</code>
<code>itsrdf:taIdentRef</code>	<code><http://dbpedia.org/resource/Organisation_for_Economic_Cooperation_and_Development>.</code>

FIGURE 3.1: An entity annotation entry from OKE data set

The data set contains annotations of NEs (belonging to DBpedia ontology classes: Person, Place and Organisation) from input texts by their start and end indices, and a link for each NE to the relevant DBpedia resource for disambiguation purpose. As it can be seen in Figure 3.1, the entity type tags are omitted in the data set, as entity typing was not required in the challenge task. Consider this example input sentence from OKE train data set:

“Germany is a member of the United Nations, NATO, the G8, the G20, and the OECD.”

For this input, the data set contains the annotations specified in Table 3.4, in NIF format.

TABLE 3.4: Annotations in OKE dataset for example sentence

Identified NE	Generated URI	Begin Index	End Index
Germany	http://dbpedia.org/resource/Germany	0	7
United Nations	http://dbpedia.org/resource/United_Nations	27	41
NATO	http://dbpedia.org/resource/NATO	43	47
G8	http://dbpedia.org/resource/Group_of_Eight_(G8)	53	55
G20	http://dbpedia.org/resource/G-20_major_economics	61	64
OECD	http://dbpedia.org/resource/Organisation_for_Economic_Co-operation_and_Development	74	78

3.3 Datasets for Relation Extraction

Many existing works reported results of FRE on un-published datasets, for example [44–49, 105], which are not available to other researchers for comparison of results and improvement of systems. The only published dataset to the best of our knowledge, having a reasonable size of vocabulary that can be used for FRE purpose is the TACRED dataset. A dataset has therefore been constructed for evaluating FRE systems, the CustFRE dataset. Next these two datasets are described.

3.3.1 The TACRED-F Dataset

The TAC Relation Extraction Dataset (TACRED) [2] is a very large and challenging RE dataset [42], built over English newswire and web text used in the

yearly Text Analysis Conference’s (TAC) Knowledge Base Population (KBP) slot filling evaluations during the period 2009-2014, and annotated through crowdsourcing via Mechanical Turk. The yearly TAC KBP slot filling tasks, starting from 2009 [115] is the most widely-known effort to evaluate knowledge base population systems [2]. TACRED contains examples of 41 relations of organizations and persons, such as, `org:founded_by`, `per:employee_of`, `per:schools_attended`, etc. and `no_relation` to represent that none of the 41 relations exists between subject and object. The `no_relation` examples constitute 79.5% of the dataset, which are demonstrated to be crucial for training high-precision RE models, as they ensure that models trained on the dataset are not biased towards predicting false positives on real world text. The average sentence length of dataset is 36.4, reflecting the complexity of contexts in which relations occur in real-world text. It is publicly available through the LDC for a small fee. Dataset is reported to have removed duplicates and examples where subject and object overlap, and is estimated to be 93.3% accurate. The dataset relation types include following 5 family relation types:

`per:other_family`, `per:parents`, `per:siblings`, `per:spouse`, `per:children`

The annotations of these family relations can be extracted from this dataset. This dataset subset, henceforth referred as TACRED-F, can be used for training, evaluating and comparing family relation extraction systems. Here F shows that it is TACRED’s family relations subset.

TACRED contains separate files for training, development and testing purpose, in conll and json formats. The examples in dataset are marked with subject and object, and the RE task for systems is to recognize that out of the 42 relations, which relation exists between marked subject and object entities. The dataset is also annotated with Stanford’s part of speech (`pos`), named entity recognition (`ner`) and dependency relation tags. Figure 3.2 gives an illustration example from the training dataset in conll format.

#	id=61b3a9d0d19bc98ba545	docid=LTW_ENG_20070103.0058.LDC2009T13	reln=per:children
1	Kollek	SUBJECT PERSON	NP PERSON nsubjpass 3
2	is	VBZ 0	auxpass 3
3	survived	VBN 0	ROOT 0
4	by	IN 0	case 6
5	his	PRP\$ 0	nmod:poss 6
6	widow	NN 0	nmod 3
7	,	,	punct 6
8	Tamar	NNP PERSON	appos 6
9	,	,	punct 6
10	a	DT 0	det 11
11	son	NN 0	appos 6
12	,	,	punct 11
13	Amos	OBJECT PERSON	NNP PERSON appos 11
14	,	,	punct 11
15	and	CC 0	cc 11
16	a	DT 0	det 17
17	daughter	NN 0	conj 11
18	,	,	punct 17
19	Osnat	NNP PERSON	appos 17
20	.	.	punct 3

FIGURE 3.2: A sample example from TACRED dataset

As it can be seen from the example, the dataset in conll format has ten columns which contain; the token number, the token, ‘SUBJECT’ for subject’s tokens and – for others, NER tag of subject and – for others, ‘OBJECT’ for object’s tokens and – for others, NER tag of object and – for others, part of speech tag of token, NER tag of token, dependency tag of token and dependency head of token.

Python script has been written that extracts only family relations’ examples from the full TACRED dataset i.e. all examples of relations: per:spouse, per:sibling, per:parent, per:other_family and per:children. Negative examples have also been extracted from the dataset, that is, examples where no family relation exists between subject and object persons. Table 3.5 summarizes the number of examples of each type of relation in TACRED-F dataset.

3.3.2 The CustFRE Dataset

The TACRED dataset is not found comprehensive in the sense that usually not all family relations in a text are annotated in the dataset. For example, for the text “Kollek is survived by his widow Tamar, son Amos and daughter Osnat.”, only

TABLE 3.5: Relation distribution of TACRED-F dataset

Relation Class	Number of Examples			
	Total	Training	Development	Testing
per:children	347	211	99	37
per:other_family	319	179	80	60
per:parents	296	152	56	88
per:siblings	250	165	30	55
per:spouse	483	258	159	66
Total positive examples	1,695	965	424	306
not_known (negative examples)	1,101	555	262	284
Total examples	2,796	1,520	686	590

three family relations are annotated in the dataset, whereas eighteen relations can be inferred from this text. The shortcomings in TACRED, the lack of available family relation datasets with a reasonable vocabulary, and the desire to make a fair evaluation of family relation extraction systems on a dataset that no system is trained on, are the motivation for creating a new comprehensive FRE evaluation dataset.

The CustFRE dataset [116] has the quality that for any text, it has been annotated with all the possible family relations. Annotation of all possible family relations in dataset is important to get true picture of systems' performance on the dataset. It is therefore made sure that in CustFRE dataset, all possible relations are annotated. For each input sentence all possible permutations of persons in the sentence have been listed in an excel file. A team of Natural Language Processing researchers then annotated each sentence permutation with the family relation between those persons (or person pronouns) from the sentence. If a text contains n persons, the number of possible relations between these persons is

$$P(n, r) = \frac{n!}{(n - r)!} \quad (3.1)$$

Where $r = 2$, as this work is concerned with binary relations, i.e. relations between

two persons. Permutations are being counted because the direction of relation is important, i.e. the relation of X to Y might be different from the relation of Y to X . As an example, consider the following text.

Mohib and Aamina tied the knot back in 2005 and also have a baby daughter, Meissa Mirza, who was born in August 2015.

This text contains three persons, *Mohib*, *Aamina* and *Meissa Mirza* and therefore six possible relations, as $P(3, 2) = 6$, as given below.

(Mohib, per:spouse, Aamina)

(Mohib, per:children, Meissa Mirza)

(Aamina, per:spouse, Mohib)

(Aamina, per:children, Meissa Mirza)

(Meissa Mirza, per:parents, Mohib)

(Meissa Mirza, per:parents, Aamina)

So all the six possible relations between these persons have been annotated in the dataset. After data annotation was completed, an expert, working in the fields of information extraction and NLP for over 30 years, verified the dataset (manually going through every single of 2,716 annotations) and checked that the dataset is correctly and completely annotated and does not contain any erroneous annotations.

The pre-defined vocabulary used for this dataset contains the family relations used in TAC KBP and TACRED, that is, the five family relations, per:spouse, per:children, per:parents, per:siblings, and per:other_family, and not_known if none of the five family relations exists between two persons. The sentences for the dataset have been collected from three type of online sources, short stories², Wikipedia³ articles and family news from news and magazines websites⁴. A total

²Taken from <https://americanliterature.com/100-great-short-stories>

³<http://www.wikipedia.org>

⁴<https://dunyanews.tv/>, <https://www.bbc.com/news>, <https://www.theguardian.com/>, <https://www.timesnews.net/>

of 248 sentences have been collected and each sentence contains two or more persons. The dataset contains a total of 2,716 annotations. Average sentence length of CustFRE, in terms of number of tokens, is 35. Number of examples of each type of relation in the dataset are given in Table 3.6.

TABLE 3.6: Relation distribution of CustFRE evaluation dataset

Relation Class	Number of Examples	Percentage of Total
per:children	347	13%
per:other_family	302	11%
per:parents	347	13%
per:siblings	282	10%
per:spouse	264	10%
not_known	1,174	43%
Total	2,716	100.0%

3.4 Need for an Explicit Method to Assess Datasets

Jha et al [117] observe that although using gold standard datasets for evaluating IE systems has greatly spurred the development of better and better IE systems, but there still are some shortcomings. Datasets do not share a common set of rules pertaining to what is to be annotated. Moreover, most of the gold standards have not been checked by other researchers after they have been published and hence commonly contain mistakes. Checking the quality of datasets is therefore important for progress in the IE field.

Additionally, while working with the IE datasets, several annotation mistakes and in-consistencies have been noticed. Knowledge extraction is an extremely important task in the field of Information Retrieval (IR). If knowledge is not correctly extracted, then it is not useful for IR systems because it will generate wrong results. Since systems are generally trained on some dataset, the dataset

must be correct so that the systems may be trained correctly. If the dataset is erroneous, the systems trained on the dataset would learn incorrect behavior and would extract incorrect knowledge. Besides, the quality of dataset also directly impacts the quality of evaluation performed using the dataset. So a criteria is needed, against which the quality of existing benchmark datasets could be judged. But no such criteria could be found in existing literature. Therefore, a criteria is proposed in this work.

3.5 Criteria to Assess NER and FRE Evaluation Datasets

To decide the criteria for assessing NER and FRE evaluation datasets, focus group has been conducted with domain experts following the guideline provided by [118]. Focus group is a qualitative research strategy in which opinions on an issue are explored through free and open discussion between members of a group and the researcher. Focus groups are facilitated group discussions in which the researcher raises issues or asks questions that stimulate discussion among members of the group. The researcher selects a group of people whom he thinks are best equipped to discuss what he wants to explore. Eight to ten people is the suggested ideal size for such group. Focus group starts with broad discussion topic or question posed by the researcher, providing a broad frame for discussion among the group members. Specific discussion points emerge as part of the discussion. The group extensively discusses the issue and arrives an agreement. A focus group is a good choice when seeking direction, explanation, or in-depth dialogue. As a criteria for assessing NER and FRE evaluation datasets is being sought, therefore focus group with domain experts has been considered a good choice.

This focus group study aims to explore the properties that NER and FRE benchmark datasets ought to have. Eight domain experts having vast experience in

Information Extraction, Natural Language Processing, and Computational Linguistics were selected for the group. Profiles of selected experts are shown in Table 3.7.

TABLE 3.7: Expert Profiles for Focus Group

Sr.	Expert Profile	Experts
1.	Professor (Information Extraction and NLP)	1
2.	Associate Professor (NLP and Computational Linguistics)	1
3.	Assistant Professor (NLP and Computational Linguistics)	2
4.	PhD Students (Information Extraction and NLP)	2
5.	Lecturers (Mathematics and Statistics)	2

A prepared script, given in Fig. 3.3, was used to welcome participants, remind them of the purpose of the group and to set ground rules.

The focus group was structured around a set of carefully predetermined questions, given in Figure 3.4, the discussion was free-flowing. Participant comments stimulated the thinking and sharing of other participants.

The group went on to discuss the characteristics they thought were important for datasets. It has been strived to gather as much information as possible, and stopped when no new information was emerging.

Five sessions of 40 minute group discussion were conducted. The sessions were conducted in a round table conference room, with facilities of white board, laptops, multimedia projector and internet access.

Three main themes emerged during the group discussions, forming the minimum criteria for assessing NER and FRE datasets. In the last session, the written criteria that emerged during discussions, was given back to the group for correction, verification and confirmation. The criteria is now stated.

For a benchmark dataset, three major properties must hold:

1. Accuracy

FOCUS GROUP INTRODUCTION

WELCOME
Thank you for agreeing to be part of the focus group. We appreciate your willingness to participate.

INTRODUCTIONS
Moderator, assistant moderator
NER and FRE tasks

PURPOSE OF FOCUS GROUP
The reason we are having these focus group discussions is to explore the properties that NER and FRE benchmark datasets ought to have. We need your input and want you to share your honest and open thoughts with us.

GROUND RULES

1. *WE WANT YOU TO DO THE TALKING*
We would like everyone to participate.
We may call on you if we haven't heard from you in a while.
2. *THERE ARE NO RIGHT OR WRONG ANSWERS*
Every person's experiences and opinions are important.
Speak up whether you agree or disagree.
We want to hear a wide range of opinions.
3. *WHAT IS SAID IN THIS ROOM STAYS HERE*
We want you to feel comfortable sharing.
4. *WE WILL BE TAKING WRITTEN NOTES OF THE GROUP*
We want to capture everything you say.
We don't identify anyone by name in our report.
You will remain anonymous.

FIGURE 3.3: The focus group introduction script

2. Completeness
3. Appropriate Size

3.5.1 Accuracy

Accuracy is the degree to which the annotations in the dataset match the valid annotations. A valid annotation must be formally defined for the dataset, and is given in Fig. 3.5 for NER datasets, and in Fig. 3.6 for FRE datasets.

Let A_D represent the accuracy of dataset D , $N_{correct}$ be the number of correct annotations in the dataset and N_{total} be the total number of annotations in the

Questions for Focus Group

Engagement Questions:

1. Which NER/RE datasets have you worked with?
2. What is your observation about the quality of datasets?

Exploration Questions:

3. What are the minimum properties that quality NER/FRE benchmark datasets must satisfy?
4. How may the properties be measured?
5. How much of a property in a dataset may be considered acceptable?

Exit Question:

6. Is there anything else you would like to add about how may the quality of datasets be assessed?

FIGURE 3.4: The questions for stimulating focus group discussion

dataset. Then the first property, that is accuracy, can be measured quantitatively, as given in 3.2.

$$A_D = \frac{N_{correct}}{N_{total}} \times 100 \quad (3.2)$$

Due to the inherent ambiguities of natural language, a 100% accurate dataset might not be possible, because at times an annotation considered correct by one annotator might be considered in-correct by another annotator. A dataset D which is at least 90% correct, can be considered acceptable, that is,

$$A_D > 90\% \quad (3.3)$$

3.5.2 Completeness

The second quality, that is completeness, refers to the comprehensiveness or wholeness of the data. All possible annotations should be annotated in the dataset. For NER datasets, all instances of person, location and organization in data should be annotated. For FRE datasets, all possible combinations of person pairs in the text should be annotated, either with a family relation or with not_known.

A **valid named entity annotation**, is a named instance of one of the three classes, person, location or organization.

1. The Person class represents people. Something is a Person if it is a person, whether they are alive, dead, real, or imaginary.
2. The Location class represents immobile things or places.
3. The Organization class represents organizations such as a school, NGO, corporation, club, etc. An organization is an entity comprising multiple people, such as an institution or an association, which has a particular purpose.

The above definitions of the classes have been taken from DBpedia/Wikipedia.

Two assumptions are also stated, which are generally assumed but at times are not clear in minds of some annotators:

1. Nested entities are not considered, only highest level NE from the type set is considered. This assumption has been taken from the CoNLL guideline. For example, "Kenneth Branagh Theatre Company West End" is one NE of type organization, the nested person entity "Kenneth Branagh" is not annotated. When the outer entity is not in type set, then consider the inner one, i.e. consider the highest/toppest level NE from the type set. For example: "The China Wars" is a movie name (movies are not in the class set), so "China" is tagged as LOC as this is the highest level NE from the type set.
2. Full name followed by its abbreviation are two occurrences of NEs e.g. the text "National Endowment for the Arts (NEA)" has two NEs "National Endowment for the Arts" and "NEA".

FIGURE 3.5: Definition of Valid Named Entity Annotation

Let C_D represent the completeness percentage of dataset D , that is, from the total number of possible annotations on D , what percentage of annotations is actually found in D . C_D is calculated as:

$$C_D = \frac{N_{total}}{N_{possible}} \times 100 \quad (3.4)$$

Where N_{total} is the actual number of annotations in D , and $N_{possible}$ is the number of possible annotations on D .

For FRE datasets, if D has m distinct input texts, T_1, T_2, \dots, T_m , and each text T_i has n_i persons, then the number of possible annotations on the dataset D can

A **valid relation annotation**, r , from a person subject s to a person object o , represented as (s, r, o) , is one of the following relations, provided the relevant information could be inferred solely from the input text, and not from an annotator's prior knowledge or other knowledge sources:

1. Children, if o is a child / step-child / adopted child (son or daughter) of s
2. Parents, if o is a parent / step-parent / adoptive parent (mother or father) of s
3. Siblings, if s and o have one or both parents common
4. Spouse, if s and o are or were spouses (husband, wife, partner, etc.)
5. Other Family, if s and o are related by any family relation other than the four described above (cousin, uncle, grandfather, sister-in-law, etc.)
6. Not Known, if no family relation can be inferred between s and o

The above definitions have been adopted from descriptions of relevant terms from Relationship vocabulary and from Wikidata properties.

FIGURE 3.6: Definition of Valid Family Relation Annotation

be calculated as:

$$N_{possible} = \sum_{i=1}^m P(n_i, 2) \quad (3.5)$$

Here $P(n_i, 2)$ is the number of 2-permutations of n_i persons and can be calculated using the permutations formula:

$$P(n_i, 2) = \frac{n_i!}{(n_i - 2)!} \quad (3.6)$$

Ideally, C_D should be 100%, but a number close to 100% can be considered acceptable, that is,

$$C_D \approx 100\% \quad (3.7)$$

3.5.3 Appropriate Size

The dataset should have an appropriate size, which is a good representation of the population. When the evaluation dataset is a good representation of the population, only then can the results be generalized to the population and hence it can be expected that systems which perform well on this dataset will also perform

well in real situations. The population where NER and FRE systems are to be used comprises general texts. Because the population is very large, Cochran's formula [119] is the most appropriate to use to determine sample size [120]. The Cochran formula is considered especially appropriate in situations with large populations and allows to calculate an ideal sample size given a desired level of precision, desired confidence level, and an estimated proportion of the attribute present in the population as:

$$n_o = \frac{Z^2 pq}{e^2} \quad (3.8)$$

Where n_o is the necessary sample size, the z - value is found in a Z table, p is the (estimated) proportion of the population which has the attribute in question, q is $1 - p$, and e is the desired level of precision or the margin of error. A 95% confidence level and $\pm 5\%$ precision has been desired for this work. A 95% confidence level gives Z value of 1.96, per the normal tables.

Population variance, p , can be estimated by conducting a pilot study [121].

For NER datasets, a pilot study was carried out on five randomly picked up news articles from a news website⁵. The articles altogether consist of 122 sentences, out of which 110 sentences have NEs. So the proportion of population which has NEs is estimated to 110/122, which is 0.9. So the required sample size is:

$$n_o = \frac{(1.96)^2(0.1)(0.9)}{(0.05)^2} = 136.3 \approx 137 \quad (3.9)$$

A sample of 137 sentences gives the dataset that is representative of the whole population, with a 95% confidence level and $\pm 5\%$ margin of error. Therefore an NER evaluation dataset must have annotations for at least 137 sentences. Let $S_{NER,D}$ represent the size of an NER dataset D, then:

$$S_{NER,D} > 137 \text{sentences} \quad (3.10)$$

For FRE datasets, the target population contains all sentences having at least two persons. A pilot study was conducted on five randomly picked up person articles

⁵<https://dunyanews.tv/>

from Wikipedia. The articles altogether contain 110 sentences, of which 48 have at least two persons (the target population). Of these 48, the sentences having family relations are 8. So the proportion of two person sentences which have any family relation is estimated to $p = 8/48 = 0.17$. So the required sample size is:

$$n_o = \frac{(1.96)^2(0.17)(0.83)}{(0.05)^2} = 213.42 \approx 214 \quad (3.11)$$

So a sample of 214 sentences from the target population (sentences having at least two persons) gives the dataset that is representative of the whole population, with a 95% confidence level and $\pm 5\%$ margin of error. Therefore an FRE evaluation dataset must have annotations for at least 214 sentences. Let S_{FRE_D} represent the size of an FRE dataset D, then:

$$S_{FRE_D} > 214 \text{sentences} \quad (3.12)$$

Next, the NER and FRE datasets selected for this research are assessed against the devised criteria.

