KNOWLEDGE EXTRACTION FROM EMAIL CONVERSATIONS AND ITS APPLICATION TO QUESTION ANSWERING

by

Parag Pravin Dakle

APPROVED BY SUPERVISORY COMMITTEE:

Dan I. Moldovan, Chair

Jessica Ouyang

Nicholas Ruozzi

R. Chandrasekaran

Copyright © 2021

Parag Pravin Dakle

All rights reserved

KNOWLEDGE EXTRACTION FROM EMAIL CONVERSATIONS AND ITS APPLICATION TO QUESTION ANSWERING

by

PARAG PRAVIN DAKLE, BE, MS

DISSERTATION

Presented to the Faculty of

The University of Texas at Dallas

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY IN

COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT DALLAS

December 2021

To my parents, Pravin and Sunita, my brother, Pushkar and my love, Shivika.

ACKNOWLEDGMENTS

A sincere thank you to my advisor, Dr. Dan Moldovan, for his keen insight and guidance. I owe him the successful and timely completion of this dissertation. I thank committee members Dr. R. Chandrasekaran, Dr. Nicholas Ruozzi, and Dr. Jessica Ouyang for their invaluable suggestions. I appreciate the crucial and relevant feedback provided by reviewers and area chairs of conferences where I submitted papers.

My deepest thanks to alumni and current members of my research group: Dr. Marta Tatu, Dr. Mithun Balakrishna, Dr. Tatiana Erekhinskaya, Dr. Takshak Desai for assisting me with my research. I also thank UT Dallas professors who taught courses that helped me comprehend fundamental concepts required to conduct research.

I am beyond grateful to my friends for being a constant source of motivation. I extend my thanks to my family - grandparents, Vidya, Chocolate, and extended family - Steve and Becky. Words cannot describe how indispensable and inspiring your encouragement has been. I dedicate this dissertation to you and your support.

November 2021

KNOWLEDGE EXTRACTION FROM EMAIL CONVERSATIONS AND ITS APPLICATION TO QUESTION ANSWERING

Parag Pravin Dakle, PhD
The University of Texas at Dallas, 2021

Supervising Professor: Dan I. Moldovan, Chair

Email communication is the exchange of messages between two or more people over the internet using electronic devices. With billions of emails exchanged every day, extracting knowledge from emails is beneficial to various email-based user applications. In any form of communication, it is important to identify the participating users or entities and their interactions throughout the conversation. Previous research on email processing has primarily focused on classification, searching, and intent detection. However, it has overlooked studying the interaction between entities participating in an email conversation.

One of the tasks to capture the interaction is entity coreference resolution. End-to-end entity coreference resolution extracts entities and their references throughout the conversation. Post extraction, it is important to use a knowledge representation format that preserves and enriches the extracted knowledge. Knowledge graphs can assist in representing these extracted intra-email interactions compactly. They can also enrich the knowledge by capturing inter-email interactions using a robust technique like matching entities across email conversations. One of the main applications to use the extracted knowledge is question answering that focuses on the entities in a conversation. These tasks, when put together, paint a simplistic yet holistic picture of a knowledge extraction pipeline for email conversations.

A deep joint learning framework is proposed for the novel task of entity coreference resolution for email conversations. Two datasets were created during the framework's development process. These datasets were used to evaluate the task difficulty and identify the limitations of the available solutions. The framework used the task of text classification for joint learning to improve the scoring of text spans. This task was also used for incorporating singletons in the result. The joint learning framework and singleton addition achieved an improvement of 4.87 and 5.26 F1 points on the two datasets, respectively.

A combination of automatic and manual methods was used to carry out relation extraction parallel to entity coreference resolution. The extracted knowledge from the tasks was used to create two knowledge graphs - one that contained the knowledge from the relation extraction task and the other that contained knowledge extracted from both tasks. The knowledge graph creation process used the NEPOMUK framework, which was created to simplify data sharing across different user applications. Changes to the NEPOMUK framework have been proposed for adding coreference knowledge to the graphs.

Lastly, previous work has investigated doing question answering using digital voice assistants. However, this dissertation explores the novel task setting of doing question answering using digital voice assistants for email conversations. The sub-task of entity resolution has been identified as essential to the proposed formulation, and a dataset evaluating the same was created using templates. A deep learning-based and two SPARQL template-based systems were used for the evaluation process. Empirical results showed an increase of 3.91% in the template-based system's accuracy when coreference information was incorporated in knowledge graphs. By laying a framework for the knowledge extraction pipeline, creating open-source datasets and benchmarks for comparison, one can hope to advance research in email processing.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Motivation

Social media applications such as text messaging, email messaging, Twitter, and Facebook play an indispensable role in communication between people. Amongst all these, email messaging is one of the oldest and widely used mediums. Although there has been a reduction in the usage of emails for personal communication, emails still play a vital role at workplaces. Emails are one of the primary mediums of communication for official communication, sharing documents, or updates. The offline nature of emails allows the recipient to respond to the email at their convenience.

Every year the number of email users and the number of emails exchanged in a year increases consistently by 3-4%. According to a survey[1] conducted in 2020, a total of 4,481M email users will be present, and about 361B emails will be exchanged every day by 2024. In addition to the large volume of email messages being exchanged, the email messages' topics are also very diverse. Both these aspects gave rise to research on numerous email processing applications. The most popular are email classification, named entity recognition, intent classification, summarization, and searching. The common feature in most of these application research has been the usage of private email data. User-data privacy has always been the primary argument corroborating the usage of private data. Though the argument is valid, it also leads to the absence of benchmarks and annotated datasets hamper future research due to the lack of comparability and inability to carry forward existing research. Thus any advancement in email processing tasks that result in creating benchmarks while preserving privacy would benefit future email research.

[1] https://www.radicati.com/wp/wp-content/uploads/2020/01/Email_Statistics_Report,_2020-2024_Executive_Summary.pdf

Text processing in emails, however, has still been very task-specific. Advances in document processing have led to the creation of multiple knowledge extraction pipelines for plain text. A popular example of this is the tool *Spacy* (Honnibal and Montani, 2017). *Spacy* provides an knowledge extraction pipeline that carries out sentence boundary detection, tokenization, part-of-speech tagging, named entity recognition, sentiment analysis, and many more configurable tools. However, such pipelines do not cater to domain-specific texts, specifically emails. The main reason for the inability to cater to emails is the absence of pipeline components that have been trained on an email dataset. A complete knowledge extraction pipeline would not only assist the system in understanding the emails better but also use the extracted knowledge to develop applications that benefit end-users. With the rapid increase in email overload, any application to assist an end-user will have an enormous impact.

Lastly, the increase in social media usage specifically micro-blogging platforms has led to the increase in research focused on conversational text[2] (Henderson et al., 2019; Mehta and Mehta, 2020; Wambsganss et al., 2020; Qamar et al., 2021). An entire conversation as compared to a single turn poses different research challenges. Understanding a conversation involves tracking multiple things like topics, references, or sentiments. These challenges have been widely investigated for Twitter conversations (Preotiuc-Pietro et al., 2012; Aktaş et al., 2018; Li et al., 2018; Van Hee et al., 2018; Aktaş et al., 2020; Bowen et al., 2020) or Reddit posts (Ghosh et al., 2018; Pant et al., 2020; Patil et al., 2020) or conversations from similar platforms (Joty and Mohiuddin, 2018). Email conversations, however, have not received the same attention. Research in email processing has primarily focused on email messages rather than email threads. Compared to email messages, the content in email threads is more conversational in nature due to the absence of Business-to-Consumer (B2C)

[2]`https://competitions.codalab.org/competitions/30312`

emails. Furthermore, extracting knowledge from these threads or conversations can allow development of richer downstream applications.

Thus, the challenge of - **building a knowledge extraction pipeline for email conversations** - constitutes the core motivation of this dissertation. Of course, this challenge is exceedingly tricky. For example, the lack of annotated datasets with real-world email conversations makes it challenging to approach several NLP problems for this domain. An entirely general-purpose pipeline that extracts all types of knowledge and handles all forms of emails is a long way off. Instead, following standard practice, our approach isolates a relevant yet realistic case of this problem.

1.2 Overview

1.2.1 Our Focus: Knowledge extraction from email conversations

As stated in the previous section, the core motivation of this dissertation is building a knowledge extraction pipeline for email conversations. Ideally, a knowledge extraction pipeline should accept email conversations in all formats. Given a set of email conversations[3] T_U for user U, the pipeline should extract all forms of knowledge from T_U. The components performing different operations like email classification, entity extraction, coreference resolution, relation extraction, intent extraction, or sentiment analysis should be available. Finally, the pipeline should then represent the extracted knowledge in any required format so that downstream applications can consume it. This dissertation groups all these requirements into four execution phases that an ideal pipeline must follow - *Preprocessing*, *Extraction*, *Representation*, and *Application*. Although the task of a knowledge extraction pipeline ends at the *Representation* phase, the *Application* phase has been added for completeness. Figure

[3]The dissertation uses the terms thread and conversation interchangeably throughout the dissertation. For the scope of this dissertation, both terms have the same semantic meaning.

1.1 shows a pipeline structure with four phases and some sample tasks in each phase. For the *Representation* phase, different representation strategies have been listed in Figure 1.1.

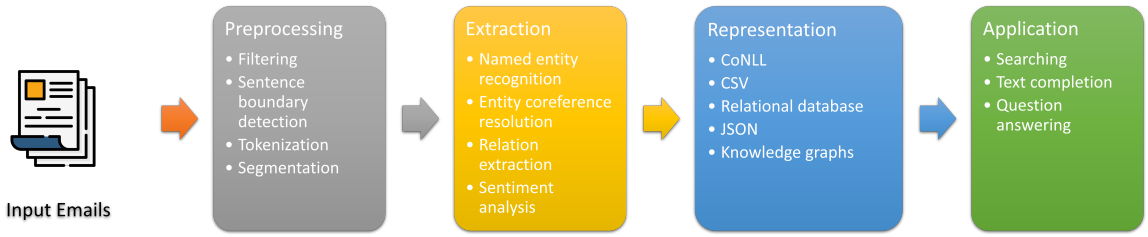

Figure 1.1. The general pipeline structure showing different phases and few example tasks carried out in each phase.

The pipeline described before is highly complex, and thus, building such a generic pipeline for email conversations is beyond the scope of this dissertation. Therefore, this dissertation focuses on a simplified but valuable case. Specifically, it focuses on the pipeline's *Extraction*, *Representation*, and *Application* phases and one crucial task for each phase. We believe this will help build a solid foundation for the ideal knowledge extraction pipeline for email conversations. Although the tasks carried out in the *Preprocessing* phase are crucial, no efforts are dedicated to automating these tasks. A mixture of manual and automated methods is used to preprocess the email conversations considered in this dissertation. Lack of publicly available preprocessed corpora containing email conversations and non-uniformity within the unprocessed corpora led to this approach. We expect the input to be preprocessed in a manner similar to that carried out in this thesis for new email conversations.

Before elaborating on the tasks researched in this dissertation, let us define the basic terminologies used throughout. An *Entity* is defined as an object or a set of objects in the

world. A *Mention* is defined as a span of text that refers to or mentions an entity. Figure 1.2 shows an example with mentions. The example highlights three mentions 'Heather', 'you', and 'the file'. These three mentions are part of two coreference chains, one containing 'Heather' and 'you', and the other containing 'the file'.

Figure 1.2. Example showing mentions and coreference chains.

In the *Extraction* phase, previous works have carried out Named Entity Recognition (NER). NER is a difficult task and has its own set of challenges. However, in this dissertation, we want to study the interaction between different entities in an email conversation. To the best of our knowledge, this entity interaction aspect in email conversations has never been explored in a generic extraction setting. In the past, social network analysis has primarily been used to study the interaction of entities. The dissertation uses the *Entity Coreference Resolution* task in an end-to-end setting to cater to the motivation of studying the interaction between entities. In simple terms, the task of end-to-end entity coreference resolution comprises two operations - detecting mentions in an email conversation and chaining mentions that refer to the same entity together. This task allows us to extract the entities via mentions and study their interaction via chained references (mentions) throughout the conversation. For this task, two datasets are created to empirically prove that the task is difficult and two models are proposed that show state-of-the-art performance on the created datasets.

Entity coreference resolution is the primary task in the *Extraction* phase. In order to extract additional knowledge, this work also considers the *Relation extraction* task. However,

this task is undertaken in a secondary capacity. Two off-the-shelf systems are explored to carry out semantic relation extraction. The approach is discarded as, for many email conversations, the obtained relation predictions are either empty or incorrect. The relation set is then simplified, and relations are extracted using a mixture of automatic and manual methods. Finally, the output of both the tasks is merged to obtain a single output for the *Extraction* phase.

The *Extraction* phase carries out knowledge extraction from each email thread individually. This helps us in understanding the entity interactions within an email thread but raises intriguing questions.

1. *What if one entity is present across multiple email threads?*

2. *If email threads share entities, can one represent the extracted knowledge compactly?*

3. *How can one add world knowledge about entities present in an email thread?*

Although the output of the *Extraction* phase can be used directly by downstream applications, the above questions emphasize the need to use a representation that compactly captures entity interactions across email threads. Additionally, the representation should also facilitate incorporating world knowledge for entities present in the email threads. In order to find a good knowledge representation strategy, the structure of email threads is examined. Email threads have a tree-like structure where each node represents an email message and link between from node a to node b represents that a is a reply to b. An email message in itself, due to its semi-structured nature, can also be represented using a graph-like structure. Every header in an email message can represent a link between a node representing the email message and one representing the header value. These structural patterns in email threads motivate us to use knowledge graphs to represent the extracted knowledge. Thus, the *Representation* phase of the pipeline is centered on creating a knowledge graph.

Any knowledge graph creation process requires the identification of nodes (entities) and the links (relations) between them. The output of the *Extraction* phase perfectly fulfills these requirements. Since the motivation to use knowledge graphs is to capture inter-email thread entity interactions, this dissertation studies the impact of adding these interactions to the knowledge graph. In the *Representation* phase, two knowledge graphs *KG-Normal* and *KG-Coref* are created. In *KG-Normal* only the extracted relations and the corresponding relation entities are presented. For *KG-Coref*, the knowledge extracted via the entity coreference resolution task is incorporated into *KG-Normal*. Creating these two knowledge graphs helps compare and analyze the two graphs in terms of correctness and compactness.

The final aspect of knowledge graphs that is important to this dissertation is ontology. An ontology is a schema or skeleton that a knowledge graph mimics. Consider two knowledge graphs KG_1 and KG_2 that use ontologies O_1 and O_2 respectively. Let $e_1 \in KG_1$ and $e_2 \in KG_2$ be two entities. Given e_1 and e_2 refer to the same real-world entity, they can be linked with each other if - O_1 and O_2 are the same ontologies or there exists a one-to-one mapping between the O_1 and O_2. Thus, the third question can be answered using an ontology that is either the same as or can be mapped to the ontologies used by knowledge graphs containing world knowledge.

Knowledge graphs have proven critical components of many modern-day applications like question answering, data visualization, and searching. In the *Application* phase of this pipeline, the dissertation explores using the created knowledge graphs in a downstream application. Two factors motivate us when selecting a downstream application for this phase. The first factor is the ability of the application to compare $KG-Normal$ and $KG-Coref$. This factor helps in measuring the impact of the entity coreference system. The second factor is the novelty of the task. Question answering for email conversations has received mediocre attention in the research literature. The task has scantily been explored in a setting where email conversations are the knowledge source for answering questions. Our approach

addresses not only this dearth of attention but also formulates the problem in a novel setting using digital voice assistants. A dataset is created that adheres to this formulation and tests a question answering system's ability to carry out coreference resolution. Using a SPARQL-based template approach, this work shows that adding coreference predictions to the knowledge graph improves the question answering system's performance.

Thus, this dissertation creates a pipeline that extracts entity coreference chains and relations from email conversations using three primary and one secondary task. It represents the extracted knowledge as knowledge graphs and then use it in a question-answering task. With this holistic or end-to-end view, we formally define our problem statement.

1.2.2 Problem Statement

The problem statement of this dissertation can be stated as - **given an email thread T, extract knowledge in the form of entity coreference chains C and relations R, represent the extracted knowledge as a knowledge graph KG and use the extracted knowledge for question answering**.

Example 1. A sample email thread.

```
Brian Heinrich 09/27/2000 01:11 PM
To: Lisa B Cousino cc:
Subject: Re: Question - Delaney Report
Actually, counting the monetized counterparty that we lumped in it would be 4
    counterparties.


Audrey Cook 09/26/2000 02:42 PM
To: Brian Heinrich cc: Bryce Baxter
Subject: Question - Delaney Report
Brian,
```

```
The answer to your question regarding the number of counterparties in "
   reconciliation status" on the Delaney Report is 3.
ajc
```

The problem statement is elaborated further using the email thread shown in Example 1. Given a user U containing u_c email threads $T_U = \{T_1, T_2, .., T_{u_c}\}$, perform the following operations:

1. Extract all entity mentions from $T \in T_U$. Chain all mentions which refer to the same entity together. For the email thread shown in Example 1 the extracted entity mentions and the corresponding chains are shown in Figure 1.3. In the figure, similar colored entities belong to the same entity chain.

2. Extract semantic relations from $T \in T_U$. This operation considers relations between different parts of the email thread T: within an email message $M \in T$ and between email messages $M_1, M_2 \in T$. In Figure 1.4, the relations extracted from the email thread in Example 1 are shown in a JSON[4] format.

3. Represent the email threads T_U as a knowledge graph $KG_U - Normal$ and $KG_U - Coref$ using the extracted entity mentions and relations. Figure 1.5 shows the graph containing the knowledge extracted from the email thread in Example 1. A section of the knowledge graph displayed has been highlighted to show the relations between the nodes. The green nodes in the graph represent mentions found in the email body. The nodes *em1* and *em2* represent the two email messages in the thread.

4. Use the created knowledge graph $KG_U - Normal$ and $KG_U - Coref$ to perform question answering on the u_c email threads. Each question is converted into a SPARQL query and executed to fetch the answer from the two knowledge graphs.

[4]Javascript Object Notation

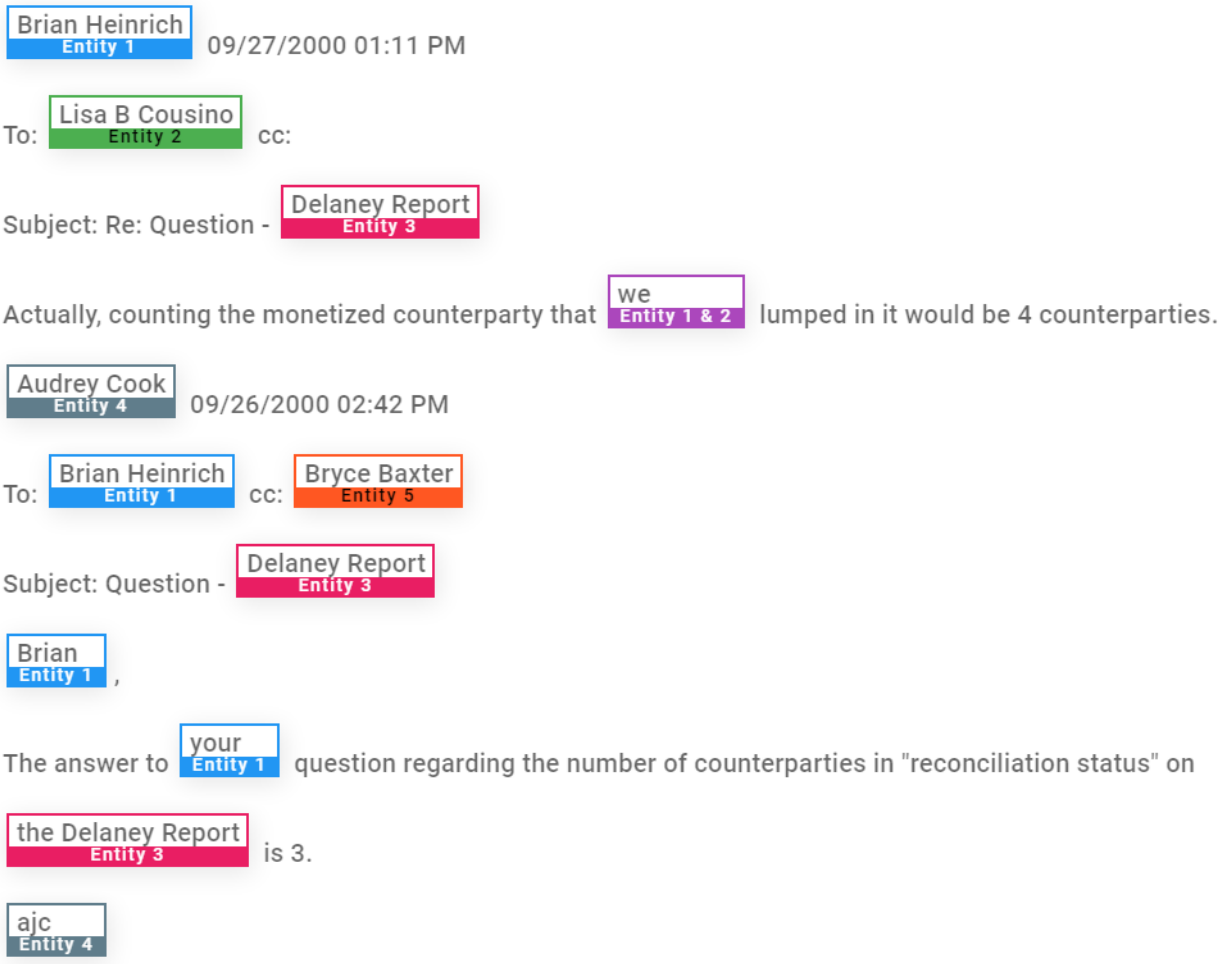

Figure 1.3. Extracting entity mentions and entity chains.

Figure 1.6 shows different phases of the pipeline and which tasks are carried out in each phase and the order of the tasks.

1.3 Contributions

Let us summarize this chapter by explicitly stating what we believe to be our significant contributions towards meeting this challenge.

1. This dissertation manually filters the Enron Email Corpus (Klimt and Yang, 2004) for email conversations using a set of constraints. The filtered corpus is then preprocessed

```
{
    "1": {
        "header-text": [
                "Brian Heinrich 09 / 27 / 2000 01 : 11 PM",
                "To : Lisa B Cousino / HOU / ECT @ ECT",
                "cc :",
                "Subject : Re : Question - Delaney Report"
        ],
        "to": ["Lisa B Cousino"],
        "from": ["Brian Heinrich"],
        "datetime": "09 / 27 / 2000 01 : 11 PM",
        "subject": "Question - Delaney Report",
        "body-text": ["Actually, counting the monetized counterparty that we lumped in it
        would be 4 counterparties."],
        "reply-to": "2"
    },
    "2": {
        "header-text": [
                "Audrey Cook 09 / 26 / 2000 02 : 42 PM",
                "To : Brian Heinrich / HOU / ECT @ ect",
                "cc : Bryce Baxter / HOU / ECT @ ECT",
                "Subject : Question - Delaney Report"
        ],
        "to": ["Brian Heinrich"],
        "from": ["Audrey Cook"],
        "cc": ["Bryce Baxter"],
        "datetime": "09 / 26 / 2000 02 : 42 PM",
        "subject": "Question - Delaney Report",
        "body-text": [
                "Brian , The answer to your question regarding the number of counterparties
                in \" reconciliation status \" on the Delaney Report is 3 .",
                "ajc"
        ]
    }
}
```


Figure 1.4. Relations extracted from the email thread in Example 1.

and made publicly available in CoNLL format. This is the first such publicly available preprocessed email conversation corpus.

2. This dissertation explores the problem of Entity coreference resolution in email conversations in a generic setting for the first time. Two datasets are created - one manually

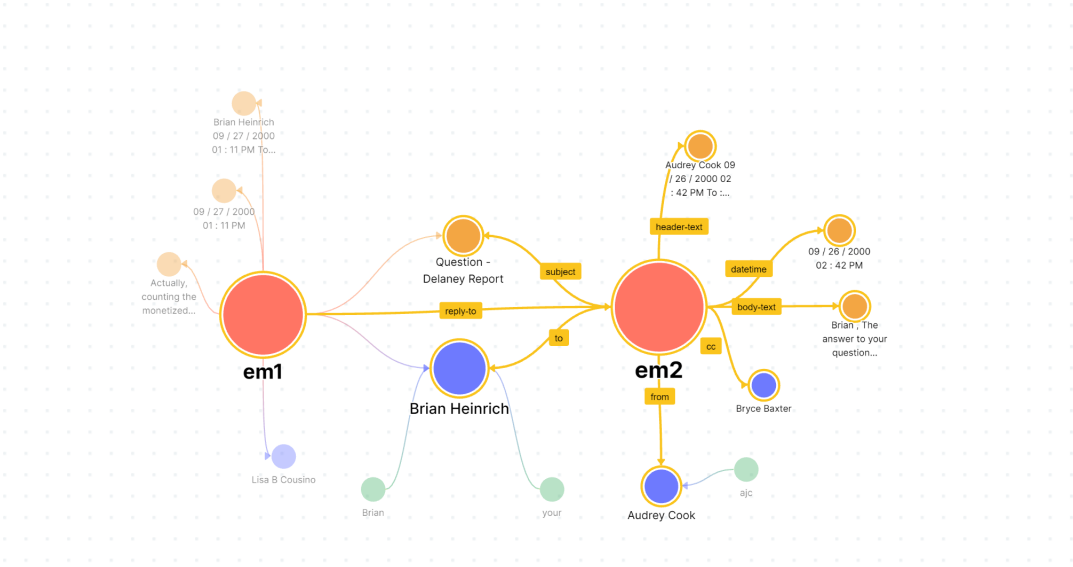

Figure 1.5. Knowledge graph showing the extraction knowledge from the email thread in Example 1.

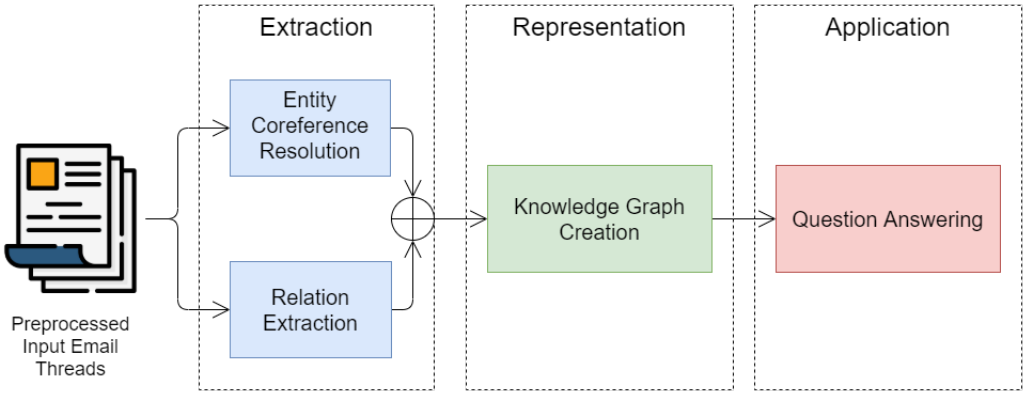

Figure 1.6. Pipeline showing the different phases and components used in this dissertation.

annotated dataset (SEED)[5] and one large-scale weakly annotated dataset (CEREC)[6] containing coreference annotations is created. A joint learning model is proposed that achieves state-of-the-art performance on the two datasets (Chapters 3 and 4).

[5] https://github.com/paragdakle/emailcoref/tree/master/data/LREC

[6] https://github.com/paragdakle/emailcoref

3. Relation extraction from email conversations is carried out both automatically and manually (Chapter 5). The resulting dataset (`ECRA`) containing the annotated relations will be made publicly available.

4. This dissertation represents the extracted knowledge using knowledge graphs and incorporate coreference information in knowledge graphs (Chapter 6). A new ontology is proposed to incorporate coreference information.

5. We formulate the Question answering task for email conversations in a practical setting using digital voice assistants. A benchmark dataset (`EMailQA`) is created containing questions that test a system's ability to perform coreference resolution. Finally, the dataset is evaluated to validate the addition of coreference information to knowledge graphs (Chapter 7).

1.4 Organization

The remainder of this dissertation is organized as follows.

Chapter 2: Related Theory

The dissertation begins by providing an in-depth overview of the knowledge extraction task for emails in Chapter 2. A brief background of the knowledge extraction task in general along with email processing is followed by an overview of deep learning. The chapter finishes by describing the SpanBERT and SBERT models used in this dissertation.