3.6 Assessment of Datasets

The NER and FRE evaluation datasets selected for this work, i.e. OKE, CoNLL, TACRED-F and CustFRE datasets, have been carefully scrutinized against the formulated criteria. For calculating accuracy and completeness of datasets, each annotation in the datasets was manually checked by the authors. Since CoNLL is a big dataset (containing 3,453 sentences in the test set), a sample was extracted for the purpose of its detailed manual analysis. Of the 231 documents in the CoNLL test dataset, 12 documents (i.e. 5% of the dataset) were used for the purpose of computing accuracy and completeness values. For CustFRE dataset, since it is made by ourselves, so it has been checked by another independent annotator for accuracy and completion. The findings are summarized in Table 3.8 and detailed next.

TABLE 3.8: Summary of evaluation datasets assessment against devised criteria

Category	Dataset	Accuracy	Completeness	Size
NER	CoNLL	96.1%	98.3%	3453
	OKE	83.5%	95.3%	176
FRE	TACRED-F	73.6%	20.8%	214
	CustFRE	99.5%	98.9%	248

3.6.1 Accuracy

When the test datasets are evaluated with respect to accuracy, 220 of the 229 NEs in CoNLL sample are found correct. The estimated accuracy of CoNLL is thus,

$$A_{CoNLL} = \frac{220}{229} \times 100 = 96.1\% \quad (3.13)$$

But many mistakes are encountered in the OKE dataset. 318 of the 381 NEs are found correct. So the accuracy of OKE is,

$$A_{OKE} = \frac{318}{381} \times 100 = 83.5\% \quad (3.14)$$

From NER datasets, the accuracy of CoNLL is found acceptable, but that of OKE dataset is not found up to the mark.

The TACRED-F test set has 590 annotations, out of which 434 are found correct. The accuracy of the dataset is thus,

$$A_{TACRED-F} = \frac{434}{590} \times 100 = 73.6\% \quad (3.15)$$

The accuracy of TACRED-F is extremely low. Since the dataset has been annotated by Mechanical Turk crowd annotation [2], it seems like, many times the annotators are not clear about the annotation criteria and are annotating based on their subjective views. As an example, consider this text from the dataset, “In addition to his(subj) wife, Meskill is survived by two daughters, Eileen Gallup of New Britain and Maureen Heneghan(obj) of ...” A *no_relation* is annotated in

the dataset, whereas it is clear from the text that a *per:children* relation exists between subject (*his*) and object (*Maureen Heneghan*).

The CustFRE dataset has 2716 annotations, of which 2702 are marked correct by the annotator. The accuracy of CustFRE is thus,

$$A_{CustFRE} = \frac{2702}{2716} \times 100 = 99.5\% \quad (3.16)$$

The accuracy of CustFRE dataset is found up to the mark.

3.6.2 Completeness

With respect to completeness, 19 such cases were encountered in the OKE test set where it was a named entity but the annotation was missing in the dataset. So,

$$C_{OKE} = \frac{381}{400} \times 100 = 95.3\% \quad (3.17)$$

The CoNLL sample was found to have 4 such cases where an NE annotation was missing in the dataset.

$$C_{CoNLL} = \frac{229}{233} \times 100 = 98.3\% \quad (3.18)$$

For TACRED-F test set, it is found that most of the times only the most apparent family relation is annotated in the dataset. For each of the 214 distinct input text T_i in the dataset, the number of persons in it were manually counted, i.e. n_i , then the number of 2-permutations of n_i persons were calculated using equations 3.5 and 3.6. These totaled to 2,842. Whereas the actual number of annotations in the dataset is 590. So the completeness percentage of TACRED-F is,

$$C_{TACRED-F} = \frac{590}{2842} \times 100 = 20.8\% \quad (3.19)$$

For CustFRE dataset, the annotator reported 30 cases of missed relation annotations. So the completeness of the dataset is,

$$C_{CustFRE} = \frac{2716}{2746} \times 100 = 98.9\% \quad (3.20)$$

So three of the four datasets are found reasonably complete. The completeness percentage of TACRED-F is found extremely low and needs improvement.

3.6.3 Size

With respect to size, all four datasets were found satisfactory. The NER test datasets, OKE and CoNLL have 176 and 3,453 sentences respectively, while the FRE evaluation datasets, TACRED-F and CustFRE have 214 and 248 sentences, respectively. As explained in previous section, NER evaluation datasets should have at least 137 sentences (refer equation 3.10), while FRE evaluation datasets should have at least 214 sentences (refer equation 3.12), so all four datasets have a reasonable size.

3.7 Improving the Datasets

Based on the devised criteria, the CoNLL and CustFRE datasets are found up to the mark, but several shortcomings have been encountered in the OKE and TACRED-F datasets. These shortcomings have been removed and these two datasets have been correctly and completely annotated.

3.7.1 The Improved OKE Dataset

The OKE dataset is collected from heterogeneous web sources like news, blogs, Web pages etc. and is therefore a valuable resource for NLP and Semantic Web researchers. Some enhancements have therefore been applied to the dataset, as described below, to make this valuable resource even more useful.

The OKE dataset has some mistakes and in-consistencies in annotations, which result in the information extraction systems not being evaluated correctly. Some example improper annotations in the dataset along with the corrections made are given in Table 3.9. Notation ‘1’ in the column cell means annotation is done, YES, and ‘0’ means it is not annotated, No. There was a need to remove the errors and also to add entity type tags in the OKE dataset. Classification of recognized NEs is usually an adjacent task to the NER task, jointly called the Named Entity Recognition and Classification (NERC) task. Since the OKE dataset does not contain entity type tags, therefore the OKE dataset has been annotated with entity type tags as well.

The data set has thus been enhanced as follows:

1. The data set contains several mistakes and inconsistencies, as mentioned in equation 3.14. These mistakes have been corrected and the annotations are made consistent. Some examples are given in Table 3.9.
2. The OKE data set does not contain entity type tags, therefore the OKE data set has been annotated with entity type tags.
3. Annotations of some NEs are not found in the dataset, as mentioned earlier in equation 3.17, so these missing annotations have been added to complete the dataset.

Some examples from the dataset now follow to elaborate the above points. Consider for instance entities such as “Irish”, “Korean”, and “American” which are annotated in the dataset, but these entities are actually nationalities and do not belong to any of the three classes: Person, Location, and Organization. Since the dataset is said to annotate instances of Person, Location, and Organization classes, therefore such annotations have been removed from the dataset. Likewise, certain entities which belong to Person, Location, or Organization class are not annotated in the dataset. For example, “Ministry of Defense”, “Korean Air” and “Persian army” are entities of type Organization but are not annotated in the dataset,

TABLE 3.9: Examples of corrections made to the OKE dataset

Named Entity	Previous Annotation	Corrected Annotation	Comments
American	1	0	<i>American</i> is type nationality, not PER, ORG or LOC. Therefore, it is removed from the dataset.
Korean	1	ORG:Korean Air	<i>Korean</i> is nationality. But the text actually has <i>Korean Air</i> , which is an organization.
Ministry of Defence	0	1:ORG	<i>Ministry of Defence</i> is an organization, but was not tagged in data set.
Yonhap news agency	1	ORG:Yonhap	<i>Yonhap</i> is name of organization, not <i>Yonhap news agency</i> .
Russia	0	1:LOC	<i>Russia</i> is a location, but was not annotated.
Rose's	1	PER:Rose	's is not part of the person name <i>Rose</i> .
King Koopa	1	0	<i>King Koopa</i> is a turtle-like fictional character and not a person, location or organization.
Persian King Xerxes	1	PER:King Xerxes	<i>Persian</i> is not part of the person name.
paraplegic Marine Jake Sully	1	PER:Marine Jake Sully	<i>paraplegic</i> is not part of the person name.
Santa	0	PER:Santa	<i>Santa</i> or <i>Santa Claus</i> is a human fictional character.
U.S.	0	1:LOC	<i>U.S.</i> is a location named entity.
Joker	0	1:PER	<i>Joker</i> is a person fictional character.
Persian army	0	1:ORG	<i>Persian army</i> is name of an organization.
Florence, Italy	1	LOC:Florence, LOC:Italy	This entity has been broken down into two location entities, <i>Florence</i> and <i>Italy</i>
FIFA	0	1:ORG	<i>FIFA</i> is acronym of an organization.

“U.S.” and “Iraq” are entities of type Location but are missed in the dataset, so such overlooked annotations have been supplemented to the OKE dataset.

Furthermore, several times, titles/words/characters describing an entity are also part of the entity’s surface form in the dataset, while other times they are not. For illustration purpose, reflect upon these entities from the dataset: “Rose’s”, “legendary cryptanalyst Alan Turing”, “Persian King Xerxes”, “Britain’s” and “paraplegic Marine Jake Sully”. In these annotations, the entities: *Rose*, *Alan Turing*, *King Xerxes*, *Britain* and *Marine Jake Sully* are augmented with additional words/characters which are actually not part of the entity, that is, the words/characters: *’s*, *legendary cryptanalyst*, *Persian*, *’s* and *paraplegic*, respectively. To the extent of OKE challenge, it did not matter because the challenge considered weak annotation for evaluation, where overlapping of entity boundaries is sufficient for correctness and exact boundary match is not required. But as NERC tasks (including CoNLL 2003 shared task) generally use strong annotation for evaluation which requires exact match of entity boundaries, for that reason the annotation surface forms have also been refined and any supplementary words which are not part of entity have been removed from them.

There are also some inconsistencies in the dataset. For instance, in text: “was born in Greenwich Village, Manhattan, New York City”, *Greenwich Village*, *Manhattan*, *New York City* is annotated as one entity. Whereas in text: “born in 1647 in Frankfurt, Germany”, *Frankfurt* and *Germany* are annotated as two separate entities. To make the dataset consistent, any unified Location entities have been broken into separate entities. The updated dataset is made available online at <https://github.com/Raabia-Asif/CustKnowledgeExtractor>.

3.7.2 The Improved TACRED-F Dataset

The TACRED-F dataset has been enhanced by correcting the annotation mistakes found in it (see equation 3.15) and by adding the relation annotations which are missing in it (see equation 3.19). Some example incorrect annotations in TACRED-F along with the corrections made are presented in Table 3.10.

TABLE 3.10: Examples of Corrections made to TACRED-F Dataset

Sentence (with Subject and Object in bold)	Relation Marked	Corrected Relation	Comments
Kelly arrives at People’s Revolution and talks to Whitney and Roxy about the shoot, which went well except for the “Brazilian bore” model.	per:spouse	not_known	No family relation is evident from this text
I noticed that everytime Jake talks about why he likes Vienna, he always says it’s because of how good she makes him feel.	per:other_family	not_known	They both refer to same person
In addition to his (subj) wife, Meskill is survived by two daughters, Eileen Gallup of New Britain and Maureen Heneghan (obj) of Haddon Heights, . . .	no_relation	per:children	His refers to Meskill and Maureen is his daughter
Along with Washburn and others, he performed some tests on Mount McKinley.	per:other_family	not_known	No family relation is evident from this text
Davis told AP the items were among many of the space-related heirlooms her husband left her when he died in 1986.	no_relation	per:spouse	He is her husband
In addition to his son Joel, Buchwald is survived by daughters Jennifer Buchwald of Roxbury, Mass.; Connie Buchwald Marks of Culpeper, Va.; sisters Edith Jaffe, of Bellevue, Wash., and Doris Kahme , of Delray Beach.	per:other_family	per:siblings	Because his sister Doris Kahme is mentioned in text

3.7.3 Evaluating the Improved Datasets

Next, the two enhanced datasets, OKE and TACRED-F, have been evaluated by two independent evaluators against the quality assessment criteria, to ensure that

the datasets have indeed been corrected and completed.

Evaluator demographics are being provided following the guideline of Bender and Friedman [122]. The evaluators are bilingual (Urdu/English) researchers working in the area of natural language processing at Capital University of Science & Technology, Islamabad, Pakistan, having ages 30+ and 40+, one is male and other is female. Based on income levels, the evaluators represent upper and middle class.

The evaluators evaluated the datasets independently in the light of proposed criteria and definitions of valid annotations. Inter-rater agreement between the evaluators is calculated in terms of Cohen's Kappa coefficient (κ) [123]. κ is a statistic commonly used for testing inter-rater agreement. Its score can range from -1 to $+1$, where 0 represents the amount of agreement that can be expected from random chance and 1 represents perfect agreement between the raters. Calculation of κ is performed using the method and formulas described by Cohen. The degree of agreement between the evaluators is found to be 0.97 for OKE and 0.96 for TACRED-F. Results are interpreted as suggested in the literature [124]. As a score above 0.9 represents that almost perfect agreement exists between annotators, thus it can be inferred that the datasets have been corrected and completed according to the criteria.

3.8 Summary

This chapter elaborated; the datasets selected for this work (OKE, CoNLL, Cust-FRE and TACRED-F), the criteria proposed for assessing the quality of evaluation datasets (includes accuracy, completeness and appropriate size), the shortcomings found in OKE and TACRED-F datasets (contain in-correct and missing annotations), and the preparation of enhanced OKE and TACRED-F datasets by overcoming the shortcomings found. The enhanced versions of datasets prepared will be used for evaluation of systems. Now the datasets are ready, next two chapters describe creating the proposed system for improved IE.

Chapter 4

CustNER - An Improved System for Named Entity Recognition

4.1 Introduction

This chapter describes the system module CustNER, proposed to meet RO1, to recognize NEs of PER, LOC and ORG types by overcoming the limitations of existing NER systems as outlined in section 1.3, by devising rules against the following regular patterns of NEs missed by existing systems:

1. Entities containing nationalities
2. Entities having corresponding resources in DBpedia
3. Acronym NEs
4. Re-occurrences of already identified NEs

Instead of starting from the scratch, outputs of existing systems have been utilized. The patterns missed by existing systems on OKE training dataset have been carefully analyzed and rules have been developed to recognize each type of pattern missed by existing systems. Rule development has been a complex process as making rule for one pattern disturbs recognitions by other rules, and requires revisiting and fine tuning the developed rules over and again. CustNER

also links NEs to corresponding DBpedia resources, and for person entities it also identifies the gender of person. The proposed system is available at the link: <https://github.com/Raabia-Asif/CustKnowledgeExtractor>

4.2 The System CustNER

CustNER does not solve the NER problem from scratch, rather it utilizes annotations made by existing systems and uses rules to focus on recognizing NEs missed by existing systems. So it first gets the input text annotated using DBpedia Spotlight [125], Stanford NER and Illinois NER. DBpedia Spotlight is a tool that annotates mentions of DBpedia resources in text. For getting text annotated using Spotlight, its confidence parameter is set to 0, so that maximum number of annotations get mentioned from which the rules could filter the correct ones. The entities identified by Spotlight and NERs are first pre-processed to remove apparent false positives, an algorithm then selects, refines and improves on these entities based on certain rules, and assistance from an external knowledge base, DBpedia [126]. A block diagram of the system is presented in Figure 4.1. The modules of CustNER are explained in coming sections.

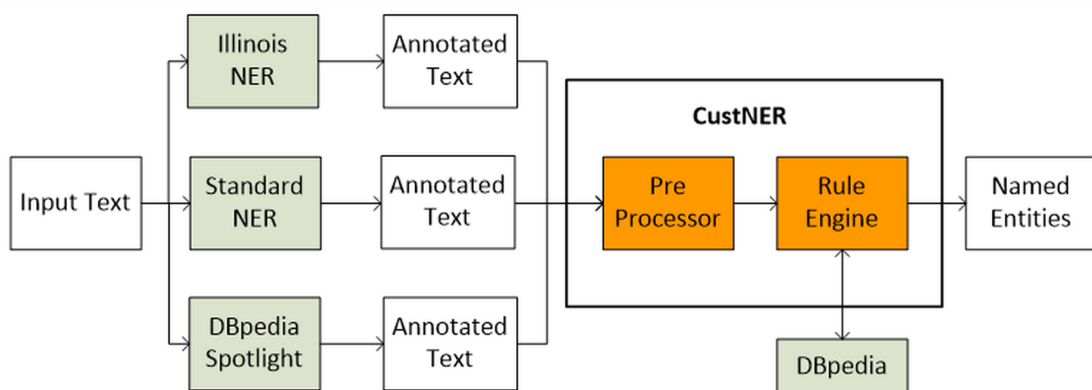


FIGURE 4.1: The System CustNER

4.2.1 Pre-Processor

Pre-processing has been done as follows:

1. The lists of entities annotated by the annotators contain some apparent false positives like he, his, goes, the etc., which need to be removed. If an annotation satisfies any of the following conditions, then it cannot be an NE and therefore be removed from the list:
 - (a) Is a single word, which is not a proper noun, or which is a title
 - (b) Starts with a conjunction or a verb or a preposition
 - (c) Has no word starting with a capital letter
 - (d) Contains \$ sign
2. If all alphabets in a sentence are capital, the first letter of each word in the sentence remains capital while all other alphabets are changed to lower case.
3. Entities identified by Stanford NER and Illinois NER are given types by these annotators. For entities identified by DBpedia Spotlight, their types are queried from DBpedia and they are typed accordingly.
4. The first word from person entities is removed if it is a title.

4.2.2 Rule Engine

Let E_P be the list of potential entities as:

$$E_P = A_{N1} \cup A_{N2} \cup A_S \quad (4.1)$$

Here A_{N1} , A_{N2} and A_S are the pre-processed lists of entities identified by Stanford NER, Illinois NER and DBpedia Spotlight respectively. Entities marked as city, country, state or province by annotators are considered locations. A new list, E_C , is then constructed from E_P , containing entities recognized by CustNER, by applying the rules in order, as explained in the coming sections. The first four

rules populate the list E_C from E_P , while the last two rules further enhance the list E_C by adding acronyms and re-occurrences of NEs in E_C .

Let e be an entity in E_P . Rule 1 and 2 add the NEs identified by NER annotators to E_C , while rules 3 to 6 annotate the NEs missed by NER annotators. From missed NE patterns identified in Introduction section, rule 3 handles pattern 1 (entities containing nationalities), rule 4 extracts missed pattern 2 (entities having corresponding resources in DBpedia), rule 5 is curated for missed pattern 3 (acronyms), while rule 6 is formulated for missed pattern 4 (re-occurrences of already identified NEs). The rules are being described in the following sections.

Rule 1 – Addition of Entities Recognized by Illinois NER

If e is annotated person, location or organization (PLO) by Illinois, add e to E_C if it is a single word person or an abbreviation, or if it is not typed in nonPLO types (types which are not person, location and organization) by DBpedia.

Rule 2 – Addition of Entities Recognized by Stanford NER

If e is annotated PLO by Stanford, add e to E_C if a resource is found for it on DBpedia which is not typed in nonPLO types.

Rule 3 – Expanding Nationality Entities

As one of the categories that the missed NEs often belong to is “entities containing nationalities”, therefore this rule expands nationality entities to check if that is a PLO NE. If e is identified as “nationality” by Stanford NER, expand e towards right to include any adjacent adjectives and nouns in the same noun phrase. For instance, consider the text: *The UK government should support Scotland remain within the European Single Market*. Here, the nationality entity *European* is expanded to “European Single Market”. Again, the knowledge base DBpedia is used for verification. Since, a corresponding organization

resource (http://dbpedia.org/resource/European_Single_Market) is found on DBpedia, therefore “org: European Single Market” is added to E_C .

Rule 4 – Addition of Mentions Having DBpedia Resources

This rule is for recognizing missed entities having corresponding resources in DBpedia. If e is annotated by Spotlight, it is checked if a PLO typed DBpedia resource can be found for it. If yes, e is added to E_C . For example, “Joker” is not identified by Stanford or Illinois NER, but since it is identified by Spotlight it is in the list E_P , and as it has a corresponding DBpedia resource, <http://dbpedia.org/resource/Joker>, of type person, therefore “per:Joker” is added to E_C .

The list E_C created as a result of applying the above rules is further enriched by applying rules 5 and 6 as explained in coming sections. Let c be an entity in E_C .

Rule 5 – Recognizing Acronyms

It has been observed that most of the time acronyms are not identified by the annotators. Usually, when an acronym first appears in a text, it is defined. This definition has some fixed patterns [104] which can be used to identify acronyms, for example “Capital University of Science and Technology, CUST”. A rule is thus formulated to identify such patterns. If in the input text, an entity c is immediately followed by one of the following patterns:

1. Capital lettered token
2. Capital lettered token in parenthesis
3. Comma followed by a capital lettered token

Then the capital lettered token (say X) is checked. If letters of X are initials of capital lettered words of c in sequence, then X is an acronym of c , and is therefore added to E_C . Consider for instance, “Reserve Bank of India (RBI)”.

TABLE 4.1: List of notations used in Algorithm 1

Notation	Description
T_1	set of PLO types $\{person, location, organization, country, city, stateOrProvince\}$
T_2	set of nonPLO Stanford NER types i.e. $\{date, time, percentage\}$
T_3	set of types might or might not be PLO $\{title, nationality, misc\}$
A_{N1}	list of texts annotated by StanfordNER
A_{N2}	list of texts annotated by IllinoisNER
A_N	$A_{N1} \cup A_{N2}$
A_S	list of texts annotated by Spotlight
E_C	list of entities annotated by CustNER
E_P	list of potential entities i.e. $A_{N1} \cup A_{N2} \cup A_S$
e	an entity in E_P
t	tokenized input text
f	the index of first token of e in t
l	the index of last token of e in t

Since “Reserve Bank of India” is already added by the system to E_C , and in the input text it is followed by a token in parenthesis whose letters are initials of capital lettered words of “Reserve Bank of India”, therefore RBI is added to E_C .

Rule 6 – Adding Re-Occurrences of Added Entities

If an entity \mathbf{c} appears again in the input text, the entity is again added to E_C . If \mathbf{c} has type person, then even if just first or just second name appears again in input text, this second mention is also added to E_C . While adding person entities, any honorifics prefixing person are ignored e.g. Mr, Mrs, Dr, etc.

Pseudocode for main rules of CustNER is given as Algorithm 1. A list of notations used in Algorithm 1 is given in Table 4.1. By applying these rules, CustNER has been able to recognize many of the NEs missed by existing NERs, for instance, the system is able to recognize all the NEs listed in Table 1.1 which are missed by most existing NERs. Some examples for rules have been presented in Table 4.2.

Algorithm 1 CustNER Rules (Page 1)

```

1: for each  $e \in E_P$  do
2:   if  $e \in A_{N2}$  and  $e.type \in T_1$  then // Rule1
3:     if ( $e.type = PER$  and  $e.noOfWords = 1$ ) or  $e.isAbbreviation = true$ 
   or  $e.dbpediaResourceType \notin nonPLO$  then
4:       Add  $e$  to  $E_C$ 
5:     end if
6:   end if
7:   if  $e \in A_{N1}$  and  $e.type \in T_1$  then // Rule2
8:     if  $e.hasDbpediaResource = true$  and  $e.dbpediaResourceType \notin$ 
    $nonPLO$  then
9:       Add  $e$  to  $E_C$ 
10:    end if
11:  end if
12:  if  $e.type = Nationality$  then // Rule3
13:     $nextTokenIndex \leftarrow l + 1$ 
14:     $nextToken \leftarrow t[nextTokenIndex]$ 
15:    if  $nextToken.pos = Adjective$  then
16:       $nextTokenIndex ++$ 
17:       $nextToken \leftarrow t[nextTokenIndex]$ 
18:    end if
19:    while  $nextToken.pos = Noun$  and  $nextToken \notin T_1$  and  $nextToken$  is
   in same NounPhrase as  $e$  do
20:       $e \leftarrow e + nextToken$ 
21:       $nextTokenIndex ++$ 
22:       $nextToken \leftarrow t[nextTokenIndex]$ 
23:    end while
24:     $resource \leftarrow link(e)$ 
25:    if  $e.hasDbpediaResource = true$  then
26:       $e.type \leftarrow resource.type$ 
27:      Add  $e$  to  $E_C$ 
28:    end if
29:  end if
30:  if  $e \in A_S$  then // Rule 4
31:    if  $e.noOfCharacters \geq 3$  and  $t[f].pos = ProperNoun$  then
32:       $resource \leftarrow link(e)$ 
33:      if  $e.hasDbpediaResource = true$  and  $e.dbpediaResourceType \in$ 
    $PLO$  then
34:         $e.type \leftarrow resource.type$ 
35:        Add  $e$  to  $E_C$ 
36:      end if
37:    end if
38:  end if
39: end for

```

CustNER Rules (Page 2)

```

40: for each  $e \in E_C$  do // Rule 5
41:    $isAcronym \leftarrow False$ 
42:    $nextTokenIndex \leftarrow l + 1$ 
43:    $nextToken \leftarrow t[nextTokenIndex]$ 
44:   if  $nextToken = '('$  or  $nextToken = ','$  then
45:      $nextTokenIndex ++$ 
46:      $nextToken \leftarrow t[nextTokenIndex]$ 
47:     if  $nextToken.isUpperCase = True$  then
48:        $X \leftarrow nextToken$ 
49:        $j \leftarrow 0$ 
50:       for each word  $w_i$  in  $e$  do
51:         if  $w_i[0].isCapital = True$  then
52:           if  $w_i[0] \neq X[j]$  then
53:              $isAcronym \leftarrow False$ 
54:             break
55:           end if
56:            $j ++$ 
57:         end if
58:       end for
59:       if  $isAcronym = True$  then
60:         Add  $X$  to  $E_C$ 
61:       end if
62:     end if
63:   end if
64: end for
65: for each  $e \in E_C$  do // Rule 6
66:   repeat
67:      $m \leftarrow getNextMatch(e)$ 
68:     if new match found then
69:       Add  $m$  to  $E_C$ 
70:     end if
71:   until  $m \neq -1$ 
72:   if  $e.type = PER$  then
73:     for each word in  $e$  do
74:       repeat
75:          $m \leftarrow getNextMatch(word)$ 
76:         if new match found then
77:           Add  $m$  to  $E_C$ 
78:         end if
79:       until  $m \neq -1$ 
80:     end for
81:   end if
82: end for

```

TABLE 4.2: Examples of Rules

Text	NE Type	Annotation by:			DBpedia	CustNER	Explanation
		Illinois	Stanford	Spotlight	Type	Annotation	
North Pole	LOC	loc: North Pole	nationality: Pole	North Pole	loc	loc :North Pole	Rule 1 applied. Resource is found on DBpedia which is not nonPLO type, so added to E_C
La La	not NE	per: La La	loc: La La	La La Land	music	not NE	Rule 1 applied. DBpedia resource is nonPLO type, so not added to E_C
Eisenheim	LOC	-	loc: Eisenheim	Eisenheim	loc	loc: Eisenheim	Rule 2 applied. Type of its DBpedia resource is not nonPLO, so added to E_C
South Korean police	ORG	misc: South Korean	nationality: South Korean	South Korean police	org	org: South Korean police	South Korean identified nationality by Stanford, so Rule 3 applied, expanded to “South Korean police”. The expanded text is on DBpedia as ORG, so is added to E_C
American	not NE	misc: American	nationality: American	-	-	not NE	Rule 3 applied. Expanded to “American singer”, but its resource is not found on DBpedia, hence it is not added to E_C

Continued on next page

Table 4.2 – continued from previous page

Text	NE	Annotation by:			DBpedia	CustNER	Explanation
	Type	Illinois	Stanford	Spotlight	Type	Annotation	
al-Bab	LOC	-	-	al-Bab	city	loc:al-Bab	Rule 4 applied. Since a PLO type (city) resource is found for it on DBpedia, hence it is added to E_C
Westler	not NE	-	-	Westler	Movie	not NE	Rule 4 applied. DBpedia resource has type Movie (i.e. nonPLO), so its not added to E_C
IS	ORG	-	-	-	country	org:IS	The input text contains “Islamic State (IS)”. Islamic State is already identified by CustNER, so the text after it is checked by Rule 5. Rule 5 identifies “IS” as acronym for “Islamic State” and adds it to E_C . Moreover, there are also other occurrences of “IS” in text, these are also added to E_C by Rule 6.
Turing	PER	-	-	-	per	per:Turing	By Rule 6, Turing is identified as a recurrence of Alan Turing which is already identified as person, so person Turing is added to E_C

4.2.3 Identifying Gender of Person NEs

The gender of a person NE is identified by applying following rules:

1. If the word immediately before person name is a male/female title (like Mr, Ms, lady, gentleman, etc.) or a male/female relation word (like mother, father), then add gender accordingly.
2. If any of the corefs of person is male/female, for example he, him, she, etc., then add gender accordingly.
3. If person has DBpedia resource, then if `rdfs:comment` contains male/female pronouns, then add gender according to first pronoun.
4. If gender is still un-decided, then look-up first name of person in an external dictionary (`gender_guesser` library for python has been used) and add gender accordingly.

4.2.4 Querying DBpedia

The rule engine uses DBpedia as a verifying source. The queries performed by CustNER against DBpedia, to check if a corresponding resource exists, are elaborated in this section.

The query given in Listing 4.1 is used to retrieve the DBpedia URI of an entity `e`. In the query `<EntityLabel>` is replaced by `e`'s text.

```
select distinct ?s AS ?URI
where {
    ?s rdfs:label "<EntityLabel>"@en.
}
```

LISTING 4.1: DBpedia Query

If a URI is found, the URI is used to retrieve `e`'s DBpedia types, labels and hypernyms by the query given in Listing 4.2

```

select distinct ?label ?hypernym ?type
where {
    {<URI> rdfs:label ?label. Filter (lang(?label)="en")}
    union    {<URI> rdf:type ?type}
    union    {<URI> purl:hypernym ?hypernym }
}

```

LISTING 4.2: Query DBpedia Type

The following rules are then applied for linking **e**.

1. If hypernym of **e** is noun (its not proper noun), then **e** is not an NE
2. If **e**'s label has upto two words and a word in label starts with a small letter, then it is not a proper noun, so **e** is not an NE
3. From the types returned from DBpedia:
 - (a) If more than five types are nonPLO types, then **e** is not an NE
 - (b) **e** is typed into the class for which more types are returned
 - (c) If no type is found in PLO classes, then **e** is not an NE

The types given in Listing 4.3 are used for deciding the class of **e**.

4.3 Experimental Setup

A core-i7, 2.20 GHz machine, having 16GB RAM, has been used for implementation and experiments. The machine has Microsoft Windows 10 Home edition, 64 bit version, installed on it. The system has been implemented in python (3.7.2) using PyCharm integrated development environment, Community Edition 2017. Table 4.3 lists the tools used by the system, the URLs of their Web demos (where user can give a text and get it annotated by these systems) and the packages which have been imported in python in order to use them in the proposed system. The tools have been used in the proposed system with their default settings.

```

    personTypes = ['Actor', 'Artist', 'MusicalArtist', '
NaturalPerson', 'Person', 'WikicatFictionalBritishPeople', '
FictionalHuman', 'FictionalPeople', 'Entertainer10', 'Singer', '
Musician']

    locationTypes = ['Place', 'City', 'Settlement', '
PopulatedPlace', 'Location', 'Building', 'Museum', '
ArchitecturalStructure', 'Area10']

    organizationTypes = ['Company', 'Organization', '
WikicatOrganisation', 'Organisation', 'Agency', '
AdministrativeUnit', 'Magazine', 'Newspaper', '
AdministrativeUnit', 'Broadcaster', 'EducationalInstitution', '
GovernmentAgency', 'EthnicGroup', 'MilitaryUnit', 'Parliament',
, 'PoliticalParty', 'ReligiousOrganisation', 'SportsClub', '
SportsLeague', 'SportsTeam', 'TradeName', 'WikicatBrands']

    nonPLOTTypes = ['Conference', 'SocietalEvent', 'Event', '
Gathering', 'Meeting', 'Crime', 'SocialEvent', 'SportsEvent', '
Tournament', 'Language', 'Action', 'Activity', 'LanguageUnit', '
Unit_of_account', 'Scheme', 'Film1', 'Music10', '
MusicalComposition', 'Song10', 'Video10', 'Movie1', '
VisualCommunication', 'TelevisionShow', 'Award1', '
WikicatDrugRings', 'Test1', 'WikicatTrials', '
WikicatFictionalDragons', 'WikicatHonorifics', '
ExpressiveStyle', 'Book1', 'Class10', 'Collection', '
AcademicDegree']

```

LISTING 4.3: Person Location Organization and nonPLO Types

As shown in Figure 4.1, the input text is first fed to Stanford NER, UIUC NER and DBpedia Spotlight, which are imported into Python using the libraries mentioned in Table 4.3. The annotated output by these three is the input to CustNER (see Figure 4.1), which has been described in the previous section 4.2 and has been implemented in Python. The CustNER rule engine uses knowledge from DBpedia, via the library mentioned in Table 4.3, and generates NEs for the input text.

TABLE 4.3: Tools used by the system module CustNER

Tool	Online Demo URL	Python Imported Package
Stanford NER	https://corenlp.run/	from pycorenlp import StanfordCoreNLP
UIUC NER	https://cogcomp.seas.upenn.edu/page/demo_view/ner , http://macniece.seas.upenn.edu:4004/	from ccg_nlpy import remote_pipeline
DBpedia Spotlight	https://www.dbpedia-spotlight.org/demo/	import spotlight
Dbpedia	http://dbpedia.org/snorql/	from SPARQLWrapper import SPARQLWrapper

The benchmark datasets (explained in chapter 3) which have been used to evaluate the performance of the proposed system are given in Table 4.4, along with the domain of dataset and the URL from where it has been downloaded.

TABLE 4.4: Datasets used for evaluation of system module CustNER

Dataset	Dataset Domain	Dataset Downloaded From
The CoNLL 2003 dataset	English newswire data	https://github.com/synalp/NER/tree/master/corpus/CoNLL-2003
The OKE 2017 dataset	Heterogeneous sources: web, news, blogs, emails, etc.	https://project-hobbit.eu/open-challenges/oke-open-challenge/

Four existing systems have been used for results comparison purpose: Stanford, UIUC, ADEL and FOX. To get comparison results for Stanford and UIUC, their packages have been imported in python as mentioned in Table 4.3 and results have been generated using default settings for both datasets. Results of ADEL and FOX have been generated using their online demos at <http://adel.eurecom.fr/api/> and <https://fox.demos.dice-research.org/index.html#!/demo> respectively, by manually giving each input text to the online demo, one by one, and saving the generated outputs in files. Results of FOX and ADEL have been obtained for

OKE dataset, and not for CoNLL dataset (as CoNLL documents are long and the online demos were giving errors on their input).

4.4 Evaluation

The system CustNER has been developed to recognize PER, LOC and ORG type NEs from text. Rules of the system have been designed very carefully by analyzing examples in OKE train dataset and by considering grammatical structure of English language, so they are applicable to any generic English text. To assess how well CustNER recognizes these NEs, it is tested on OKE evaluation, CoNLL03 train and CoNLL03 evaluation datasets. The OKE dataset contains annotations for PER, LOC and ORG classes, whereas the CoNLL dataset contains annotations for one more class, i.e. MISC. For evaluating our system, the MISC tags from CoNLL dataset have been ignored. The following sections describe the evaluation measures used for this work, report results of evaluation achieved, along with a comparison with other NER systems and an analysis of system errors.

4.4.1 Evaluation Measure

The evaluation measures used to gauge the performance of NERC systems have unanimously been precision, recall and F1 measure. F1 score, or simply F measure is the harmonic mean of precision and recall.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4.2)$$

Precision is the fraction of annotations made by the system that are correct, while recall refers to the percentage of total annotations correctly identified by the system.

$$Precision = \frac{TP}{TP + FP} \quad (4.3)$$

$$Recall = \frac{TP}{TP + FN} \quad (4.4)$$

Here TP, FP and FN stand for true positives, false positives and false negatives, respectively. For multi class problem, precision and recall are usually micro-averaged over all classes. When three classes, PER, LOC and ORG are considered, then:

$$Precision_{micro} = \frac{TP_{PER} + TP_{LOC} + TP_{ORG}}{TP_{PER} + TP_{LOC} + TP_{ORG} + FP_{PER} + FP_{LOC} + FP_{ORG}} \quad (4.5)$$

$$Recall_{micro} = \frac{TP_{PER} + TP_{LOC} + TP_{ORG}}{TP_{PER} + TP_{LOC} + TP_{ORG} + FN_{PER} + FN_{LOC} + FN_{ORG}} \quad (4.6)$$

There are some variations in the definition of correct annotation, though. Four major variations found in literature are as follows:

1. NER considering Weak Annotation: An annotation is considered correct if its surface form overlaps with gold standard’s surface form, irrespective of its type (i.e. even if its boundaries do not match exactly in gold standard and even if it is not typed correctly). This technique is used by OKE challenge, for instance.
2. NER considering Strong Annotation: An annotation is considered correct if its surface form matches with gold standard’s surface form, irrespective of its type (i.e. if its boundaries match exactly in gold standard, but it might not be typed correctly). Han et al. [104] have reported their results using this scheme.
3. NERC considering Weak Annotation: An annotation is considered correct if its surface form overlaps with gold standard’s surface form, and its type is same as in gold standard (i.e. even if its boundaries do not match exactly in gold standard, but it is typed correctly). Jiang et al. [127] evaluated using this method.
4. NERC considering Strong Annotation: An annotation is considered correct if its surface form and type match with gold standard’s surface form and type (i.e. if its boundaries match exactly in gold standard and it is also typed correctly). This approach is followed by CoNLL shared task, and is used the most in existing literature.

To understand these better, consider this example from OKE train dataset.

It was a few minutes before halftime of the [ORG] **FIFA** Confederations Cup final between host [ORG] **Brazil** and the world and European champion [ORG] **Spain** at the refurbished [LOC] **Maracanã stadium**.

The NER task concerns only recognizing NEs, and does not include classifying them, so its sufficient that all four of the bold NEs in above example have been identified. If weak annotation is considered, the NE boundaries do not matter, so for example, annotations *host Brazil* and *champion Spain* by a system are also considered correct, while for strong annotation match such annotations are considered incorrect. The NERC task concerns recognizing as well as classifying NEs, so an annotation is considered correct if it is typed correctly. If *Spain* is marked as *location* by a system for above example, it is incorrect.

The proposed system has been evaluated using precision, recall and F1 scores following CoNLL guideline, that is, for NERC considering Strong Annotation.

4.4.2 Results

The results of CustNER on the benchmark datasets are encouraging. Table 4.5 summarizes the Named Entity Recognition and Classification results of CustNER and baseline systems on enhanced OKE evaluation dataset, considering strong annotation match. All results are reported in percentages. Per class and aggregate precision, recall and F1 (micro) scores for the systems have been generated using the segeval scorer script [128].

It can be seen from Table 4.5 that F1 of CustNER is 81.03 on OKE evaluation dataset, highest compared to all existing systems. On average CustNER scores 10.5 F1 points better than others. Compared to the systems whose output is used by CustNER (Stanford and Illinois), the rules have successfully raised the F1 from 67.16 and 76.43 to 81.03.

TABLE 4.5: NERC Results Comparison on OKE test set

System	Evaluation Metric	Named Entity Class			Average (micro)
		PER	LOC	ORG	
Stanford	Precision	58.90	66.91	71.19	63.11
	Recall	78.53	70.54	57.53	71.77
	F1 score	67.31	68.68	63.64	67.16
Illinois	Precision	86.54	85.47	60.00	81.07
	Recall	76.27	77.52	53.42	72.30
	F1 score	81.08	81.30	56.52	76.43
ADEL	Precision	59.86	57.76	46.55	56.70
	Recall	49.72	51.94	36.99	48.02
	F1 score	54.32	54.69	41.22	52.00
FOX	Precision	88.12	80.83	68.52	82.34
	Recall	79.66	75.19	50.68	72.56
	F1 score	83.68	77.91	58.27	77.14
LUKE	Precision	88.97	91.84	74.65	86.62
	Recall	72.88	69.77	72.60	71.77
	F1 score	80.12	79.30	73.61	78.50
CustNER	Precision	90.30	84.55	64.79	83.29
	Recall	84.18	80.62	63.01	78.89
	F1 score	87.13	82.54	63.89	81.03

Analyzing the results of systems with respect to NE classes, it may be observed that for all systems, F1 for ORG type is much lower compared to types PER and LOC. So the ORG type NEs are the hardest for NER systems to identify. It may further be noticed that, except Stanford, for all other systems recall is much lower compared to precision. Among all systems, precision of LUKE is highest, where as recall and F1 are highest for CustNER.