Chapter 3: Corpus

This chapter discusses the creation of the base corpus that will be used by all components of this dissertation. Starting with an overview of the available email corpora, this chapter describes the extraction, filtering, and preprocessing steps carried out for

the construction of the base corpus. Challenges faced during the creation of the corpus along with detailed statistics are also presented in this chapter. The Enron Email Corpus is used for creation of the base corpus.

Chapter 4: Entity coreference resolution

In this chapter, the dissertation delves into the main components of the pipeline, beginning with the Entity Coreference Resolution task. The chapter presents the annotation process used, statistics of the corpus created, and describes the challenges associated with the task. Furthermore, the proposed joint learning model for the task, experiments, results, and error analysis are presented.

Chapter 5: Relation extraction

The chapter first describes the relations what will be extracted and then elaborates the extraction process. The chapter also describes the alternative methods that were evaluated for relation extraction.

Chapter 6: Knowledge representation

This chapter focuses on representing the knowledge extracted in Chapters 4 and 5. It discusses different representation formats used previously and then provides a succinct introduction to knowledge graphs. The chapter then describes the process to create two knowledge graphs - *KG-Normal* and *KG-Coref*.

Chapter 8: Future work and conclusions

This chapter suggest areas for future research and summarize the dissertation contributions and conclusions.

Appendix A: Additional experiment results

Additional experimentation results for experiments performed using the joint learning model have been provided.

Appendix B: Sample EMailQA questions

Example questions for each complexity level in the EMailQA dataset have been provided.

Appendix C: SPARQL templates

SPARQL template used by *SimpleQuery* and *ComplexQuery* for each template used to generate questions in *EMailQA* has been provided.

Appendix D: Relation extraction example

The chapter provides an example of the complete output generated by the relation extraction process on an email thread excerpt.

Appendix E: TACRED relations

The list of relations in the TACRED dataset along with examples are included.

Appendix F: DocRED relations

The list of relevant relations in the DocRED dataset along with description are included.

CHAPTER 2

RELATED THEORY

2.1 Knowledge extraction

Knowledge Extraction (KE) or Information Extraction (IE) is one of the fundamental and earliest tasks of Natural Language Processing. The task can be defined as extracting structured information from unstructured or semi-structured text. One of the earliest systems developed to tackle the problem was FRUMP (DeJong, 1979). FRUMP (Fast Reading Understanding and Memory Program), designed to work on news articles, processes articles from a UPI news wire belonging to different domains, and creates a summary for the users. Another system on similar lines was JASPER (Andersen et al., 1992). JASPER (Journalist's Assistant for Preparing Earnings Reports) was developed by Carnegie Group, Inc. for Reuters Ltd. The system uses a template-driven approach and partial understanding techniques to extract real-time financial information from PR Newswire.

The task of KE was introduced and explored on a larger scale with the introduction of the Message Understanding Conference series. MUC-1 (1987) explored KE from documents without any evaluation criterion. MUC-2 (1989) defined the task as template filling with predefined event types. MUC-2 introduced the evaluation metrics of precision and recall for the task. The data in both MUC-1 and MUC-2 comprised of sanitized military messages. MUC-3 to MUC-5 added complexity to the task by increasing template slots and incorporating nesting in the template structure. MUC-6 (Sundheim, 1995) introduced three sub-tasks to ensure that the proposed systems had a deeper understanding of language: coreference, word sense disambiguation, and predicate-argument structure. The MUC series ended with MUC-7 (Chinchor, 1998), in which the task of relation extraction from the given text was added. The relations were restricted to those found within an organization - *employee_of*, *product_of* and *location_of*. The MUC series was followed by Semantic Evaluation or SemEval

tasks (1998 - ongoing), CoNLL shared tasks (1999 - ongoing), and Automatic Content Evaluation or ACE tasks (2000 - 2008). Besides fostering research in the KE domain, these tasks have also helped establish benchmarks and metrics for evaluating and comparing different KE systems.

One of the earliest works to provide an overview of the task as well as discuss different systems carrying out KE was done by Cowie and Lehnert (1996). Cowie and Lehnert (1996) highlight the impact of KE systems and how their introduction assists NLP researchers to work on large-scale systems with actual human language data compared to the artificial data used by many systems previously. Simoes et al. (2004) provide a possible decomposition of KE in 5 tasks: *Segmentation*, *Classification*, *Association*, *Normalization* and *Coreference Resolution*. Given a text document, *Segmentation* is dividing the document into segments (tokens, sentences, or chunks). The *Classification* task then aims to classify each of these segments into different classes (e.g., NER), which is followed by *Association* that focuses on identifying relationships between the segments. *Normalization* and *Coreference Resolution* are the final tasks in which the first converts some segments into a standard format (e.g., date or time), and the second chains all segments which refer to the same real-world entity together.

In the early days, KE in emails primarily focused on the classification of emails. Cohen et al. (1996) in their work on the classification of personal emails, compare the performance of two learning methods. The first method is a classical Information Retrieval based method using TF-IDF weighting and the second method is a variation of the RIPPER rule learning algorithm (Cohen, 1995). Another work carried out to compare different classifiers for email classification was done by Brutlag and Meek (2000). The authors elaborate on the challenges of the email domain and use an email collection of sparse email categories for evaluation.

Manco et al. (2002) also explored the email classification problem but proposed an adaptive email classifier system. The authors propose a User Agent (like Thunderbird or Outlook),

which is autonomous. It organizes the messages in homogeneous groups/hierarchies and is adaptive as it incrementally refines the message organization. The work of Klimt and Yang (2004) has been instrumental for email classification as well as other email processing tasks as they introduce the Enron Email Corpus. Klimt and Yang use SVM for email classification and validate their results on the Enron Corpus using another corpus containing email messages exchanged between several students and a faculty at the Language Technology Institute at CMU. Alkhereyf and Rambow (2017) annotate the Enron Corpus and another corpus into business and personal categories. The authors report good performance on cross-corpus classification using conventional classifiers like SVM and Extra-Trees with pre-trained word embeddings. Apart from email classification, emails have been part of other tasks like named entity recognition (Minkov et al., 2005; Jung et al., 2015), intent classification (Cohen et al., 2004), searching (Soboroff et al., 2006; Minkov et al., 2008), clustering (Huang and Mitchell, 2008), and summarization (Muresan et al., 2001; Lam, 2002; Newman and Blitzer, 2003; Nenkova and Bagga, 2004; Corston-Oliver et al., 2004; Rambow et al., 2004; Carenini et al., 2007; Ulrich et al., 2008). Many of these works can be categorized into the first four tasks of the KE decomposition described before. To the best of our knowledge, no previous work has explored the last task, Coreference Resolution, with respect to email conversations.

2.2 Deep Learning

2.2.1 Overview

Deep learning is a branch of machine learning that focuses on algorithms inspired by the working and structure of the brain. Although neural networks in general attempt to do that, the *depth* in deep learning models take a step further in that direction. In contrast to the traditional machine learning algorithms that are linear, deep learning algorithms stack functions to create a hierarchy increasing in complexity and abstraction. The larger the stack,

the *deeper* the network is considered to be. In general, due to the increased complexity, deep learning algorithms require a large amount of data for training.

A simple and general representation of a deep learning model is shown in Figure 2.1. The basic structure consists of the following parts/sections:

1. Pre-processing: A sequence of words from the corpus cannot be taken directly as input; it needs to be converted into a format which the model can accept. Tokenization, stop-words removal, and lemmatization or stemming are some examples of pre-processing functions.

2. Encoder: After the corpus is pre-processed, it is given as input to the encoder. An encoder's core or basic task is converting the raw data or weak features into strong ones. The decoder can then use these strong features for simple tasks like classification or regression. The encoder consists of two basic parts:

    - Tail: The tail component of an encoder consists of transformations that help extract strong features from weak features. Although the basic functionality of the Tail and Body is identical, they are thought of as separate because the Tail for each domain is relatively constant. For example, the Tail for models in Language domain problems is an embedding layer that consists of one or more types of embeddings.

    - Body: This is the main feature encoding component of the encoder. Deep neural models generally have varying body styles. A prominent feature is that it has multiple repetitions of the same layer or groups of layers. Multiple repetitions help in improved feature extraction.

3. Decoder / Head: The decoder layers perform classification or regression using the features extracted by the encoder. Designing a decoder, also known as a Head, can

be considered as designing a classifier or regressor on a strong set of features. The components of a decoder, to some extent, rely heavily on the encoder to output a strong set of features. The ideal scenario for classification will be an input to the decoder, which is linearly separable and has a large margin between classes. An example of an extremely simple decoder is a Fully Connected Layer (FCN) with a softmax layer. The softmax layer can be removed for regression.

4. Post-processing: Many times, the result or output of the decoder or head still has some errors. For example, in a sequence tagging problem that includes span detection, if the first tag is B-X and the remaining tags in the span have a couple of I-Y's, this can be fixed by changing all the I-Y to I-X's. Here X and Y can be any distinct classes. The logic behind this processing is that a span cannot have tags of two different classes for one labeling.

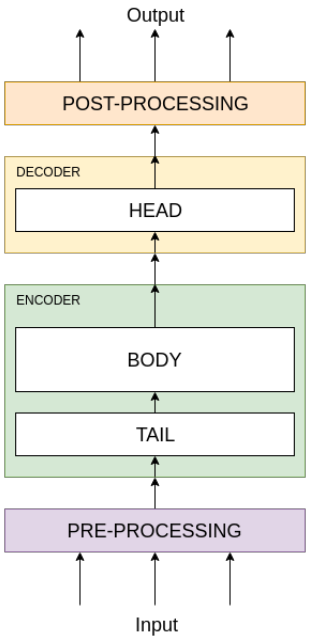

Figure 2.1. Outline of a deep learning model

Although conceptually, deep neural networks have been present since the 1970s, it is recently that the advancement in hardware has led to creating highly complex and deep neural networks. Primarily the domain of deep learning has been image processing; however, recently, deep learning has had a significant impact in the domain of natural language processing due to the creation of language models. The work of Vaswani et al. (2017) has been crucial in the creation of these language models. One of the most breakthrough architectures for language models was BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019). BERT, trained on two language modeling tasks, consists of 12-24 encoders with 12-24 bidirectional self-attention heads. Next, this section discusses SpanBERT, a variant of BERT that is used in this dissertation.

2.2.2 SpanBERT

The SpanBERT (Joshi et al., 2019) model extends the BERT model by changing the nature of the training tasks. The training tasks focus on *token spans* to obtain richer span representations for span-selection tasks like question answering and coreference resolution. Next, a brief overview of the SpanBERT model architecture is provided and that is followed with the description of the training tasks. For a detailed explanation of the model, experiments, and results, this dissertation refers the readers to Joshi et al. (2019).

Architecture

The SpanBERT model uses the same architecture as the BERT model. It consists of multiple layers of the Transformer model (Vaswani et al., 2017) and depending on the number of layers, SpanBERT has two variants - `base` with 12 Transformer layers and `large` with 24 Transformer layers. Figure 2.2 shows the architecture of the Transformer model[1]. For a

[1]The figure has been taken directly from the original work.

detailed description of the Transformer model, this dissertation refers the readers to Vaswani et al. (2017).

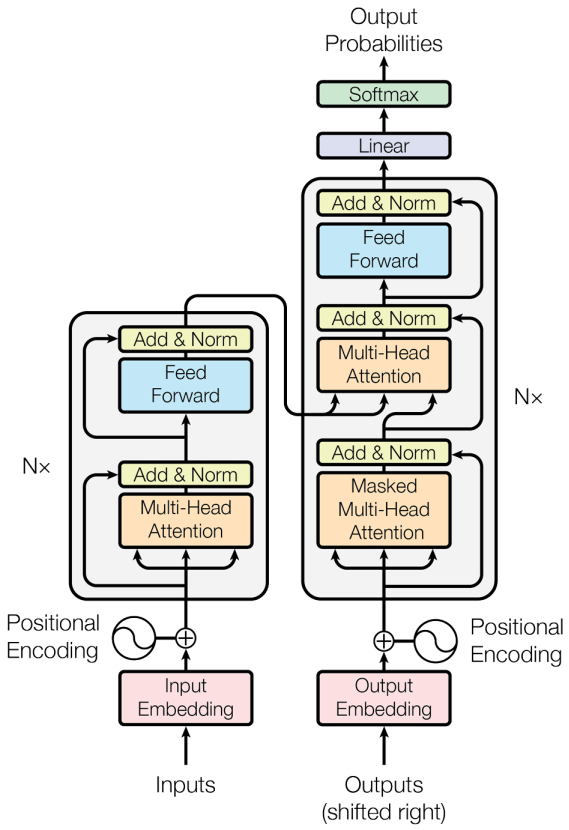

Figure 2.2. The Transformers model architecture.

Training tasks

The SpanBERT model uses Masked Language Model as the primary training task and Span Boundary Objective (SBO) as the secondary training task. Figure 2.3 shows the SpanBERT training tasks using an example[2]. The span *an American football game* is masked and the model tries to predict the token *football* using the equation 2.1. In the equation, $L_{MLM}(football)$ is the probability that the predicted token is *football* given the embedding

[2]The figure has been taken directly from the original work.

representation x_7. Next, $L_{SBO}(football)$ is the probability that the predicted token is *football* given the embedding representations x_4 and x_9, and positional embedding p_3 for the position of the token in the masked span. The example shows how the SBO task trains the model to learn span boundary embeddings (x_4 and x_7) that (i) together represent the span and (ii) be used to generate the span tokens.

$$L(football) = L_{MLM}(football) + L_{SBO}(football) \tag{2.1}$$

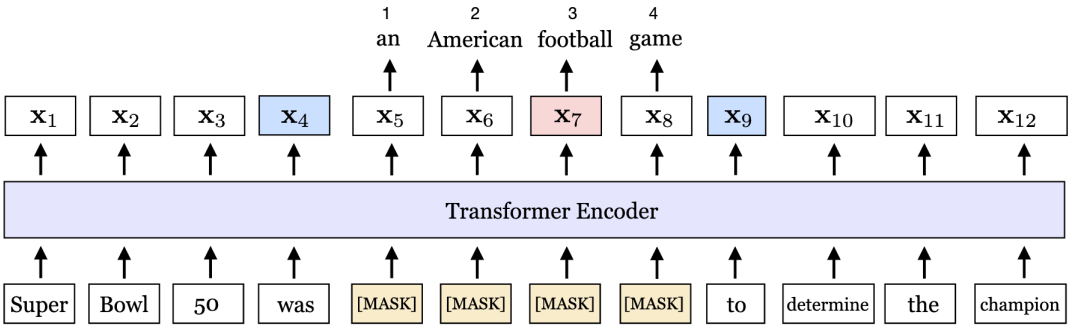

Figure 2.3. An example illustrating SpanBERT training.

2.2.3 SBERT

Lee et al. (2017) proposed `e2e-coref`, the first end-to-end neural entity coreference resolution model. The proposed model used GloVe embeddings (Pennington et al., 2014) for tokens and a bidirectional LSTM that used the token embeddings to generate span embeddings. A mention score is computed for each span, and for each valid antecedent:mention pair, an antecedent score is computed. Eventually, using these scores, the probability of an antecedent belonging to a chain is computed. The model was fine-tuned and evaluated on the CoNLL 2012 shared task (Pradhan et al., 2012) and GAP (Webster et al., 2018) corpus. The model was improved further by the authors using higher-order inference (Lee et al., 2018). The authors used an additional scoring function that is computationally inexpensive to do coarse

pruning. The improved model, `c2f-coref`, also used a language model to generate token embeddings in place of GloVe. Figures 2.4 and 2.5[3] show the `e2e-coref` architecture.

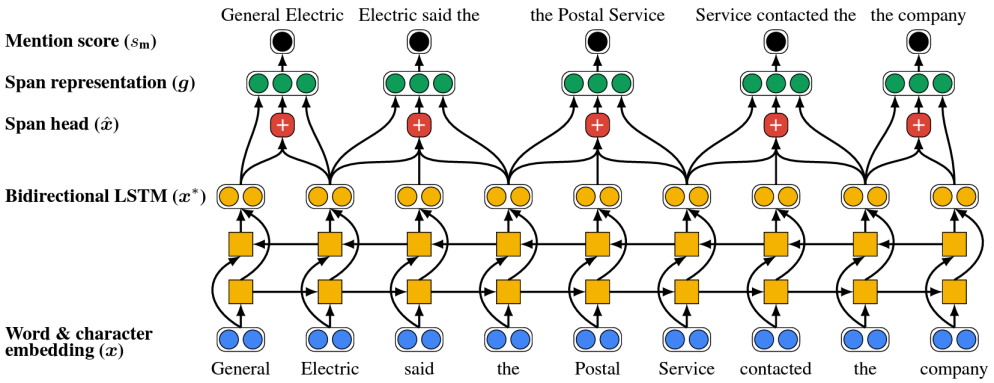

Figure 2.4. Span embedding generation and mention scoring in the architecture proposed by Lee et al. (2017).

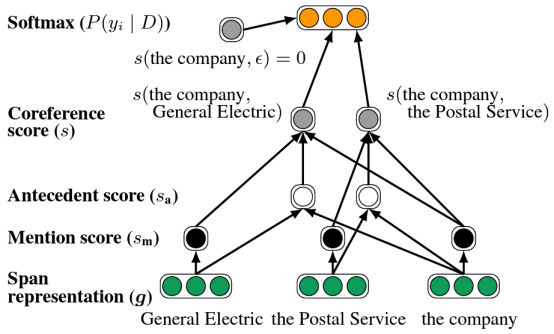

Figure 2.5. Antecedent distribution computation in the architecture proposed by Lee et al. (2017).

Although multiple works have improved upon this model, the work of Joshi et al. (2019) has been used most frequently in this dissertation. Joshi et al. replaced the bidirectional LSTMs used in `c2f-coref` with SpanBERT for obtaining richer embedding representations. The improvement in embedding representations resulted in a +3.09% increase in performance. This model is referred to as **SBERT** throughout this dissertation.

[3]The figures have been taken directly from the original work.

The task definition used in the SBERT architecture is explained in the remaining part of this section. Given a document D, the task of Coreference Resolution is to extract the set of coreference chains C from D. This is done by learning a distribution $P(\cdot)$ over possible antecedent spans $y \in Y$ for each mention span x.

$$P(y) = \frac{e^{s(x,y)}}{\sum_{y' \in Y} e^{s(x,y')}} \tag{2.2}$$

Here, $s(x, y)$ is a scoring function that generates a pairwise score for a coreference link between spans x and y. It uses fixed-length span representations g_x and g_y. For a span x, g_x is a concatenation of four vectors: embedding representations of the start and end token/word piece token of the span, embedding encoding the width of the span, and an attention vector computed over the span tokens. The scoring function $s(x, y)$ is computed using equation 2.3.

$$s(x, y) = s_m(x) + s_m(y) + s_c(x, y) + s_a(x, y) \tag{2.3}$$

In equation 2.3, $s_m(x)$ and $s_m(y)$ computes how likely spans x and y are mentions respectively. The function $s_a(x, y)$ computes the joint compatibility score of spans x and y. Equations 2.4 and 2.6 show how each of these scores are computed. Here, $\texttt{FFNN}(\cdot)$ is a feed forward neural network and $\phi(x, y)$ represent embedding which encodes speaker and metadata information.

$$s_m(x) = \texttt{FFNN}_m(g_x) \tag{2.4}$$

$$s_c(x, y) = g_x^T W_c g_y \tag{2.5}$$

$$s_a(x, y) = \texttt{FFNN}_c(g_x, g_y, \phi(x, y)) \tag{2.6}$$

As mentioned before, the authors proposed using a coarse-to-fine strategy for better and improved pruning (Lee et al., 2018). Pruning is carried out as a three stage process:

1. First stage: Select top M spans using their mention scores.

2. Second stage: Select top K antecedents of each remaining span i using the sum of $s_m(i) + s_m(j) + s_c(i, j)$.

3. Third stage: The overall coreference s(i,j) is computed based on the remaining span pairs.

CHAPTER 3

CORPUS

3.1 Introduction

3.1.1 Background

Numerous corpora have been used for email processing over time. University emails (Cohen et al., 1996, 2004), email users survey (Whittaker and Sidner, 1996; Brutlag and Meek, 2000), private emails (Manco et al., 2002; Corston-Oliver et al., 2004), simulated emails (Lam, 2002), and email archives (Nenkova and Bagga, 2004) are few of the initial sources for email corpora. Some of the popular email corpora are described below.

Avocado Research Email Collection: This corpus (Douglas Oard, 2015) consists of emails and attachments from 279 accounts of a now-defunct IT company. The accounts primarily belong to the employees of the company. The collection consists of 2.03M items divided into personal folders. These items are further divided into emails, attachments, appointments, contact, or report. The Avocado Collection is highly structured and consists of a large amount of metadata for every item. Although this corpus has been used in multiple email processing studies, it is not freely available[1].

PW CALO Corpus (Cohen et al., 2004): A collection of 222 email messages was generated over a four-day exercise. The exercise involved a group of 6 users taking part in group activities assuming different work roles.

W3C Mailing List Corpus: The W3C mailing list corpus was released as part of the TREC Enterprise search task in 2005 (Craswell et al., 2005). The corpus contains 198,394 emails in an HTML-ised format. However, parsed versions of the HTML

[1]https://catalog.ldc.upenn.edu/LDC2015T03

corpus are also available[2]. Ulrich et al. (2008) further annotate 30 email threads from the corpus for performing email summarization.

3.1.2 Dissertation Contributions

The significant contributions of this chapter can be summarized as follows:

1. The Enron Email Corpus (Klimt and Yang, 2004) is manually filtered for email conversations using a set of constraints. Language of email message contents, the validity of the email thread, and duplicate email threads are the primary constraints used in the filtering process.

2. The filtered corpus is then preprocessed using sentence boundary detection and tokenization. The preprocessed output is made publicly available in CoNLL format. This is the first publicly available preprocessed English email conversation corpus.

The remainder of the chapter describes the construction process of the base corpus used for this dissertation, and the challenges observed during the creation of the corpus are discussed. These challenges make knowledge extraction and knowledge representation a difficult task.

3.2 Enron Email Corpus

The Enron Email Corpus[3] (Klimt and Yang, 2004) is one of the few publicly available email corpora containing actual user interactions. It was created by the CALO Project [4] (A Cognitive Assistant that Learns and Organizes). The Enron Email Corpus is a multi-lingual

[2] `https://tides.umiacs.umd.edu/webtrec/trecent/parsed_w3c_corpus.html`

[3] `https://www.cs.cmu.edu/~./enron/`

[4] `http://www.ai.sri.com/project/CALO`

corpus with the majority of email threads in English. It contains emails of 150 employees, organized in a directory structure. Each employee directory is further organized into folders like inbox, drafts, deleted_items, sent_items, and other folders created by the employee. Over the years, contents of the corpus have been filtered to remove sensitive information like names, emails, or attachments. This corpus has been further annotated for various tasks like intent classification[5], hierarchy prediction (Agarwal et al., 2012), summarization (Carenini et al., 2007), and email classification (Alkhereyf and Rambow, 2017).

3.3 Terminology

Before the base corpus construction process is described, let us define the standard terms or concepts used throughout this work.

3.3.1 Email Message

An email message is split into three parts:

1. Header: An email header is a section containing the meta-data for the email message. Meta-data for an email message comprises information like sender and receiver details, date and time, subject, or message type. Example 2 shows an email header (red-colored text).

2. Footer: For this dissertation, any legal or confidentiality statement or notice inserted at the end of an email message body is considered an email footer. Example 2 shows an email footer (orange colored text).

3. Body: The remaining part of the email message is the email body. Example 2 shows an email body (blue colored text).

[5] `https://github.com/ParakweetLabs/EmailIntentDataSet`

Example 2. Example showing different sections of an email message.

```
Subject:  Hello

Date:  Mon, 30 Jul 2001 10:40:17 -0500

From:  "Davis, Dana" <Dana.Davis@ENRON.com>

To:  <mfoster@grti.tec.ar.us>


Good Morning Aunt Mae -

I got your email.  It was a wonderful surprise to see.

....

....

Anyway I be emailing you.  Love ya.

Dana

**********************************************************************

This e-mail is the property of Enron Corp.  and/or its relevantaffiliate and may
    contain confidential and privileged material for thesole use of the intended
    recipient (s).  Any review, use, distribu>

**********************************************************************
```

3.3.2 Email Thread

An email conversation is a sequence of email messages exchanged over time. Email conversations owing to their multi-recipient nature, are not turn-based, i.e., the conversation participants do not follow a specific order for responding to a previous email message. This reply-to nature of the conversation creates a tree-like structure when ordering the email messages sequentially. An email thread T is a path in this tree structure from the root to a leaf node.

Table 3.1. Top 10 directories for all users and their email count.

Directory Name	**Email Count**
all_documents	128,103
discussion_threads	58,609
sent	57,653
deleted_items	51,356
inbox	44,859
sent_items	37,921
notes_inbox	36,665
_sent_mail	30,109
calendar	6,133
archiving	4,477
Total	455,885

3.4 Base Corpus Construction

The primary challenge for any email processing task is the availability of a preprocessed corpus. Few tasks like email classification and summarization, due to work done on the tasks, have established corpora and baselines. For a new email processing task, the lack of a preprocessed corpus is a significant hurdle. Here, preprocessing comprises sentence boundary detection, tokenization, and output representation in a standard format like CoNLL.

The Enron Email Corpus is used as the email corpus for this dissertation. The corpus is available publicly and contains 517,394 email threads or 1.07M email messages. Compared to the W3C Corpus, the Enron Corpus is larger, and the email content is more balanced and not very technical. Tables 3.1 and 3.2 list the top 10 directories and the number of emails in them, and the distribution of email threads based on the thread length respectively. Although the corpus is primarily English, multiple email messages contain Spanish, Russian, German, and French content.

An email message in the Enron Email Corpus is further divided into two types (see Table 3.3):

Table 3.2. Distribution of email threads based on the thread length.

Thread Length	**Email Thread Count**
1	292,892
2-3	155,928
4-5	45,556
6-10	20,193
11-15	2,031
16-20	410
21-30	183
31-50	80
51-75	22
76-100	7
100+	2
Total	517,394

1. Email message with a full or normal header: An email message containing a header as shown in Examples 3 or 4 belongs to this type. The header format for both full and normal is nearly consistent across the entire email corpus, thereby facilitating parsing.

Example 3. Example of a full email header.

```
Message-ID: <16159836.1075855377439.JavaMail.evans@thyme>
Date: Fri, 7 Dec 2001 10:06:42 -0800 (PST)
From: heather.dunton@enron.com
To: k..allen@enron.com
Subject: RE: West Position
Mime-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
X-From: Dunton, Heather </O=ENRON/OU=NA/CN=RECIPIENTS/CN=HDUNTON>
X-To: Allen, Phillip K. </O=ENRON/OU=NA/CN=RECIPIENTS/CN=Pallen>
X-cc:
```

```
X-bcc:
X-Folder: \Phillip_Allen_Jan2002_1\Allen, Phillip K.\Inbox
X-Origin: Allen-P
X-FileName: pallen (Non-Privileged).pst
```

Example 4. Example of a normal email header.

```
From:  Allen, Phillip K.
Sent:  Friday, December 07, 2001 5:14 AM
To:    Dunton, Heather
Subject:      RE: West Position
```

2. Email message with a compressed header: A compressed email header contains only partial information like sender, recipient, date-time, or subject. Example 5 shows samples of such headers found in the Enron Email Corpus. The lack of additional information like recipients and subject affects the information extraction task.

   Example 5. Examples showing compressed headers found in the email corpus.

   ```
   ---------------------- Forwarded by Kirk McDaniel/HOU/EES on 11/27/2001
      10:04 AM --------------------------


   >>> "Bailey, Susan" <Susan.Bailey@ENRON.com> 11/16/01 09:23 AM >>>


   >>> <Kim.Ward@enron.com> 10/16/01 10:37AM >>>
   ```

The remainder of the section describes the steps to create the base corpus used for all the other tasks. This base corpus will finally be represented in the CoNLL format. This representation allows the direct testing and usage of multiple off-the-shelf models available for various NLP tasks.

Table 3.3. Distribution of email messages based on email message type.

Type	**Email Message Count**
Full Header	931,178
Compressed Header	138,825
Total	1,070,003

3.4.1 Extraction and Filtering

The first step in creating the larger corpus is to shortlist email threads from the Enron Email Corpus. An email thread conversation is a valid conversation if it contains four or more email messages. However, to increase the size of this shortlisted pool of email threads, we do not restrict the scope only to email threads in the *inbox* directory. For each user, email threads in all directories except *all_documents*, *discussion_threads*, *drafts*, *deleted_items*, *sent_items*, *sent*, *_sent_mail*, and *_sent* are considered. Email threads in previous directories are omitted as they are either auto-generated, discarded, or are part of other email threads. A total of 9,724 email threads with a minimum of 4 email messages in each thread are obtained after including additional directories.