Since it was hypothesized that English grammar does not change, and regular patterns for entities exist in natural language texts, therefore CustNER is further evaluated on a different dataset (CoNLL03) for which rules have not been built.

Table 4.6 presents results of CustNER and baseline systems on the CoNLL03 evaluation dataset.

TABLE 4.6: NERC Results Comparison on CoNLL03 test set

System	Evaluation Metric	Named Entity Class			Average (micro)
		PER	LOC	ORG	
Stanford	Precision	75.45	87.55	87.87	83.11
	Recall	91.70	90.34	81.36	87.77
	F1 score	82.78	88.92	84.49	85.38
Illinois	Precision	98.17	94.22	95.20	95.81
	Recall	93.76	96.82	85.43	92.00
	F1 score	95.91	95.50	90.05	93.87
LUKE	Precision	96.83	95.58	92.87	95.06
	Recall	97.19	94.78	94.96	95.63
	F1 score	97.01	95.18	93.91	95.34
CustNER	Precision	97.48	92.51	93.74	94.51
	Recall	96.57	97.06	90.89	94.83
	F1 score	97.02	94.73	92.29	94.67

Here again, F1 for ORG type is slightly lower compared to types PER and LOC. So it can be concluded that the ORG type NEs are generally the hardest for NER systems to identify. But in contrast with OKE, on CoNLL, there is no significance difference between recall and precision of systems. It means that systems are mostly able to recognize the NEs in CoNLL dataset, but are unable to extract many NEs from OKE dataset.

It can be seen from Table 4.6 that F1 of CustNER on CoNLL dataset is also very good, 94.67, which is higher than Stanford and Illinois and is very close to LUKE. On average CustNER scores 3.14 F1 points better than other systems.

It is important to note here that CustNER is trained on OKE dataset whereas LUKE is fine-tuned on CoNLL dataset. On OKE dataset, CustNER performs better than LUKE by 2.53 F1 points. On the CoNLL dataset, LUKE performs better than CustNER, by 0.67 F1 points. So on average, CustNER performs 0.93 F1 points better than LUKE (which is the previous highest F1 reporter for English NER of general classes, to the best of our knowledge).

So it can be said that usually English grammar does not change, as rules made by identifying patterns on OKE dataset perform well on CoNLL dataset too.

Moreover, CustNER is implemented on a simple personal computer with single Intel core i7 CPU, and does not require any exceptional resources to run. The deterministic algorithm is very time efficient as opposed to its DL counterpart, LUKE, which is extremely resource expensive. LUKE is fine-tuned on CoNLL and is pre-trained on Wikipedia dataset containing 3.5 billion words and 11 million entity annotations. LUKE's pre-training took 30 days on a server with 2 Intel Xeon Platinum 8168 CPUs (each containing 24 cores) and 16 NVIDIA Tesla V100 GPUs. Further, LUKE's training on CoNLL took 203 minutes on a server with 2 Intel Xeon E5-2968 v4 CPUs (each having 16 cores) and 8 V100 GPUs.

After the pre-training and fine-tuning, running LUKE on CoNLL test set takes 4 hours to produce results, whereas CustNER only takes 6 minutes to produce results on the same dataset in the same settings. So, the deep learning system LUKE performs slightly better on one dataset (CoNLL) but has a lot of computation and data cost. CustNER is a simple system and still performs better than LUKE on one dataset (OKE), on average (of two datasets) performs better than LUKE and takes less time to annotate.

A comparison of the time taken by the two systems to produce output on evaluation datasets is given in Table 4.7. Compared to LUKE, CustNER takes 3 times less time to produce results on OKE evaluation dataset, and 38 times less time on CoNLL evaluation dataset, in same settings.

TABLE 4.7: Comparison of empirical run time for NER task

Dataset	Time taken by CustNER	Time taken by LUKE
OKE eval	3.3 minutes	10 minutes
CoNLL eval	6.2 minutes	236 minutes

4.4.3 Analysis of Errors

The incorrect annotations that CustNER makes have been analyzed. The most frequent errors are the following ones:

4.4.3.1 Errors from NERs Used

Some errors are propagated from the underlying systems used for NER, Illinois and Stanford. For example, *Costa* (PERSON) is tagged LOCATION by Illinois, *pope* (not an NE) is annotated PERSON by Stanford. But few of such cases get propagated to our system as we have used DBpedia for verification and have some checks in the rules to avoid such errors.

4.4.3.2 Errors from Part of Speech Tags

As CustNER rules are made for English language grammar and use part of speech (PoS) tags (which are given as part of CoNLL dataset and are produced using Stanford tagger for OKE dataset), some annotation errors by CustNER occur because the PoS tag was incorrect. Some example incorrect PoS tags from CoNLL dataset are presented in Table 4.8. The examples in the Table are all names of PER, LOC or ORG, so their correct PoS is proper noun, but they are incorrectly PoS tagged as mentioned against each in the Table.

TABLE 4.8: Examples of incorrect PoS tags

Example	NE type	PoS Tag
Khaled	person	verb
Real Madrid	organization	adjective properNoun
Italy	location	adjective
Rome	location	verb
Bitar	person	adjective
St Louis	organization	personalPronoun verb
New Zealand	location	adjective properNoun

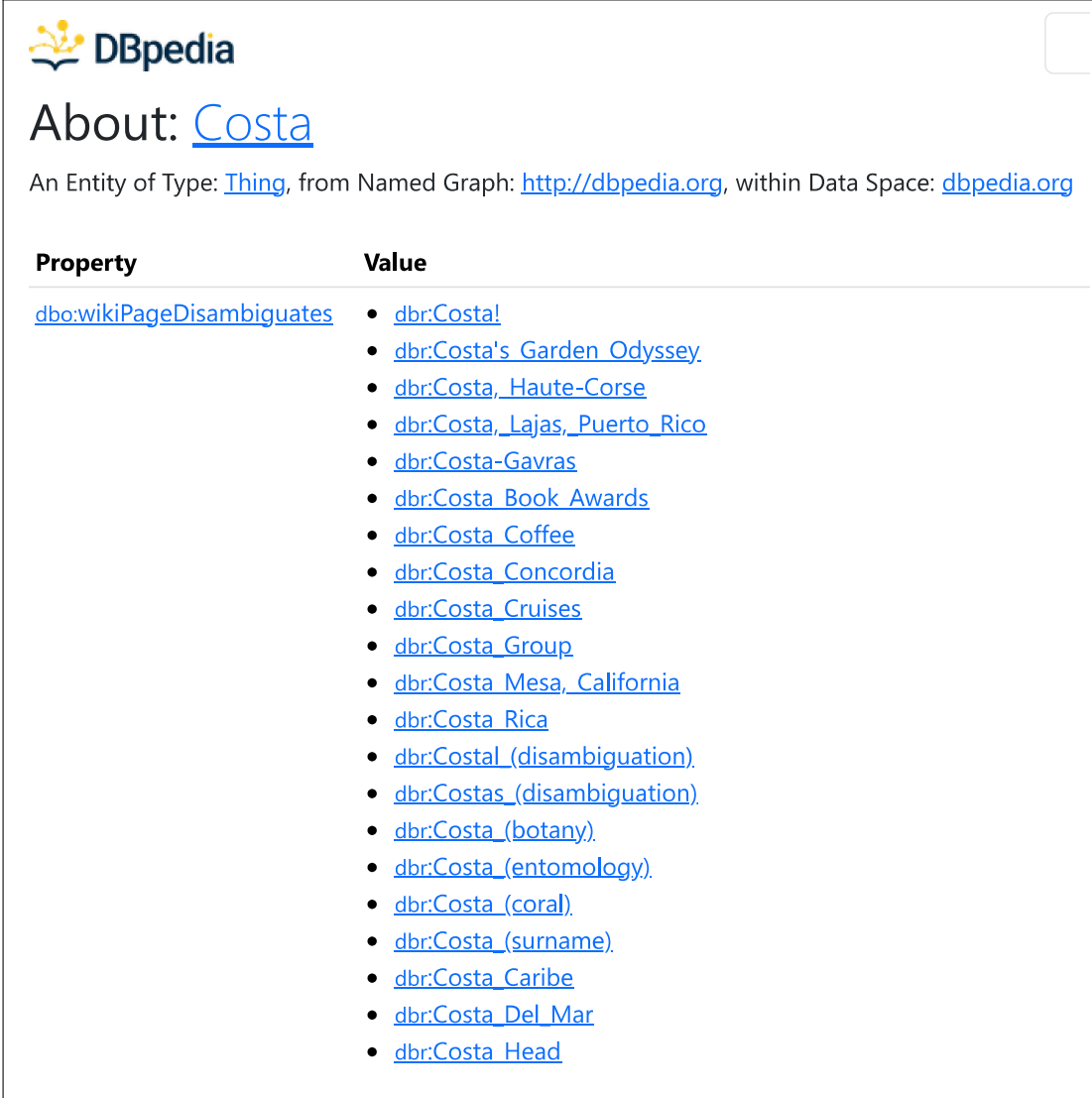
4.4.3.3 Incorrect Type from DBpedia

Type of entity on DBpedia is sometimes incorrect leading to some erroneous annotations by CustNER. For example, *Swiss bank accounts* is not an NE but its DBpedia resource (https://dbpedia.org/page/Swiss_bank_accounts) is incorrectly typed as organization, hence it is incorrectly tagged organization by CustNER.

Some errors also occur for the reason that an exact match of the entity label is used for associating to DBpedia resource. A partial match could be more appropriate but the threshold of similarity needs to be determined, and this could have a cost in terms of performance. For example, the entity *The German federal prosecutor's office* is not annotated by any of the annotators, neither a resource with exact label exists on DBpedia, but DBpedia has a resource for it with a slightly different label *Public Prosecutor General (Germany)*.

4.4.3.4 DBpedia Disambiguation Pages

Some errors occur because for some entities, the corresponding resource found on DBpedia is actually a disambiguation page which lists DBpedia resources that the label might represent. An example for entity label *Costa* is given in Figure 4.2. A strategy needs to be devised to decide which DBpedia resource from the list on disambiguation page refers to the entity.



The screenshot shows a DBpedia page titled "About: [Costa](#)". Below the title, it states: "An Entity of Type: [Thing](#), from Named Graph: <http://dbpedia.org>, within Data Space: dbpedia.org".

Property	Value
dbo:wikiPageDisambiguates	<ul style="list-style-type: none"> • dbr:Costa! • dbr:Costa's Garden Odyssey • dbr:Costa, Haute-Corse • dbr:Costa, Lajas, Puerto Rico • dbr:Costa-Gavras • dbr:Costa Book Awards • dbr:Costa Coffee • dbr:Costa Concordia • dbr:Costa Cruises • dbr:Costa Group • dbr:Costa Mesa, California • dbr:Costa Rica • dbr:Costal (disambiguation) • dbr:Costas (disambiguation) • dbr:Costa (botany) • dbr:Costa (entomology) • dbr:Costa (coral) • dbr:Costa (surname) • dbr:Costa Caribe • dbr:Costa Del Mar • dbr:Costa Head

FIGURE 4.2: Example of DBpedia disambiguation page

4.4.3.5 Ambiguous Cases

Some errors come from such NEs which have different types in different contexts. In general, typing errors occur when an NE type depends on its context. CustNER does not consider context for deciding type and hence makes some mistakes in such cases. For example, *Zootopia* is name of a movie as well as name of a city in the movie. Its DBpedia resource has type movie so the system does not annotate it because movie is not person, location or organization type. But the context tells that Zootopia here refers to location, not movie.

*From the largest elephant to the smallest shrew, the city of **Zootopia** is a mammal metropolis where various animals live and thrive.*

Moreover, there are many cases of Sports news in CoNLL where locations are tagged organizations because of representing teams. An example is given below.

NHL ICE HOCKEY - STANDINGS AFTER FRIDAY'S GAMES.

(tabulate under won, lost, tied, goals for, goals against, points):

W L T GF GA PTS

HARTFORD(org) 12 7 6 77 76 30

BUFFALO(org) 13 12 2 78 77 28

MONTREAL(org) 11 14 4 99 104 26

BOSTON(org) 10 11 4 74 84 24

PITTSBURGH(org) 10 13 3 86 94 23

OTTAWA(org) 7 12 6 64 77 20

In this example from the CoNLL dataset, Hartford, Buffalo, Montreal, Boston, Pittsburgh and Ottawa are location names but marked organizations, CustNER annotates them as locations.

4.5 Summary

This chapter has presented the system CustNER for the named entity recognition and classification of PER, LOC and ORG type NEs from English text. Carefully analyzing the missed annotations of existing NER systems on OKE training dataset, regular patterns of NEs have been identified. Most of the NEs not annotated by existing systems either contain nationalities, have corresponding resource in DBpedia, are acronyms or are re-occurrences of other NEs. Rules have been formulated against these patterns of NEs missed by existing systems, by carefully analyzing grammatical structure of English language on OKE train dataset. The output of existing systems and DBpedia KB has been utilized to recognize NEs

which are not recognized by most existing systems. The system has been tested and compared with SOTA systems on OKE and CoNLL (without retraining) evaluation datasets. These benchmark datasets are constructed by the organizers of top-tier conferences in the NLP domain and contain texts from different domains. Since generic datasets have been chosen, and CustNER scores good F1 on these standard datasets, it will perform well on any general dataset.

Rules are generally considered static. But CustNER is not dataset specific as rules have been carefully designed against regular patterns of English language on generic dataset. The results have been compared with SOTA models, proving that the designed system is not dataset specific. CustNER outperforms existing NER systems, with F score 10.5 points better than baselines (on average) on OKE dataset and 3.1 points better on average on CoNLL dataset. The system is able to correctly recognize the NEs missed by existing systems given in Table 1.1. This chapter has thus addressed the first research question and has successfully achieved the first research objective (both re-stated below).

RQ1: How to formulate rules to recognize NEs which are missed by existing NER systems?

RO1: Devise a technique to recognize named entities from text, specially those instances which are missed by existing NER systems

Chapter 5

CustRE - An Improved System for Relation Extraction

5.1 Introduction

This chapter describes the system module CustRE, proposed to achieve RO2, to extract family relations of types parents, siblings, spouse, children, other_family, and not_known from text. It has been observed that existing systems are making many mistakes which can be avoided by simple rules, like: *If A children B, Then B parents A*. The syntactic patterns in which family relations appear in text in TACRED-F training dataset have been carefully analyzed. Rules have been formulated to recognize the patterns.

Family relations are usually expressed by using specific words e.g. parent relation is conveyed by using words like father, mother, step-father, parent, papa, mom, etc. CustRE extracts such words from text and decides which family words are binding which persons from text into the family relation triples. Using domain knowledge and analyzing training data, common cases of how (subject, family_relation, object) triples appear in text have been figured out. The method is based on pattern matching through regular expressions for extracting family relations explicitly

mentioned in text (direct relations) and employs propagation to infer implicit family relations (reverse, transitive and coref relations). The proposed system is available at the link: <https://github.com/Raabia-Asif/CustKnowledgeExtractor>

5.2 The System CustRE

The system CustRE is proposed for extracting family relations from English text. CustRE is mainly designed from our world knowledge of how family relations appear in text, and is not dataset specific. Thus it does not need to be redesigned to extract family relations from a new dataset. NLP tasks are usually regarded as a pipeline process, where some kind of information is extracted at each stage [28, 49, 55]. This module focuses the relation extraction stage of the pipeline. The earlier steps of the pipeline, like PoS, NER, dependency tagging, co-reference resolution have been achieved using Stanford and NeuralCoref¹ systems.

A family relation between two persons is conveyed in text by using family words like sister, mama, wife, grandfather, etc. A list of such words has been compiled which express family relations, and is given in Appendix A. The system extracts all family words from input text and decides which family words are binding which persons from text into the family relation triples. First, following initial rules are applied to the input text:

1. If text contains Mr someone and Mrs someone, then they are spouses
2. If there is no family relation word in sentence, there is no relation between its persons
3. If x is pronoun, and coref of person y, then x and y refer to same person and hence have no relation

The following rules are then used to decide which list word of input text \mathbf{t} is connecting which persons. The result is a list of family triples extracted from the text.

¹<https://github.com/huggingface/neuralcoref>

Rule 1 – The Usual Case: subject relation object

If **t** has two persons and a family relation word between them, this is the simplest case, just connect the first person as subject to the second person (as object) with the relation.

Ahmed has a son Ali, Ahmed’s son Ali, Ali’s father Ahmed, Ali’s wife Amna, are all examples of this case. The triples formed by this rule are (Ahmed, per:children, Ali), (Ahmed, per:children, Ali), (Ali, per:parents, Ahmed), and (Ali, per:spouse, Amna), respectively.

Complexity of the problem increases as the number of relation words and persons increases in **t**. It then becomes difficult to decide which relation word connects with which person.

Rule 2 – The Multiple Relations Case: subject relation object relation object relation object ...

If persons and relation words appear alternatively in **t**, connect the first person (as subject) with every next relation word and person pair.

For example, the triples extracted from the text “Hina’s sister Anam, brother Ali and mother Alina were in the car” by this rule are (Hina, per:siblings, Anam), (Hina, per:siblings, Ali), and (Hina, per:parents, Alina).

Rule 3 – The Numbered Relations Case: subject number relations object1, object2... objectD.

This case handles the numbered relations cases. When **t** has a person followed by a plural relation word (e.g. sons, children, sisters, etc.), followed by more than one persons, and a number **d** modifies the relation word, then connect first person as subject and the relation word as predicate to **d** many next persons as objects.

Texts that have a number specified with relation word, for example “Amna’s four children, Bilal, Dina, Fari and Hadi, ...”, the number many relations are generated from them having same subject and relation but different objects, for this example four relations are generated: (Amna, per:children, Bilal), (Amna, per:children, Dina), (Amna, per:children, Fari), and (Amna, per:children, Hadi).

Rule 4 – The Plural Relation Case: subject relations object1, object2...

When **t** has a person followed by a plural relation word, followed by more than one persons, then connect first person as subject and the relation word as predicate to each of the next persons as object.

For instance, for the text “Amna’s children, Bilal, Dina, Fari and Hadi, ...”, multiple relations are generated having same subject and relation but different objects: (Amna, per:children, Bilal), (Amna, per:children, Dina), (Amna, per:children, Fari), and (Amna, per:children, Hadi).

All the above cases are to handle situations where subject person appears before relation word and object person appears after relation word.

Rule 5 – The Inverted Relation Case: object relation inverter subject

At times the relation word is followed by a word such as ‘of’ which inverts the direction of relation, that is, the first person becomes the object in such case while the second person befits the subject. Consider as an example, “Ahmed is son **of** Ali”. The relation that should be extracted from this example is (Ali, per:children, Ahmed), and should not be (Ahmed, per:children, Ali). To handle this situation, a small list of inverting words has been compiled. For any of the above cases, if a relation word is followed by any of the inverting words, then the direction of the extracted relation is inverted.

Rule 6 – The Indirect Case 1: object subject relation

If more than one person appear before relation word in \mathbf{t} , then it is checked if the first person is followed by one of “,” , “(“ , or “and” . This is the indirect case, so the second person is connected as subject to the first person (as object) with the relation.

“Hadi, Ali’s son . . .” , “Danial and Farhan, the brothers went to . . .” , “Hadi (Ali’s son) came to . . .” are examples of such cases. The relations generated according to this rule are, (Ali, per:children, Hadi), (Farhan, per:siblings, Danial), and (Ali, per:children, Hadi), respectively.

Rule 7 – The Indirect Case 2: relation subject object

If relation word appears before persons in \mathbf{t} , then it is checked if the first person is followed by one of “,” , “(“ , or “and” . This is another indirect case. The relation word is connected as predicate to the former person as subject and the latter person as object.

Examples of such cases include, “The brothers, Danial and Farhan, went to . . .” , “Wife of Hadi, Sana, came to . . .” , “Son of Ali (Hadi) inherited . . .” . The relations extracted according to this rule are (Danial, per:siblings, Farhan), (Hadi, per:spouse, Sana) and (Ali, per:children, Hadi) respectively.

Next, for each triple $(\mathbf{s}, \mathbf{r}, \mathbf{o})$ extracted by the above rules, it is checked if \mathbf{s} or \mathbf{o} have any co-referents. For each co-referent \mathbf{cS} of \mathbf{s} , also add the triple $(\mathbf{cS}, \mathbf{r}, \mathbf{o})$ to the list of triples. Similarly, for each co-referent \mathbf{cO} of \mathbf{o} , also add the triple $(\mathbf{s}, \mathbf{r}, \mathbf{cO})$ to the list of triples. Then it is checked if the triples in the list of triples can be propagated to form new relation triples. For example, if list has triples $(x, \text{children}, y)$ and $(y, \text{sibling}, z)$, then the triple $(x, \text{children}, z)$ is also added to the list.

In the next section, the implementation details of the proposed system are explained.

5.3 System Implementation

Architecture of the proposed system, CustRE, is given in Fig. 5.1. Text \mathbf{t} is input to the system, and the system generates a list of family relation triples as output. Input text \mathbf{t} is first pre-processed and basic NLP annotations for PoS, NER, dependency relations and coreference have been obtained using external systems, Stanford corenlp and NeuralCoref. The system uses lists of common relation words, \mathbf{L} , as lexical clues. \mathbf{L} has been mainly compiled by taking words that represent family relations from Wikidata properties, and is further enriched by adding such words from TACRED-F training dataset. Using \mathbf{L} , the Pattern Extractor module extracts from \mathbf{t} a string \mathbf{pat} to represent the pattern of important parts of \mathbf{t} .

A Regex Base of regular expressions (**regexs**) has been compiled to identify family relation patterns. If any of these **regexs** is matched in \mathbf{pat} , then the Explicit Triples Generator module, with the help of the Extraction Rules, generates a list of family relation triples explicitly mentioned in \mathbf{t} . This list is forwarded to the Implicit Triples Generator module, which uses the Coref and Propagation Rules to enrich the triples list to include triples for those family relations as well which are implicitly implied in \mathbf{t} .

The major modules of the system are now explained.

5.3.1 Relation Words Lists \mathbf{L}

To identify relevant patterns in \mathbf{t} for FRE, the words that express family relations are very important lexical clues [105]. Small lists are therefore compiled, one for each relation, making a total of 180 words (given in Appendix A), comprising common words that are used to represent each of the relations. The online Wikidata Properties² include a list “Also known as” for each property. These lists for Wikidata properties were obtained and mapped to the family relations as given in

²<https://www.wikidata.org/wiki/Wikidata:Property>

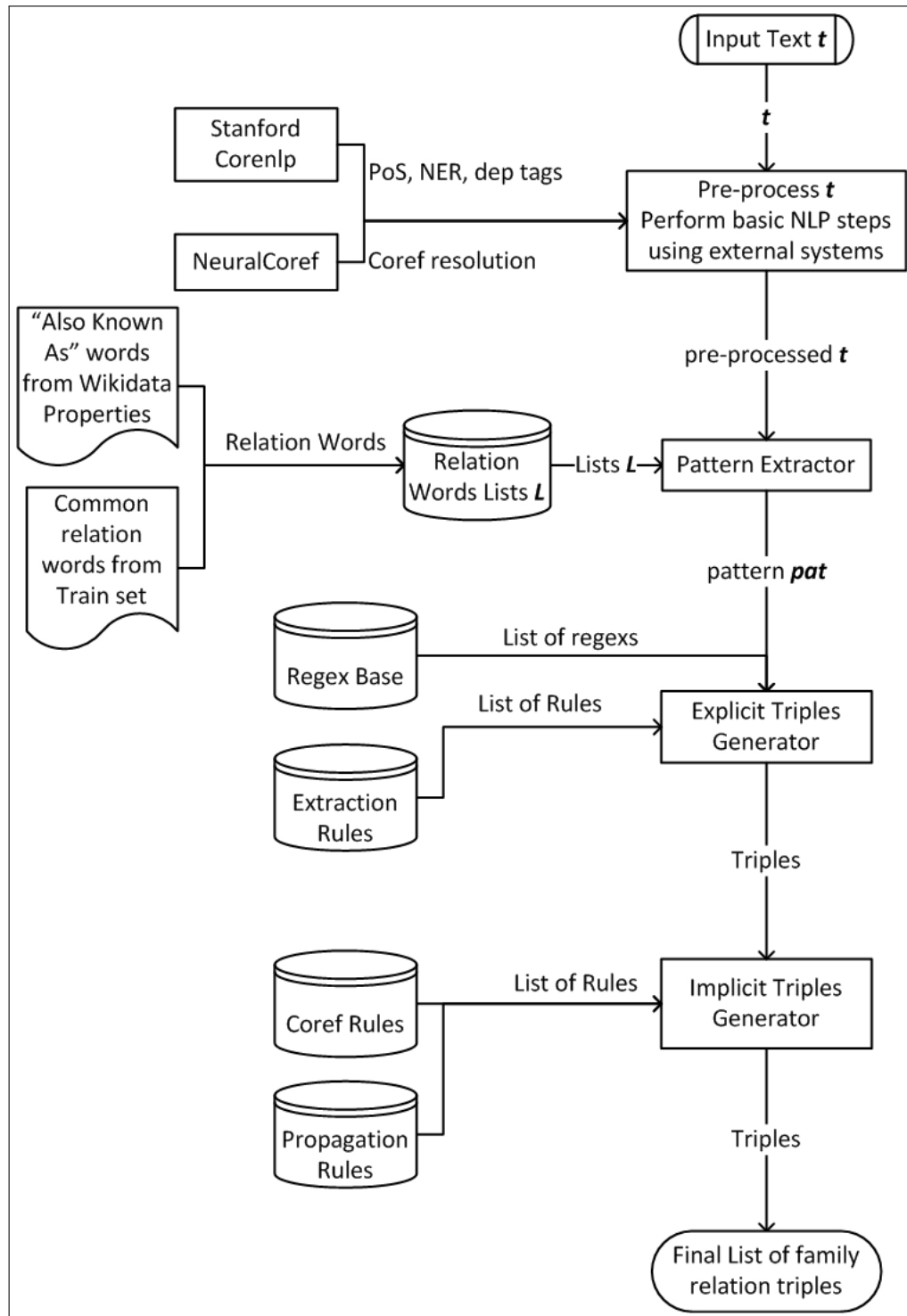


FIGURE 5.1: Architecture of the proposed system, CustRE

Table 5.1. These lists make the core of the lists. The lists are further enriched by adding to them common words used for family relations in the training dataset, and any other words that were thought to represent family relations.

TABLE 5.1: Mapping of Wikidata Properties to family relations

Wikidata Property	Mapped to relation
P22 father, P25 mother, P3448 stepparent	per:parents
P26 spouse, P451 partner	per:spouse
P40 child	per:children
P3373 sibling	per:siblings
P1038 relative	per:other_family

5.3.2 Pattern Extractor

The Pattern Extractor module extracts from \mathbf{t} a string \mathbf{pat} to represent the pattern of important parts of \mathbf{t} , by replacing any persons (names or pronouns) with a generic \mathbf{P} , any relation words with \mathbf{w} , and so on. A complete list of the features extracted from \mathbf{t} , along with the replacements made is given in Table 5.2.

TABLE 5.2: Features extracted from \mathbf{t} and the symbols used to represent them

No.	Feature Extracted	Replacement Symbol
1	person name or person pronoun	\mathbf{P}
2	punctuation mark	The punctuation mark's symbol for example ; : ., () etc.
3	and	$\&$
4	Numbers having nummod dependency on a relation word	The number in Arabic numerals, let us call it \mathbf{d}
5	word from relation list L	\mathbf{w}

Consider the input text, “*Kollek is survived by **his** widow **Tamar**, son **Amos** and daughter **Osnat**.”. Only important parts of this text, as specified in Table 5.2 are extracted from this text, i.e. “**Kollek his** widow **Tamar**, son **Amos** and daughter **Osnat**.”. Next, according to Table 5.2, the persons (written in bold in text, Kolle, his, Tamar, Amos, Osnat) are replaced by \mathbf{P} 's, the family words (widow, son, daughter) are replaced by \mathbf{w} 's, 'and' is replaced by $\&$, and punctuation marks (.) remain as it is. Hence the pattern string generated by this module for the example text is “ $\mathbf{P P w P, w P \& w P.$ ”*

5.3.3 Regex Base

A Regex Base has been compiled, i.e. a set of regular expressions, **regexs**, defined to recognize family relation patterns in **t**. For this purpose, the common syntactic patterns in which family relations occur have been figured out. These patterns have been formalized in the form of regular expressions.

Many times, the patterns in which family relations are found are similar, regardless of the family relation they express. For example, the sentences “Ali’s son Ahmed”, “Ali’s wife Salma”, and “Ali’s sister Sana”, all have same syntactic structure but convey different family relations. They all have a person followed by a relation word followed by another person, and the first and second persons are related by the relation word between them. A generic rule can be therefore formulated to handle this pattern: *person relationWord person*. Common syntactic patterns of family relations have been identified and are given in Table 5.3. Most of the times, the subject of relation appears in the text before the object, and the relation word relating subject and object appears in between them. The first three **regexs** are to handle these usual cases. But there are other cases when relation word appears either after or before both subject and object, these are handled by 4th and 5th **regexs** below. Regular expressions are written in python to identify these patterns in texts. Here, the symbol + means one or more occurrences.

5.3.4 Explicit Triples Generator

If any of the **regexs** in the Regex Base is matched in **pat**, then the Explicit Triples Generator module generates a list of (*subj*, *rel*, *obj*) triples for the family relations explicitly mentioned in **t**, as defined by the Extraction Rules given in Table 5.4. These extraction rules are elaborated with the help of some examples in Table 5.5. At times the relation word is followed by a word such as ‘**of**’ which inverts the direction of relation, that is, P_1 becomes the object in such case while P_2 befits the subject. For example, Ahmed is son **of** Ali. The relation that should be extracted from this example is (Ali, per:children, Ahmed), and should not be

TABLE 5.3: List of regular expressions

No.	Regex	Explanation
1	$P_s (w P_o)+$	The usual case: <i>subject relation object</i> , or The multiple relations case: <i>subject relation object relation object</i>
2	$P_s d w P_{o1}, P_{o2}, \dots, P_{od}$	The numbered relations case: <i>subject number relation object1, object2. . . , objectD</i> .
3	$P_s w P_o+$	The plural relations case: <i>subject relation object1, object2. . . .</i>
4	$P_o [, (\&] P_s w$	Indirect case 1: <i>object subject relation</i> . Two persons are encountered before the relation word, and the first person is followed by one of ‘,’ ‘(’, or ‘and’
5	$w P_s [, (\&] P_o$	Indirect case 2: <i>relation subject object</i> . The relation word is encountered before both persons, and the first person is followed by one of ‘,’ ‘(’, or ‘and’

TABLE 5.4: Extraction Rules for matched regular expressions

#	Regex Matched	Extraction Rule
1	$P_s (w P_o)+$	For n occurrences of the pattern (wP_o) matched in pat , extract the triples $(P_s, rel, P_{o1}), (P_s, rel, P_{o2}) \dots (P_s, rel, P_{on})$. Here P_s is the subject, rel is the predicate (and is the relation represented by relation word w), and P_{oi} is the object of the relation triple (P_s, rel, P_{oi}) , where $i \in 1 \dots n$.
2	$P_s d w P_{o1}, P_{o2}, \dots, P_{od}$	Extract d triples: $(P_s, rel, P_{o1}), (P_s, rel, P_{o2}) \dots (P_s, rel, P_{od})$. The subject and the predicate of all these d triples are P_s and rel respectively, but the objects are $P_{o1}, P_{o2} \dots P_{od}$.
3	$P_s w P_o+$	Extract as many triples as the number of P 's that follow w in pat : $(P_s, rel, P_{o1}), (P_s, rel, P_{o2}) \dots$. The subject and the predicate of all these triples are P_s and rel respectively, but the objects are $P_{o1}, P_{o2} \dots$
4	$P_o [, (\&] P_s w$	Extract the triple (P_s, rel, P_o) .
5	$w P_s [, (\&] P_o$	Extract the triple (P_s, rel, P_o) .

TABLE 5.5: Examples of triples extraction

Example text	GeneratedRegex		Extracted Triples
	Pattern	Matched	
Ahmed has a son Ali.	$P w P.$	$P_s (w P_o)+$	(Ahmed, per:children, Ali)
Ali, who is son of Bano	$P, w P$	$P_s (w P_o)+$	(Bano, per:children, Ali)
Bushra's husband Majid, daughter Salma and brother Adil attended the dinner.	$P w P, w P \& w P.$	$P_s (w P_o)+$	(Bushra, per:spouse, Majid) (Bushra, per:children, Salma) (Bushra, per:siblings, Adil)
Ali's four sons, Bilal, Danial, Farhan and Hadi, ...	$P 4 w, P, P, P \& P,$	$P_s d w P_{o1}, P_{o2}, \dots, P_{od}$	(Ali, per:children, Bilal) (Ali, per:children, Danial) (Ali, per:children, Farhan) (Ali, per:children, Hadi)
Ali's children, Bilal, Danial, Farhan and Hadi, ...	$P w, P, P, P \& P,$	$P_s w P_o+$	(Ali, per:children, Bilal) (Ali, per:children, Danial) (Ali, per:children, Farhan) (Ali, per:children, Hadi)
Hadi, Ali's son ...	$P, P w$	$P_o [, (\&) P_s w$	(Ali, per:children, Hadi)
Danial and Farhan, the brothers went to ...	$P \& P, w$	$P_o [, (\&) P_s w$	(Farhan, per:siblings, Danial)
Hadi (Ali's son) came to ...	$P (P w)$	$P_o [, (\&) P_s w$	(Ali, per:children, Hadi)
The brothers, Danial and Farhan, went to ...	$w, P \& P,$	$w P_s [, (\&) P_o$	(Danial, per:siblings, Farhan)
Wife of Hadi, Sana, came to ...	$w P, P,$	$w P_s [, (\&) P_o$	(Hadi, per:spouse, Sana)
Son of Ali (Hadi) inherited ...	$w P (P)$	$w P_s [, (\&) P_o$	(Ali, per:children, Hadi)

(Ahmed, per:children, Ali). To handle this situation, a small list of inverting words has been compiled. For direct cases, if a relation word is followed by any of the inverting words, then the direction of relation is inverted.

The list of triples generated by this module is forwarded to the Implicit Triples Generator module, which uses the Coref and Propagation Rules to further enrich the triples list by including triples for those family relations as well which are implied in \mathbf{t} .

5.3.5 Implicit Triples Generator

For each of the extracted triples, the Implicit Triples Generator module applies Coref Rules to generate triples for co-references of subject and object too. For example, for an extracted triple (A, rel, B) , the triple (C_A, rel, B) is also added to the list of triples, for each co-reference C_A of A , and the triple (A, rel, C_B) is also added to the list, for each co-reference C_B of B .

Next, this module infers new relations from the extracted relation triples by applying a set of Propagation Rules. As an example, from the extracted triples $(X, \text{per:spouse}, Y)$ and $(Y, \text{per:children}, c)$, the relation $(X, \text{per:children}, c)$ is also inferred and added to the set of extracted relations. These rules also add the inverse of extracted relations, so for the example just discussed, the relations $(Y, \text{per:spouse}, X)$, $(c, \text{per:parents}, Y)$, and $(c, \text{per:parents}, X)$ are also added. The triple set is now in its final form.

An example run of the main system for input text “Kollek is survived by his widow Tamar, son Amos and daughter Osnat.” is given in Fig. 5.2. The main rules of the system are also given in Algorithm 2.

5.4 Evaluation Results and Analysis

The system CustRE has been developed in Python using Pycharm’s community edition. The baseline Stanford KBP relation annotator [55] has been installed using the Python wrapper, `pycorenlp`³, and its relation extractor results have been obtained on evaluation datasets. The second baseline is the deep learning system

³<https://pypi.org/project/pycorenlp/>

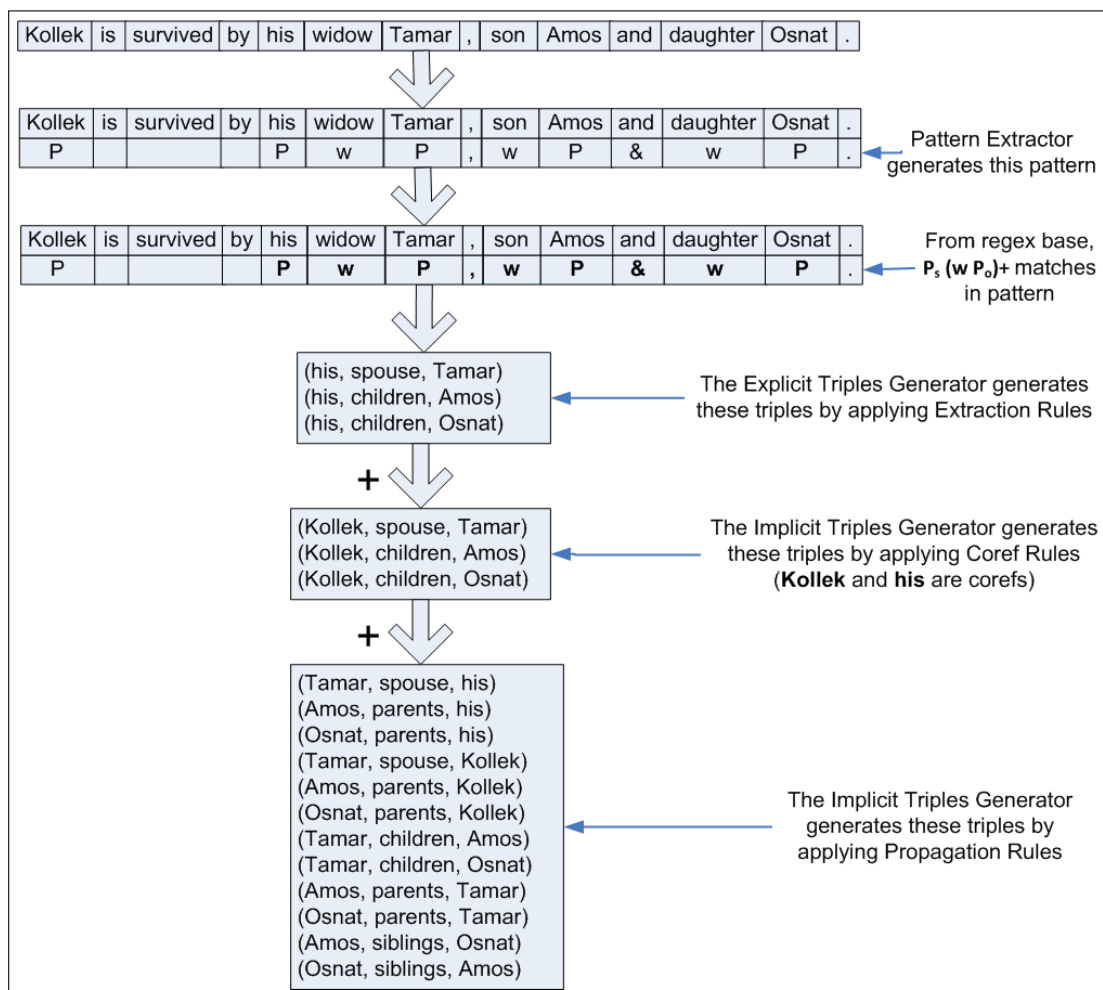


FIGURE 5.2: An example run of CustRE for the input “Kollek is survived by his widow Tamar, son Amos and daughter Osnat.”

by Zhang et al. [2], who are the authors of TACRED dataset [129]. This system has been henceforth referred as TACRED-PA. Its pytorch implementation has been downloaded from github⁴, and trained on TACRED-F dataset. Implementations for the baselines SpanBert [42] and LUKE [56] have also been downloaded from github⁵, which contain pre-trained models. These models have been fine-tuned on TACRED-F dataset. Per relation and aggregate precision, recall and F1 (micro) scores for CustRE and the baseline systems have been generated using TACRED scorer script. The result percentages of systems’ performance on enhanced TACRED-F test set have been summarized in Table 5.6. The results of the system are quite encouraging. CustRE scores highest compared to existing

⁴<https://github.com/yuhaozhang/tacred-relation>

⁵<https://github.com/facebookresearch/SpanBERT>, <https://github.com/studio-ousia/luke>

Algorithm 2 CustRE Rules: Main

```

1: pattern ← ExtractPattern(inputText, RelationWordsLists)
2: regexNo ← 1
3: for each regex in regexs do
4:   matches = getMatches(regex, pattern) //gets all non-overlapping matches
   of regexs in pattern
5:   for each match in matches do
6:     i ← 0
7:     d ← 0
8:     n ← len(match)
9:     ADDTRIPLES(match, i, d, n)
10:  end for
11:  regexNo ++
12: end for
13: addImplicitTriples(triples) //adds transitive, reverse and coref triples

```

TABLE 5.6: Results Comparison for FRE task on TACRED-F test set

System	Evaluation Metric	Family Relation Class					Average (micro)
		Children	Other family	Parents	Siblings	Spouse	
Stanford	Precision	0.0	33.3	3.6	78.5	80.3	39.6
	Recall	0.0	3.6	0.9	18.9	17.4	8.4
	F1 score	0.0	6.5	1.5	30.5	28.6	13.9
TACRED-PA	Precision	36.7	27.8	33.0	48.7	40.8	37.9
	Recall	16.3	55.8	44.3	61.9	59.7	46.0
	F1 score	22.6	37.1	37.8	54.5	48.5	41.6
SpanBert	Precision	23.6	21.8	31.0	58.6	55.9	36.3
	Recall	29.5	59.4	49.1	64.1	56.6	50.2
	F1 score	26.2	31.8	38.0	61.2	56.3	42.1
LUKE	Precision	76.5	24.9	67.4	72.9	64.0	61.7
	Recall	69.6	48.6	72.9	61.9	60.9	64.7
	F1 score	72.9	32.9	70.0	66.9	62.4	64.7
CustRE	Precision	73.6	88.4	73.7	76.9	61.6	72.0
	Recall	69.6	82.6	69.3	59.3	69.7	68.8
	F1 score	71.5	85.4	71.4	67.0	65.4	70.4

systems, on average 30.2 F1 points higher than others. Among the five systems, CustRE achieves the highest F score overall and for other_family, parents and spouse relation classes. For children and siblings classes, Luke performs the best.