On obtaining the initial set of candidate email threads, the following types of email threads are filtered manually from the resulting set:

1. Duplicates: An email thread that is part of a larger email thread or is a duplicate belongs to this category. The multi-recipient nature of email conversations results in one email thread possibly being present in directories of multiple users.

2. No content: An email thread containing m email messages and >m/2 email messages containing no email body falls in this category.

3. Invalid attachments: The Enron Email Corpus consists of email threads with inline document attachments. Some email threads contain attachments as long hexadecimal strings and are hence labeled as invalid content.

Table 3.4. Distribution of email threads per filtering category.

Email Thread Category	**Email Thread Count**
Duplicates	2,867
No content	564
Invalid attachments	75
Non-English content	54
Accepted Email Threads	6,164
Total	9,724

4. Non-English content: Email threads in the Enron Corpus consists of messages or text in English, Spanish, Russian, German, and French. The scope of this work being restricted to English, email threads containing text in any other language are discarded.

After filtering from the initial set, 6164 email threads are obtained. Table 3.4 gives a distribution of the initial email threads in each of the filtering categories.

3.4.2 Preprocessing

Post extraction and filtering, preprocessing is carried out on the selected email threads. The first operation is cleaning up the email threads of special characters or sections which do not contribute to the email thread. Example 6 provides some instances of such type.

Example 6. Examples of special characters or sections present in the corpus.

```
1. =09=09=09=09=09  Search Over 100,000 Artists ArtistAl....
2. Regards, Shmuel.
########################################
Shmuel S. Oren
3. =20
Your use of Yahoo! Groups is subject to the Yahoo! Terms of Service.=20
4. ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
```

C L I C K Z T O D A Y

Friday, June 8 2001

The Internet's Leading Resource http://www.clickz.com/

for Doing Business Online_____

- -

This e-mail is optimized for mono-spaced typefaces.

For maximum legibility, specify Courier as your e-mail font.

- -

for subscription options, see the bottom of this mailing.

- -

The next operation is sentence boundary detection (SBD). This is a crucial task for both entity detection as well as relation extraction. For SBD, the pysbd[6] package is used. Example 7 shows the result of performing SBD. It is important to note that the result of SBD is not optimal or correct for multiple instances. At this stage, however, these errors are not rectified and this dissertation accepts the operation's output.

Example 7. Example of SBD output.

Before:

> 1127 PST APS sent the following WSCCnet message:

> APS has met all WSCC USF procedure requirements for Path 23 unscheduled

> flow accommodation. All local controllable devices have been utilized.

> APS now requests the use of the Coordinated Controllable devices. Please

> check your schedules.

After:

1127 PST APS sent the following WSCCnet message: APS has met all WSCC USF procedure requirements for Path 23 unscheduled flow accommodation.

[6]https://pypi.org/project/pysbd/

```
All local controllable devices have been utilized.
APS now requests the use of the Coordinated Controllable devices.
Please check your schedules.
```

The final operation in preprocessing is the tokenization of extracted sentences. For the tokenization operation, a white-space tokenizer is used. Using a white-space tokenizer on the sentence *'All local controllable devices have been utilized.'* the tokens {*'All', 'local', 'controllable', 'devices', 'have', 'been', 'utilized', '.'*} are obtained.

3.4.3 Feature Annotations

Past work done on emails has highlighted the importance of various features for different tasks. This dissertation studies and evaluates the following features for an email thread:

1. Message identifier (MI): For an email thread T containing N email messages, message identifier for a token x belonging to message i ($i \in \{0, 1, ..., N\text{-}1\}$) is i.

2. Section information (SI): An email message is divided into three sections: header, body, and footer. This feature assigns one of the header(1), body(2), and footer(3) classes to each token in an email message.

3. Reversing an email (REV): Reversing email messages in a thread refers to ordering the messages as per time in the email header. This is expected to enhance understanding of the conversation flow in the thread.

A small subset of 171 email threads is chosen from the base corpus for performing feature annotations. The feature annotations are first annotated automatically using keywords found in an email header like *From*, *To*, *Subject*, *Forwarded By* and *Original Message*. Next, the obtained annotations are then corrected manually. In addition to the annotations, sentence boundaries for these email threads are also corrected. The need for manual correction is highlighted in the challenges described in the next sub-section.

3.5 Challenges

3.5.1 Email addresses

An email address is a unique identifier for every user having an email account and hence is a mention representing a Person or an Organization entity. An email message in its entirety that is header and body most certainly contains the email addresses of the sender and recipient(s). However, it may or may not contain the name of the sender, the name(s) of the recipient(s), or both. Thus, it is crucial to identify and link an email address to the entity it represents.

Generally, email addresses bear some lexical similarity to the name of the entity it represents. However, there are also instances when there is no overlap between the entity's name and email address. Additionally, an email address can represent a group of individuals as a whole. Such email addresses are called aliases. The difficulty of tracing conversations increases when an alias is involved in an email conversation. Example 8 shows various types of email addresses along with the corresponding name of the entity, if available, it represents.

Example 8. Examples of different types of email addresses along with the names of their corresponding entities.

g..barkowsky@enron.com - Barkowsky, Gloria G.

theresa.staab@enron.com - Staab, Theresa

smu-betas@yahoogroups.com - SMU Betas

fackel@yahoo.com - Leah

3.5.2 Different email thread structures

A consistent email header, body, and thread structure ease the preprocessing task and extraction of various features. It also helps in faster error analysis. The emails in the Enron corpus have varied header as well as email thread structures. A fixed structure of an email

thread plays an important role in deciding email boundaries and thereby the scope of different pronouns that are local to an email message in the thread. Threads in the Enron Corpus generally follow a time-based last to first ordering. However, multiple instances of out-of-order threads as well as different email structures are seen. Example 9 shows one such structure.

Example 9. An example showing an out-of-order email thread structure present in the Enron Corpus.

```
...
 -----Original Message-----
Sent:   Friday, May 25, 2001 12:35 PM
...
email contents
....
---------------------- Forwarded by Jaime Sanabria/ENRON_DEVELOPMENT on
   05/25/2001 12:42 PM
   ---------------------------
on 05/21/2001 03:49:00 PM
To:   "ENRON: Sanabria, Jaime" <jaime.sanabria@enron.com>
...
```

3.5.3 Name abbreviations and variations

Identifying the name of a PER or an ORG entity is crucial for correct coreference chain identification and tasks like anaphora resolution or question answering. The semi-structured nature of email messages adds to the complexity of identifying all names referring to the same entity. The names present in the email message headers for PER type are either full names or the names that are registered in the system. However, in an email message body,

name abbreviations or variations are used between frequent or known participants. For an ORG entity, the signature found at the end of an email message contains a non-abbreviated version of the name compared to the names found in the message subject or body. Examples 10 and 11 shows a few name abbreviations and variations observed in the base corpus.

Example 10. *Frazier, Perry* referred to as *PT*, *Kimberly* as *Kim*, *Miller, Mary Kay* as *MK* and *Transwestern Commercial Group* as *TW*.

Example 11. *Robert Superty* $\longleftrightarrow$ *Bob Superty* and *William E. Brown* $\longleftrightarrow$ *Bill Brown*.

3.5.4 Fragmented email headers

An email header which does not follow the format shown by Examples 3, 4 and 5 is termed as a fragmented email header. Example 12 shows an instance of an invalid email header and the correct version of the same header. The effect of a fragmented email header is first observed in the output of the SBD operation. For the header in Example 12, the sentences obtained as the output of SBD are {'Laura A. de la Torre', 'To: Mery L.', 'Brown/Internal/Accenture@Accenture, Sheri A. 11/20/2001 05:35 PM Righi/Internal/Accenture@Accenture', 'cc:', 'Subject: Conversion and', 'Arbitrage Q&A'}.

All three features considered in Section 3.4.3 are dependent on correctly identifying the start of an email header which in turn is dependent on the correct identification of sentence boundaries. From the output of the SBD task given before, it can be seen that of the detected 7, only two sentence boundaries are correct. In addition to the incorrect SBD output, the missing keyword *'From'* in the email header adds ambiguity as *'Laura A. de la Torre'* can either be the name of the sender of the previous email or the current email (header shown in Example 12). Due to these errors, manual correction of the annotations obtained using rules needs to be performed.

Apart from impacting the feature annotation process, the errors in the SBD output also affect the entity extraction task as fragmented headers often split an entity into parts

resulting in multiple entities or no entity being identified. For the relation extraction task, a similar effect is observed.

Example 12. Example of a fragmented and correct email header

Fragmented:

```
Laura A. de la Torre
                                    To:     Mery L.
Brown/Internal/Accenture@Accenture, Sheri A.
       11/20/2001 05:35 PM          Righi/Internal/Accenture@Accenture
                                    cc:
                                    Subject:    Conversion and
Arbitrage Q&A
```

Correct:

```
Laura A. de la Torre
11/20/2001 05:35 PM
To: Mery L. Brown/Internal/Accenture@Accenture, Sheri A. Righi/Internal/
    Accenture@Accenture
cc:
Subject: Conversion and Arbitrage Q&A
```

CHAPTER 4

ENTITY COREFERENCE RESOLUTION

4.1 Introduction

4.1.1 Background

Entity Resolution has received attention in the natural language research community since the 1960s, with noun-phrase and pronomial resolution being the early forms of the task. Shared tasks such as CoNLL 2012 (Pradhan et al., 2012) and MUC (Grishman and Sundheim, 1996) define it as linking referring spans of text that point to the same discourse entity. Over the years, numerous corpora have been released for coreference resolution, with MUC-6 (Grishman and Sundheim, 1996), MUC-7 (Chinchor, 1998), ACE (Doddington et al., 2004) and OntoNotes being the popular ones. OntoNotes 2.0 and OntoNotes 5.0 were used in Task-1 of SemEval 2010 (Recasens et al., 2010) and CoNLL 2012 shared task (Pradhan et al., 2012) respectively. However, each of these corpora either fully or mainly comprise of news articles.

Although multiple corpora released over the years contain a small fraction of telephonic speech text, only a few corpora have focused on studying the task in a purely conversational setting. Character Identification Corpus (Chen and Choi, 2016) was the first corpus to focus on the entity-linking task in this setting. It was constructed using TV show transcripts with annotations for speakers in a multi-party conversation. One of the earliest works to consider annotating coreference chains for emails was done by Goldstein et al. (2006). The authors proposed annotating 2000 emails from the Enron Email Corpus. However, the project webpage[1] is still under construction, and no contact could be established with the authors. The Manually Annotated Sub-Corpus (MASC) (Ide et al., 2008) project is

[1] `https://jikd-email.umiacs.umd.edu/corpus/`

another corpus to consider annotating coreference chains for emails. The corpus includes 45 emails from the Enron Email Corpus (Klimt and Yang, 2004), 96 spam emails, and 35 w3c email digests. However, no coreference annotations were released as a part of the corpus. Furthermore, the emails considered from the Enron Corpus were single messages compared to our work which focuses on email threads. Aktaş et al. (2018) used a Twitter corpus to study the performance of Stanford statistical coreference system (Clark and Manning, 2015). They evaluated a corpus with 185 threads containing 278 coreference chains and reported a mediocre performance.

Previous works in the literature have explored specific cases of entity coreference resolution for email conversations. Abadi (2003) was one of the earliest works to explore coreference resolution in email conversations. The author performs anaphora resolution on an e-commerce email dataset. The resolution process was carried out only for third-person pronouns and not all entity mentions. Culotta et al. (2005) extract names and email addresses of PER entities from email headers and also carry out coreference resolution to group extracted names. However, the authors extract from email headers and not the entire email body as the work focuses on social network analysis. Compared to this, considering all mentions in the email thread allows us to capture entity interactions. Diehl et al. (2006) carry out name reference resolution on a subset of Enron emails. Their work aims at resolving first-name references for all Enron employees only. Although this is a crucial step in identifying and chaining entities across emails, the focus on only first-name references and Enron employees eliminates a large chunk of mentions. The work of Elsayed and Oard (2006) comes closest to our work. Elsayed and Oard (2006), via identity modeling, carry out person name, nickname, and email address resolution using email headers, salutations, and signatures found in the email body. The resolution process, however, does not consider pronouns. Thus, in a general sense, we consider entity resolution as an unexplored problem for email conversations. The main challenge to solving the problem is the lack of an annotated corpus. An annotated corpus can be used to perform qualitative and quantitative error analyses of existing solutions.

This analysis can help identify the limitations of those solutions and provide a direction for developing a new solution. It is essential to note that all previous works focusing on reference resolution created an annotated corpus but failed to release it publicly.

4.1.2 Dissertation Contributions

The main contributions of this chapter are as follows:

1. This chapter explores the problem of entity coreference resolution in a generic setting for the first time. A manually annotated coreference resolution dataset `SEED` is created and used to demonstrate the difficulty of the problem empirically. `SEED` is the first publicly available dataset containing manual entity coreference resolution annotations.

2. Using `SEED`, a large-scale dataset (`CEREC`) containing weak entity coreference annotations was created. This publicly available dataset contains 60,383 coreference chains and 445,762 annotated mentions.

3. Two limitations of the current models on `SEED` and `CEREC` are identified. The work proposes a joint learning framework and singleton post-processing to overcome the limitations and report an increase of 5.26 and 4.87 F1 points on `SEED` and `CEREC`.

4.2 Entities

Before defining the task, the types of entities that will be considered in this dissertation[2] are defined:

1. Person (PER): A single individual or a group of individuals can be annotated as a Person. A Person can be specified by name (John Doe), email address (johndoe@abc.com),

[2] `https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-entities-guidelines-v6.6.pdf`

first name (John), last name (Doe), occupation (the accountant), family relation (dad), pronoun (he), or by a combination of these. All fictional human characters appearing in movies, TV, or books are considered a Person entity. A group of individuals that do not meet the requirements for an Organization entity can be annotated as a Person entity. For example, "Analysts", "IBM's lawyers," "The family," or "The house painters." References of Person entities, marked red, are shown in Examples 13 and 14.

2. Location (LOC): Places defined on a geographical basis, those that constitute a political entity, and where a person or an organization can be found are Location entities. An address, desk locations, one-dimensional location like a border between two other locations, water-body, natural land-regions, non-named locations ("southern Africa"), and general regions like "part of the city," or "airspace." References of Location entities, marked orange, are shown in Example 14.

3. Organization (ORG): An organization entity must have some formally established association. Typical examples are businesses, government units, sports teams, and formally organized music groups. A department inside a company can also be termed as an organization. References of Organization entities, marked green, are shown in Example 14.

4. Digital (DIG): A digital entity is a media or pointer to a media present on some form of digital storage. For example, email attachments, URLs, directory addresses. References of Digital entities, marked blue, are shown in Example 13.

Example 13. Example of an email with Person and Digital entity references.

```
From:  Dunton, Heather
Sent:  Tuesday, December 04, 2001 3:12 PM
```

To: Belden, Tim; Allen, Phillip K.

Cc: Driscoll, Michael M.

Subject: West Position

Attached is the Delta position for 1/18, 1/31, 6/20, 7/16, 9/24

<< File: west_delta_pos.xls >>

Let me know if you have any questions.

Heather

Example 14. Example of an email with Person, Organization and Location entity references.

From : Toews , John

Sent : Wednesday , November 07 , 2001 3:10 PM

To : Johnson , Richard C ; Chanley , Earl

Subject : Tie in

I received a call form Bob Bradly who is representing Vern E Falkner , and he needs to talk to someone about possibly doing a tie - in to our pipeline in Hansford County Texas . His phone number is 405 - 842 - 4334 .

I attempted to contact both of you via phone and was unsuccessful , hence the e - mail . Did not wish for this to fall through the cracks . Thanks , John Toews , OCC

4.3 Problem Definition

We now formally define the entity resolution task for email threads. Let T be an email thread containing N email messages and M be the set of all mentions in T and E be the number of unique entities present in T. Let C be a set of chains of mentions $\{c_1, c_2, ..., c_E\}$, where each chain contains mentions referring to a unique entity. The term chain is analogous to a

coreference cluster. Here, an entity belongs to one of the following classes: Person (PER), Organization (ORG), Location (LOC), or Digital (DIG). Compared to the CoNLL 2012 Shared Task, all singleton chains in T are considered to be a part of C. A singleton chain contains a single mention. Therefore, given an email thread T, the entity resolution task is to identify C. Example 15 shows a sample email message and the corresponding entities. The colored tokens represent entities, and the entities with similar colors belong to the same coreference chain.

Example 15. Example of entity resolution task in email conversations

```
Date: Mon, 17 Dec 2001 14:28:03 -0800 (PST)
From: g..barkowsky@enron.com
To: theresa.staab@enron.com
Subject: RE: Final Statements and Invoices for November
X-From: Barkowsky, Gloria G.
X-To: Staab, Theresa
yes, I 'll do this. Do you have anything for Crestone and Lost Creek?
```

4.4 Problem Evaluation

4.4.1 SEED Dataset

The base corpus created in 3.4.2 is used as the input for the annotation process. A subset of 46 email threads containing 245 email messages is selected by randomly choosing 16 users and then considering all the emails for those users. For the scope of this task, the entity types PER, LOC, ORG, and DIG are used. When marking a mention, the following guidelines are observed:

- The part of speech of a mention can be one of *Nouns*, *Noun Phrases* and *Pronouns*.

- **No** event or verb is to be annotated.
- **No** date, time, or date-time is to be annotated.
- When deciding on the width of a mention, the shortest width which describes the entity is chosen.

An email conversation, owing to the *To*, *Cc* and *Bcc* fields, can result in having participants with different levels of involvement. The participants in the *To* field are deemed to be directly involved in the conversation, and those in *Cc* and *Bcc* to be indirectly involved. Pronouns such as 'you,' 'team,' 'everybody,' and 'your' refer to each direct participant individually. This approach is similar to the one followed by Zhou and Choi (2018) in their work on the resolution of plural mentions. The annotated corpus is referred to as `SEED`.

Compared to the OntoNotes 5.0 corpus, annotations for singleton chains are present in `SEED` to help the model understand email address and name mentions in the email header during fine-tuning. As an exception to this, singleton pronoun chains are excluded.

The annotation process for `SEED` was carried out manually as a two-step process: identifying the mentions and chaining them. For the complete process, three annotators were used. Inter-annotator agreement on the Fleiss et al. (2003) Kappa statistic was $\kappa = 0.87$. A high κ value is due to a large number of unambiguous email addresses and names in `SEED`. All cases where no agreement was reached were resolved by discussion.

Table 4.1 gives details on the size of annotations in `SEED`. The distribution of mentions per entity type is given in Table 4.2.

Note that `SEED` also contains speaker annotations. Before manually annotating `SEED`, an evaluation of weakly annotating email threads using the SBERT model with few manually annotated samples was carried out. Poor performance on this evaluation led to manually annotating `SEED`.

Table 4.1. Details on the size of annotations in SEED.

Statistic	**Value**
Email Threads	46
Email Messages	245
Coreference Chains	866
Annotated Mentions	5834
Annotated Pronouns	981
Length of longest coreference chain	77
Average Length of coreference chains	6.73
Singleton chains	106

Table 4.2. Mention and entity distribution per entity type.

	PER	**ORG**	**LOC**	**DIG**
Mentions	76%	14%	4%	6%
Unique Entities	69%	17%	8%	6%

4.4.2 Challenges in Email Conversations

The problem of anaphora resolution for Twitter conversations (Aktaş et al., 2018) exhibits characteristics similar to the problem in consideration. Email conversations are similar to Twitter conversations in the tree structure, constructed by the conversation's 'reply-to' nature. Furthermore, Twitter handles are analogous to email addresses and retweeting to forwarding. Lastly, both emails and tweets show some basic structure for a header and body present in every sample.

Nevertheless, there are numerous differences between the two. Firstly, the use cases that the two mediums serve are very different. Twitter is a microblogging and social networking platform in which, by default, all conversations are public and intended for a much larger audience. On the other hand, an email or "electronic mail" is intended, like regular mail, directly for just the recipient individually or as part of a group. Secondly, the language in tweets uses many character reducing strategies or *textisms* (Lyddy et al., 2014) due to the

character limit constraint. Since no explicit word limit is set for a single email, the text is often more elaborate and descriptive.

In addition to the challenges described in Section 3.5, the challenges observed in general email conversations are described below:

1. Typos affecting referring expressions.

   Example 16. *They will also be proposing that the Commission switch from long run marginal cost to embedded cost principles for allocating costs of service among its customers.* ***The*** *also propose a $187 million or 12.5% rate increase annually, compared to present rates.*

2. Speaker references: Email conversations are multi-user conversations by nature. Due to this, third-person pronouns are used very frequently in an email conversation. Aktaş et al. (2018) is the only work in our knowledge considering this phenomenon in a conversational setting. Although an email thread can be viewed as a turn-based sequential conversation over time, the time sequencing may not align with the flow of the conversation, thereby adding to the complexity of the task.

3. Ambiguity with first-person plural pronouns: In an email conversation, the participants represent a larger group or an organization, especially in a formal setting. These cases add ambiguity to the resolution of first-person plural pronouns. Consider the pronoun '*we*'. It can resolve to both the sender and recipient together or the entity the sender is representing.

4.4.3 Experiments

Models

Entity resolution on `SEED` is evaluated by considering both within document (WD) and cross-document (CD) formulations of the task. For the WD formulation, the SBERT model

described in 2.2.3 is used. We refer to the SBERT model trained on the OntoNotes 5.0 corpus as **OntoSpanBERT**. For the CD formulation of the task, the model proposed by Barhom et al. (2019) is used. The model was trained jointly on ECB+ corpus for both event and entity resolution tasks and is the current state-of-the-art for the ECB+ corpus. The model iteratively performs event and entity coreference resolution. The results of each subtask are alternately used to merge predicted chains in each iteration. The authors use mention lexical span, surrounding context, and event-entity mention relations via predicate-arguments structures to obtain predictions.

Setting

For the corpus evaluation in the WD setting, the independent variant of OntoSpanBERT[3] has been used. Since the original CoNLL 2012 task does not include singleton chains in its training and predictions, the scores with and without singleton chains in the corpus are reported during the computation of performance metrics. The input to the models is in CoNLL 2012 format. The experiments were run on a GPU environment comprising of 8 cores of Nvidia GTX 1080 Ti with 12 GB of memory per core.

Evaluation

The majority of the recent work done on entity coreference uses the MUC, B^3 and CEAFE metrics (Pradhan et al., 2012). Moosavi and Strube (2016) show the shortcomings of each of these metrics and propose the Link-based Entity Aware (LEA) metric[4]. The results using all four metrics for comparability as well as correctness have been reported. The

[3] `https://github.com/mandarjoshi90/coref`

[4] Zhou and Choi (2018) propose variations of B^3 and CEAFE, which may be more appropriate here since this work follows a similar annotation scheme. However, for ease of comparison, this dissertation skips using these variations.

official scorer[5] provided by CoNLL 2012 shared task is used. The official scorer raises a non-crashing duplicate reference error when a single-token mention belongs to more than one chain. This error is also observed on the OntoNotes corpus, and hence this work reports the scores ignoring the errors.

Both models deal with more entity types than those defined here, like Facility, Event, Product, or Vehicle. However, since the model does not output the entity type of a chain, no chain from the predictions is removed. Furthermore, since none of the models were trained on the DIG entity type, scores excluding DIG annotations have also been reported. Tables 4.3 and 4.4 show the empirical results of the experiments. For the WD setting of the task, the OntoSpanBERT performs best when the test data contains only PER, ORG, and LOC entity types and no singleton chains. The +0.99 F1 increase after removing singleton chains is understandable as removing singleton chains leads to a reduction in the size of expected final chaining resulting in a higher recall than the general setting. Likewise, the drop in precision for all metrics after removing the DIG entity type shows that the current model already captures the type even if the training corpus did not contain annotations for the DIG type.

Next, a SBERT model is fine-tuned on `SEED` using an 80:20 train-test split. This model is referred to as **SeedSpanBERT**. The results obtained show that SeedSpanBERT exhibits the best performance. However, it is essential to note that the test set contains merely ten email threads.

Note that fine-tuning OntoSpanBERT on `SEED` was attempted, and it did not result in any improvement in the results observed before the additional fine-tuning. The small size of `SEED` results in only slight weight perturbations, which is not significant enough to change the predicted chaining.

[5] `https://github.com/conll/reference-coreference-scorers/tree/LEA-scorer`

Table 4.3. Evaluation results for OntoSpanBERT and SeedSpanBERT on SEED. Avg. F1 score is computed using MUC, B^3 and CEAFE metrics.

Model	MUC			B^3			CEAFE			Avg.
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**F1**
OntoSpanBERT	58.7	40.8	46.5	50.9	23.1	30.1	29.7	30.7	28	34.9
w/o singletons	58.7	40.9	46.6	50.8	23.6	30.5	29.6	34.5	29.8	35.6
w/o DIG	57.6	42.2	47.2	49.7	23.8	30.4	28.9	31.4	27.8	35.2
w/o both	57.6	42.2	47.2	49.6	24.2	30.9	28.7	**35.2**	29.4	35.9
SeedSpanBERT	**79.5**	**64.8**	**70.9**	**62.3**	**46.6**	**52.1**	**57.4**	31.7	**39.1**	**54**

Table 4.4. Evaluation results using LEA metric for OntoSpanBERT and SeedSpanBERT on SEED.

Model	**P**	**R**	**F1**
OntoSpanBERT	47.6	20.6	27
w/o singletons	46.2	20.4	26.8
w/o DIG	46.2	21	27.3
w/o both	29.4	48.3	19.6
SeedSpanBERT	**57.4**	**42.6**	**47.8**

In a cross-document formulation, an email thread is viewed as a collection of email messages. Since SEED does not include event annotations, the predictions obtained were not meaningful and thus, could not be used for evaluation.

4.4.4 Error Analysis

Error analysis presented here has been performed on the predictions of OntoSpanBERT and SeedSpanBERT. The predictions obtained by the OntoSpanBERT model were assessed with different variations of SEED (without singletons, DIG entity type, or both). There are five general types of errors observed. Primary error analysis is performed on the predictions of the OntoSpanBERT model, and changes observed in the predictions on the test set using SeedSpanBERT are reported. Table 4.5 gives more information on the statistics of each

error type. It is important to note that since the error categories are not mutually exclusive, the possibility of a span of text contributing to more than one error category exists. The objective here is to get an insight into the type of errors observed and the individual statistics.

Figure 4.1 shows a comparison between OntoSpanBERT and SeedSpanBERT in the distribution of the first and fifth category errors per entity type on the test set. The comparison considers only email threads in the test set. It can be seen that a high percentage of PER mentions in the corpus largely influence the learning of the model. Additionally, since SEED compared to the OntoNotes 5.0 corpus does not contain sufficient ORG mentions, OntoSpanBERT will likely perform better on ORG mentions. From the performance of the models on the DIG entity type, it can be inferred that OntoSpanBERT, in terms of mention span identification, does a better job at implicitly capturing the entity type.

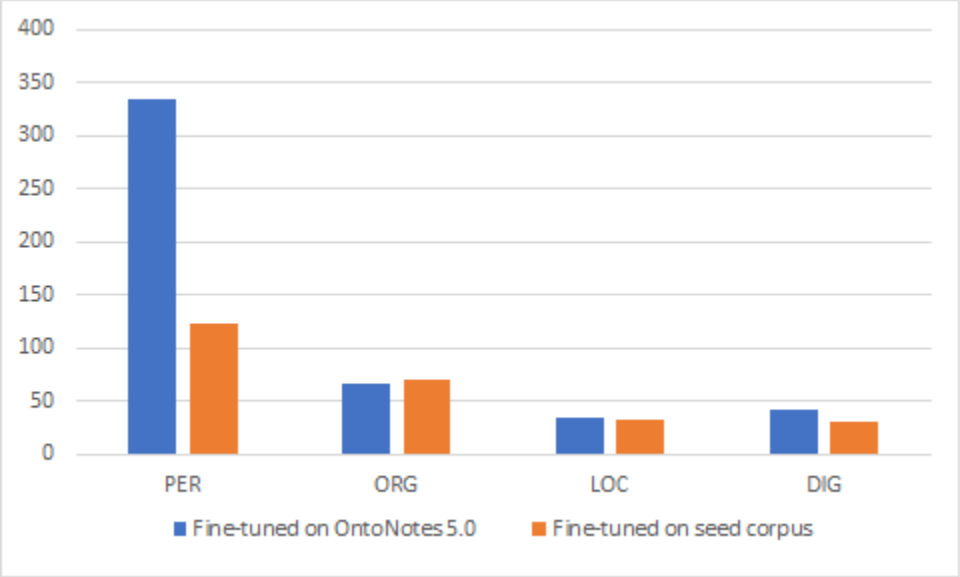

Figure 4.1. Comparison of error distribution per entity type between OntoSpanBERT and SeedSpanBERT

Missing references in the chain

1. Missing references in the email header - The English language section of the OntoNotes 5.0 corpus consists of texts from one of the following categories: newswire, magazine ar-

Table 4.5. Statistical information of errors observed. Count_f column reports numbers observed on the full SEED dataset, Count_1 on the test set with OntoSpanBERT, and Count_2 on the test set with SeedSpanBERT respectively.

Category	**Count$_f$**	**Count$_1$**	**Count$_2$**
Missing references in the chain	1587	476	254
Decomposition of a single chain	135	42	22
Wrong items in the chain	245	90	254
Missing chains	222	65	27
Incorrect and irrelevant mention spans	1054	271	27

Table 4.6. Distribution of missing references per pronoun type.

	1st Person	**2nd Person**	**3rd Person**	**Other**
%	36%	52%	2%	10%

ticles, broadcast news, broadcast conversations, web data, conversational speech data, and English translation of the New Testament. None of these categories have texts with a header close or similar to an email message header. Additionally, the corpus contains no email addresses, thereby making an email address an unfamiliar concept to the trained model. 45% of missing references in the chain belong to this sub-category, of which missing email address mentions are 56%, and missing name mentions are 44%.

2. Missing pronomial references - This error sub-category contributes to 11% of this error category. Table 4.6 gives a distribution of the errors per pronoun category.

3. Other missing references in the chain - The final sub-category consists of the remaining missing references. These errors are further divided into their corresponding entity types to understand the missing references better. Table 4.7 shows the corresponding breakdown. The results show that even though the model was never trained on the DIG entity type, it partially or completely predicted 210 mentions out of 348 present in SEED.

Table 4.7. Distribution of missing references per entity type.

	PER	**ORG**	**LOC**	**DIG**
%	37%	31%	13%	19%

Table 4.8. Detailed statistics of error reductions with SeedSpanBERT as compared to OntoSpanBERT. 1: Count increased from 5 to 116. 2: Count increased from 4 to 23.

Type	**% change**
Missing references in the chain	-46
Missing references in the email header	-78
Missing email references	-77
Missing name references	-80
Missing pronomial references	-63
Other missing pronomial references	-19
Decomposition of a single chain	-48
Wrong mentions in the chain	+182
Pronouns	$+2220_1$
Other PER entity mentions	$+475_2$
Missing chains	-59
Chains of length 2	-86
Incorrect and non-relevant mention spans	-92
Incorrectly identified mention spans	-65
Non-relevant mention spans	-93
Duplicate name mention spans	-100

From the results of SeedSpanBERT, it can be inferred that the training process helped the model learn the importance of email headers and focus on the relevant entity mention types. It does not help in solving the mention identification problem in the rest of the email body.

Decomposition of a single chain

This error category represents the cases when a single coreference chain in the annotated corpus was present as multiple chains in the predictions. However, taking a union of all

these chains does not necessarily obtain the original annotated chain. Of the decomposed chains, 58% are split into two parts, 33% into three parts, and 8% into four parts. One instance of decomposition into five parts is seen. Even though there is a reduction in the number of decomposed chains with SeedSpanBERT, the fine-tuning process creates longer chains composed of multiple single entity chains. Singleton chains are the dominant ones to be absorbed in other chains. The small size of `SEED` is a factor that attributes to this behavior.

Wrong mentions in the chain

This error category indicates that an incorrect mention is identified as part of a coreference chain. The majority of the errors on the entire seed corpus are pronouns (68%) and other PER entity mentions (17%). Post-fine-tuning, the number of wrong mentions in the test set increased by 182%. Merging chains of length 2-3 into bigger chains or other chains of similar lengths is the primary factor for this significant increase. The scores on the MUC metric show that the OntoSpanBERT model does a better job at chaining mentions but lacks the knowledge of identifying mentions in an email corpus. On the other hand, the SeedSpanBERT model, post-fine-tuning on `SEED`, learns how to identify mentions but fails at the chaining task.

Missing chains

Since the CoNLL 2012 shared task corpus does not contain singleton chains, they have been excluded from this error category. Additionally, chains that have only one of their elements predicted are tagged as missing chains. Table 4.9 shows the breakdown of the missing chains by the length of the individual chains, respectively. Chains of length between 2 and 3 dominate this error sub-category. Most of these chains consist of an email address and the corresponding name of the entity referred only in the header of one email message in the email

Table 4.9. Breakdown of missing chains by the length of the chain.

Length	2-3	4-5	6-10	10+
Count	172	27	17	6

thread. Few examples of these types of chains are *['dutch.quigley@enron.com', 'Quigley, Dutch'], ['ed.mcmichael@enron.com', 'McMichael Jr., Ed']*. The results of SeedSpanBERT do not imply that the entire chain is present as expected in the predictions but instead implicate that the elements of a previously missing chain are present either in a single chain or as parts of another chain.

Incorrect and irrelevant mention spans

1. Incorrectly identified mention spans

   These types of errors consist of predicted mention spans whose width differs from the expected mention spans. Email headers consist of the full names of the sender as well as recipients. However, the entire span of these names is not predicted on multiple occasions (71%). Example 17 consists of a sample prediction where *Barkowsky, Gloria G.* was the expected mention prediction, but the system returned *Gloria G.*.

   Example 17. Example showing partial and duplicate name mention predictions

   Date: Mon, 17 Dec 2001 14:28:03 -0800 (PST)

   From: g..barkowsky@enron.com

   To: theresa.staab@enron.com

   Subject: RE: Final Statements and Invoices for November

   X-From: Barkowsky, ***Gloria G.***

   X-To: ***Staab, Theresa***

   X-cc:

X-bcc:

yes, I'll do this. Do you have anything for Crestone and Lost Creek?

2. Irrelevant mention spans

   The SpanBERT model used to obtain predictions is fine-tuned on the CoNLL 2012 shared task consisting of additional entity types. This results in additional or irrelevant mentions being predicted by the model that are considered errors. One of the contributing factors to the increase in precision of SeedSpanBERT was the 93% reduction seen in these spans. Fine-tuning the model on the seed corpus helped exclude the learning of the other entity types present in the OntoNotes 5.0 corpus, thereby not predicting spans representing those types.

3. Duplicate name mention spans

   When a sub-span of a name mention span is predicted as part of another chain or the same chain, the sub-span is considered a duplicate one as the full name is considered the entity representative span. Example 17 shows the scenario where the expected mention is just *Staab, Theresa*, but *Theresa* is also predicted as a mention span.

4.5 Corpus for Entity Resolution in Email Conversations (CEREC)

4.5.1 Annotation

The annotation procedure is divided into two parts: mention annotation and coreference annotation, and for both parts, SBERT is used. `SEED` is used as the starting point[6].

[6]Only 43 email threads out of the 46 have been used in this work as three email threads were discarded due to their overlap with the other email threads in `SEED`

Table 4.10. Statistics for changes done during manual correction of predictions obtained on 143 email threads.

Statistic	**Value**
Added Mentions	2,106
Corrected Mentions	344
Deleted Mentions	325
No-change/Predicted Mentions	12,056
Total Mentions	13,837
Precision	0.93
Recall	0.86
F1-score	0.89

Mention Annotation

Given an email thread, correctly identifying spans of text which refer to an entity is the task of mention identification. Here, the mention identification task is framed as identifying a single coreference chain that consists of all spans of text referring to a valid entity. A valid entity is an entity of the type PERSON, ORGANIZATION, LOCATION, or DIGITAL. Consider Example 15, here the single coreference chain will be *["g..barkowsky@enron.com", "theresa.staab@enron.com", "Barkowsky, Gloria G.", "Staab, Theresa", "I", "you", "Crestone and Lost Creek"]*. Framing the task in this manner helps speed up the annotation process as it eliminates the need to perform architectural changes and carry out experiments to test each change.

First, an SBERT model is trained on `SEED` for the mention identification task. Next, this trained model is used to obtain predictions on the unlabelled corpus. From these predictions, approximately 2% (143 email threads) are manually corrected. A training set of 94 email threads and a validation set of 49 email threads is created[7]. Table 4.10 shows the count of the type of changes done during the manual correction of these 143 email threads and

[7]Note that these 143 email threads are selected from the 171 email threads used for performing feature annotations(See 3.4.3)

Table 4.11. Results of two models trained on 94 gold annotated and 6,001 weakly annotated documents respectively.

Model	**P**	**R**	**F1**
M-SBERT$_{94}$	94	82	87.58
M-SBERT$_{6001}$	95	80.8	87.37

the corresponding precision, recall, and F1-score of the trained model. The remaining 6,001 email threads will be referred to as mention annotated corpus (MAC). The motivation to create a training and validation set is to compare models trained on gold annotated (94 email threads) and weakly annotated (MAC) training sets. These models will be referred to as M-SBERT$_{94}$ and M-SBERT$_{6001}$ respectively. Table 4.11 reports the results of these two models on `SEED`. From the results, two inferences can be drawn:

1. The model M-SBERT$_{6001}$ performs equally well than its counterpart trained on a gold annotated corpus. Weak annotations, by definition, are either incomplete or contain incorrect annotations. However, based on the correction evaluation statistics (Table 4.10) and experiment results, an assumption that they are gold mention annotations for obtaining weak coreference annotation can be made.

2. The performance of the model M-SBERT$_{6001}$ illustrates the robustness of the model to ignore the noise in the weakly annotated corpus.

Finally, both `SEED` and the training set containing 94 email threads are used to train an SBERT to obtain mention annotations on 6001 email threads, thereby improving the quality of mention annotations.

Coreference Annotation

The next step after completing mention annotation is to perform entity coreference annotation. For this task, an approach similar to the one undertaken for obtaining mentions

Table 4.12. CEREC statistics.

Statistic	Value
Number of email threads	6001
Number of email messages	36,448
Number of words	6,569,227
Coreference Chains	60,383
Annotated Mentions	445,762
Annotated Pronouns	145,615
Length of longest coreference chain	388
Average Length of coreference chains	7.3822

annotations is used. First, a gold validation set is created to assist in evaluating the training performance. A set of 34 email threads is selected from the validation set used for mention annotation. Two annotators performed annotation only on the previously gold-annotated mentions. Second, an SBERT model is trained on the coreference annotations of SEED to obtain annotations on the MAC. Mention annotations from MAC are provided as input during the coreference annotation process. The final annotated corpus will be referred to as CEREC. Table 4.12 provides different corpus statistics. Although the corpus contains a large number of mention annotations, 47,013 mentions added during the mention annotation process have not been annotated by the model in this step.

4.6 Corpus Analysis

4.6.1 Baselines

Header baseline1 (Hb1): A simple baseline of resolving pronouns based on the participants in the email header is constructed. All first person singular pronouns ("I", "me", "my", "mine", "myself") are chained to the sender, and second-person pronouns ("you", "your", "yours", "yourself", "yourselves") to the recipients respectively. First-person plural pronouns ("we", "us," "our," "ours," "ourselves") are linked to both the sender and the

recipients of the email message. In addition, all non-pronomial mentions that are the same or have overlapping words are chained together. This baseline is rule-based and does not consider the surrounding context.

Header baseline2 (Hb2): This is similar to Header baseline1 except for how first-person plural pronouns are resolved. In this baseline, all first-person plural pronouns in an email message are chained together into one coreference chain and not to the sender or recipients of that message. Furthermore, each first-person plural pronoun chain in an email message is merged with the corresponding chains in every other message of that email thread.

c2f-coref (C2F): The `c2f-coref` model described in 2.2.3 is used for this baseline.

SBERT: The SBERT model described in 2.2.3 is used for this baseline. The SBERT model trained on `CEREC` is referred to as **CerecSBERT**.

4.6.2 Experiments

The training set for these experiments is `CEREC` containing 6001 email threads, and the validation set contains 34 email threads, the one used for coreference annotation. The `SEED` containing 43 email threads is used as the test set. Mention detection and coreference resolution are the two tasks evaluated in these experiments. The following three experiments are carried out:

- Exp1: Use the Hb1 and Hb2 baselines for evaluating coreference resolution given mention annotations as input. These baselines also use section information (SI) to identify mentions present in an email header.

- Exp2: Use the C2F and CerecSBERT baselines to evaluate both mention detection and coreference resolution tasks. Compared to the CerecSBERT baseline, the C2F baseline does not enforce a maximum sentence length restriction and has a higher hyperparameter value for maximum training sentences.

The *genre* feature is removed for both C2F and CerecSBERT baselines since it does not apply to this corpus. For the C2F baseline, the hyperparameters *max_span_width*, *max_training_sentences* and epochs are set to 20, 30 and 10 respectively. This is done to make training tractable on the environment. For the CerecSBERT baseline, the `spanbert_base` model is used with a maximum segment length of 256, and training is carried out on an NVIDIA GeForce GTX 1080 Ti GPU with 8 12GB cores. This work follows the standard experimental setup used in the CoNLL 2012 Shared task. A preliminary evaluation is done using the metrics used in 4.4.3.

4.6.3 Results

Tables 4.13 and 4.14 show results of Exp1 and Exp2 for all baselines and all metrics. First, it can be seen that how first-person plural pronouns are resolved in the header baselines does not significantly impact the average F1 score. Second, the average F1 score of CerecSBERT is 0.28 F1 points higher than the C2F baseline. This shows that increasing the maximum sentence length and maximum training sentences do not help C2F in outperforming CerecSBERT. Both models perform equally well. Compared to the results reported in 4.4.3, the CerecSBERT shows an improvement of 5.25 F1 points. Finally, the large difference in F1 scores of the Exp1 and Exp2 baselines is because Exp1 baselines use mention annotations and the SI feature.

4.6.4 Error Analysis

This section presents error analysis performed on the predictions obtained by the baselines on a subset of 15 email threads selected randomly from `SEED`. The selected 15 email threads contain a total of 282 coreference chains with 1261 mentions. Human evaluation is performed to gain an in-depth understanding of the errors. Errors are broadly divided into four categories. These are similar to the ones used in 4.4.4. Table 4.15 shows the distribution of errors into these categories for each of the baselines.

Table 4.13. Evaluation results on SEED. Avg. F1 score is computed using MUC, B^3 and CEAFE metrics.

Model	MUC			B^3			CEAFE			Avg.
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**F1**
Hb1	90.2	75.1	**81.9**	82	65.3	72.7	61.6	74.4	**67.4**	74
Hb2	91.3	74	81.8	87.1	64.2	**74**	59.2	76.6	66.8	**74.2**
C2F	86.8	64.3	73.9	72.5	46.3	**56.6**	67.2	35.4	46.3	58.9
CerecSBERT	87.2	64.3	**74**	76.1	46.7	57.9	63	35.9	**45.7**	**59.2**

Table 4.14. Evaluation results on SEED using the LEA metric.

Model	P	R	F1
Hb1	71.1	62	66.3
Hb2	75.7	60.3	67.1
C2F	69.4	45.5	55
CerecSBERT	73.9	44.6	**55.6**

Missing mentions in the chain

Hb1 and Hb2 baselines use mention annotations as input to perform coreference chaining. Hence, only the deep learning baselines are considered for this error category. Missing mentions are further divided into three types to understand the limitations of the baselines.

1. Missing pronoun mentions: This error type contributes to 8% of all missing mentions.

2. Missing mentions in email header: A missing email address or name of a participant in the email message present in the email header is considered in this type. This error type contributes to 12% of all missing mentions.

3. Other missing mentions: All missing non-pronomial mentions present in the email body are considered in this error type. For C2F and CerecSBERT, the distribution range of these missing mentions for entity types is PER - 40%, ORG - 24%, LOC - 8-9%, and DIG - 26-28%.

Missing chains

In this error category, coreference chains present in the gold annotations but absent in the predictions are considered. Since Hb1 and Hb2 use mention annotations as input, counts for this error category for these baselines are not reported. The models C2F and CerecSBERT in the original work (Lee et al., 2018; Joshi et al., 2019) were trained on CoNLL 2012 shared task corpus, which did not contain any singletons. Both C2F and CerecSBERT baselines report similar numbers for this error category. About 68-85% of chains in this error category are of lengths 1 or 2.

Incorrectly chained mentions

All mentions in a coreference chain are considered to refer to the same entity. A mention or reference in a predicted coreference chain that does not refer to the same entity is considered to be incorrectly chained. These mentions are further broken down into pronoun mentions and other mentions. All baselines except C2F report a close count for pronoun mentions. CerecSBERT owing to its higher context capturing capabilities does a better job at resolving pronomial mentions than C2F. For other mentions, C2F and CerecSBERT baselines report approximately 1.5 times the counts reported by Hb1 and Hb2. This highlights the effectiveness of rule-based approaches and the possible benefits of having a hybrid approach.

Decomposed chains

Counts are reported for both the number of original chains and the number of chains that are created. It is evident by the high number of decomposed chains for C2F and CerecSBERT baselines that deep learning models do a worse job of linking chains across email messages and handling paraphrasing.

Table 4.15. Error statistics of baselines for different error categories.

Error Category	Hb1	Hb2	C2F	CerecSBERT
Missing mentions in the chain				
Missing pronoun mentions	-	-	102	101
Missing mentions in email header	-	-	155	160
Other missing mentions	-	-	224	197
Missing chains	-	-	131	116
Incorrectly chained mentions				
Pronouns	60	82	139	98
Other	150	137	195	201
Decomposed chains				
Number of chains decomposed	50	46	42	63
Number of new chains	134	115	108	156

4.6.5 Ablation Study

Training using additional features like speaker information and genre indicators on top of coreference annotations has proved to be helpful in the past. On the same lines, an ablation study is performed for three features specific to conversational texts, which have a thread-like structure. The feature-annotated corpus created in Section 3.4.3 used for this study. SBERT is used as the evaluation model, and `SEED` with 43 email threads is used as the training set. For validation and testing, 14 and 20 email threads are used, respectively. This extended dataset is referred to as `ExSEED`. Table 4.16 reports results of experiments with permutations of all features using the CoNLL average F1 metric (described in 4.4.3). An embedding size of 20 is chosen to encode EI and SI for all feature addition experiments.

Table 4.16 shows that the addition of SI improves the performance of the model in all scenarios. SI provides information which is useful in identification mentions used for pronoun resolution. All mentions in *To* or *Cc*, or the mention in *From* are used to resolve pronouns like *I*, *you*, *me*, *us*, etc[8].

[8]This excludes the cases when the sender or an alias of the sender is one of the recipients of the email

Table 4.16. SBERT evaluation results for all permutations of additional features.

Feature	**Avg. F1 (conll)**
SBERT	55.57
+ MI	54.40
+ SI	**56.53**
+ REV	53.94
+ REV + MI	52.15
+ REV + SI	54.18
+ MI + SI	55.29
+ REV + MI + SI	52.94

Reversing the email thread (REV) in temporal order reduces the average F1. This disproves the hypothesis presented before. However, it is essential to note that the test size for these experiments consisted of only 20 email threads. Finally, the addition of MI does not help the model. MI provides the model with message boundary information that can be used to merge inter email message clusters but fails to impact the current setting positively.

4.6.6 Mention Scoring and Singleton Problem

The error analysis presented before and the work of Timmapathini et al. (2021) highlight the limitation of the SBERT model's mention extraction component on `CEREC` and SciERC (Luan et al., 2018) corpora, respectively. Table 4.15 highlighted that in the obtained predictions, about 28% of the coreference chains present in the analyzed email threads were missing, and about 13% of the mentions were missing. The missing chains and mentions directly resulted in low recall scores of B^3, CEAFE, and LEA metrics. The analysis also highlighted the limitation of the SBERT model's mention extraction component for a domain-specific corpus. An email message from a larger email thread highlighting this limitation is shown in Example 18. The mentions highlighted in green received higher span scores than the mentions

Table 4.17. Span scores for gold mentions highlighted in Example 18.

Mention	SBERT
Germany, Chris	0.2
McMichael Jr., Ed	0.42
Zisman, Stuart	0.3
Concannon, Ruth	0.1
Miller, Don	0.1
Asset Mktg	0.04
Germany, Chris	0.1
Bridgeline	0.07
Anita Patton	0.1
Bridgeline	0.2
Rita Wynne	0.4
ENA	0.2

highlighted in red. Additionally, only the mentions highlighted in green are present in the predictions. Table 4.17 shows individual scores for each of these mentions.

Example 18. Example from CEREC shows an email message with mentions highlighted span score and presence in the predictions.

From: Germany, Chris

Sent: Tuesday, April 23, 2002 3:03 PM

To: McMichael Jr., Ed; Zisman, Stuart

Cc: Concannon, Ruth; Miller, Don (Asset Mktg); Germany, Chris

Subject: RE: Bridgeline Storage & Transport

Per Anita Patton at Bridgeline - the balance is 1,930,552 dth. Per Rita Wynne - ENA is showing a balance of 1,986,972 dth.

4.7 Proposed Solution

4.7.1 Model design

In 4.5.1, it can be seen that the SBERT model achieves high scores for the mention/span identification task. Bamman et al. (2020) in their work also report the same observation. With this as motivation, we hypothesize that improving the performance of the SBERT model for scoring of spans can help the model better adapt to a new domain. This hypothesis is tested with a joint learning approach using two tasks - Coreference Resolution and Span Classification. For the coreference resolution task, the task definition provided in 2.2.3 is used. The remainder of this section provides a detailed description of the span classification and the joint learning models. Figure 4.2 shows a pictorial representation of the model.

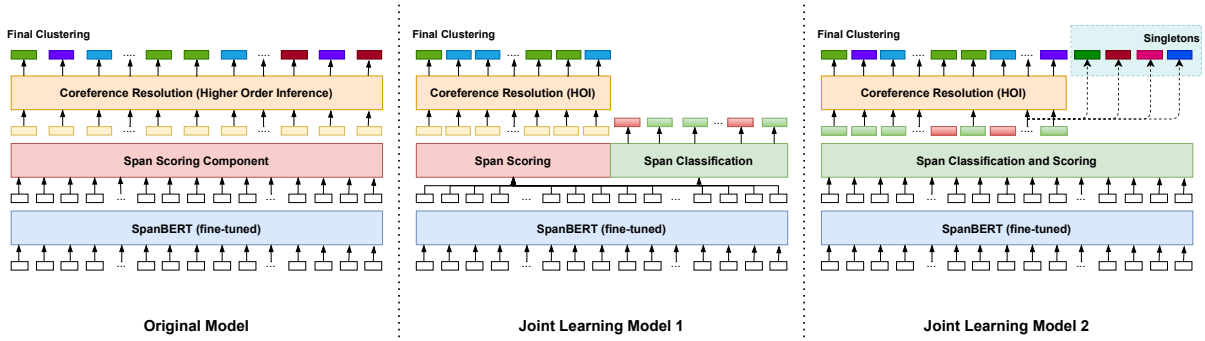

Figure 4.2. Architectures of the original and proposed joint learning models. The third model also provides a visual representation of the proposed post-processing for incorporating singletons.

Span Classification

Given a document D, for each span x, the span classification task is defined as identifying if x is a mention or not. This is computed using a two-layer feed-forward neural network given by the following equation:

$$s_x = \texttt{softmax}(\texttt{FFNN}_{sc}(g_x)) \tag{4.1}$$

Here, s_x is a softmax distribution over two classes: `MENTION` and `NOT-A-MENTION`.

Joint Learning

We propose two models that jointly learn the span classification and coreference resolution tasks. The loss function of these models is given in equation 4.2. L_{cr} is the loss of the model for the coreference resolution task, and L_{sc} is the loss of the model for the span classification task.

$$L = L_{cr} + L_{sc} \tag{4.2}$$

Joint Learning Model 1 (JM1)

In this joint learning model, the span classification task is added as an additional module/component parallel to the SBERT architecture. The second model in Figure 4.2 shows the architecture of a **JM1** model.

Joint Learning Model 2 (JM2)

The SBERT model uses Equation 2.4 to obtain mention scores and then selects the top k spans to obtain the scores using Equation 2.3. To fully observe the impact of using span classification for improving mention scoring, the existing span scoring component of SBERT is replaced with the span classification component. In other words, given a set of candidate spans, the spans are first classified and then using s_x on the spans classified as mentions, pairwise scores are obtained using Equation 2.3. The JM2 model, compared to JM1, has a lesser number of trainable parameters but has approximately the same number of parameters as SBERT[9]. The last model in Figure 4.2 shows the architecture of a **JM2** model.

[9]The difference between the two models is the span scoring function. In SBERT, a single score is computed for each candidate span, whereas JM2 classifies each span into two classes.

4.7.2 Post-processing: Singleton Addition

The work of (Chen et al., 2018) and the evaluation results discussed in 4.4.3 show how addition of singletons to the dataset degrades the performance of the SBERT model or models using the architecture proposed by Lee et al. (2017). Since the proposed models already learn span classification, a simple post-processing step is introduced to incorporate singletons. Let C_P be the coreference chaining obtaining as the output of the coreference resolution task. Let C_P^m be the set of mentions in C_P. Let SC_M be the set of spans classified as mentions by the span classification task. The following steps show how singletons are added to C_P:

1. Compute $Sg = SC_M - C_P^m$.
2. Update C_P using $C_P = \{C_P \cup (m) | m \in Sg\}$. Here (m) is a singleton chain containing a mention $m \in Sg$.

Singleton addition post-processing is carried out on top of the JM1 and JM2 models, and the corresponding models are referred to as **JM1+S** and **JM2+S** respectively. The third model in Figure 4.2 gives a pictographic representation of JM2+S.

4.8 Experiments

4.8.1 Datasets

In order to test the performance of the joint models, experiments were carried out on `CEREC` (**CD**) and `ExSEED` (**ED**). However, since the mention scoring problem has also been observed on other domain-specific corpora, the LitBank dataset (Bamman et al., 2020) is included as part of the experimentation setup. Furthermore, as the SBERT model was originally trained and tested on the OntoNotes 5.0 dataset, is it also incorporated to evaluate the impact of the joint learning models.

Table 4.18. Description of how the data is distributed in each dataset and if the dataset contains singleton annotations.

Dataset	Train	Dev	Test	Singletons
CD	6001	34	43	Yes
ED	43	14	20	Yes
LD	80	10	10	Yes
OD	2802	343	348	No

1. LitBank (Bamman et al., 2020) containing containing prefixes of 100 different works of English-language fiction (**LD**).

2. OntoNotes 5.0 English dataset[10] from the CoNLL 2012 shared task (**OD**).

Some statistics on the data are included in Table 4.18.

4.8.2 Baselines

SBERT

The SBERT model[11] described in 2.2.3 is used for this baseline. For `CEREC` and `ExSEED`, the scores reported in 4.6.3 and 4.6.5 are used for the SBERT baseline.

U-MEM

Toshniwal et al. (2020) in their work on long document coreference resolution, evaluate bounding memory in terms of the number of entities in the memory for a linear runtime in the length of the document. Their memory-augmented neural network with unbounded

[10] `https://catalog.ldc.upenn.edu/LDC2013T19`

[11] `https://github.com/mandarjoshi90/coref`

memory **(U-MEM)** achieves the best results[12] on the LitBank corpus. In this work, the U-MEM model is not retrained, and only the scores reported in the paper are used for comparison.

CorefQA

Framing the coreference resolution task as a question answering task, Wu et al. (2020) propose the **CorefQA** model that achieves state of the art results on the OntoNotes and GAP (Webster et al., 2018) datasets. Similar to **U-MEM**, only the scores reported in the paper[13] are used for comparison.

4.8.3 Evaluation Setting and Metrics

The experiments evaluating the performance of the proposed models and the baselines described before are carried out in two phases. In `PHASE I`, `ExSEED` is used to evaluate the performance of SBERT, JM1, and JM2 models. These experiments are used to choose between JM1 and JM2 for `PHASE II` experiments. `PHASE II` experiments evaluate all baselines described above and the proposed models on the datasets described above.

All datasets are converted into the CoNLL 2012 format. The F1 scores for MUC, B3, and CEAF metrics are reported using the CoNLL-2012 official scripts[14]. The average F1 score is the unweighted average of the individual metric F1 scores. For `ExSEED`, `CEREC`, and OntoNotes, the best scores of 5, 3, and 3 independent runs of the model are reported, re-

[12]Although the current state-of-the-art results on the LitBank are reported by Xia and Van Durme (2021), this work uses U-MEM as Toshniwal et al. (2020) report individual metric scores and the difference between the two results is only 0.2 F1 points.