Algorithm 2 CustRE Rules: Add Triples

```

1: function ADDTRIPLES(match, i, d, n)
2:   if regexNo is 1 or 2 or 3 then
3:     subj ← match[i].text
4:   else if regexNo is 4 then
5:     obj ← match[i].text
6:   else if regexNo is 5 then
7:     pred ← match[i].text
8:     rel ← getFamRel(pred) //maps pred to vocabulary
9:   end if
10:  i ++
11:  while i ≠ n do
12:    if regexNo is 1 or 2 or 3 then
13:      if match[i] is a number then
14:        d ← match[i]
15:      else if match[i] = 'w' then
16:        pred ← match[i].text
17:        rel ← getFamRel(pred)
18:      else if match[i] = 'P' then
19:        obj ← match[i].text
20:        if ((regexNo is 1 or 3) or (regexNo is 2 and d > 0)) then
21:          Add (subj, rel, obj) to triples
22:          if d > 0 then
23:            d --
24:          end if
25:        end if
26:      end if
27:    else if regexNo is 4 then
28:      if match[i] = 'P' then
29:        subj ← match[i].text
30:      else if match[i] = 'w' then
31:        pred ← match[i].text
32:        rel ← getFamRel(pred)
33:        Add (subj, rel, obj) to triples
34:      end if
35:    else if regexNo is 5 then
36:      if match[i] = 'P' then
37:        subj ← match[i].text
38:        if i > 1 then
39:          if match[i] = 'P' then
40:            obj ← match[i].text
41:            Add (subj, pred, obj) to triples
42:          end if
43:        end if
44:      end if
45:    end if
46:    i ++
47:  end while
48: end function

```

Stanford’s system performs the worst among all and surprisingly has not been able to extract correctly even a single instance of children relation, thus an F1 score of 0. A comparison of F1 scores of the systems is given in Fig. 5.3. It can be seen that CustRE clearly outperforms all baselines. One surprising observation is that the each baseline system has different F1 score for parents and children relations, which are simply inverse of each other. If X has child Y , then Y has parent X . So if a system is able to extract parent or child relation in one direction, it is not difficult to reverse that relation and extract the second of the two relations. But surprisingly, the ML based systems have failed to do this.

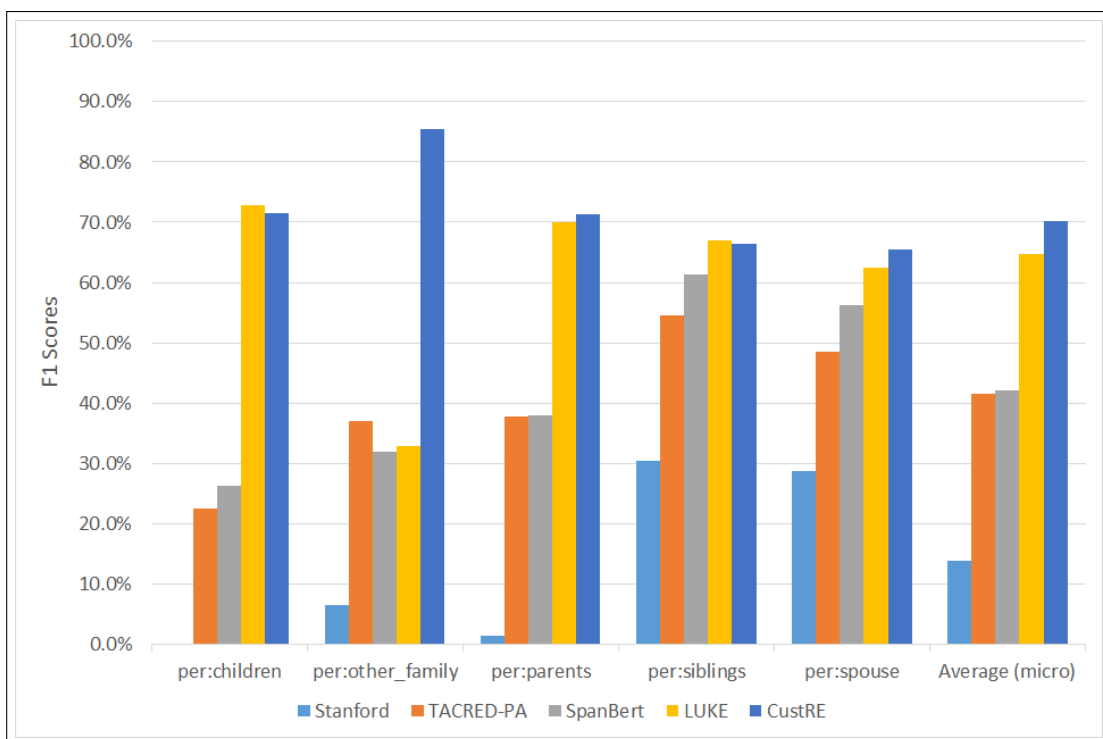


FIGURE 5.3: F1 Scores Comparison of FRE task on TACRED-F test set

Since it was hypothesized that English grammar does not change, and regular patterns for family relations exist in natural language texts, therefore CustRE has been further evaluated on a different dataset (CustFRE [116]) for which rules have not been built. CustRE extracts family relations from any text, and does not need redesigning to extract family relations from a new dataset. The result percentages of systems’ performance on the new dataset are given in Table 5.7.

TABLE 5.7: Results Comparison for FRE task on CustFRE evaluation dataset

System	Evaluation Metric	Family Relation Class					Average (micro)
		Children	Other family	Parents	Siblings	Spouse	
Stanford	Precision	6.8	45.8	1.8	58.3	87.5	43.0
	Recall	1.2	7.3	0.3	14.9	24.0	8.6
	F1 score	2.0	12.6	0.5	23.7	37.6	14.3
TACRED-PA	Precision	46.3	22.5	42.7	43.3	42.3	41.5
	Recall	26.8	8.9	40.6	45.7	63.9	36.2
	F1 score	33.9	12.8	41.7	44.5	50.9	38.7
SpanBert	Precision	34.0	33.2	31.5	49.2	49.4	40.0
	Recall	32.9	33.8	33.7	66.7	73.4	46.3
	F1 score	33.4	33.5	32.6	56.6	59.0	42.9
LUKE	Precision	64.4	23.8	65.4	55.1	61.9	59.6
	Recall	65.7	8.0	77.8	51.8	78.0	56.7
	F1 score	65.1	11.9	71.1	53.4	69.0	58.1
CustRE	Precision	84.2	72.1	84.0	84.4	73.2	79.7
	Recall	75.5	62.6	74.4	73.1	84.0	73.7
	F1 score	79.6	67.0	78.9	78.3	78.2	76.6

Again CustRE system’s F1 score is highest, 76.6%, which is 18.5% better than LUKE, 33.7% higher than SpanBert, 37.9% higher than TACRED-PA, and 62.3% higher than Stanford. On average CustRE F1 is 38.1 points better than others. When the performance of systems on TACRED-F is compared to that on a new unseen dataset, i.e. CustFRE, the proposed system’s performance has raised by 6.3 F1 points, because the system is designed keeping in view domain knowledge and is not dataset specific. No significant change is observed in F1 scores of Stanford and Spanbert, whereas the performance of remaining two systems drops significantly on the new dataset. LUKE’s performance came down by 6.6 F1 points, and Taced-PA’s by 2.9 points. Thus these machine learning (ML) based systems which learned/tuned on TACRED dataset, are found not to perform well on a new dataset (CustFRE).

The proposed rule-based family relation extraction system has outperformed existing ML-based systems and the performance gap is significant. This is because rules have been designed keeping in view domain knowledge, and are not specific to any training dataset. Whereas ML systems learn from training data, which many times have many mistakes as well [117], resulting in the system learning those incorrect examples too. But this is not the case with rule-based systems, as rule-based system is designed by human expert keeping in view the domain knowledge, which is correct.

Moreover, the family relations domain is not changing, the rules of the domain are fixed and already exist in natural language texts, we only need to identify them and make rules to recognize them, so it is better to make a rule-based system. But for other domains that are changing, rule-based systems designed by domain experts might not suffice, an ML-based system might perform better for changing domains. Furthermore, the rule-based approach dominates the commercial market despite being largely ignored by the research community, as concluded by researchers from International Business Machines Corporation (IBM), Chiticariu et al. [130], after surveying IE systems from industry and literature. They suggest the academic NLP community to stop treating rule-based IE as a dead-end technology, because rule-based systems serve the industry needs better than the latest ML techniques, because of their interpretability and ability to easily incorporate domain knowledge specific to the problem.

The complexity of FRE task generally increases as the number of persons and hence the number of possible relations increases in text. To see how the performance of systems is affected as the complexity of task increases, F1 scores of systems on CustFRE dataset have been drawn for different number of persons in Fig. 5.4. It can be seen that performance of all systems degrades as the number of persons in input text increase, though the degrade rate of the proposed system is lowest compared to others.

When the output of the top three performers on CustFRE dataset is analyzed, see Table 5.8, it is found that most of the times Luke and SpanBert mark some

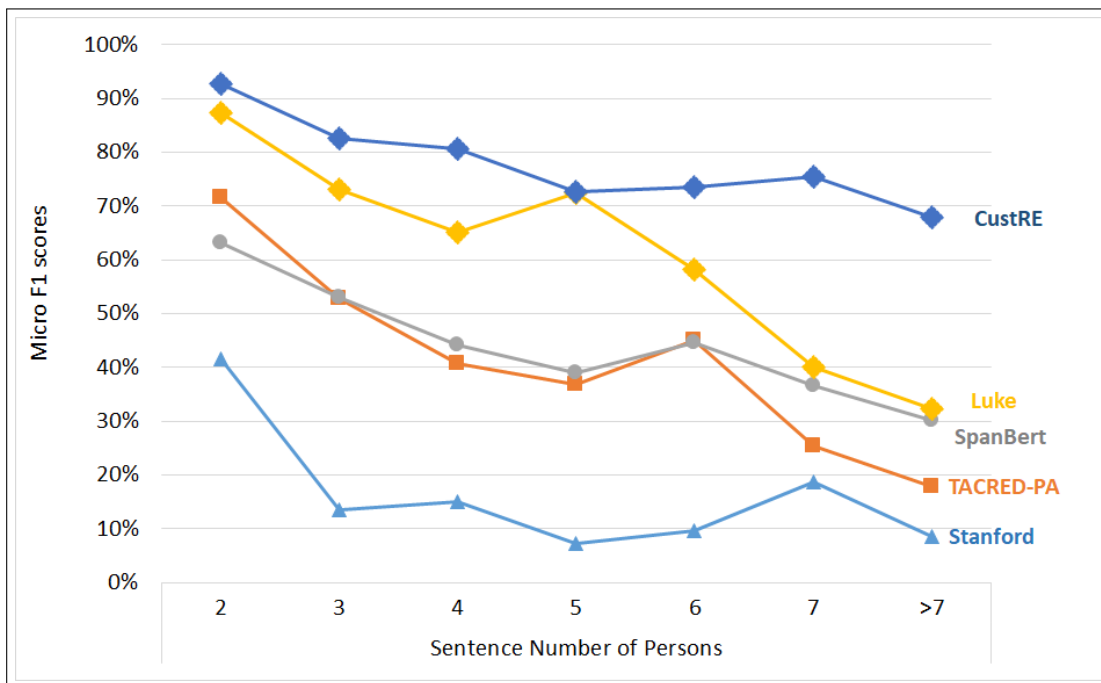


FIGURE 5.4: F1 Comparison with respect to number of persons in sentence

annotations in a sentence correct and some incorrect (for 138 and 156 sentences respectively, out of a total of 247 sentences). While CustRE most of the times marks all annotations in a sentence correctly (for 168 sentences). This could be because all annotations of a sentence are not usually marked in training datasets on which ML systems learn.

TABLE 5.8: Output analysis of three top performing systems

No. of Sentences where:	CustRE	LUKE	SpanBert
All annotations of a sentence are marked correct	168	99	52
All annotations of a sentence are marked incorrect	8	10	39
Some annotations of a sentence are marked correct and some are marked incorrect	71	138	156

Moreover, very surprisingly, many times the DL systems mark correctly the relations in one direction, but the inverted relation is incorrectly annotated, or a relation which can be simply inferred from two already annotated relations is incorrectly marked. For instance, consider annotations by the three systems for

the input sentence “**She** had a daughter, **Maureen** in 1941 and adopted a son, **Michael** in 1945.”, given in Table 5.9.

TABLE 5.9: Annotations by three top performing systems for input sentence “**She** had a daughter, **Maureen** in 1941 and adopted a son, **Michael** in 1945.”

No.	(Subject, Object)	Correct Annotation	Annotation by System:		
			CustRE	LUKE	SpanBert
1	(She, Maureen)	per:children	per:children	per:children	per:children
2	(She, Michael)	per:children	per:children	per:children	per:children
3	(Maureen, She)	per:parents	per:parents	per:parents	per:children
4	(Maureen, Michael)	per:siblings	per:siblings	per:children	per:children
5	(Michael, She)	per:parents	per:parents	per:parents	per:children
6	(Michael, Maureen)	per:siblings	per:siblings	per:children	per:children

Here, first two relations are correctly marked by all three systems. Relation number 3 and 5, which are simply inverse of first two, are marked incorrectly by SpanBert. Relation number 4 and 6, which can be simply inferred from relation number 1 and 2 (1 and 2 are already correctly detected by all systems), are marked incorrectly by both Luke and SpanBert. As ML systems are a black box, it could not be found out why they are making such mistakes. The proposed system has simple rules for such cases though, when a relation is detected, its inverted relation is also annotated, and existing relations are propagated to find new relations (*She child Maureen* and *She child Michael* propagate to annotate *Maureen sibling Michael*), hence CustRE does not make such mistakes.

So rules made by identifying patterns on TACRED-F dataset perform well on CustFRE dataset too. CustRE is implemented on a simple personal computer. The deterministic algorithm only takes 23 seconds to produce results on TACRED-F test set where LUKE takes 260 seconds to produce results on the same dataset in same settings. A comparison of the time taken by the two systems to produce output on evaluation datasets is given in Table 5.10. Compared to LUKE, CustRE takes 11 times less time to produce results on TACRED-F evaluation dataset, and 15 times less time on CustRE evaluation dataset, in same settings.

TABLE 5.10: Comparison of empirical run time for FRE task

Dataset	Time taken by CustNER	Time taken by LUKE
TACRED-F	23 seconds	260 seconds
CustFRE	20 seconds	297 seconds

Thus the output of proposed system can be used for construction of more accurate and complete KGs. An example KG for text “Kollek is survived by his widow Tamar, son Amos and daughter Osnat.” is given in Fig. 5.5. For this text, the proposed system module CustNER recognizes the person NEs; *Kollek*, *Tamar*, *Amos* and *Osnat*, and the proposed system module CustRE extracts the following family relations between these person NEs:

(Kollek, per:spouse, Tamar)
 (Kollek, per:children, Amos)
 (Kollek, per:children, Osnat)
 (Tamar, per:spouse, Kollek)
 (Tamar, per:children, Amos)
 (Tamar, per:children, Osnat)
 (Amos, per:parents, Kollek)
 (Amos, per:parents, Tamar)
 (Amos, per:siblings, Osnat)
 (Osnat, per:parents, Kollek)
 (Osnat, per:parents, Tamar)
 (Osnat, per:siblings, Amos)

These triples can be saved in RDF, and can be visualized as a KG, as in Fig. 5.5. Once data is converted to knowledge graph, precise queries can be made on it by using SPARQL and the relevant information can be precisely extracted. For instance, such queries like “brother of Osnat” can be precisely answered from KG.

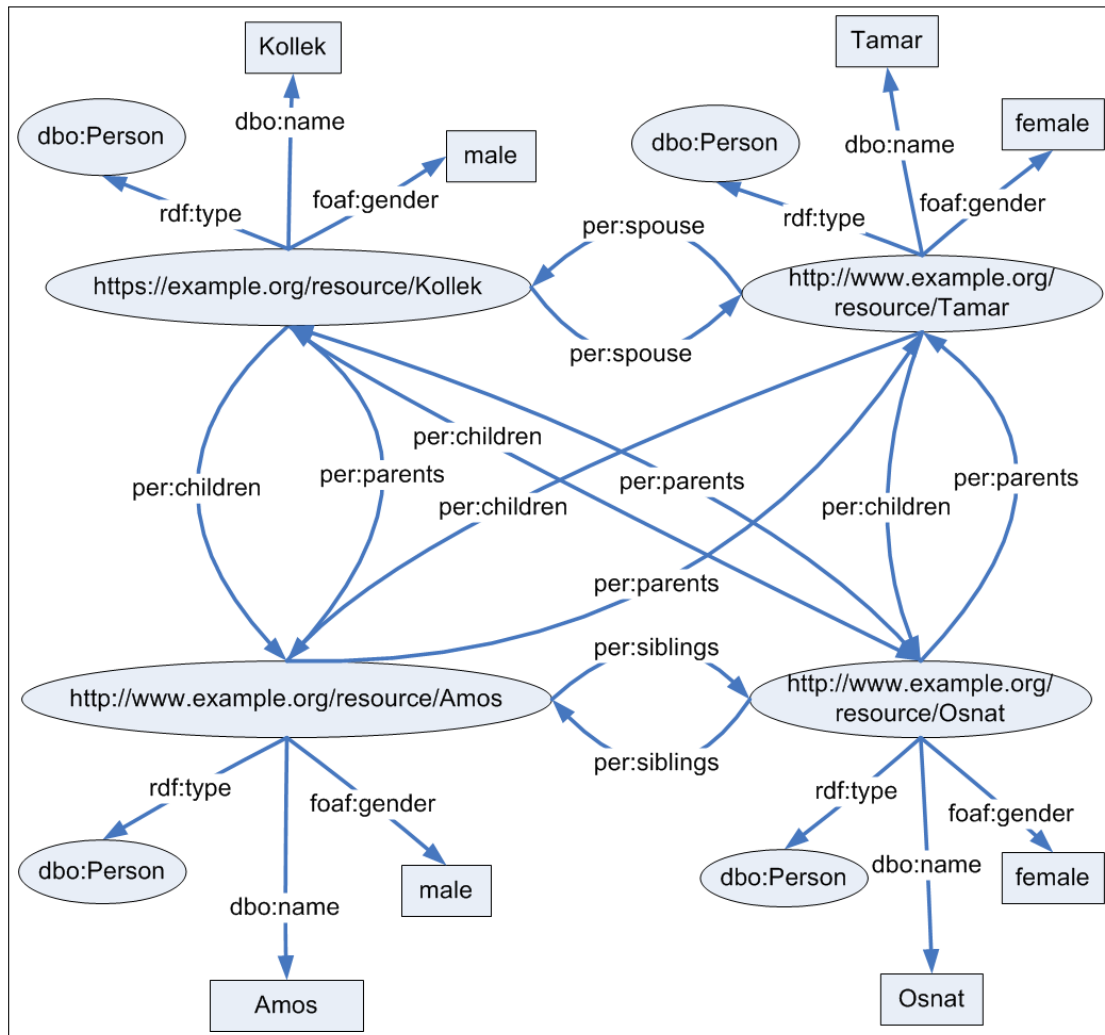


FIGURE 5.5: KG for text “Kollek is survived by his widow Tamar, son Amos and daughter Osnat.”

5.5 Analysis of Errors

The incorrect annotations that the proposed system makes have been analyzed. The most frequent errors are the following ones:

5.5.1 Coref Errors

Many times the errors of the system are because of incorrect anaphora resolution. Anaphora resolution is an essential step in relation extraction, for which CustRE depends on an external system. The errors of the co-reference resolution system get propagated to CustRE. Consider this text,

*Pratt also has a 7-year-old son with **his** first wife, **Anna Faris**.*

For this text, CustRE correctly extracts the spouse relation between *his* and *Anna Faris*. But since the corefs *Pratt* and *his*, are not detected corefs by the coref system, hence the spouse relation of *Anna Faris* to *his* does not extend to *Pratt*, resulting in in-correct annotation (*Pratt, not_known, Anna Faris*).

Improving the performance of co-reference resolution would avoid such errors and improve the performance of the proposed system.

5.5.2 Overlapped Triple Errors

There are cases where subject or object word is missing in the input text, and the relation word is predicate as well as object/subject. Consider this text,

***Her husband** never intended to harm the government or the people of Cuba.*

Here, the relation between *Her* and *husband* is spouse relation, the relation word *husband* is also object of relation triple. Such cases are not handled in present work. They will be dealt in future work.

5.5.3 Other Errors

There are other errors that occur because the rules do not always generate the correct relation. There are situations where the rules fail and the correct relation is not annotated by the system. Consider this example,

*She and **her** husband would drop **him** off at a kindergarten in the morning and
pick him up after work.*

Here, the system would annotate spouse relation between *her* and *him*, because of the *husband* relation word appearing between them. The correct annotation is *not_known*. In future, it would be explored if sentence dependency parse or constituency parse might help with dealing such situations.

5.6 Summary

This chapter detailed the system module, CustRE, for better extraction of family relations from any generic English text. Regular patterns of family relations have been identified by analyzing English language texts from TACRED-F dataset. Rules have been formulated against those patterns to extract family relations from text, those which are explicitly mentioned in text, as well as those which are only implied in the text. Great care has been taken while designing rules to make them generic. To check this, the rule-based system has been tested on un-seen dataset too, without re-training. The results have been compared with SOTA models, proving that the designed system performs well on FRE. CustRE outperforms the baseline systems by achieving an F1 score 30.2 points higher than others on average, on TACRED-F dataset, and 38.1 points higher on unseen CustFRE dataset. When evaluated on new unseen dataset, the performance of CustRE is even better, whereas the F1 scores of all the baselines have either remained unchanged or have dropped, for instance LUKE's performance came down by 6.6 points. Thus it cannot be said that ML systems trained on one dataset, can perform the same when a new dataset is encountered. The proposed system is able to correctly recognize the relations missed by existing systems given in Table 1.3. This chapter has thus addressed the second research question and has successfully achieved the second research objective (both re-stated below).

RQ2: How to formulate rules to extract family relations which are incorrectly extracted by existing RE systems?

RO2: Devise a technique for better extraction of family relations from text.

Chapter 6

Conclusion and Future Work

In this dissertation, it has been argued that effective extraction of NEs and relations from text is crucial for populating KG from presently un-structured text on the web, and a system has been proposed for doing it. With the proposed system for improved information extraction, knowledge graph can be more accurately and more completely populated. Relevant literature is first thoroughly reviewed and benchmark datasets and available existing systems have been identified. It was found that performance of existing systems is not adequate for effective population of KG. The available datasets were also found erroneous and incomplete. A criteria was therefore devised, by conducting focus group with researchers, for assessment of IE dataset quality. It was observed that dataset annotators lack a clear understanding of valid annotation, therefore valid annotations for datasets were also formally defined. Benchmark datasets were assessed and enhanced with the devised criteria. The enhanced (corrected and completed) datasets were used for evaluating and comparing the system.

Since performance of existing systems was not found satisfactory, an IE system for better extraction of entities and relation has been proposed. Regular patterns for PER, LOC, ORG NEs and family relations already exist in natural language texts, which have been identified by analyzing training datasets, and rules have been formulated to recognize the regular patterns. The system has two major modules; CustNER and CustRE. CustNER performs recognition of PER, LOC,

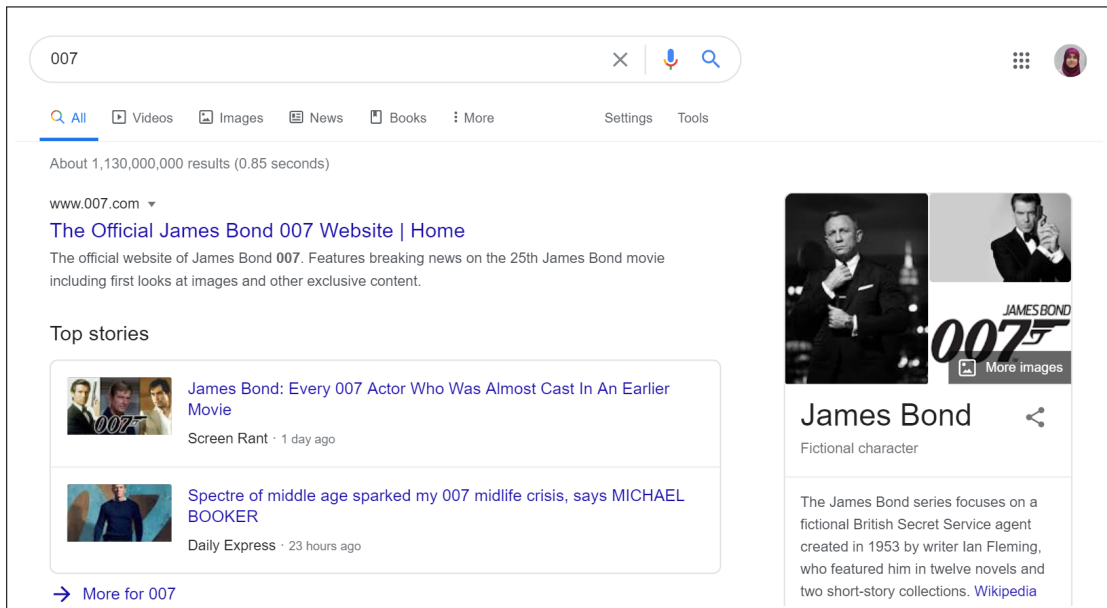
ORG NEs, and classifies PER NEs as male or female. It was noticed that the kind of NEs missed from annotation by existing systems (the false negatives) are mostly similar. CustNER therefore utilizes output of existing systems and knowledge from DBpedia, and works on recognizing these missed NEs. Rules have been designed for specifically recognizing the missed NE classes. Evaluation of CustNER on OKE dataset (whose source is mainly web) reveals that the proposed system is able to recognize the NEs missed by other systems, by making a recall of 78.89% and an F1 score of 81.03%. To verify that the rule-based system is not dataset specific, it has been evaluated on an unseen dataset which is based on a different source, i.e. the CoNLL dataset (whose source is newswire). Again the proposed system performed reasonably well, with an F1 measure 94.67%. Moreover, CustNER is a deterministic time efficient system and takes at least 3 times less time to produce results as compared to its DL counterpart, LUKE. Through CustNER thus, the RO1 has been successfully met. The results suggest that incorporating knowledge from DBpedia and making rules specific to the kind of entities missed, can improve recognition of generally missed NEs.

The module CustRE extracts family relations between persons and classifies them into; children, parents, siblings, spouse and other family classes. If no family relation between two persons is inferred from the input text, it is annotated as *not_known*. It was observed that family relations usually appear in fix kinds of syntactic patterns in text. Common patterns in which family relations appear in text were identified and rules were formulated for identifying such patterns and extracting relations accordingly. This approach has worked very well, as is apparent from CustRE's performance for FRE on the well-known and widely used TACRED dataset (F1 score 70.4%). This is a great improvement over existing systems, with at least 5.7 F1 points higher than other systems. On another unseen evaluation dataset, CustFRE dataset, the system performed F1 score 76.6%, which is at least 18.5 F1 points higher than existing systems. Moreover, CustRE is a deterministic system and takes at least 11 times less time to produce results on evaluation datasets compared to its DL counterpart, LUKE. Through CustRE thus, the RO2 has been successfully met. It is important to note here that the

performance gap between CustRE and baselines is significant, because the performance of ML systems which performed well on TACRED, fell down greatly on the un-seen dataset. Thus the generally claimed assertion that, ML and DL systems trained on one dataset can perform the same when a new dataset is encountered, is found to be false.

Analyzing the errors made by the system, it is found that CustNER needs rules for correctly identifying ambiguous NEs such as location names representing organizations. To avoid tying errors, context of NEs should also be considered. The DBpedia linking part also has margin for improvement by devising a scheme which incorporates semantic similarity of concepts based on their properties and can associate a surface form from text to relevant DBpedia resource even when their labels do not exactly match. Instead of designing a new partial match algorithm, Google's algorithm may be utilized. For example giving a query *wikipedia: The German federal prosecutor's office* to google gives *Public Prosecutor General (Germany)*, which can in turn be used to retrieve the correct DBpedia resource. Google may also assist in identifying some other NEs which none of the NERs is able to identify, for example *007*. Google result of the query *007* is given in Figure 6.1. A human can see from this result that *007* refers to the person James Bond. A mechanism needs to be devised that utilizes google results, or google knowledge panel giving precise information about query towards top right, to interpret the entity's type. CustNER also does not go further when DBpedia returns a disambiguation page instead of a resource page. A scheme needs to be developed that decides which resource from the disambiguation list corresponds to the concerned NE. These limitations will be worked on in future.

The annotation mistakes made by module CustRE are mostly propagated from incorrect co-reference resolution. Improving co-reference resolution would thus improve the system performance. Some errors occur when the subject or object does not appear as a separate word, but the same word represents relation as well as subject/object. Rules need to be devise for extraction of such overlapped triples. Moreover, the rules sometimes fail to make correct annotation. Overcoming these limitations will be worked on in future. The possibility of incorporating sentence

FIGURE 6.1: Google result for query *007*

dependency parse and constituency parse would also be explored to further improve the performance of the rules.

This dissertation has handled extraction of PER, LOC, ORG NEs from text, and extraction of family relations among persons. Correct extraction of NEs and relations from text is important for structuring information as KG, which is in turn important for correctly answering precise queries. Extraction of family relations holds great importance in construction of family graphs in biomedical domain, which is important to assess the risk of inherited medical conditions and to improve patient care and decision making [50, 51, 53, 54]. Many times, existing systems fail to make correct extractions even in seemingly straightforward cases [2]. With the proposed system for improved IE, KG and family trees can be more accurately and more completely populated. The proposed system is not restricted to the datasets discussed in this work, and is able to extract NEs and family relations from any general texts, as has been demonstrated by good results of system on unseen datasets too. In future, the rule-based technique would be extended to extract other kinds of relations among persons too, and afterwards relations between other types of NEs too (i.e. LOC and ORG).

Bibliography

- [1] H. Paulheim, “Knowledge graph refinement: A survey of approaches and evaluation methods,” *Semantic web*, vol. 8, no. 3, pp. 489–508, 2017.
- [2] Y. Zhang, V. Zhong, D. Chen, G. Angeli, and C. D. Manning, “Position-aware Attention and Supervised Data Improve Slot Filling,” in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 35–45, 2017.
- [3] R. Grishman and B. Sundheim, “Message Understanding Conference - 6 : A Brief History,” in *The 16th International Conference on Computational Linguistics.*, 1996.
- [4] N. A. Chinchor, “OVERVIEW OF MUC-7 / MET-2 Overviews of English and Multilingual Tasks,” in *Proceedings of the Message Understanding Conference (MUC-7)*, 1998.
- [5] E. T. K. Sang and F. De Meulder, “Introduction to the conll-2003 shared task: Language-independent named entity recognition,” in *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pp. 142–147, 2003.
- [6] G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel, “The Automatic Content Extraction (ACE) Program Tasks , Data , and Evaluation,” in *Conference on Language Resources and Evaluation*, 2004.

-
- [7] P. McNamee and H. Dang, “Overview of the TAC 2009 knowledge base population track,” in *Text Analysis Conference (TAC)*, vol. 17, pp. 111–113, 2009.
- [8] H. Ji, R. Grishman, H. T. Dang, K. Griffitt, and J. Ellis, “Overview of the tac 2010 knowledge base population track,” in *Third text analysis conference (TAC 2010)*, 2010.
- [9] G. Giannakopoulos, M. El-Haj, B. Favre, M. Litvak, J. Steinberger, and V. Varma, “Tac 2011 multiling pilot overview,” in *Proceedings of Text Analysis Conference (TAC2011)*, 2011.
- [10] H. Ji, J. Nothman, B. Hachey, *et al.*, “Overview of tac-kbp2014 entity discovery and linking tasks,” in *Proceedings of Text Analysis Conference (TAC2014)*, pp. 1333–1339, 2014.
- [11] C. Schmitt, V. Walker, A. Williams, A. Varghese, Y. Ahmad, A. Rooney, and M. Wolfe, “Overview of the tac 2018 systematic review information extraction track.,” in *Proceedings of Text Analysis Conference TAC*, 2018.
- [12] H. Ji, A. Sil, H. T. Dang, I. Soboroff, J. Nothman, and S. I. Hub, “Overview of tac-kbp 2019 fine-grained entity extraction.,” in *Proceedings of Text Analysis Conference TAC*, 2019.
- [13] R. Speck, M. Roder, S. Oramas, L. Espinosa-Anke, and A.-C. N. Ngomo, “Open Knowledge Extraction Challenge 2017,” *Semantic Web Evaluation Challenge*, pp. 35–48, 2017.
- [14] H. Zhang, F. Boons, and R. Batista-navarro, “Whose story is it anyway? Automatic extraction of accounts from news articles,” *Information Processing and Management*, no. February, pp. 1–12, 2019.
- [15] D. Nagalavi and M. Hanumanthappa, “The NLP Techniques for Automatic Multi-article News Summarization Based on Abstract Meaning Representation,” *Emerging Trends in Expert Applications and Security*, pp. 253–260, 2019.

-
- [16] L. A. Pizzato, D. Moll, and C. Paris, “Pseudo Relevance Feedback Using Named Entities for Question Answering,” in *Australian Language Technology Workshop*, pp. 83–90, 2006.
- [17] D. Nagrale, V. Khatavkar, and P. Kulkarni, “Document Theme Extraction Using Named-Entity Recognition,” *Computing, Communication and Signal Processing*, pp. 499–509, 2019.
- [18] T. Ma, H. Zhou, Y. Tian, and N. Al-Nabhan, “A novel rumor detection algorithm based on entity recognition, sentence reconfiguration, and ordinary differential equation network,” *Neurocomputing*, vol. 447, pp. 224–234, 2021.
- [19] S. Ganguly, *Exploiting Textual Content in Academic Citation Networks*. PhD thesis, Doctoral dissertation, International Institute of Information Technology Hyderabad, 2019.
- [20] K. Humphreys, M. Calcagno, and K. Powell, “Creating a document index from a flex-and Yacc-generated named entity recognizer,” 2006.
- [21] S. Ananiadou, D. B. Kell, and J.-i. Tsujii, “Text mining and its potential applications in systems biology,” *TRENDS in Biotechnology*, vol. 24, no. 12, 2006.
- [22] B. Settles, “Biomedical Named Entity Recognition Using Conditional Random Fields and Rich Feature Sets,” in *International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, pp. 107–110, 2004.
- [23] A. Pal and A. Mustafi, “Vartani spellcheck – automatic context-sensitive spelling correction of ocr-generated hindi text using bert and levenshtein distance,” 2020.
- [24] S. Remstam, “A novel low annotation-cost interactive framework for named entity recognition,” Master’s thesis, KTH, School of Electrical Engineering and Computer Science (EECS), 2020.

- [25] A. Goyal, V. Gupta, and M. Kumar, “Recent Named Entity Recognition and Classification techniques : A systematic review,” *Computer Science Review*, vol. 29, pp. 21–39, 2018.
- [26] S. Sarawagi, “Information Extraction,” *Foundations and Trends® in Databases*, vol. 1, no. 3, pp. 261–377, 2008.
- [27] J. R. Finkel, T. Grenager, and C. Manning, “Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling,” in *43rd annual meeting on association for computational linguistics*, 2005.
- [28] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, “The stanford corenlp natural language processing toolkit,” in *Association for Computational Linguistics (ACL) System Demonstrations*, pp. 55–60, 2014.
- [29] L. Ratinov and D. Roth, “Design Challenges and Misconceptions in Named Entity Recognition,” in *Computational Natural Language Learning*, pp. 147–155, 2009.
- [30] T. Redman, M. Sammons, and D. Roth, “Illinois Named Entity Recognizer: Addendum to Ratinov and Roth ’09 reporting improved results,” tech. rep., University of Illinois, Urbana-Champaign, USA, 2016.
- [31] R. Speck and A.-C. C. N. Ngomo, “Ensemble learning of named entity recognition algorithms using multilayer perceptron for the multilingual web of data,” in *Proceedings of the Knowledge Capture Conference, K-CAP 2017*, pp. 10–13, 2017.
- [32] J. Plu, R. Troncy, and G. Rizzo, “ADEL @ OKE 2017 : A Generic Method for Indexing Knowledge Bases for Entity Linking,” *Semantic Web Evaluation Challenge*, pp. 49–55, 2017.
- [33] J. Plu, G. Rizzo, and R. Troncy, “Enhancing Entity Linking by Combining NER Models,” *Semantic Web Evaluation Challenge*, vol. 1, pp. 17–32, 2016.

- [34] N. Bach and S. Badaskar, “A Review of Relation Extraction,” *Literature review for Language and Statistics II*, vol. 2, pp. 1–15, 2007.
- [35] F. Suchanek, G. Kasneci, G. Weikum, F. Suchanek, G. Kasneci, G. Weikum, Y. A. Core, F. M. Suchanek, and G. Weikum, “Yago : A Core of Semantic Knowledge Unifying WordNet and Wikipedia,” in *16th International World Wide Web Conference*, pp. 697–706, 2007.
- [36] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, “Freebase : A Collaboratively Created Graph Database For Structuring Human Knowledge,” in *SIGMOD*, pp. 1247–1249, 2008.
- [37] A. Fader, S. Soderland, and O. Etzioni, “Identifying Relations for Open Information Extraction,” in *conference on empirical methods in natural language processing*, pp. 1535–1545, 2011.
- [38] O. Etzioni, A. Fader, J. Christensen, and S. Soderland, “Open Information Extraction : The Second Generation,” in *Twenty-Second International Joint Conference on Artificial Intelligence*, pp. 3–10, 2011.
- [39] A. Yates, M. Banko, M. Broadhead, M. Cafarella, O. Etzioni, and S. Soderland, “TextRunner : Open Information Extraction on the Web,” in *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics*, no. April, pp. 25–26, 2007.
- [40] P. Gamallo, M. Garcia, and S. Fern, “Dependency-Based Open Information Extraction,” in *13th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 10–18, 2012.
- [41] G. Angeli, M. Johnson, and P. Christopher, “Leveraging Linguistic Structure For Open Domain Information Extraction,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 344–354, 2015.