[13]Since this work uses SpanBERT-base for all its experiments, only CorefQA scores for SpanBERT-base are reported for fairness.

[14]`https://github.com/conll/reference-coreference-scorers`

Table 4.19. Additional training statistics for `PHASE II` experiments. All SBERT models were trained on GPU and JM2 models on CPU.

	Model	**Avg. Epochs**	**Avg. Training Time**
ED	JM2	28	56 mins
CD	JM2	8	27 hours 36 mins
OD	SBERT	5	5 hours 10 mins
	JM2	9	16 hours 21 mins
LD	SBERT	16	24 mins
	JM2	14	77 mins

spectively[15]. However, for LitBank, the average scores across 10-fold cross-validation results are reported.

The experiment models were implemented using Tensorflow. For all datasets, `spanbert-base` is used as the encoder for the SBERT baseline and all joint learning models. For each experiment, the best-performing model on the development set for the coreference resolution task is selected and evaluated on the test set. The training process was stopped using early stopping over the development set accuracy (coreference resolution task) for five epochs.

All hyperparameter values unless specified here are the same as specified by the original works. The maximum segment length was 256. The activation function of all feed-forward neural networks for the coreference resolution task was *elu*. A two-layer feed-forward neural network is used with a hidden layer of 3000 dimensions and linear activation for the span classification task. All SBERT model experiments were carried out on an NVIDIA GeForce GTX 1080 Ti GPU with 8 12GB cores. All joint model experiments were carried out on a 528GB RAM CPU environment. Table 4.19 shows additional training statistics for all datasets and models.

[15]The number of runs varies owing to the size of each dataset.

Table 4.20. PHASE I evaluation results on the ExSEED dataset. The results are reported without and with the singleton post-processing component.

Avg. F1.	SBERT	JM1	JM2
Normal	55.57	55.49	**56.95**
+Singleton	-	60.02	**62.21**

4.9 Results

PHASE I experiment results are given in Table 4.20. Compared to the SBERT baseline, JM1 shows a decrease of -0.08, and JM2 shows an increase of +1.48 in the average CoNLL F1. The addition of singleton chains results in an increase of +4.53 and +5.26 F1 in the scores of JM1 and JM2 models, respectively. Finally, the +1.46 increase in F1 of JM2 compared to JM1 and the absence of the additional span classification layers resulted in JM2 being the model of choice for PHASE II experiments.

PHASE II experiments are carried out on all four datasets. Table 4.21 shows results of PHASE II experiments for all models and from these results, some key observations are drawn. First, the results on ExSEED, LitBank, and OntoNotes show that the JM2 model outperforms the SBERT model by +1.38, +0.22, and +2.79 F1, respectively. Second, adding singletons to JM2 increases the F1 scores on the ExSEED, CEREC, and LitBank by +5.26, +4.87, and +14.91, respectively. The significant increase in the LitBank F1 score demonstrates the poor job SBERT does on singletons. The scores using JM2+S achieve new state-of-the-art results on the ExSEED and CEREC datasets. Lastly, the addition of singletons decreases the average F1 for OntoNotes as the dataset does not contain any singletons.

4.10 Error analysis

This section analyzes the predictions obtained by the SBERT, JM2, and JM2+S models on the test set of all four datasets. In line with the error analysis presented in 4.6.4, the

Table 4.21. PHASE II experiment results for all models on all four datasets. Appendix A contains results for each run and fold.

		MUC			B^3			CEAFE			Avg.
	Model	P	R	F1	P	R	F1	P	R	F1	**F1**
ED	SBERT	83.3	62.1	71.2	69.7	44.3	54.2	58.2	31.9	41.2	55.57
	JM2	81.2	64.5	**71.9**	64.8	49.6	56.2	64.6	31.8	42.7	56.95
	JM2+S	81.2	64.5	**71.9**	66.7	54.3	**59.8**	54.5	55.1	**54.8**	**62.21**
CD	SBERT	87.2	64.3	**74.06**	76.1	46.7	57.9	63.04	35.9	45.7	59.25
	JM2	84.2	64.9	73.3	65.9	48.03	55.5	67.7	32.3	43.7	57.56
	JM2+S	86.1	63.8	73.3	70.7	50	**58.6**	55.5	55.06	**55.3**	**62.43**
LD	U-MEM	90.8	85.7	**88.2**	80	72.1	**75.9**	65.1	66	**65.5**	**76.5**
	SBERT	89.5	86.4	87.9	72.8	58.6	64.8	64.2	17.3	27.2	59.99
	JM2	89.1	86.5	87.8	72.3	58.2	64.4	65.5	18.1	28.3	60.21
	JM2+S	89.1	86.5	87.8	72.9	72.4	72.6	61.6	69.3	64.8	75.12
OD	CorefQA	85.2	87.4	**86.3**	78.7	76.5	**77.6**	76	75.6	**75.8**	**79.9**
	SBERT	83.2	78.08	80.5	73.2	67.3	70.1	71.5	60.1	65.3	72.03
	JM2	84.1	79.7	81.9	76.2	70.4	73.2	73.7	65.4	69.3	74.82
	JM2+S	84.1	79.7	81.9	72.6	71.5	72.1	55.4	70.6	62.1	72.05

errors are divided into four categories - the first two categories showcase the model's ability concerning mention extraction. In contrast, the last two categories evaluate how well the model performs mention linking. Although richer span embedding representations should assist the model in both mention extraction and linking, this work considers the impact of improving span representations should be more significant for the last two categories. For LitBank, the average counts over ten folds are reported.

4.10.1 Missing mentions in the chain

A *missing mention* is a mention which is present in a gold coreference chain but absent in all predicted coreference chains. Using JM2 results in a slight improvement for this category for all datasets except CEREC. For JM2+S, however, reductions of 37%(ExSEED), 17%(CEREC), 70%(LitBank), and 37%(OntoNotes 5.0) are observed. A considerable decrease

Table 4.22. Error analysis statistics on the output of SBERT, JM2 and JM2+S models for PHASE II experiments.

	Error Category	SBERT	JM2	JM2+S
ED	Missing mentions in the chain	777	748	**490**
	Missing chains	207	203	**60**
	Decomposed chains	**67**/193	73/**187**	73/**187**
	Incorrectly chained mentions	**534**	548	731
CD	Missing mentions in the chain	1187	1,243	**988**
	Missing chains	305	334	**166**
	Decomposed chains	142/350	144/379	**131/329**
	Incorrectly chained mentions	**900**	905	1,144
LD	Missing mentions in the chain	768.1	735.8	**226.3**
	Missing chains	543.2	526.6	**120.4**
	Decomposed chains	**81.7**/832.5	88.1/826.2	88.1/**396.4**
	Incorrectly chained mentions	453	**448.2**	477.9
OD	Missing mentions in the chain	4,363	3,382	**2,765**
	Missing chains	1,140	797	**469**
	Decomposed chains	490/1,092	**448/1,003**	**448/1,003**
	Incorrectly chained mentions	3,224	**2,776**	3,168

in the number of missing mentions usually leads to an increase in recall scores, as seen in Table 4.21, especially for the CEAFE metric.

4.10.2 Missing chains

Chains that are present in gold coreference chains but absent in predicted chains belong to this category. It is important to note that a chain is only considered missing when none of its mentions are present in the predicted chains. Like the previous error category, the JM2 model slightly reduces the counts reported by the SBERT baseline for all datasets except CEREC. The JM2+S model exhibits a decrease of 71%, 46%, 78%, and 59% in the number of missing chains for ExSEED, CEREC, LitBank, and OntoNotes, respectively.

4.10.3 Decomposed chains

For this error category, both the number of chains decomposed and the number of decomposed parts are computed. The reported statistics show that using JM2+S improves the model's ability to link chains, reducing the number of decomposed parts for all datasets. No significant increase or decrease was observed in the number of decomposed chains.

4.10.4 Incorrectly chained mentions

A mention referring to entity E in a predicted chain is an incorrectly chained mention if it is part of a chain in which the majority of the mentions refer to some other entity E'. For LitBank and OntoNotes, a slight reduction in incorrectly chained mentions is observed when JM2 is used. For `ExSEED` and `CEREC`, higher counts for this category paired with lower counts for missing mentions and missing chains show that although the model correctly identified mentions, it failed to chain them correctly.

4.11 Discussion

This chapter presented two joint learning models and extensively tested the JM2 and JM2+S models on four datasets. The previous section highlighted the impact of the joint learning approach using error statistics. With other statistics and examples, this section demonstrates the effect of jointly learning the two tasks and identifies avenues for further improvement. For LitBank, the statistics for the best fold are used in this section for analysis.

The spans are first divided into three categories:

1. *Non-gold span (NG)*: A candidate span that is not a gold (annotated) mention.
2. *Gold span (G)*: A candidate span that is a gold mention.
3. *Predicted span (P)*: A candidate span that was present in the predicted coreference chains. In simple terms, the model predicts it to be a mention.

It is important to note that the span categories are defined for the coreference resolution task. Let the sets containing Non-gold, Gold, and Predicted spans be S_{NG}, S_G, and S_P, respectively. Ideally, a coreference resolution system, with $s_x(x)$ as the mention scoring function, should assign low scores to a span in S_{NG} and high scores to spans in S_P. Additionally, the system should also have $S_G \cup S_P = S_G$. Finally, the system should be able to chain identified mentions correctly.

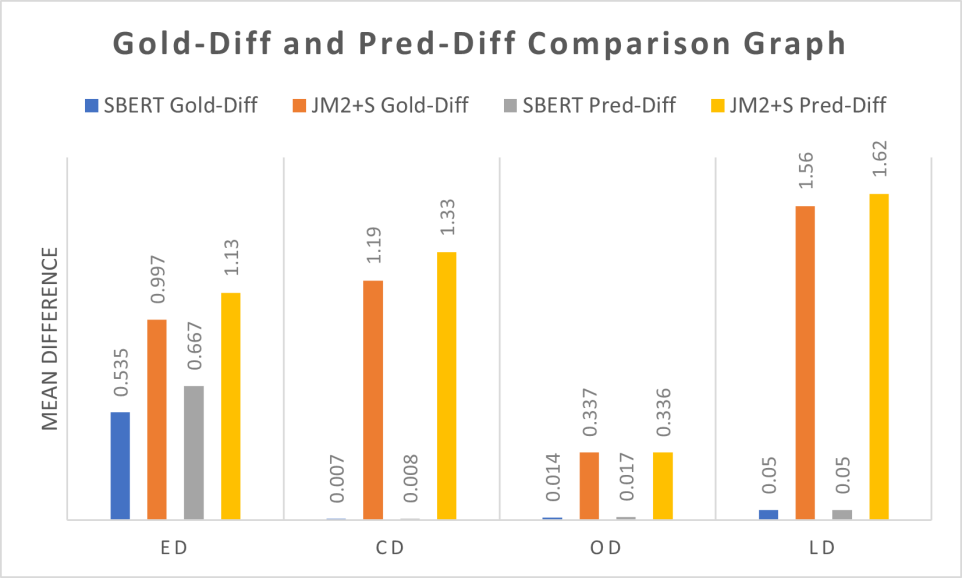

Figure 4.3. Graph comparing SBERT and JM2+S using $M_G - M_{NG}$ (`Gold-Diff`), and $M_P - M_{NG}$ (`Pred-Diff`) for all four datasets.

Let M_{NG} be the mean span score[16] of the spans in S_{NG}. Similarly, M_G and M_P are computed. Figure 4.3 shows a graph which compares the values $M_G - M_{NG}$ and $M_P - M_{NG}$ obtained using SBERT and JM2+S models on all four datasets[17]. For all datasets, using JM2+S results in a minimum of ≈2x increase in the difference between the mean span scores. More statistics showing the impact of the JM2+S model are shown in Table 4.23 and their details are as follows:

[16]A softmax function is applied over the span scores obtained using the SBERT model for comparison as JM2+S model also applies a softmax over the positive class span scores.

[17]All displayed scores are multiplied by a factor of 10^2 for representation.

1. Added mentions: The statistic highlights how many correct mentions the span classification task added as singletons to the output of the coreference resolution task. Both the number of correctly added mentions and the total number of added mentions are provided. It can be seen that about 37% (lowest) to 83% (highest) of the added mentions are correct. The impact of this addition is dependent on the number of mention annotations in the dataset. `CEREC` and OntoNotes having more mention annotations see a smaller impact on the final F1 scores.

2. NG (non-gold) pronouns: The statistic counts how many pronouns[18] that were not annotated in the original dataset and were added as singletons to the output of the coreference resolution task. Since the test set in all datasets except OntoNotes contain singleton annotations, a high value is seen only for the OntoNotes dataset.

3. Added singletons: This statistic counts how many of the total singletons present in the dataset were correctly added by the span classification task output. The values show that the singleton post-processing correctly added 37% (lowest) to 72% (highest) of the total mentions. This not only highlights the limitation of the SBERT model but also the impact of the singleton post-processing. This statistic is not computed for OntoNotes as it does not contain singleton annotations.

4. Mention Extraction Avg. F1: This statistic reports the average conll F1 score on the mention extraction task. This statistic shows that the impact of the JM2+S model is more significant on smaller datasets, especially one with singleton annotations. Additionally, the improved performance on the mention extraction task corroborates the effectiveness of the proposed comprehensive model.

Lastly, this section demonstrates that the metric scores are not truly indicative of the impact of the joint learning models. Revisiting Example 18, the scores of the highlighted

[18]The considered pronouns are *I, you, we, us, me, she, her, he, him, they, them.*

Table 4.23. Additional statistics highlighting the impact of the JM2+S model on the test set of each dataset. The brackets contain the improvement by JM2+S over SBERT.

Statistic	**ED**	**CD**	**LD**	**OD**
Added mentions	258/ 312	312/ 380	628/ 806	617/ 1,763
NG pronouns	1	3	6	60
Added singletons	71/ 117	111/ 219	472/ 654	N/A
Avg. F1. Mention Extraction	88.7 (+6)	87.2 (+3.6)	90.8 (+7.8)	85 (+1.1)

Table 4.24. Span scores for gold mentions highlighted in Example 18. The scores in the SBERT and JM2+S columns are obtained using corresponding models.

Mention	**SBERT**	**JM2+S**
Germany, Chris	0.2	0.7
McMichael Jr., Ed	0.42	0.45
Zisman, Stuart	0.3	2.1
Concannon, Ruth	0.1	0.6
Miller, Don	0.1	0.3
Asset Mktg	0.04	5.5e-4
Germany, Chris	0.1	0.2
Bridgeline	0.07	1.5e-3
Anita Patton	0.1	0.4
Bridgeline	0.2	0.1
Rita Wynne	0.4	0.5
ENA	0.2	0.02

mentions are analyzed to understand the impact of jointly learning span classification with coreference resolution. In Table 4.24, the scores[19] highlighted in red are absent, and green are present in the corresponding model predictions. Emboldened mentions represent mentions that observe an increase in their respective span scores. The table shows that eight out of the twelve spans observed an increase in their score, and two out of the twelve spans that were not part of SBERT predictions were added by the JM2+S model.

[19]The scores are multiplied by a factor of 10^2 for representation.

Table 4.25 show sample sentences with highlighted spans. These spans were added as singletons in the post-processing phase of the JM2+S model. For each dataset, two types of examples are presented - Correct and Incorrect mention predictions. In examples for correct mention predictions, the added mentions are present in the gold annotations. On the contrary, the added mentions in incorrect mention predictions are absent in the gold annotations. The examples for incorrect mention predictions show that the added mentions, although incorrect, refer to a valid entity. The majority of the incorrect mentions can be grouped into two categories:

1. *Absence of annotation*: The dataset in consideration does not contain annotations for the chain type, i.e., singletons, the entity type that is being referenced or missed annotation.

2. *Text structure*: In `ExSEED` and `CEREC`, the mentions in email footers are not annotated, and thus if a mention in the footer is added, it is considered incorrect.

In conclusion, this chapter investigates the first component of the knowledge extraction pipeline - Entity coreference resolution. A small manually annotated corpus - `SEED` was created and used to show that entity coreference resolution in email conversations is not an unsolved problem. The experiments using `SEED` showed the limitations of the SBERT model and that the task deserves attention from the research community. Next, the chapter discussed the creation of `CEREC` - a large-scale corpus containing weak entity coreference annotations. Two joint learning models JM2 and JM2+S, were proposed to improve the performance on `CEREC`. The JM2+S model reported an increase of 4.87 F1 points over SBERT. The model also outperformed SBERT on three other datasets. For all experiments performed in this chapter, thorough error analysis was presented. In the next chapter, the relation extraction component of the pipeline is discussed.

Table 4.25. Examples drawn from the predictions of the JM2+S model on the test set of each dataset.

	Correct mention predictions	**Incorrect mention predictions**
ED	- To: Justin Boyd/LON/ECT@ECT, Janine Juggins/LON/ECT@ECT - The lady that I wanted to attend was out today. - If possible all assumptions to be agreed with the affected producers (PB, BG, Vintage and Chaco).	- I had a glance over the written paper from Sullivan & Cromwell as to FX Transactions - Visit our energy trading website at http://www.ubswenergy.com - My suggestion would be for the FX business be run out of a NON FCM firm in the states (NY) since it is a non-regulated product.
CD	- To: Tony DiGirolamo; Tim Brandi; Charlie Creswell; Bill Horvath; Jeff Bollinger; Heath Wenrich; Shawn Kulaga; Randy Sweigart - Yom Kippur was nice–I spent it with my family in KY. - I received a call form Bob Bradly who is representing Vern E Falkner, ...	- please contact the sender or reply to Enron Corp. at enron.messaging.administration@enron .com and ... - To: <douglass“.”dan@enron.com>, steven.harris@enron.com, ... - Here is the final version for distribution to the parties.
LD	- But then I am a timid man, and dislike violence; - The Director of Companies was our captain and our host. - And Mr. Lucian Gregory, the red-haired poet, was really (in some sense) a man worth listening to, even if one only laughed at the end of it.	- then lying at the docks waiting for the _Edinburgh Castle_ due in from England. - Hunters for gold or pursuers of fame, they all had gone out on that stream, ... - father and son lived up at grampa's in Tarrytown.
OD	- Regarding the Oct. 3 letter to the editor from Rep. Tom Lanthos, chairman of the House Subcommittee on Employment and Housing, alleging: - I never did write that story. - and frankly it doesn't work with the Chinese.	- She and her daughter Jordan often shuffle through these pictures of him taken before he left. - He is best known among his neighbors in Paris as a painter. - And Chris Hill our ambassador was in China a few days ago.

CHAPTER 5

RELATION EXTRACTION

5.1 Introduction

5.1.1 Background

Whittaker and Sidner (1996) explored the problem of *email-overload* regarding personal information management in emails. The authors highlight *threading* as one of the key requirements for asynchronous communication. *Threading* means grouping a set of email messages using some criterion. Ideally, the email messages grouped are part of the same conversation. Threading helps in regenerating the conversational context and the management of conversational history. The Zawinski algorithm (Zawinski, 1997) is one the most popular email message threading algorithms. It was used in Netscape Mail and News 2.0 and 3.0 and was later merged into RFC 5256 (Crispin and Murchison, 2008). The algorithm makes use of the *In-Reply-To* and *References* header fields to carry out threading. However, since the fields are optional for email clients, the algorithm cannot be used universally.

Lewis and Knowles (1997) argue that threading is a language processing task and thus requires sophisticated solutions. The authors used the email message's subject, quoted, and unquoted content to carry out threading. Wang et al. (2008) use the Zawinski algorithm to create a basic thread outline and fill in missing messages in the thread using the subject header. Erera and Carmel (2008) consider a more inclusive approach by using various email attributes such as message subject, participants, date of submission, and message content. Another task that is similar to threading is thread reconstruction. Cowan-Sharp (2009) use topic modeling for the thread reconstruction task. The authors use Latent Dirichlet Allocation to classify email messages to threads by detecting topic and topic change in the threads. Ailon et al. (2013) carry out *causality threading* for machine-generated emails by extracting causal relations between email messages. Dehghani et al. (2013) propose two

feature-driven supervised learning methods that learn to extract linear and tree structures of email conversations, respectively. The learning methods use email content, subject, date, and participants as features.

Besides the `reply-to` relation, few works have also focussed on extracting other relations from email conversations. Diehl et al. (2007) propose a supervised learning ranking model to identify manager-subordinate relationships from email threads in the Enron Email Corpus. Boufaden et al. (2005) extract `to` and `from` relations as part of their preprocessing tasks to create a privacy protection system for emails. Mahlawi and Sasi (2017) use regular expressions to extract `to`, `from`, `date`, `url` and `phone number` relations from a small subset of the Enron Email corpus. Dredze et al. (2006) create an attachment prediction system to reduce the volume of missing attachment mail. A corpus annotated with `attachment` relations is used to train the prediction system. The annotated corpus, however, is not available publicly.

As highlighted in Chapter 1, this dissertation considers the relation extraction task in a secondary capacity. The task is explored in an extraction setting compared to the work carried out for the entity coreference resolution task.

5.1.2 Dissertation Contribution

The main contribution of this chapter is the creation of the `ECRA` dataset. The dataset contains annotations for 11 relations for 762 email threads and is created using a mixture of manual and automatic methods. This is the first publicly available dataset containing relation annotations for email conversations in the Enron corpus.

5.2 Relations

Mathematically, a relation R holds between two sets if there exists a non-empty collection of ordered pairs containing one object from each set. If a relation R holds between $x \in X$ and

$y \in Y$, it can be read as xRy where X is the domain of R, and Y is the range of R. Before a description of different relations considered in this dissertation is provided, it is important to put forth assumptions required to understand these relations.

- An entity class is an abstract class that represents a collection of similar objects. It can also be interpreted as a template that every object in the collection follows. It is important to highlight the difference between an entity class and an entity type. This dissertation considers entity type to be a subset of the entity class set.

- Email Message is an entity class, and every email thread contains email messages which are unique instances of the Email Message class.

Relations present in an email thread are divided into two categories:

1. Relations present in an email body: For the text in the email body, there are two ways to look at the relation extraction problem: consider the email body text independently or as a text representing different entities and how those entities are related to each other. Aligning to the goal of capturing entity interactions, the second approach is considered and the work specifically focuses on the `COREFERENCE` and `ATTACHMENT` relations. The relation `xCOREFERENCEy` holds if both `x` and `y` refer to or are talking about the same real-world entity. This relation will be extracted implicitly via the Entity coreference resolution task (See Chapter 4).

2. Relations present in email metadata: The relations present in email metadata are independent of the content in email messages present in the email thread. These relations are further divided as follows:

   (a) Intra-email message relations: Relations that are present between an email message instance and its header fields belong to this sub-category. *From*, *To*, *Cc*, *Bcc*, *Date*, and *Subject* are the relations in this sub-category.

(b) Inter-email message relations: Relations between two email message instances belong to this sub-category. Currently, only the *Reply-to* relation is present in this sub-category as it links two different email messages in an email thread.

Table 5.1 provides a detailed description of the relations present in the email metadata and the `ATTACHMENT` relation. It is important to note that the cardinality property provided in Table 5.1 is the expected one and may or may not reflect in the actual corpus. The domain cardinality will always be validated, but the range cardinalities for all relations are the maximum values. For example, an email message may not have a `BODY-TEXT` or `FOOTER-TEXT` relation. Example 19 shows the extracted relations using an email message (see Example 2). The email message as a whole is represented using the object `EM0`. Thus, using the entities and relations defined in this chapter, this work extracts knowledge from email conversations. Next, this section discusses in detail the email corpus that is used from the extraction process.

Example 19. Sample extracted relations for Example 2.

```
EM0 - FROM - Davis, Dana
EM0 - FROM - Dana.Davis@ENRON.com
EM0 - TO - mfoster@grti.tec.ar.us
EM0 - SUBJECT - "Hello"
EM0 - DATE-TIME - Mon, 30 Jul 2001 10:40:17 -0500
EM0 - HEADER-TEXT - "Subject:    Hello
Date:         Mon, 30 Jul 2001 10:40:17 -0500
From:         \"Davis, Dana\" <Dana.Davis@ENRON.com>
To:           <mfoster@grti.tec.ar.us>"
EM0 - BODY-TEXT - "Good Morning Aunt Mae -
I got your email. It was a wonderful surprise to see.
....
```

Table 5.1. Detailed description of relations extracted (EM - Email Message).

Relation	**Domain (x)**	**Range (y)**	**Cardinality (n≥0)**	**Description**
FROM	EM	PER, ORG	1:2	An email message is FROM a Person or an Organization.
TO	EM	PER, ORG	1:n	The email message is sent TO a Person or an Organization. The recipient is a direct recipient or a primary intended recipient.
CC	EM	PER, ORG	1:n	A carbon copy (CC) email message is sent to a Person or an Organization. A cc is a copy of an email message whose recipient appears on the recipient list, so that all other recipients are aware of it.
BCC	EM	PER, ORG	1:n	A blind carbon copy (BCC) email message is sent to a Person or an Organization. A bcc is a copy of an email message sent to a recipient whose email address does not appear in the message.
SUBJECT	EM	String	1:1	The SUBJECT of an email message.
DATE-TIME	EM	Date-time	1:1	The DATE-TIME the email message was received.
HEADER-TEXT	EM	String	1:1	An email message has HEADER-TEXT as a string representing the entire header section.
BODY-TEXT	EM	String	1:1	An email message has BODY-TEXT as a string representing the entire body section.
FOOTER-TEXT	EM	String	1:1	An email message has FOOTER-TEXT as a string representing the entire footer section.

Table 5.1 continued

Relation	Domain (x)	Range (y)	Cardinality (n≥0)	Description
ATTACHMENT	EM	DIG	1:n	An email message has a file (DIGITAL entity type) as an ATTACHMENT. For the attached file, only the filename is considered and not the actual binary file.
REPLY-TO	EM	EM	1:1	An email message is a REPLY-TO another email message.

```
Anyway I be emailing you. Love ya.

Dana"

EM0 - FOOTER-TEXT -

"************************************************************************
This email is the property of Enron Corp. and/or its relevant

affiliate and may contain confidential and privileged material for the

sole use of the intended recipient (s). Any review, use, distribu...

************************************************************************"
```

5.3 Extraction Process

In this section, the process used to extract relations from email threads is described. Mahlawi and Sasi (2017) perform knowledge extraction on 500 email messages in the Enron Email Corpus. The authors extract sentiment and entities from the emails to construct a graphical representation of the email. The weighted graph is then used to extract a summary of the email. The authors use a set of regular expressions to extract information from an email message. Of the extracted information, the important fields for this research are 'to,' 'from,' 'cc,' and 'date' (date-time). The regular expressions used for each of these fields are

Table 5.2. Regular expressions used for relation extraction from email messages.

Relation Name	Regular Expression	Added
FROM	from:([a-z0-9_\.-]+@[\da-z\.-]+\.[a-z\.]2,6)	No
	forwarded by ([a-zA-Z-_,\s]+)	Yes
	from: ([a-zA-Z0-9-_\.,@\s]+[,;]*)	Yes
TO	to:([a-z0-9_\.-]+@[\da-z\.-]+\.[a-z\.]2,6)	No
CC	cc: ([a-z0-9_\.-]+@[\da-z\.-]+\.[a-z\.]2,6)	No
BCC	bcc: ([a-z0-9_\.-]+@[\da-z\.-]+\.[a-z\.]2,6)	Yes
DATE-TIME	date: (\w+), ([0-9]+) (\w+) ([0-9]+)	No
	sent: (\w+), ([0-9]+) (\w+) ([0-9]+)	Yes
	(\d2\s[/|-]\s\d2\s[/|-]\s\d2,4\s\d2\s:\s\d2)(\s[A|P]M)*	Yes
SUBJECT	subject : (.*)?	Yes
ID	message-id: .*?(\d+\s\.\s\d+).*?	Yes

shown in Table 5.2. It is important to note that these regular expressions are not sufficient in extracting all the information and coverage. Section 3.5 outlines the nuances in email message formats or structures and owing to these nuances the regular expressions given in Table 5.2 cannot be used directly. The table also shows what updates were made to the existing regular expressions. The last column in the table shows if the regular expression was added to the original list created by Mahlawi and Sasi (2017).

For the relations ATTACHMENT, HEADER-TEXT, BODY-TEXT, and FOOTER-TEXT the extraction process was carried out manually. The variations in the header formats and footer formats resulted in the extraction process for these four relations to be carried out manually. Thus, first the process of selection of email threads from CEREC is discussed. The email threads in CEREC are grouped by user, and then the users are divided into buckets based on the number of email threads a user has. Table 5.3 shows the distribution of 6001 CEREC threads with 131 users into 10 buckets. Each bucket is also assigned a weight using Equation 5.1. With the amount of manual effort required along with obtaining sufficient bucket coverage, the number of users n_u is randomly picked from [1, 10] and then randomly pick n_u users from the weighted buckets. The selected users and the corresponding number of email threads are -

Table 5.3. Distribution of users into buckets based on number of email threads.

Bucket Size (No. of Email Threads)	**User Count**	**Weight**
0-49	103	0.24
50-99	15	0.16
100-149	2	0.03
150-199	3	0.09
200-249	2	0.07
250-299	1	0.04
300-349	1	0.05
350-399	1	0.06
400-449	1	0.07
450-499	2	0.15

'beck-s' (115), 'dasovich-j' (472), 'haedicke-m' (50), 'lay-k' (19), 'sager-e' (90), and 'skilling-j' (16).

$$\texttt{Weight} = \frac{\texttt{Total number of email threads in the bucket}}{\texttt{6001}} \tag{5.1}$$

Post selection of the email threads, manual annotation of the 762 email threads or 4,525 email messages was carried out for `HEADER-TEXT`, `BODY-TEXT`, `FOOTER-TEXT`, and `ATTACHMENT` relations. A single annotator carried out the annotation process as the process did not involve resolving any ambiguity. Post the manual annotation process, the remaining relations are extracted using the regular expressions described earlier. However, the output of the extraction process is verified manually due to the variations in the email header formats. The statistics of the extracted relations are shown in Table 5.4. The final output of the extraction process is represented in Javascript Object Notation (JSON). The dataset containing the JSON representations of 762 email conversations is referred as `ECRA`. Appendix D shows an excerpt from an email thread and the corresponding JSON representation.

Table 5.4. Statistics of relations in ECRA.

Relation Name	Number of instances
FROM	6,135
TO	23,584
CC	9,056
BCC	1,724
SUBJECT	3,608
DATE-TIME	4,465
ATTACHMENT	412
HEADER-TEXT	4,519
BODY-TEXT	3,636
FOOTER-TEXT	127
REPLY-TO	3,763

5.4 Alternative methods

Before using the extraction process described in the previous section, two other automatic relation extraction methods were evaluated on the SEED corpus.

5.4.1 LUKE

The TACRED dataset (Zhang et al., 2017) is a large-scale supervised dataset for relation extraction. The dataset was constructed over six years (2009-2015) and contains TAC KBP relation annotations. Statistics for the TACRED dataset are provided in Table 5.5. TACRED consists of relations extracted between two entities, which aligns well with the type of relations in the Enron Email Corpus. Appendix E lists all the relations present in the TACRED dataset. It also provides details for each relation, like examples and whether the relation will be used in the extraction process.

Since this dissertation focuses primarily on entity extraction and coreference resolution, we explored using off-the-shelf systems for relation extraction. For TACRED, Yamada et al. (2020) propose the model LUKE that uses BERT as the base model and update the pre-

Table 5.5. TACRED Dataset Statistics

Statistic	**Value**
Number of Examples	106,264
Train	68,124
Dev	22,631
Test	15,509
Percentage of Negative Examples	70.5
Number of Relations	41
Average Sentence Length	36.1

training step by adding entities to the randomly masked word prediction task. They also propose an *entity-aware* self-attention mechanism that considers the type of tokens (normal or entities) when computing attention scores. This pre-trained BERT model obtains state-of-the-art performance in various entity-related tasks. This model reports an F1-score of 72.7, which, however, is not the best score. The model proposed by Cohen et al. (2020) reports an F1-score of 74.8. For this dissertation, the model proposed by Yamada et al. is selected since the code and pre-trained model is open-sourced. For each email message, the following process was used to extract relations using LUKE:

1. Extract all entity mentions using the JM2+S model.
2. Construct input for LUKE[1] using the extracted entity mentions in the email body.
3. Obtain predictions using LUKE and perform post-processing to keep only the required relations (see Appendix E).
4. Extract entity-relation triples from the predictions as output.

For obtaining the predictions, the `luke-large-finetuned-tacred` variant of the LUKE model was used. The experiments were carried out on an NVIDIA GeForce GTX 1080 Ti

[1] `https://github.com/studio-ousia/luke`

GPU with 8 12GB cores. However, the predictions obtained were extremely poor, and for most of the email threads, no relations or incorrect relations were extracted. Example 20 shows the predictions obtained on an email message. The text spans colored green are the entities extracted using the JM2+S model and given as input to LUKE. It can be seen that only the last one of the four predictions obtained is correct as *Steve* and the people comprising *we* are employees of *Enron* or *Transwestern Pipeline Company*. Due to the significant manual correction involved in using the output of LUKE, it was not used for the relation extraction.

Example 20. Example showing the predictions obtained using LUKE.

```
Input email message body:
Because of the scheduled Employee Meeting at the Hyatt, we will move Steve's
    meeting to 9:00a on Tuesday.
Please adjust your calendars accordingly.
adr
Audrey D. Robertson
Transwestern Pipeline Company
email address: audrey.robertson@enron.com
(713) 853-5849
(713) 646-2551 Fax


Predictions:
the Hyatt - per:employee_of - Steve
we - per:employee_of - the Hyatt
we - per:employee_of - Steve
Audrey D . Robertson - per:employee_of - Transwestern Pipeline Company
```

Table 5.6. DocRED Dataset Statistics

Statistic	**Value**
Number of Examples	63,443
Train	38,269
Dev	12,332
Test	12,842
Number of Relations	96
Average Sentence Length	24.87

5.4.2 JEREX

The DocRED dataset (Yao et al., 2019) is a large-scale dataset constructed from Wikipedia and Wikidata (Vrandečić and Krötzsch, 2014). The dataset contains named entity and relation annotations and offers both human-annotated and large-scale distantly supervised data. Extracting the named entities and relations requires reading multiple sentences in a document. Statistics for the DocRED dataset are provided in Table 5.6. Of the 96 Wikidata relations, only 31 relations are of relevance concerning the Enron Email Corpus. These 46 relations are described in Appendix F (See Table F.1).

JEREX (Eberts and Ulges, 2021), a joint entity-level relation extraction model that uses BERT for generating span embeddings and, using these embeddings, carries out entity extraction, coreference resolution, and relation extraction. The model, when released, was state-of-the-art on the DocRED dataset with an F1 score of 60.40. For these experiments, the `bert-base-cased` variant of the JEREX[2] model is used. The model was used in the *Multi-instance Relation Classifier (MRC)* setting. The experiments were carried out on an NVIDIA GeForce GTX 1080 Ti GPU with 8 12GB cores.

A sample email message and the obtained JEREX predictions are shown in Example 21. In the sample email message, relations between only two were found out of the extracted

[2]`https://github.com/lavis-nlp/jerex`

twelve mentions. Although the extracted relations are correct, the dearth of relations for most of the email messages led to this model not being used for relation extraction. For the email message shown in Example 20, JEREX was unable to obtain any relations.

Example 21. Example showing the predictions obtained using JEREX.

```
Input email message body:
I received a call form Bob Bradly who is representing Vern E
    Falkner, and he needs to talk to someone about possibly doing a tie-in to our
    pipeline in Hansford
    County Texas. His phone number is 405-842-4334.
I attempted to contact both of you via phone and was unsuccessful, hence the e-
    mail. Did not wish for this to fall through the cracks.
Thanks, John Toews, OCC

Predictions:
Hansford County - contains administrative territorial entity - Texas
Texas - located in the administrative territorial entity - Hansford County
```

CHAPTER 6

KNOWLEDGE REPRESENTATION

6.1 Introduction

Before delving into the chapter, it is essential to emphasize the shift in thinking for the remainder of the dissertation. The Enron Email Corpus was visualized as a single corpus consisting of email conversations belonging to various users. In other words, there was no divide in the email conversations with respect to the users. However, from now, this dissertation will consider the Enron Email Corpus as one containing a set of users that further contain some email conversations individually. The focus now will be on the user and the user's email conversations. The motivation behind this shift is how downstream email applications function. For emails, the downstream applications for a user generally focus on the user's emails and not emails of the entire organization or all users in general. Additionally, privacy is also an important aspect that needs to be considered when using user email data, as sharing information between users can prove to be a violation. Thus, the dissertation will focus on representing the knowledge extracted from a user's email conversations and using that in downstream applications.

6.1.1 Background

Template or wrapper induction is the technique of generating skeletal representations of repeated content from previously seen data. These skeletons can then be used to extract information from the previously unseen documents. Although email template induction is a widespread technique to extract information from emails, the technique was originally used on web documents (Kushmerick, 1997; Sarawagi, 2002; Arasu and Garcia-Molina, 2003; Hachenberg and Gottron, 2013). The technique is followed naturally for emails as most of the emails are system generated or Business-to-consumer (B2C). These emails use HTML

templates with a few variable fields in each template. One of the earlier works using a simpler form of template induction using email subject to carry out email threading was done by Ailon et al. (2013). Zhang et al. (2015) do template induction to extract product names from synthetically generated data. Wendt et al. (2016) perform email classification using hierarchical template representation. Moving away from HTML-based template induction, Gupta et al. (2019) use visual and semantic information to carry out template induction in place of the HTML DOM tree.

Viégas et al. (2006) create a tool *Themail* that portrayed relationships using the interaction histories preserved in email archives. The tool's interface shows a series of columns of keywords extracted from the user's email content, arranged along a timeline. Agarwal et al. (2012) use a weighted graph to create an Enron Organizational hierarchy. The graph represents all employees as nodes and links two employees who have communicated via email. The weight of the link represents the number of emails exchanged between the two employees (nodes) the link connects. The work of Beseiso et al. (2012) comes closest to the work presented in this chapter. The authors jointly perform ontology learning and knowledge extraction and use knowledge graphs to represent the extracted knowledge. However, the authors only focus on Business emails in the Enron Email Corpus and single email messages. This work focuses on email threads and does not ignore email threads based on the category of the email thread content.

6.1.2 Dissertation Contributions

The main contributions of this chapter are as follows:

1. In this chapter, the existing NEPOMUK ontology is modified to incorporate entity coreference resolution information in the knowledge graphs.

2. Using the extracted relations and entity coreference clusters for 6 users in the Enron corpus, two knowledge graphs *KG-Normal* and *KG-Coref* are created for each user. These knowledge graphs will be open-sourced.

The chapter is organized as follows: Section 6.2 describes previous works on knowledge graphs along with definitions of various terms used in the chapter. A detailed description of the ontologies used for creating the knowledge graphs along with added updates is outlined in Sections 6.4 and 6.4. Finally, Section 6.5 explains the knowledge graph creation process.

6.2 Knowledge Graphs

Richens (1956) introduced the concept of semantic nets in the context of mechanical translation. The earliest works to use the term "knowledge graphs" was done by Schneider (1973). This work followed that of Kingsley et al. (1969) that used "knowledge spaces" or "knowledge maps." Kingsley et al. use graph theory in their work on creating a metalanguage of communicable knowledge. However, it was the release of Google's Knowledge Graph (Singhal, 2012) that got the attention of the research community. This release highlighted the shift in perspective of seeing tokens as not just search strings but objects or things. The Knowledge Graph contained 500M objects and 3.5B facts about and relationships between different objects when launched. Today the Knowledge Graph contains about 5B entities and 500B facts.

The rise in the popularity of knowledge graphs has resulted in creation of numerous large scale knowledge graphs like OpenCyc, WordNet (Miller, 1995), Freebase (Bollacker et al., 2008), WikiData (Vrandečić and Krötzsch, 2014), BabelNet (Navigli and Ponzetto, 2012), DBPedia (Lehmann et al., 2015) and YAGO (Rebele et al., 2016). Domain-specific knowledge graphs have also gained much attention due to the significance of domain expertise represented in the knowledge graphs. Examples of domain-specific knowledge graphs are listed in

Table 6.1. In addition to domain-specific knowledge graphs, various task-specific knowledge graphs have been utilized in the industry. Previously discussed Google's Knowledge Graph is used in web search, knowledge graphs by Amazon (Dong, 2019), eBay (Pittman et al., 2017), Airbnb (Chang, 2018) and Uber (Hamad et al., 2018) for Commerce, and knowledge graphs by Facebook, LinkedIn (He et al., 2016) and Pinterest (Gonçalves et al., 2019) for Social Networks respectively. This dissertation refers readers to Noy et al. (2019) for a detailed read on five diverse industrial knowledge graphs.

Before defining various terms and concepts used in this chapter, this section elaborates on the motivation of using knowledge graphs for knowledge representation -

- Structure - Using knowledge graphs as the representation method can embed some structure in the representation via ontologies. This assists in simplifying the application of downstream tasks such as social network analysis, question answering, and searching.

- Compactness - Using knowledge graphs can make the representation compact. Common people across email threads can be grouped to use a common node in the graph, increasing the links and decreasing the node count.

- Adding knowledge - The availability of various large-scale knowledge graphs with world knowledge (See Table 6.1) and the usage of common ontologies makes it easy to add world knowledge to knowledge graphs. The task of identifying if an entity is common between two knowledge graphs is called Entity Typing. Previous works (Huang et al., 2015; Balaneshin-kordan and Kotov, 2016; Marino et al., 2016; Logan et al., 2019; Sharma et al., 2019; Futia and Vetrò, 2020) have highlighted the impact of adding world knowledge using knowledge graphs.

- Inferring Knowledge - Addition of rules using SWRL (Semantic Web Rule Language)[1] can help in reasoning and inferring new knowledge from the existing knowledge. A

[1] `https://www.w3.org/Submission/SWRL/`

Table 6.1. Domain specific Knowledge Graphs.

Domain	**Knowledge Graph**	**Status**	**Size**
Academic	OpenCitations Corpus (Peroni et al., 2015)	Active	As of November 03, 2020, the OCC has ingested the references from 326,743 citing bibliographic resources and contains information about 13,964,148 citation links to 7,565,367 cited resources[a].
	SciGraph[b]	Active	Currently, SciGraph is projected to contain about 1.5B to 2B triples.
	Microsoft Academic Knowledge Graph (Färber, 2019)	Active	As of 2018, the knowledge graphs contains 209M papers, 146M citations, and 253M authors.
Geographic	LinkedGeoData (Stadler et al., 2012)	Active	As of 2015, LinkedGeoData contains 1.2 billion triples.
Life Sciences	Bio2RDF (Belleau et al., 2008)	Active	As of July 2014, Bio2RDF contains 35 datasets with 11B triples.
User-generated content	Revyu (Heath and Motta, 2007)	Offline	As of 2014, Revyu contained 20,000 triples.
Tourism	La Rioja Turismo (Alonso-Maturana et al., 2018)	Active	675,368 triples
	Tyrolean Knowledge Graph (Kärle et al., 2018)	Active	As of 2018, the knowledge graph contains 1.5B triples.
Music	DBTune.org[c]	Active	1.1M+ triples
Law	Lynx (Montiel-Ponsoda et al., 2018)	Active	-

[a] `https://opencitations.net/corpus`

[b] `https://www.springernature.com/gp/researchers/scigraph`

[c] `http://dbtune.org/jamendo/`

`CORRESPONDED` property can be added to a conversational knowledge graph using a simple rule like if X is the sender and Y is a direct or indirect recipient, then X and Y have `CORRESPONDED` with each other.

6.2.1 Definitions

This section aims to provide concise definitions of knowledge graphs and their related terms. These definitions have been selected considering the goal of this dissertation. For a complete and comprehensive read about knowledge graphs, this dissertation refers readers to the works of Hogan et al. (2020) and Ji et al. (2021).

Knowledge Graph

The term knowledge graph has been defined differently by multiple works in the literature. Bergman (2019) provides a good description of the different definitions present. This dissertation uses the following definition (Hogan et al., 2020): *knowledge graph is a graph of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and whose edges represent relations between these entities.* Before we formally define what a *graph of data* is, it is important to define the term *knowledge*. *Knowledge*, in this dissertation, refers to something that is known. Knowledge can be factual statements like "Gloria wrote this email" (see Example 17) or quantified statements like "An email message can have only one sender."

A *graph of data* is a directed edge-labeled graph. A directed edge-labeled graph consists of two components: nodes and edges. In simple words, *nodes* represent entities and *edges* represent relations between those entities. However, since the term entity has been defined previously, a detailed definition of a node is needed. A *node* can be a mention referring to an entity defined in Section 5.2 or an entity that is part of the email metadata like sender, recipient, date, or body. An *edge* is a binary relation that holds between the two entities

connected by the edge, and the direction of the relation is given by the direction of the edge. Figure 6.1 shows few nodes and edges created from the email in Example 17.

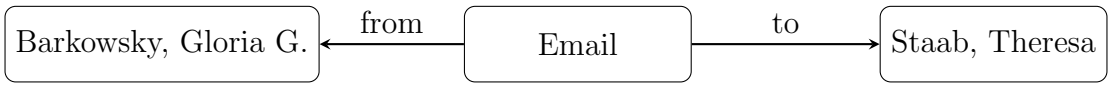

Figure 6.1. Sample directed edge-labeled graph for Example 17.

The terms discussed previously are formally defined in the Resource Description Framework or RDF. RDF is a standard model for exchanging data over the web. RDF is written using XML and hence not suitable for easy human interpretation. RDF uses a graph-based data structure for representing the data where the atomic representational structure is a triple. Each triple consists of a subject, predicate, and object. A set of such triples is termed an RDF graph. Figure 6.2 shows an example of a triple. The corresponding RDF/XML representation is shown after Figure 6.2. This representation is similar to the previously mentioned directed edge-labeled graph.