- [42] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, “Spanbert: Improving pre-training by representing and predicting spans,” *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 64–77, 2020.
- [43] I. Efremova, B. Ranjbar-Sahraei, F. Oliehoek, T. Calders, and K. Tuyls, “Investigation of a baseline method for genealogical entity resolution,” in *Proceedings of the Workshop on Population Reconstruction, Organized in the Framework of the LINKS Project*, International Institute for Social History IISH, 2014.
- [44] J. Efremova, A. Montes García, A. B. Iriondo, and T. Calders, “Who are my ancestors? Retrieving family relationships from historical texts,” in *Communications in Computer and Information Science*, vol. 573, pp. 121–129, Springer Verlag, 2016.
- [45] J. Efremova, A. M. García, J. Zhang, and T. Calders, “Towards population reconstruction: extraction of family relationships from historical documents,” in *First International Workshop on Population Informatics for Big Data (21th ACM-SIGKDD PopInfo’15)*, pp. 1–9, 2015.
- [46] D. Kokkinakis and M. Malm, “Character Profiling in 19th Century Fiction,” in *Proceedings of Language Technologies for Digital Humanities and Cultural Heritage Workshop*, no. September, (Hissar, Bulgaria), pp. 70–77, 2011.
- [47] A. Makazhanov, D. Barbosa, and G. Kondrak, “Extracting family relationship networks from novels,” *arXiv preprint arXiv:1405.0603*, 2014.
- [48] K. Janakiraman, “Extracting Character Relationships From Stories,” in *Proceedings of the Tenth Annual AAAI conference on AIIDE*, 2014.
- [49] V. Devisree and P. C. R. Raj, “A hybrid approach to relationship extraction from stories,” *Procedia Technology*, vol. 24, pp. 1499–1506, 2016.

- [50] Y. Kim, P. M. Heider, I. R. Lally, and S. M. Meystre, “A hybrid model for family history information identification and relation extraction: Development and evaluation of an end-to-end information extraction system,” *JMIR Med Inform*, vol. 9, p. e22797, Apr 2021.
- [51] J. F. Silva, J. R. Almeida, and S. Matos, “Extraction of family history information from clinical notes: Deep learning and heuristics approach,” *JMIR Med Inform*, vol. 8, p. e22898, Dec 2020.
- [52] X. Yang, H. Zhang, X. He, J. Bian, and Y. Wu, “Extracting family history of patients from clinical narratives: Exploring an end-to-end solution with deep learning models,” *JMIR Med Inform*, vol. 8, p. e22982, Dec 2020.
- [53] K. He, J. Wu, X. Ma, C. Zhang, M. Huang, C. Li, and L. Yao, “Extracting kinship from obituary to enhance electronic health records for genetic research,” in *Proceedings of the Fourth Social Media Mining for Health Applications (#SMM4H) Workshop & Shared Task*, (Florence, Italy), pp. 1–10, Association for Computational Linguistics, Aug. 2019.
- [54] K. He, L. Yao, J. Zhang, Y. Li, and C. Li, “Construction of genealogical knowledge graphs from obituaries: Multitask neural network extraction system,” *J Med Internet Res*, vol. 23, p. e25670, Aug 2021.
- [55] Y. Zhang, A. Chaganty, A. Paranjape, D. Chen, J. Bolton, P. Qi, and C. D. Manning, “Stanford at TAC KBP 2016 : Sealing Pipeline Leaks and Understanding Chinese,” *Proceedings of the Ninth Text Analysis Conference (TAC 2016)*, 2016.
- [56] I. Yamada, A. Asai, H. Shindo, H. Takeda, and Y. Matsumoto, “LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Online), pp. 6442–6454, Association for Computational Linguistics, November 2020.
- [57] A. Dresch, D. P. Lacerda, and J. A. V. Antunes, “Proposal for the conduct of design science research,” in *Design Science Research: A Method for Science*

- and Technology Advancement*, pp. 117–127, Cham: Springer International Publishing, 2015.
- [58] S. Cucerzan, “Large-Scale Named Entity Disambiguation Based on Wikipedia Data,” in *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pp. 708–716, 2007.
- [59] K. Riaz, “Rule-based Named Entity Recognition in Urdu,” in *Proceedings of the 2010 named entities workshop (NEWS)*, no. July, (Uppsala, Sweden), pp. 126–135, Association for Computational Linguistics, 2010.
- [60] U. Singh, V. Goyal, and G. S. Lehal, “Named Entity Recognition System for Urdu,” in *Proceedings of COLING 2012*, (Mumbai, India), pp. 2507–2518, The COLING 2012 Organizing Committee, December 2012.
- [61] S. Hakimov, S. A. Oto, and E. Dogdu, “Named Entity Recognition and Disambiguation using Linked Data and Graph-based Centrality Scoring,” in *Proceedings of the 4th international workshop on semantic web information management*, pp. 1–7, 2012.
- [62] F. B. Mesmia, K. Haddar, N. Friburger, and D. Maurel, “CasANER: Arabic Named Entity Recognition Tool,” in *Intelligent Natural Language Processing: Trends and Applications* (K. Shaalan, A. E. Hassanien, and F. Tolba, eds.), pp. 173–198, Cham: Springer International Publishing, 2018.
- [63] N. M. S. Wahyunia and N. A. S. ERa, “Rule-based named entity recognition (ner) to determine time expression for balinese text document,” *Jurnal Elektronik Ilmu Komputer Udayana*, vol. 9, no. 4, pp. 555–562, 2021.
- [64] N. P. A. S. A. Sugiartaa and N. A. S. ERa, “Location named-entity recognition using rule-based approach for balinese texts,” *Jurnal Elektronik Ilmu Komputer Udayana*, vol. 9, no. 3, pp. 435–442, 2021.
- [65] K. Kurniadia and N. A. S. ERa, “Person named entity recognition in balinese,” *Jurnal Elektronik Ilmu Komputer Udayana*, vol. 10, pp. 99–104, 2021.

- [66] A. Prasad and N. Sharma, “Rule-based recognition of associated entities in hindi text: A domain centric approach,” in *Information and Communication Technology for Competitive Strategies (ICTCS 2020)* (A. Joshi, M. Mahmud, R. G. Ragel, and N. V. Thakur, eds.), pp. 373–383, Singapore: Springer, 2022.
- [67] S. A. TARMIZI and S. SAAD, “Named entity recognition for quranic text using rule based approaches.,” *Asia-Pacific Journal of Information Technology & Multimedia*, vol. 11, no. 2, 2022.
- [68] G. Luo, X. Huang, C.-y. Lin, and Z. Nie, “Joint Named Entity Recognition and Disambiguation,” in *Empirical Methods in Natural Language Processing*, no. September, pp. 879–888, 2015.
- [69] Y. Sharma, R. Bhargava, and B. V. Tadikonda, “Named entity recognition for code mixed social media sentences,” *International Journal of Software Science and Computational Intelligence (IJSSCI)*, vol. 13, no. 2, pp. 23–36, 2021.
- [70] S. Sulaiman, R. A. Wahid, S. Sarkawi, and N. Omar, “Using Stanford NER and Illinois NER to Detect Malay Named Entity Recognition,” *International Journal of Computer Theory and Engineering*, vol. 9, no. 2, pp. 147–150, 2017.
- [71] S. Pudasaini, S. Shakya, S. Lamichhane, S. Adhikari, A. Tamang, and S. Adhikari, “Application of nlp for information extraction from unstructured documents,” in *Expert Clouds and Applications* (I. Jeena Jacob, F. M. Gonzalez-Longatt, S. Kolandapalayam Shanmugam, and I. Izonin, eds.), (Singapore), pp. 695–704, Springer Singapore, 2022.
- [72] D. C. Liu and J. Nocedal, “On the limited memory bfgs method for large scale optimization,” *Mathematical programming*, vol. 45, no. 1-3, pp. 503–528, 1989.
- [73] S. K. Siencnik, “Adapting word2vec to Named Entity Recognition,” in *20th nordic conference of computational linguistics, nodalida*, pp. 239–243, 2015.

- [74] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, “Neural architectures for named entity recognition,” in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 260–270, 2016.
- [75] D. Nadeau, “Balie—baseline information extraction: Multilingual information extraction from text with machine learning and natural language techniques,” tech. rep., Technical report, University of Ottawa, 2005.
- [76] M. Xiaofeng, W. Wei, and X. Aiping, “Incorporating token-level dictionary feature into neural model for named entity recognition,” *Neurocomputing*, vol. 375, pp. 43–50, 2020.
- [77] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.
- [78] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, “Translating embeddings for modeling multi-relational data,” *Advances in neural information processing systems*, vol. 26, pp. 2787–2795, 2013.
- [79] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, “Complex embeddings for simple link prediction,” in *International conference on machine learning*, pp. 2071–2080, PMLR, 2016.
- [80] B. Yang, S. W.-t. Yih, X. He, J. Gao, and L. Deng, “Embedding entities and relations for learning and inference in knowledge bases,” in *Proceedings of the International Conference on Learning Representations (ICLR) 2015*, 2015.
- [81] I. Yamada, H. Shindo, H. Takeda, and Y. Takefuji, “Joint learning of the embedding of words and entities for named entity disambiguation,” in *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 250–259, 2016.

- [82] I. Yamada, H. Shindo, H. Takeda and Y. Takefuji, “Learning distributed representations of texts and entities from knowledge base,” *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 397–411, 2017.
- [83] Y. Cao, L. Huang, H. Ji, X. Chen, and J. Li, “Bridge text and knowledge by learning multi-prototype entity mention embedding,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1623–1633, 2017.
- [84] O.-E. Ganea and T. Hofmann, “Deep joint entity disambiguation with local neural attention,” in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2619–2629, 2017.
- [85] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, “Ernie: Enhanced language representation with informative entities,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1441–1451, 2019.
- [86] L. B. Soares, N. Fitzgerald, J. Ling, and T. Kwiatkowski, “Matching the Blanks: Distributional Similarity for Relation Learning,” in *57th Annual Meeting of the Association for Computational Linguistics*, pp. 2895–2905, 2019.
- [87] M. E. Peters, M. Neumann, R. Logan, R. Schwartz, V. Joshi, S. Singh, and N. A. Smith, “Knowledge enhanced contextual word representations,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 43–54, 2019.
- [88] X. Wang, T. Gao, Z. Zhu, Z. Zhang, Z. Liu, J. Li, and J. Tang, “Kepler: A unified model for knowledge embedding and pre-trained language representation,” *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 176–194, 2021.
- [89] R. Wang, D. Tang, N. Duan, Z. Wei, X.-J. Huang, J. Ji, G. Cao, D. Jiang, and M. Zhou, “K-adapter: Infusing knowledge into pre-trained models with

- adapters,” in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 1405–1418, 2021.
- [90] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, (New Orleans, Louisiana), pp. 2227–2237, Association for Computational Linguistics, June 2018.
- [91] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of NAACL-HLT*, pp. 4171–4186, Association for Computational Linguistics, 2019.
- [92] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized bert pre-training approach,” *arXiv preprint arXiv:1907.11692*, 2019.
- [93] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, “Xlnet: Generalized autoregressive pretraining for language understanding,” in *Advances in neural information processing systems*, pp. 5753–5763, 2019.
- [94] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, “Albert: A lite bert for self-supervised learning of language representations,” in *International Conference on Learning Representations*, 2019.
- [95] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, “Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880, 2020.
- [96] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified

- text-to-text transformer,” *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.
- [97] I. Harrando and R. Troncy, “GraphNER : Named Entity Recognition as Graph Classification,” in *ESWC 2021*, (Heraklion, Greece), 2021.
- [98] S. Wang, X. Li, Y. Meng, T. Zhang, R. Ouyang, J. Li, and G. Wang, “kNN-NER: Named Entity Recognition with Nearest Neighbor Search,” *arXiv preprint arXiv:2203.17103*, 2022.
- [99] B. Chen, Z. Hao, J. Zhu, and G. Xie, “Embedding Logic Rules Into Recurrent Neural Networks,” *IEEE Access*, vol. 7, pp. 14938–14946, 2019.
- [100] A. Sil and A. Yates, “Re-ranking for Joint Named-Entity Recognition and Linking,” in *ACM international conference on Information & Knowledge Management*, pp. 2369–2374, 2013.
- [101] M. Chabchoub, M. Gagnon, and A. Zouaq, “Collective disambiguation and Semantic Annotation for Entity Linking and Typing,” in *Semantic Web Challenges*, pp. 33–47, Cham: Springer International Publishing, 2016.
- [102] J. Plu and G. Rizzo, “ADEL : ADaptable Entity Linking,” *SemanticWeb*, vol. 1, pp. 1–5, 2017.
- [103] M. Marrero and J. Urbano, “A Semi-Automatic and low-cost method to learn patterns for named entity recognition,” *Natural Language Engineering*, vol. 24, no. 1, pp. 39–75, 2017.
- [104] Y. Han, W. Chen, X. Xiong, Q. Li, Z. Qiu, and T. Wang, “Wide & Deep Learning for improving Named Entity Recognition via Text-Aware Named Entity Normalization,” in *The AAAI-19 Workshop on Recommender Systems and Natural Language Processing*, vol. 1, 2019.
- [105] D. Santos, N. Mamede, and J. Baptista, “Extraction of Family Relations between Entities,” *INForum*, pp. 549–560, 2010.

- [106] A. Romadhony, A. Purwarianti, and D. H. Widyantoro, “Rule-based indonesian open information extraction,” in *2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA)*, pp. 107–112, 2018.
- [107] I. A. Norabid and F. Fauzi, “Rule-based text extraction for multimodal knowledge graph,” *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 5, 2022.
- [108] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, “Aligning books and movies: Towards story-like visual explanations by watching movies and reading books,” in *Proceedings of the IEEE international conference on computer vision*, pp. 19–27, 2015.
- [109] S. Zhang, P. Ng, Z. Wang, and B. Xiang, “REKnow: enhanced knowledge for joint entity and relation extraction,” *arXiv preprint arXiv:2206.05123*, 2022.
- [110] E. Strubell, A. Ganesh, and A. McCallum, “Energy and policy considerations for deep learning in nlp,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3645–3650, 2019.
- [111] O. Loyola-Gonzalez, “Black-box vs. white-box: Understanding their advantages and weaknesses from a practical point of view,” *IEEE access*, vol. 7, pp. 154096–154113, 2019.
- [112] G. Angeli, V. Zhong, D. Chen, A. T. Chaganty, J. Bolton, M. J. J. Premkumar, P. Pasupat, S. Gupta, and C. D. Manning, “Bootstrapped self training for knowledge base population,” in *TAC*, 2015.
- [113] L. Ratinov, D. Roth, D. Downey, and M. Anderson, “Local and global algorithms for disambiguation to Wikipedia,” *ACL-HLT 2011 - Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, pp. 1375–1384, 2011.

- [114] B. M. Sundheim, "OVERVIEW OF RESULTS OF THE MUC-6 EVALUATION," in *Proceedings of the 6th conference on Message understanding*, pp. 13–31, 1995.
- [115] P. McNamee and H. Dang, "Overview of the TAC 2009 knowledge base population track," in *Text analysis conference (TAC)*, vol. 17, pp. 111–113, 2009.
- [116] R. Mumtaz, M. A. Qadir, and A. Saeed, "CustFRE: An annotated dataset for extraction of family relations from English text," *Data in Brief*, vol. 41, p. 107980, 2022.
- [117] K. Jha, M. Roder, and A.-C. C. N. Ngomo, "All that glitters is not gold – Rule-based curation of reference datasets for named entity recognition and entity linking," in *Lecture Notes in Computer Science*, vol. 10249 LNCS, pp. 305–320, 2017.
- [118] R. Kumar, *Research methodology: a step-by-step guide for beginners*. SAGE Publications, Inc, 2011.
- [119] W. G. Cochran, *Sampling techniques*. John Wiley & Sons, 1977.
- [120] T. S. Nanjundeswaraswamy and S. Divakar, "Determination of Sample Size and Sampling Methods in Applied Research," *Proceedings on Engineering Sciences*, vol. 3, no. 1, pp. 25–32, 2021.
- [121] J. E. Bartlett, J. W. Kotrlik, and C. C. Higgins, "Organizational research: Determining appropriate sample size in survey research," *Information technology, learning, and performance journal*, vol. 19, no. 1, pp. 43–50, 2001.
- [122] E. M. Bender and B. Friedman, "Data Statements for Natural Language Processing: Toward Mitigating System Bias and Enabling Better Science," *Transactions of the Association for Computational Linguistics*, vol. 6, pp. 587–604, 2018.
- [123] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and psychological measurement*, vol. 20, no. 1, pp. 37–46, 1960.

- [124] M. L. McHugh, “Interrater reliability: the kappa statistic,” *Biochemia medica*, vol. 22, no. 3, pp. 276–282, 2012.
- [125] P. N. Mendes, M. Jakob, A. García-silva, and C. Bizer, “DBpedia Spotlight : Shedding Light on the Web of Documents,” in *Seventh international conference on semantic systems*, pp. 1–8, 2011.
- [126] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, “Web Semantics : Science , Services and Agents on the World Wide Web DBpedia - A crystallization point for the Web of Data,” *Web Semantics: Science, Services and Agents on theWorldWideWeb*, vol. 7, pp. 154–165, 2009.
- [127] R. Jiang, R. E. Banchs, and H. Li, “Evaluating and Combining Named Entity Recognition Systems,” in *Sixth Named Entity Workshop*, pp. 21–27, 2016.
- [128] H. Nakayama, “seqeval: A python framework for sequence labeling evaluation,” 2018. Software available from <https://github.com/chakkiworks/seqeval>.
- [129] V. Zhong, Y. Zhang, D. Chen, G. Angeli, and C. Manning, “TAC Relation Extraction Dataset, Web Download. Philadelphia: Linguistic Data Consortium,” 2018.
- [130] L. Chiticariu, Y. Li, and F. R. Reiss, “Rule-based Information Extraction is Dead ! Long Live Rule-based Information Extraction Systems !,” in *Conference on Empirical Methods in Natural Language Processing*, no. October, pp. 827–832, 2013.

Appendix A

The Relation Words Lists

Parent Words

1. parent
2. adopter
3. born
4. bornin
5. bornt
6. Father
7. dad
8. daddy
9. papa
10. mama
11. mother
12. mum
13. mom
14. stepfather
15. step-father
16. stepmother
17. step-mother
18. step-parent
19. stepparent

Children Words

20. child
21. children
22. adopt

23. adopting
24. adopted
25. adoption
26. adoptive
27. kid
28. offspring
29. son
30. daughter
31. stepson
32. step-son
33. stepdaughter
34. step-daughter

Spouse Words

35. couple
36. ex
37. ex-couple
38. spouse
39. ex-spouse
40. husband
41. ex-husband
42. hubby
43. ex-hubby
44. widow
45. ex-widow
46. better-half
47. significant-other
48. wife

49. ex-wife
50. wives
51. ex-wives
52. wed
53. wedding
54. wedded
55. marry
56. marries
57. marrying
58. married
59. marriage
60. remarry
61. remarries
62. remarrying
63. remarried
64. life-partner
65. marriage-partner
66. newlywed
67. girlfriend
68. ex-girlfriend
69. boyfriend
70. ex-boyfriend
71. concubine
72. ex-concubine
73. lover
74. ex-lover
75. fiancé
76. fiancée
77. betroth
78. betrothed
79. affianced

Appendix B

Search Strategy for Literature Review

For literature review, the search strategy was primarily directed towards finding relevant published research papers from Google Scholar¹. The relevant task name was used to create search strings; these include “named entity recognition,” “NER”, “rule-based named entity recognition,” “named entity classification”, “NERC”, “information extraction”, “relation extraction,” “relation classification”, “family relation extraction,” and “family relation classification”. Time range in Google Scholar was set to year 2005 onwards. Around 1.5 million (for NER) and 89 thousand (for RE) results were returned. Articles were sorted by relevancy and citation count. We explored till page 8 for each search string result by reading titles and keywords. 334 items were found to be potentially relevant, 186 for NER and 148 for FRE. The selected articles were filtered manually according to the inclusion exclusion criteria, by reviewing abstracts, and skimming the contents of the papers. 39 articles were finally selected, and are categorized and discussed in Chapter 2.

¹<https://scholar.google.com/>