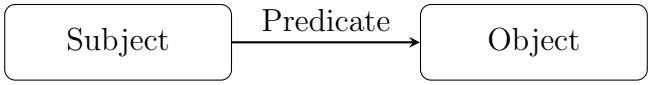

Figure 6.2. RDF triple example.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<rdf:Description rdf:about="Subject">
  <rdf:Predicate>Object</rdf:Predicate>
</rdf:Description>
</rdf:RDF>
```

A node in an RDF graph can be of three types: *IRIs*, *literals*, and *blank nodes*. For this dissertation, IRIs and literals are critical. Although blank nodes may be used when the

Object value of some Predicate is missing, the other two types are of statistical significance. An IRI or a literal, also known as a *resource* denotes something in the real world. Physical objects, abstract concepts, digital objects, numbers, and strings are examples of things that can be resources. The RDF W3C Recommendation defines the following concepts:

1. The resource denoted by an IRI is called its *referent*.
2. The resource denoted by a literal is called its *literal value*.
3. Literals have *datatypes* that define the range of possible values.
4. An *RDF Statement* is an assertion of an RDF triple indicating that some relationship, indicated by the predicate, holds between the resources denoted by the subject and object.
5. A predicate denotes a *property* and is a resource thought of as a binary relation.
6. An *RDF vocabulary* is a collection of IRIs. For example, the RDF Schema W3C Recommendation (Brickley et al., 2014) contains IRIs that form the RDF Schema vocabulary and can be used to define and document additional RDF vocabularies. In the RDF/XML example, *"http://www.w3.org/1999/02/22-rdf-syntax-ns#"* is a Namespace IRI for the RDF built-in vocabulary.

Ontology

A specification of a representational vocabulary for a shared domain of discourse - definitions of classes, relations, functions, and other objects - is called an ontology (Gruber, 1993). An ontology provides a generic and broader view of the knowledge being represented. The role that a schema plays in database systems is what an ontology plays for knowledge graphs. The definition of an ontology can be simplified as:

1. An ontology explains which entity classes are present and what properties each class object has.

2. It defines the relationships which exist between different classes. A relation definition comprises the range, domain, directionality, and cardinality.

Although today ontologies are used largely in conjunction with knowledge graphs, an ontology on its own is not bound to any knowledge representation strategy or logic. An ontology, being generic, provides a specification that a knowledge-base or knowledge-sharing system can follow. Today, when creating a knowledge graph, numerous open-source ontologies like schema.org[2], Wikidata[3], or DBPedia[4] are available which can be used as the base ontology for creating the knowledge graph. However, the NEPOMUK Message Ontology is best suited for the domain of emails.

6.3 NEPOMUK

Decker and Frank (2004) proposed the paradigm of Social Semantic Desktop (SSD) inspired by research in Semantic Web, Peer-to-Peer Networks, and Social Networks. At the core, SSD simplifies data sharing across applications on a computer and across the network. Using Semantic Web, a unified data representation can be created that applications can use. This representation facilitates improved data processing and helps in using the data to provide better or enhanced functionality. The Networked Environment for Personal Ontology-based Management of Unified Knowledge (NEPOMUK) project created such a data representation.

[2] `https://schema.org/`

[3] `https://www.wikidata.org/wiki/Wikidata:WikiProject_Ontology`

[4] `https://wiki.dbpedia.org/services-resources/ontology`

6.3.1 NEPOMUK Message Ontology (NMO)

The NEPOMUK Message Ontology is an extension of the NEPOMUK Information Element Framework. This ontology focuses on the domain of messages, specifically emails and instant messages. NMO models the structure of Emails using an ontology, thereby enabling Emails to be semantically linked to other domain ontologies. In this dissertation, the section of the ontology on emails has been used. Figure 6.3 and Table 6.2 provide a pictorial and detailed overview of the NEPOMUK Message Ontology section related to email messages, respectively. In Figure 6.3, the nodes colored red are Classes, and those colored in blue are Properties.

6.3.2 NEPOMUK Contact Ontology (NCO)

The NEPOMUK Contact Ontology aims to capture contact information that is common between multiple desktop applications. Inspired by the Vcard Ontology[5], this ontology extends the VCARD specification (Dawson and Howes, 1998) to provide a comprehensive framework that can be used to organize contact information. For this dissertation, the section of the ontology linked to NMO is considered. An overview from a graphical and classes/properties perspective of the considered section is provided Figure 6.4 and Table 6.3.

6.3.3 NEPOMUK File Ontology (NFO)

The NEPOMUK File Ontology deals with files and other desktop resources. The fundamental idea is that files are a sequence of bytes stored in a Filesystem or on a Network. The ontology supports files that reside on a filesystem and embedded or attached files in messages or files in the trash folder. Since the research presented here focuses on emails, the usage of this ontology has been very simplistic. Table 6.4 provides details regarding the elements

[5]https://www.w3.org/TR/vcard-rdf/

Table 6.2. Important classes and properties of the NEPOMUK Message Ontology (NCO - NEPOMUK Contact Ontology, NFO - NEPOMUK File Ontology).

Name	**Type**	**Comments**	**Domain**	**Range**
Message	Class	A message. Could be an email, instant messanging message, SMS message etc.	N/A	N/A
Email	Class	An email	N/A	N/A
messageSubject	Property	The subject of a message	Message	string
emailTo	Property	The primary intended recipient of an email	Email	ContactMedium (NCO)
emailCc	Property	A Contact that is to receive a cc of the email	Email	ContactMedium (NCO)
emailBcc	Property	A Contact that is to receive a bcc of the email	Email	ContactMedium (NCO)
messageFrom	Property	The sender of the message	Message	ContactMedium (NCO)
messageId	Property	An identifier of a message	Message	string
receivedDate	Property	Date when this message was received	Message	date-time
sentDate	Property	Date when this message was sent	Message	date-time
plainTextMessageContent	Property	Plain text representation of the body of the message	Message	string
inReplyTo	Property	Signifies that a message is a reply to another message	Message	Message
hasAttachment	Property	Links a message with files that were sent as attachments	Message	Attachment (NFO)

Table 6.3. Important classes and properties of the NEPOMUK Contact Ontology.

Name	**Type**	**Comments**	**Domain**	**Range**
Contact	Class	A Contact. A piece of data that can provide means to identify or communicate with an entity.	N/A	N/A
PersonContact	Class	A Contact that denotes a Person	N/A	N/A
Name	Class	A class representing a contact's name	N/A	N/A
PersonName	Class	A class representing a person contact's name. Possesses various attributes	N/A	N/A
ContactMedium	Class	A superclass for all contact media - ways to contact an entity represented by a Contact instance	N/A	N/A
EmailAddress	Class	An email address	N/A	N/A
emailAddress	Property	-	EmailAddress	string
fullname	Property	To specify the formatted text corresponding to the name of the object the Contact represents	Name	string
hasEmailAddress	Property	An address for electronic mail communication with the object specified by this contact	PersonContact	EmailAddress
hasPersonName	Property	Attaches name information to a person contact	PersonContact	PersonName
nickname	Property	A nickname of the Object represented by this Contact	Name	string

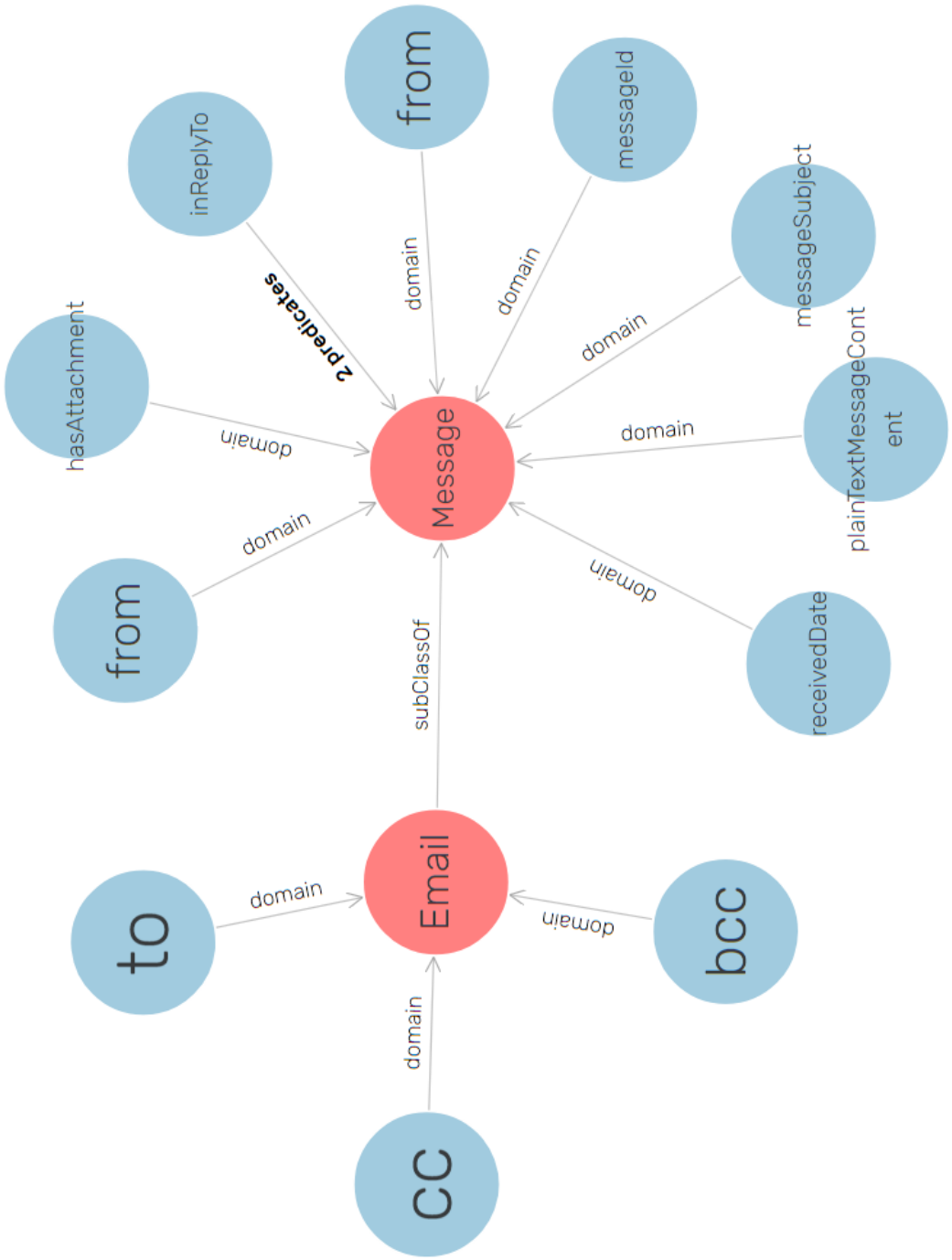

Figure 6.3. NEPOMUK Message Ontology.

of the ontology used, whereas Figure 6.5 displays a brief graphical overview of some crucial elements of this ontology. An overview of the links between the three ontologies - NMO,

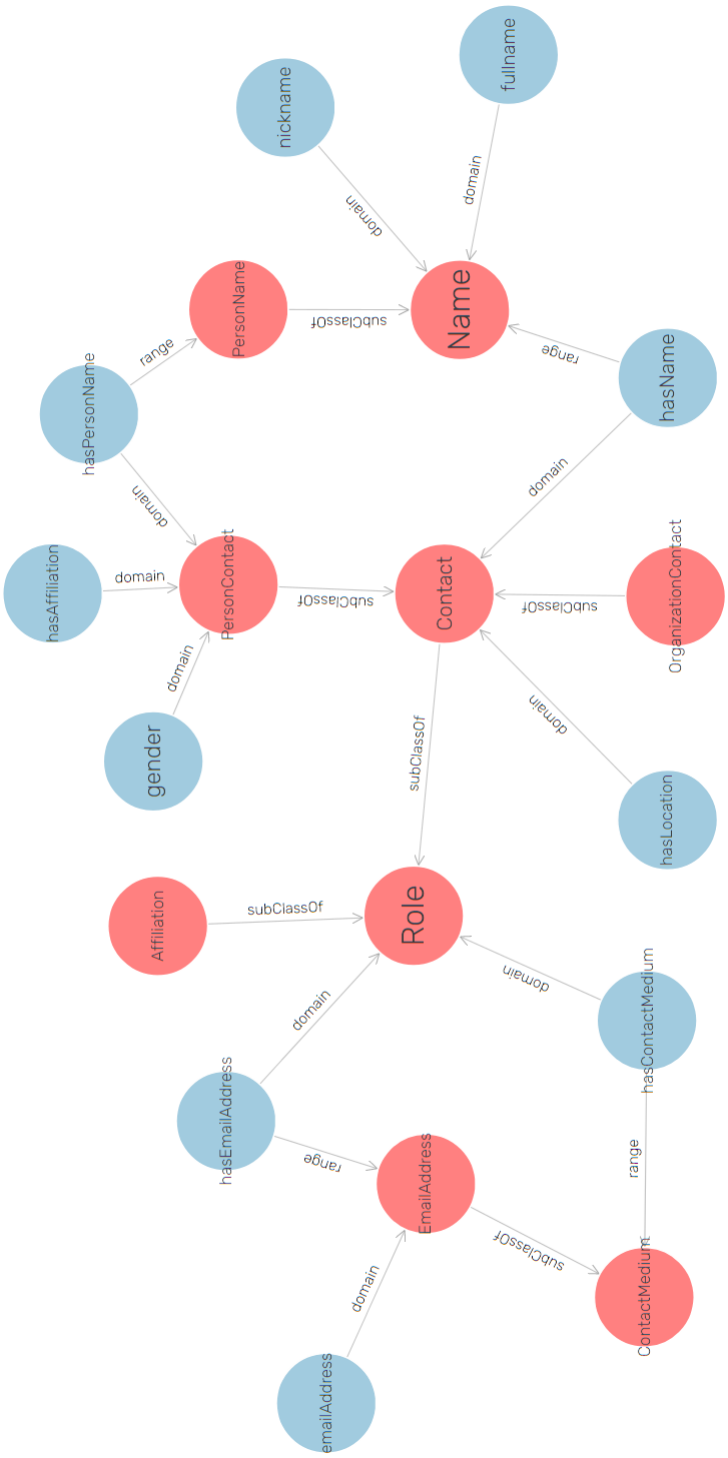

Figure 6.4. NEPOMUK Contact Ontology.

NCO, and NFO is shown in Figure 6.6. The figure's left and right dividers show the linking Classes and Properties between NCO-NMO and NMO-NFO, respectively.

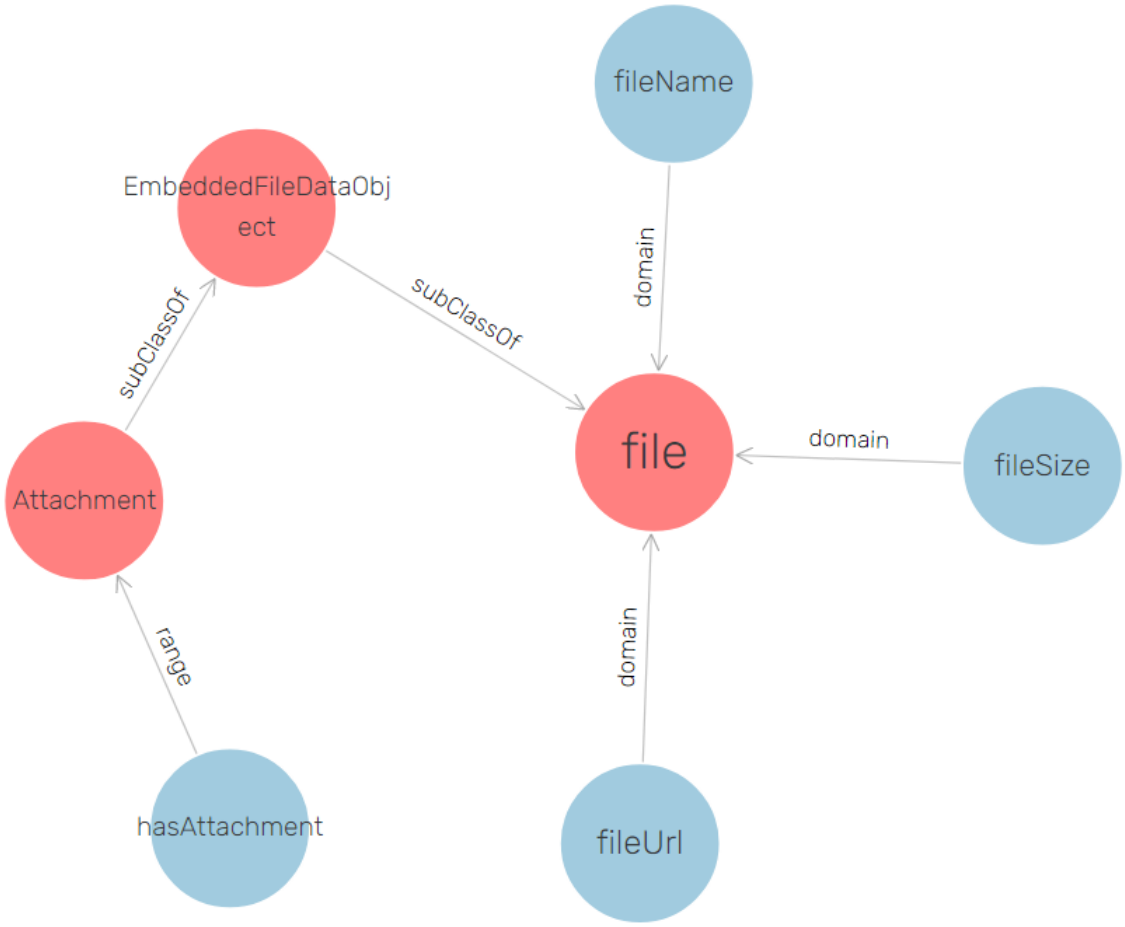

Figure 6.5. NEPOMUK File Ontology.

6.4 Ontology Updates

The three NEPOMUK ontologies described before are used as the base ontologies for creating the knowledge graphs. However, the ontologies in themselves are not sufficient to completely represent the email threads present in the annotated corpus created in Section 5.3. Table 6.5 shows details of the added elements. The bold-faced elements are the new additions to base ontologies. Figure 6.7 shows the added elements from a graphical perspective.

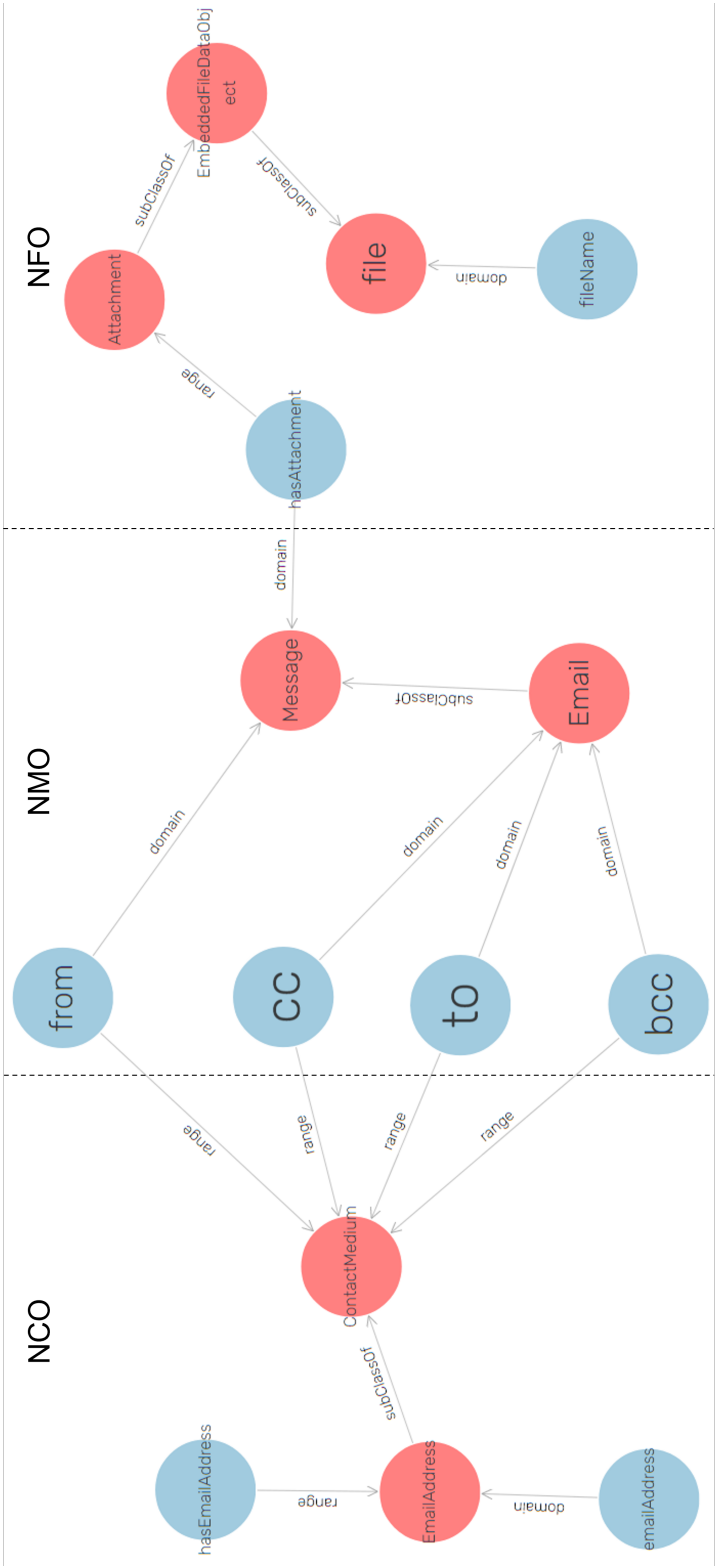

Figure 6.6. NEPOMUK inter-ontology links.

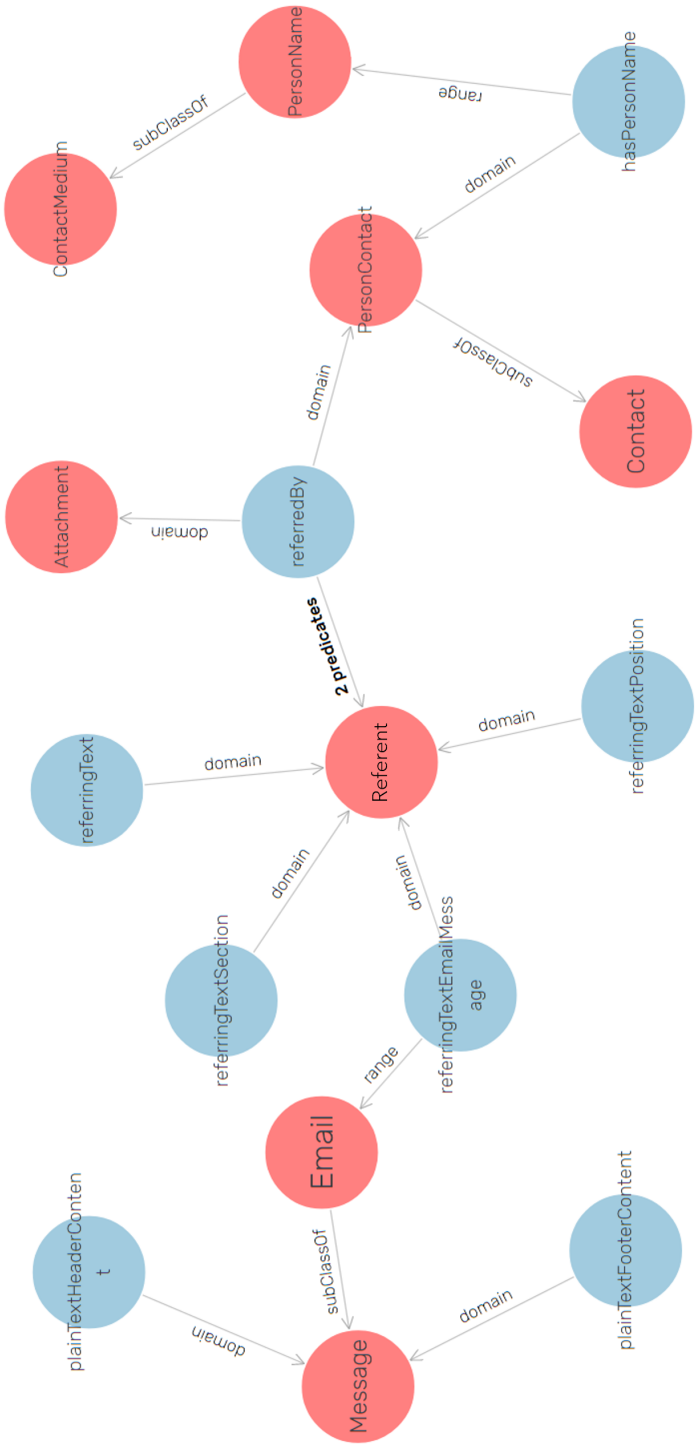

Figure 6.7. Added elements to the base ontologies.

Table 6.4. Important classes and properties of the NEPOMUK File Ontology.

Name	Type	Comments	Domain	Range
FileDataObject	Class	A resource containing a finite sequence of bytes with arbitrary information, that is available to a computer program and is usually based on some kind of durable storage	N/A	N/A
Attachment	Class	A file attached to another data object	N/A	N/A
fileName	Property	Name of the file, together with the extension	FileDataObject	string

6.5 Knowledge Graph Creation

This section describes the process for the creation of knowledge graphs. For this process, the ontologies described previously have been used as the skeleton or schema. The remainder of the section discusses the creation of two knowledge graphs - *KG-Normal* and *KG-Coref*. The second knowledge graph *KG-Coref* is created to evaluate and analyze the impact of adding coreference predictions obtained using the JM2+S model (See Section 4.7) to the *KG-Normal*. Python 3.6 was used along with `rdflib`[6] to create the knowledge graphs.

6.5.1 KG-Normal

The *KG-Normal* knowledge graph is created using the `ECRA` dataset. A total of 762 email threads from 6 users is taken as input for the knowledge graph creation process (See Section 5.3 for more details on the input data). In order to create the knowledge graph, first, a mapping between the extracted relations and the RDF properties described in the Sections 6.3 and 6.4 is defined (See Table 6.6). Next, using this mapping, each email is processed and

[6] `https://rdflib.readthedocs.io/en/stable/gettingstarted.html`

Table 6.5. Classes and Properties used in addition to the NEPOMUK Ontologies.

Name	**Type**	**Comments**	**Domain**	**Range**
plainTextHeaderContent	Property	Plain text representation of the header of the message	Message	string
plainTextFooterContent	Property	Plain text representation of the footer of the message	Message	string
PersonName	Class **(Parent- ContactMedium)**	A class representing a person contact's name	N/A	N/A
Referent	Class	A referent	N/A	N/A
referingText	Property	The text span that is refering an entity	Referent	string
referringTextPosition	Property	The position of the text span that is referring an entity	Referent	integer
referringTextSection	Property	The section of the email that the referent text is present	Referent	string
referringTextEmailMessage	Property	The email message that the referent text is present	Referent	Email
referredBy	Property	Entity being referred to by a Referent	Referent; PersonContact; Attachment	Referent

the extracted relation is mapped to an ontology property to create a knowledge graph. Along with the mapping, one additional property *nmo:messageId* is used. For all email messages containing a message id as "Message-ID: <15043288.1075859.JavaMail.evans@thyme>", the id '15043288.1075859' is extracted. For email messages without an explicit message id, a Universally Unique Identifier (UUID) is used as the message id.

Algorithm 1 outlines this creation process in a succinct manner. In the algorithm, the constructs $Email(e)$, $PersonContact(S)$, $EmailAddress(P, S)$, $PersonName(P, S)$ and $FileDataObject(a)$ create a unique IRI of the respective class using the given object that is then added to the knowledge graph. The algorithm takes an email thread T, a knowledge graph G, and a hashmap $persom\_map$ that maps person names or email addresses to their PersonContact IRIs. Lines 1-7 create a new NMO Email object and add different email attributes to the object. Finally, on line 8, the email object is added to the graph G. Lines 9-21 iterate over each sender S. If the sender S is not present in $person\_map$ (lines 10-14), a new NCO PersonContact object is created and added to the hashmap and the graph. In lines 16-20, depending on S being an email address or an entity name, an object of NCO EmailAddress or NCO PersonName is created and added to the graph. This loop from 9-21 is repeated for recipients in to, cc and bcc fields, respectively. Finally, lines 23-27, iterating over each attachment, create a new NFO FileDataObject, set the fileName attribute, and add the attachment object to the graph.

6.5.2 KG-Coref

KG-Coref is created by incorporating the entity coreference resolution predictions for the email threads in `ECRA` with *KG-Normal*. The creation process differs in how a PersonContact/EmailAddress/PersonName is added to the graph and how the mentions identified in the email subject/body are added to the graph. The addition of the mentions is carried out in a manner that preserves the following information - the email message the mention

Algorithm 1 Algorithm to create KG-Normal

Require: Email Thread T, KG G and Hashmap $person\_map$

for email e in thread T **do**
  $E \leftarrow Email(e)$ ▷ Create nmo:Email object
  $E.subject \leftarrow subject$
  $E.receivedDate \leftarrow date$
  $E.plainTextHeaderContent \leftarrow headerText$
  $E.plainTextBodyContent \leftarrow bodyText$
  $E.plainTextFooterContent \leftarrow footerText$
  $G \leftarrow E$ ▷ Add the email object to the Graph
  for sender S in from **do**
    if S does not exists in $person\_map$ **then**
      $P \leftarrow PersonContact(S)$
      $person\_map \leftarrow S, P$
      $G \leftarrow E, P$ ▷ Add PersonContact object as nmo:messageFrom to the Graph
    end if
    $P \leftarrow person\_map(S)$
    if S is an email address **then**
      $G \leftarrow EmailAddress(P, S)$ ▷ Add PersonContact email address to the Graph
    else if S is a name **then**
      $G \leftarrow PersonName(P, S)$ ▷ Add PersonContact name to the Graph
    end if
  end for
  ... ▷ Repeat the process for to, cc and bcc recipients in the email
  for attachment a in attachments **do**
    $A \leftarrow FileDataObject(a)$
    $A.fileName \leftarrow a$
    $G \leftarrow E, A$
  end for
end for

Table 6.6. Mapping between ECRA relations and NEPOMUK properties.

Relation	Ontology Property
from	nmo:messageFrom
to	nmo:emailTo
cc	nmo:emailCc
bcc	nmo:emailBcc
subject	nmo:messageSubject
date-time	nmo:receivedDate
	nmo:sentDate
attachment	nmo:hasAttachment
header-text	nmo:plainTextHeaderContent
body-text	nmo:plainTextMessageContent
footer-text	nmo:plainTextHeaderContent
reply-to	nmo:inReplyTo

belongs to, the section the mention is located in, the position in the section text, and the PersonContact node that the mention is referring to. The main differences between the two algorithms are as follows:

1. Adding PersonContact - Lines 11-13 in Algorithm 1 are replace by the function given in Algorithm 2. For *KG-Coref*, the check on line 10 of Algorithm 1 is done, and if the check fails, other mentions in the same coreference cluster are checked in $person\_map$ and if a match is found, the corresponding PersonContact IRI is used (lines 2-6 in Algorithm 2). If no match is found, a new NCO PersonContact object is created and added to the hashmap and the graph (lines 7-11 in Algorithm 2).

2. Adding remaining mentions - Algorithm 3 is used for adding the remaining mentions to the graph G. This algorithm follows line 28 of Algorithm 1. Lines 1 and 2 iterate over each chain $c \in C$ and each mention $m \in c$. If m is not present in the $person\_map$, a new Referent node is created and the corresponding referringText, referringTextEmailMessage, referringTextSection, and referringTextPosition properties

are set (Lines 3-9). Next, using $GetClusterEntity$ a mention $CE \in c$ is searched such that $CE \in person\_map$. If CE exists, the referredBy property is set (Lines 10-13). Finally, on lines 14 and 15, the referent R, mention m, and email E are added to the $person\_map$ and graph G.

Algorithm 2 Algorithm to add a sender S or recipient R to the knowledge graph

Require: Sender or Recipient S, KG G, Hashmap $person\_map$ and Coreference Clusters C

1: $P \leftarrow NULL$
2: **for** Cluster c in C **do**
3: **if** $S \in c$ and $\{m_i$ in $person\_map | \ m_i \in c, m_i \neq S\}$ **then**
4: $P \leftarrow person\_map(m_i)$
5: **end if**
6: **end for**
7: **if** P is $NULL$ **then**
8: $P \leftarrow PersonContact(S)$
9: **end if**
10: $person\_map \leftarrow S, P$
11: $G \leftarrow E, P$

Table 6.7 shows the comparison between the two knowledge graphs using various statistics. The reduction in the number of PersonContact nodes highlights the compactness of *KG-Coref*. In addition, the linking of various PersonContact nodes together due to coreference resolution reduces the number of links (to, cc, bcc) between Email nodes and PersonContact nodes.

Algorithm 3 Algorithm to add remaining mentions to the knowledge graph

Require: Email thread T, KG G, Hashmap $person\_map$ and Coreference Clusters C

for chain c in C **do**
  for mention m in c **do**
    if m not in $person\_map$ **then**
      $R \leftarrow Referent()$
      $R.referringText \leftarrow m$
      $E \leftarrow GetEmailInThread(m, T)$
      $R.referringTextEmailMessage \leftarrow E$
      $R.referringTextSection \leftarrow GetSectionInEmail(m, E)$
      $R.referringTextPosition \leftarrow GetPositionInEmail(m, E)$
      $CE \leftarrow GetClusterEntity(c)$
      if CE is not $NULL$ **then**
        $CE.referredBy \leftarrow R$
      end if
      $person\_map \leftarrow m, R$
      $G \leftarrow E, R$
    end if
  end for
end for

Table 6.7. Statistics comparing KG-Normal and KG-Coref

Statistic	**KG-Normal**	**KG-Coref**
Email nodes	4,525	4,525
PersonContact nodes	8,474	7,704
Attachment nodes	397	397
Email links	57,930	57,523
PersonContact links	8,672	7,967
Recipient Links		
From	5,557	5,979
To	22,520	22,305
Cc	8,650	8,487
Bcc	1,720	1,691
Referents	-	79,932

CHAPTER 7

QUESTION ANSWERING IN EMAIL CONVERSATIONS

7.1 Introduction

An intelligent personal assistant is an implementation of an animated computer interface agent with social intelligence that assists a user in operating a computing device and using application programs on a computing device (Gong, 2003). These programs use a combination of speech processing, natural language processing and understanding, and machine learning to understand the input, process it, and generate an appropriate response. Here, this dissertation specifically focuses on a type of intelligent personal assistants - digital voice assistants (DVAs) like Apple's Siri (Team, 2017), Amazon Alexa[1], Google Assistant[2], and Microsoft Cortana[3]. Various studies (Tulshan and Dhage, 2018; Song et al., 2019; Terzopoulos and Satratzemi, 2020) and surveys have studied the impact of these assistants and projected significant growth in their future usage. Juniper Research, in their 2018 market research report, estimate the number of digital voice assistants in use to rise to 8 billion by the end of 2023[4]. Google in 2018 reported that 72% of the people who owned a voice-activated speaker use it as part of their daily lives (Kleinberg, 2018). Edison Research and NPR in their Smart Audio Report[5] tell that 63% of the total U.S. online population (18+) use a voice-operated personal assistant on any device.

Since the late 1990's the problem of email overload has been well-documented (Whittaker and Sidner, 1996; Jackson et al., 2002; Dabbish and Kraut, 2006; Soucek and Moser, 2010;

[1] https://developer.amazon.com/en-US/alexa

[2] https://assistant.google.com/

[3] https://www.microsoft.com/en-us/cortana/

[4] https://www.juniperresearch.com/researchstore/content-digital-media/voice-assistants-market-research-report

[5] https://www.nationalpublicmedia.com/insights/reports/smart-audio-report/

Barley et al., 2011; Group and Group, 2012; The Radicati Group, 2015). This drove the research on email processing (Bellotti et al., 2003; Gupta et al., 2004; Park et al., 2019), email classification (Grbovic et al., 2014; da Silva, 2016; Mujtaba et al., 2017) and spam-ham filtering (Youn and McLeod, 2007; Awad and ELseuofi, 2011). The increase in the number of emails and usage of DVA's leads us to explore the prospect of using DVAs for email-based applications. Bendersky et al. (2021) in their work also highlight the importance and need for a question answering system for answering questions over personal emails.

In this chapter, the Question answering (QA) task is formulated in a setting where the entity asking a question is a Person, and the entity answering a question is a DVA. In the real world, the questions asked will be complex, and the eventual outcome is an action taken or information provided by the DVA. Consider the following exchange between Bob (Person) and his DVA:

```
Bob: Can you forward the resume Alice sent to Mark?
DVA: I found two resumes. One was sent by Alice1 and the other by Alice2. Which
    one do you want me to forward?
Bob: The one sent by Alice1.
```

In the question `Can you forward the resume Alice sent to Mark?`, there are multiple entities involved - you (DVA), the resume, Alice, and Mark. There is also a resulting explicit action involved in forwarding the document. There are implicit actions involved in searching the document and resolving the entities mentioned. The DVA can unambiguously resolve Mark but needs assistance in resolving Alice due to two entities named Alice. In order to address these scenarios and sub-tasks, in this chapter, we target one of the crucial sub-tasks of resolving entities given the textual transcript of the question. This work aims to create a question answering dataset that would test the resolution skills of the answering system. Multiple baselines are also tested on the dataset to observe their performance and identify future avenues for improvement. Figure 7.1 shows a sample scenario for this simplified task.

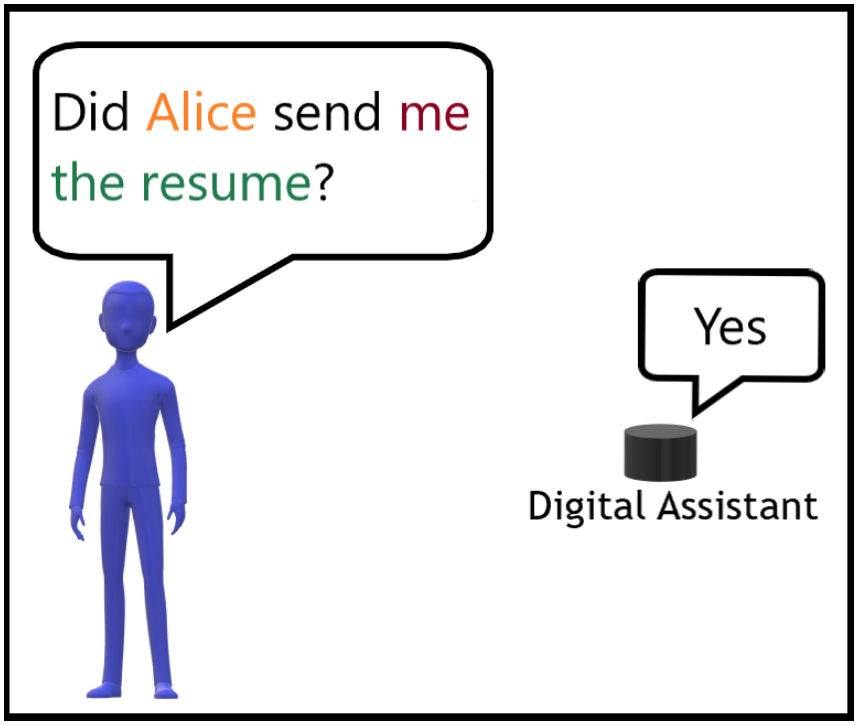

Figure 7.1. Pictorial representation of the QA environment setting.

7.1.1 Background

The question answering task for email conversations has not been carried out in the traditional sense previously. The traditional sense focuses on answering questions within the context of a given set of documents. The size of the given set of documents can vary, but the general nature of the task is the same. The approach for emails, however, has been different. Yang et al. (2018) extract question-answer pairs already present in the Avocado Email Collection to create a QA dataset. The authors extract questions and candidate answers from the email collection and foresee using the dataset for performing text similarity to answer previously seen questions. The work of Zylich et al. (2020) comes closest to our work. Zylich et al. (2020) create an open-domain question answering system for teaching assistance. They create the dataset using course material like the syllabus, lecture slides, course emails, and prior discussion forum posts. The dataset primarily contains 2004 discussion forum posts, and only a minor fraction of the additional 288 documents were announcement emails. Our

work, however, focuses solely on email conversations and, specifically, email conversations that took place in the real world containing a mix of business and personal emails.

7.1.2 Dissertation Contributions

The main contributions of this chapter are as follows:

1. This dissertation explores the task of Question Answering in email conversations in a novel setting using digital voice assistants. The sub-task involving entity coreference resolution is identified as crucial and chosen as the focus of the question answering task.

2. We create the *EMailQA* dataset containing 29,574 questions. These questions are created using five templates that target the coreference resolution skills of a QA system. Manual annotation is carried out to extract unique alias-name and alias-attachment name pairs from the email threads in the `ECRA` dataset. Three levels of complexity are introduced in the created questions using these alias pairs. A total of 8,648 questions are created with a complexity level of 1 or higher. This dataset will be released publicly.

3. Using the created questions and two simple SPARQL query-based baselines, the impact of the joint learning models proposed in sub-section 4.7.1 is evaluated. The knowledge graphs containing entity coreference information result in a 3.91% increase in the accuracy of the baseline system.

7.2 EMailQA dataset

This section describes the steps carried out to create a Question-Answering dataset - `EMailQA` for the same. This dissertation aims to achieve the following goals via `EMailQA`:

1. Create a corpus containing questions that provide information about the email collection.

2. Add complexity to the dataset by using aliases in place of names and email addresses. This goal will help in testing the ability of a system to carry out coreference resolution.

3. Add complexity to the dataset by using attachment description in place of the attachment file names. This further tests the system's coreference resolution capabilities.

4. Rank the questions based on the added complexities for assisting in qualitative analysis of a QA system.

5. Create a benchmark for Question answering for email conversations. We hope that this would drive future research in corpus development and improve performance on the QA task for this domain.

Alrashed et al. (2018) show that the top three reasons for an email revisit are - Instructions to perform a certain task (24.1%), Attachments or links (22%), and an answer to a question that was previously asked (16.3%). Following this analysis, attachments are targeted as the primary topic for the questions. Table 7.1 shows the question templates that were used in creating questions. Along with the templates, the table also lists some properties and examples for each template.

The JSON representation of 762 email threads created as the output of the relation extraction process is used as input for the question generation process.

The generation process for each template is carried out as follows:

1. **Template 1 & 4** - Extract all email messages containing a sender, at least one attachment, and a received date-time. For each attachment as A and sender as X, create the question with the date-time as the answer. For each attachment A, create a question with the sender X as the answer.

2. **Template 2** - Extract all email messages containing at least one recipient (to/cc/bcc) and at least one attachment. For each recipient as X and each attachment as A, create

Table 7.1. Templates used for question generation along with properties and example questions.

	Template	**Properties**	**Examples**
T1	When did X send A?	X - Person A - Attachment Answer type - Date-time	When did romero, araceli send Estate1 _ 08 . ppt? When did tom bauer send Cmpltnss & AccrcyJS . doc?
T2	Did X receive A?	X - Person A - Attachment Answer type - Yes/No	Did sgillesp@sequentenergy.com receive vng _ release _ 1 _ 0 . zip? Did mike jordan receive Vision1 . ppt?
T3	Did X send A to Z?	X,Z - Person A - Attachment Answer type - Yes/No	Did tom bauer send Cmpltnss & AccrcyJS . doc to shawn kilchrist? Did mary solmonson send Vision1 . ppt to mike jordan?
T4	Who sent A?	A - Attachment Answer type - Person-/Org	Who sent Estate1 _ 08 . ppt? Who sent n381RED . DOC?
T5	What is X's email address?	X - Person Answer type - Email address	What is glover, sheila's email address? What is mills, scott's email address?

a question with the answer as *Yes*. Negative examples are created using recipient-attachment pairs that do not exist in the email collection.

3. **Template 3** - Extract all email messages containing at least one recipient (to/cc/bcc), a sender, and at least one attachment. For each recipient Z, each attachment A and sender X create a question with the answer as *Yes*. Similar to template 2, negative examples are created using recipient-sender-attachment pairs that do not exist in the email collection.

4. **Template 5** - Extract email messages that contain a sender name and sender email address. Using the name as X, create a question with the email address as the answer.

For the email threads in consideration, all Person/Organization/Attachment aliases are extracted from the each email thread. Let T be an email thread containing M email messages $EM_1, EM_2, .., EM_M$. Let $P_1, P_2, .., P_N$ be the N text spans consisting of names/email addresses of sender/recipient Person/Organization entities. Let $A_1, A_1, .., A_K$ be the K attachments present in T. From the M email messages, all tokens from the email message body are extracted and tokens with a partial match to the N name/email address text spans are identified. These partial matches are then manually checked and filtered to obtain all valid alias-name or alias-email address pairs. An alias-name or alias-email address pair is considered valid if, for a given email collection, the same alias is not mapped to another person/organization's name or email address of another Person/Organization. For example, in the pair *mahesh-"mahesh lakhani"* the alias *mahesh* represents a Person X with name *mahesh lakhani*. This pair is valid if the alias *mahesh* is not mapped to a name or email address of another Person X'. The manual checking and filtering results in the extraction of 908 alias-name or alias-email address pairs.

For attachment aliases, the body of the email message containing the attachment is extracted. Next, the span of text describing the attachment is selected manually. If the email message body does not contain a clear description of the attachment, the corresponding attachment-description pair is discarded. The extracted description is treated as an alias of the attachment, and for a given email collection, all valid alias-attachment pairs are extracted. An alias-attachment pair is valid if the alias does not map to another attachment in the same email collection. This process results in extracting 124 alias-attachment pairs.

The extracted alias pairs are then used to introduce complexity to the questions generated using the steps mentioned before for each template. For each template, if an alias for a name/email address/attachment exists, then it is used in place of the original text. Depending on the number of substitutions made in a question, a complexity level is assigned. This level indicates how many coreference resolutions the QA system needs to perform to answer

the question correctly. Example 22 shows the questions generated using the extracted aliases and their corresponding complexity levels. Tables 7.2 and 7.3 show the statistics of the generated questions. This dissertation refers to the entire dataset as `EMailQA-Full` and the subset containing questions with a complexity of one or more as `EMailQA-Coref`. Additional examples for each complexity level are provided in Appendix B.

Example 22. Example showing questions generated using alias substitution and their complexity levels.

```
Complexity Level 1:
Original - Did tom bauer send Cmpltnss & AccrcyJS . doc to shawn kilchrist?
With Alias - Did tom bauer send Cmpltnss & AccrcyJS . doc to shawn?

Original - What is peggy mahoney's email address?
With Alias - What is peggy's email address?

Original - When did tom bauer send Cmpltnss & AccrcyJS . doc?
With Alias - When did tom bauer send the list of items we recently discussed in
    our Trading status meeting?

Complexity Level 2:
Original - Did shawn kilchrist receive Cmpltnss & AccrcyJS . doc?
With Alias - Did shawn receive the list of items we recently discussed in our
    Trading status meeting?

Original - Did william gang send ~0064565edwards.doc to jeff dasovich?
With Alias - Did bill gang send ~0064565edwards.doc to dasovich?
```

Original - Did steffes, james d. send CA Surcharge Matrix 10 - 09 . doc to dasovich, jeff?

With Alias - Did jim steffes send CA Surcharge Matrix 10 - 09 . doc to dasovich?

Original - When did choate, heather send Estate . ppt?

With Alias - When did heather send the Estate Org Chart?

Complexity Level 3:

Original - Did piper, greg send n1bx11 ! . DOC to koehler, anne c.?

With Alias - Did greg send the latest draft of all purchase and sale documents on the sale of NetCo to anne?

Original - Did william gang send the draft of ~0064565edwards.doc to jeff dasovich?

With Alias - Did bill gang send the draft of the transaction description for the privatization of the electrical distribution system at Edwards AFB to dasovich ?

Original - Did matt.pagano@sce.com send DA Proposal . doc to dasovich, jeff?

With Alias - Did matt send the SCE ' s proposal to settle past Direct Access Credit issues to dasovich?

7.3 Baselines

In this section, the baseline systems evaluated on `EMailQA-Full` and `EMailQA-Coref` are described. It is important to highlight that although the same question-answer pairs are used across all the baselines, the input document or the email collection format is not the

Table 7.2. Statistics of `EMailQA-Full` and `EMailQA-Coref`.

Statistic	**Value**
Total number of questions generated	29,574
Number of questions generated using Template 1	346
Number of questions generated using Template 2	10,804
Number of questions generated using Template 3	17,533
Number of questions generated using Template 4	331
Number of questions generated using Template 5	740
Number of questions with complexity 0	21,106
Number of questions with complexity 1	7,944
Number of questions with complexity 2	649
Number of questions with complexity 3	55
Number of negative questions added	7,676

Table 7.3. Distribution of questions with respect to users.

User	**Email Thread Count**	**Attachment Count**	**Question Count**
Beck-s	115	21	1,129
Dasovich-j	472	260	25,248
Haedicke-m	50	28	2,313
Lay-k	19	10	115
Sager-e	90	77	919
Skilling-j	16	1	30

same (plain text vs. RDF). The motivation behind different input formats is to observe the impact of using knowledge graphs as the representation method.

7.3.1 UnifiedQA

Khashabi et al. (2020) build a single pre-trained model, UnifiedQA, that is trained on four different types of question answering formats. A total of eight different datasets are used for training. The authors train the T5 (Raffel et al., 2020) model on different types of QA formats - extractive, abstractive, multiple-choice, and yes/no. The UnifiedQA model is trained in a text-in text-out fashion similar to the T5 model. It reports state-of-the-art results on

several QA datasets. In both `EMailQA` datasets, extractive and yes/no types of questions are present. The ability of the UnifiedQA model to handle different types of questions and its state-of-the-art performance makes it an excellent deep learning based baseline.

7.3.2 SimpleQuery

In this baseline, given a question, a SPARQL query is generated for that question. The generated query is used to extract the answer from *KG-Normal*. Since the questions are generated using templates, a corresponding SPARQL query template was created for each question template. For this baseline, given a question, the placeholders in the corresponding template are extracted. The extracted placeholders and the corresponding query template are used to generate a SPARQL query. A sample question template and corresponding SPARQL query template used for this baseline are provided below. Appendix C.1 lists all question templates and their corresponding SPARQL query templates.

```
Question Template: When did X send A?
Query Template:
PREFIX nco: <http://www.semanticdesktop.org/ontologies/2007/03/22/nco#>
PREFIX nmo: <http://www.semanticdesktop.org/ontologies/2007/03/22/nmo#>
PREFIX nfo: <http://www.semanticdesktop.org/ontologies/2007/03/22/nfo#>
select ?date where {
    VALUES (?nametype) {
        (nco:fullname)
        (nco:nickname)
        (nco:emailAddress)
    }
    ?emailmessage nmo:messageFrom ?person .
    ?person ?nametype "X" .
    ?emailmessage nmo:sentDate ?date .
```

```
    ?emailmessage nmo:hasAttachment ?attachment .
    ?attachment nfo:fileName "A" .
}
```

7.3.3 CorefQuery

KG-Coref builds on *KG-Normal* by incorporating the coreference resolution predictions. This baseline aims to evaluate the impact of the addition of coreference predictions. Similar to the SimpleQuery baseline, this baseline uses a SPARQL query template for each question type. The SPARQL query template used for template T1 is provided below. Appendix C.2 lists all query templates used in this baseline.

```
Question Template: When did X send A?
Query Template:
PREFIX nco: <http://www.semanticdesktop.org/ontologies/2007/03/22/nco#>
PREFIX nmo: <http://www.semanticdesktop.org/ontologies/2007/03/22/nmo#>
PREFIX nfo: <http://www.semanticdesktop.org/ontologies/2007/03/22/nfo#>
PREFIX kefec: <http://example.org/kefec/ontology#>
select ?date where {
    VALUES (?contactmedium) {
        (nco:hasPersonName)
        (nco:hasEmailAddress)
    }
    ?email nmo:messageFrom ?sender .
    ?person ?contactmedium ?sender .
    ?email nmo:sentDate ?date .
    ?email nmo:hasAttachment ?attach .
    {
        VALUES (?nametype) {
```

```
            (nco:fullname)
            (nco:nickname)
            (nco:emailAddress)
        }
        ?sender ?nametype "X" .
    }
    UNION {
        ?person kefec:referredBy ?ref .
        ?ref kefec:referringText "X" .
    }
    {
        ?attach nfo:fileName "A" .
    }
    UNION {
        ?attach kefec:referredBy ?ref1 .
        ?ref1 kefec:referringText "A" .
    }
}
```

7.4 Experimentation and results

For evaluating the baselines, the `EMailQA-Full` and `EMailQA-Coref` datasets are used. To evaluate the UnifiedQA[6] baseline, the context for the answer (email message) is extracted and given as input to the model. The `unifiedqa-t5-small` variant of the model is used and evaluated on an NVIDIA GeForce GTX 1080 Ti GPU with 8 12GB cores. For the

[6] `https://github.com/allenai/unifiedqa`

Table 7.4. Experiment QA experiment results for all baselines.

Baseline	**Correct**$_{COREF}$	**Accuracy**$_{COREF}$	**Correct**$_{FULL}$	**Accuracy**$_{FULL}$
UnifiedQA	18	0.2%	53	0.17%
SimpleQuery	976	11.28%	17968	60.38%
CorefQuery	**1131**	**13.49%**	**19106**	**64.21%**

SimpleQuery and CorefQuery baselines, the knowledge graphs were stored into GraphDB[7]. For each baseline, the SPARQL queries for all questions were generated and executed via GraphDB API calls using Python 3.6. The accuracy metric is used to compare the models. The answers returned by the baselines are matched to the gold answers to compute the accuracy.

Table 7.4 shows results of the experiments on both `EMailQA-Full` and `EMailQA-Coref`. The table shows the number of questions correctly answered and the corresponding accuracy. On `EMailQA-Full` the best performance of 64.21% was reported by CorefQuery compared to 60.38% reported by SimpleQuery. On `EMailQA-Coref` also, CorefQuery reports the best performance of 13.49% compared to SimpleQuery's 11.28%, For both datasets, a +3.83% and +2.21% increase corroborates the impact of the coreference resolution system. The UnifiedQA baseline performs poorly on both datasets with an accuracy of just 0.17% and 0.2%. However, it is important to note that this model was not fine-tuned on either `EMailQA` dataset. Finally, the experiment results for each user are shown in Table 7.5. The table shows that for five out of six users, CorefQuery reports higher accuracy and, on average, shows an improvement of +15.05% over the SimpleQuery baseline.

[7] `https://www.ontotext.com/products/graphdb/`

Table 7.5. Comparison of the SimpleQuery and CorefQuery results with respect to users.

User	SimpleQuery	CorefQuery
Beck-s	**949** **84.13%**	855 75.79%
Dasovich-j	15556 61.61%	**15964** **63.22%**
Haedicke-m	1170 54.82%	**1815** **85.05%**
Lay-k	45 39.13%	**86** **74.78%**
Sager-e	237 25.76%	**370** **40.21%**
Skilling-j	11 36.66%	**16** **53.33%**
Average Accuracy	50.35%	**65.40%**

7.5 Error Analysis

In this section, an in-depth quantitative and qualitative error analysis is presented. First, the performance of the baselines for each template is analyzed. Table 7.6 reports these statistics for each template where each cell reports the count of the number of questions answered correctly and the accuracy for that baseline and template type. From the results, it can be seen that the increase in accuracy for T1 (+14.45%), T3 (+14.39%), and T4 (+7.86%) templates results in a better performance for the CorefQuery baseline. Also, for the 21,106 questions with zero complexity, SimpleQuery reports an accuracy of 80.75% and ComplexQuery an accuracy of 85.34%. The improvement of +4.59% highlights the impact of the coreference system to link mentions in the email header (to, from, cc, and bcc) correctly.

Next, Table 7.7 shows performance of the baselines on questions with complexity of 1, 2 and 3. Although CorefQuery demonstrates best performance, the low accuracy scores of 64.30% on `EMailQA-Full` and 12.93% on `EMailQA-Coref` shows a big room for improvement. Additionally, the inability of any baseline to answer a question with complexity 3

Table 7.6. QA experiment results per template in the `EMail-Full` dataset. T1, T2, T3, T4 and T5 are the templates used to create the questions.

Baseline	Correct				
	T1	**T2**	**T3**	**T4**	**T5**
UnifiedQA	0	0	0	53	0
	0%	0%	0%	16.01%	0%
SimpleQuery	120	**7817**	9654	151	**226**
	34.68%	**72.35%**	55.06%	45.61%	**30.54%**
CorefQuery	**171**	6390	**12177**	**178**	190
	49.42%	59.14%	**69.45%**	**53.77%**	25.67%

Table 7.7. Distribution of results based on the complexity of questions in `EMail-Coref`.

Baseline	Correct		
	1	**2**	**3**
UnifiedQA	18	0	0
	0.22%	0%	0%
SimpleQuery	951	25	0
	11.97%	3.85%	0%
CorefQuery	**1131**	**36**	0
	14.23%	**5.54%**	0%

emphasizes the need for a better coreference resolution system. The table also shows the poor performance of the UnifiedQA baseline on complex questions.

In the remaining part of the section, qualitative error analysis is performed. The error analysis presented attempts to narrow down error contributing factors. During the dataset creation process, 124 alias-attachment pairs were used to introduce complexity in the created questions. These pairs resulted in the creation of 6,661 questions across all templates except T5. A successful resolution of these 124 pairs is thus crucial in answering 22.38% questions in the dataset. A manual analysis revealed that of the 124 pairs, only 31 attachments had referring mentions, of which only 2 had the expected alias as the referring mention. Failure to successfully resolve attachments is thus one of the major contributing factors to the errors.

Treading along the same grounds we analyse how incorrectly chained email addresses[8] impact the T5 template questions (questions of the form - 'What is X's email address?'). In order to quantify this impact, the concept of *Transitive Chaining* is introduced. Let X, Y and Z be three entities and $m_1^x, m_2^x, .., m_{c1}^x$, $m_1^y, m_2^y, .., m_{c2}^y$ and $m_1^z, m_2^z, .., m_{c3}^z$ be the corresponding entity mentions. Here, $c1$, $c2$ and $c3$ are number of mentions for X, Y and Z entities. Now, X and Z are said to be *chained transitively* ($X - Y - Z$) if X and Y are chained together in one email thread, and Y and Z are chained together in another email thread. In the *transitive chain*, $X - Y - Z$, X and Z are the start and end of the chain. The length of a *transitive chain* is defined as the number of entities in between the start and end of the chain. A chain of length zero is a *direct chain*. A chain of length one or more is an *indirect chain*.

For an incorrectly answered T5 template question, analyzing the length of a transitive chain with the baseline answer and entity mention in the question should help us understand the spread of incorrect coreference chaining. Table 7.8 provides statistics for all incorrectly answered T5 questions in terms of transitive chain lengths. Out of all the incorrectly answered T5 template questions, no answer was found for 411 questions, and 42, 4, and 30 questions observed transitive chain lengths of 0, 1, and 2, respectively. The remaining 57 questions had a transitive chain length of 3 or more. These statistics show how incorrect coreference chaining clubbed with a compact representation like knowledge graphs can generate incorrect answers. The ability of knowledge graphs to link entities across email threads for given user results in creating transitive chains with non-zero length. However, this also shows that any improvement in coreference chaining will improve performance on the QA task.

Lastly, for the UnifiedQA baseline, this work believes the lack of structural knowledge (header-body-footer) and email addresses hampers the model's ability to answer the ques-

[8]Here chain is a coreference chain.

Table 7.8. Transitive chain length statistics for incorrectly answered T5 questions.

Transitive Chain Length	**Count**
0	42
1	4
2	30
3+	57
Empty Answer	411
Total Incorrect	554

tions. However, the keyword "from" does assist the model in identifying the sender for some T4 template questions.

This chapter concludes the last task of the dissertation concerning the knowledge extraction pipeline for email conversations. In this dissertation, each pipeline task was described in detail, starting with the background of the task, the task setup, the dataset used, the annotation steps carried out, experiments, results, and thorough error analysis. We investigated two novel tasks, created four datasets for three different tasks, proposed two models that demonstrate state-of-the-art performance on two datasets, and empirically showed the positive impact of our proposed model using a downstream application. Thus, the work presented in this dissertation manages to successfully undertake the realistic case presented in Chapter 1. After undertaking the realistic case, the next chapter summarizes the dissertation and discusses future research directions for the knowledge extraction problem.

CHAPTER 8

FUTURE WORK AND CONCLUSIONS

8.1 Dissertation Summary

Let us review this dissertation by discussing our three main contributions.

Contribution 1: Investigating two new tasks in email conversations for the first time.

This dissertation investigated the Entity coreference resolution problem in email conversations in a generic setting for the first time. It evaluated the problem (see Section 4.4) using a small manually annotated dataset - `SEED`. The evaluation showed that the problem is challenging and one that current state-of-the-art models perform poorly on. Post evaluation, the dissertation used `SEED` to create a large-scale weakly annotated dataset called `CEREC`. Experiments were performed on `CEREC` using different baselines and the observed errors were analysed to identify mention scoring as one of the significant limitations of SBERT. This dissertation then proposed a joint model to improve SBERT's mention scoring component. The model learned coreference resolution and span classification jointly and reported new state-of-the-art results on `CEREC` and `SEED`. The proposed JM2+S model achieved an improvement of 4.87% and 5.26% on the `CEREC` and `SEED` datasets respectively.

In Chapter 7, this dissertation investigated the Question answering task for email conversations in a novel setting. It formulated the problem as an interaction between a person and a digital voice assistant and, for the first time, evaluate the task in the new setting using two large datasets - `EMailQA-Full` and `EMailQA-Coref`. The dissertation evaluated datasets on three baselines - UnifiedQA, SimpleQuery, and CorefQuery. UnifiedQA is a deep learning baseline, and SimpleQuery and CorefQuery are SPARQL-based baselines. The SPARQL baselines used templates to generate queries for each

question. The results demonstrated the impact of adding coreference resolution to the knowledge extraction pipeline. The addition increased the accuracy of SimpleQuery and ComplexQuery by +3.91% and +2.22%, respectively.

Contribution 2: New datasets.

One of the significant hurdles this dissertation encountered while working on email conversations was the lack of publicly available annotated datasets. Thus, one of our contributions is that we have made two datasets publicly available and will be releasing two more soon. Making the datasets public would ensure that others can research the same topic without facing the same hurdles. The first dataset that to be released publicly was `SEED`. `SEED` was the first human-annotated dataset containing email conversations with named entity, mention, and coreference annotations. It was created using the Enron Email Corpus and includes 46 email threads with 866 coreference chains and 5,834 mentions. The dataset also introduced a new entity type - DIGITAL to include attachments and other digital entities in the email corpus. Next, the weakly annotated `CEREC` dataset was released. The dataset contains 6001 email threads, 38,996 coreference chains, and 422,081 annotated mentions. We first trained the SBERT model on `SEED` for the mention extraction task and then obtained mention annotations on `CEREC`. This was followed by training the SBERT model on `SEED` for the coreference resolution task and, with the mention annotations as input, obtained coreference annotations on `CEREC`.

The other two new datasets are `ECRA` and `EMailQA`. The `ECRA` dataset contains annotations for 11 relations (defined in Section 5.2), and four relations were manually annotated and seven relations automatically annotated. and contains annotations for 11 relations. It contains 762 email threads with 61,209 relation annotations. Next, in Chapter 7, we created `EMailQA`, a question-answering dataset containing two parts - `EmailQA-Full` and `EmailQA-Coref`. The dataset was created using five templates

that majorly focus on email attachments. Complexity was introduced on a scale of 0-3 in the dataset via name/email address/attachment aliases. `EmailQA-Full`, the entire dataset, contains 29,574 question-answer pairs. `EMailQA-Coref` contains 8,648 questions with a complexity level of one or more. Lastly, knowledge graphs were created for six users in the Enron Email Corpus, representing 762 email threads (Chapter 6). These knowledge graphs incorporate the knowledge extracted in the form of entities, mentions, and relations. This dissertation believes these knowledge graphs will be instrumental in research using knowledge graphs directly or via embeddings.

Contribution 3: Knowledge extraction pipeline for email conversations.

This work created a pipeline that focuses on three out of four phases and four tasks across these phases. The pipeline is the first one to focus on email threads compared to the previous works using email messages. The pipeline presented in this dissertation carries out entity extraction via coreference resolution, followed by relation extraction, and finally represents the entities and relations using a knowledge graph. The application of the knowledge graphs to question answering completes a holistic implementation of the pipeline. Thus, the knowledge extraction pipeline presented in this dissertation advances the frontiers of research in email conversations and provides a strong foundation for future work to extend the pipeline.

While this summary describes our progress on the knowledge extraction task, in many ways, our contributions in terms of datasets, models, and benchmarks barely scratch the surface. Next, this chapter suggests promising directions for future research.

8.2 Future Work

8.2.1 Short-to-medium term ideas

In Section 7.5, the impact of the coreference resolution system was highlighted. However, along with the impact, this dissertation also showed that improving the coreference system

will significantly impact the QA task. This room for improvement has also been discussed in Section 4.9. Thus, the possibility of an immediate impact makes entity coreference resolution an ideal task to focus on next. Two key directions for future work on the entity resolution task have been identified:

Improving span representations - The joint model proposed in Chapter 4 works towards improving the mention scoring component of the SBERT model. However, in the same chapter, the dissertation also identified that one of the drawbacks of the SBERT model is that it generates poor span representations. Gandhi et al. (2021) also highlight the problem and propose two loss functions to obtain richer span representations. The authors test the two loss functions on the medical notes dataset, released as a part of the i2b2/VA Shared-Task and Workshop in 2011 (Uzuner et al., 2011). Incorporating these two functions in `JM2+S`, using a different language model like CorefBERT (Ye et al., 2020), using named entity tags as features (Khosla and Rose, 2020), and using the section information feature (see 4.6.5) are few of the avenues we would like to pursue.

Coreference resolution as a text-to-text task - Raffel et al. (2020) create a single model for several NLP tasks by framing the tasks in a text-in text-out framework. Their proposed T5 model achieves state-of-the-art results on various benchmark datasets. The authors evaluate the T5 model on the WSC (Levesque et al., 2012) dataset and report an accuracy of 90.8 (using `T5-11B`). The WSC task, however, is different when compared to the coreference resolution task for email conversations. Firstly, we do not focus only on pronouns. Next, no mention is provided as input to the model, and thus, the model needs to carry out mention detection and coreference resolution. We can reformulate the coreference resolution task in the text-in text-out framework by either

using chain masks in the output for entities in the same chain or adding markers near entities in the text output (see Example 23).

Example 23. Example showing different text-to-text formulations of the CR task.

```
Text Input (Email message excerpt):
"Taft, Sheldon A." <SATaft@vssp.com> on 08/24/2000 10:47:18 AM
To: "‘bmerola@enron.com’" <bmerola@enron.com>, "‘pmikuls@wpsr.com’" <
    pmikuls@wpsr.com>, "‘bkorandovich@newenergy.com’" <bkorandovich@newenergy
    .com>, "‘mayer@taftlaw.com’" <mayer@taftlaw.com>, "‘kurt@theoec.org’" <
    kurt@theoec.org>


Text Output 1:
"[0]" <[0]> on 08/24/2000 10:47:18 AM
To: "‘[1]’" <[1]>, "‘[2]’" <[2]>, "‘[3]’" <[3]>, "‘[4]’" <[4]>, "‘[4]’"
    <[4]>


Text Output 2:
"Taft, Sheldon A.[0]" <SATaft@vssp.com[0]> on 08/24/2000 10:47:18 AM
To: "‘bmerola@enron.com[1]’" <bmerola@enron.com[1]>, "‘pmikuls@wpsr.com[2]’"
     <pmikuls@wpsr.com[2]>, "‘bkorandovich@newenergy.com[3]’" <
    bkorandovich@newenergy.com[3]>, "‘mayer@taftlaw.com[4]’" <mayer@taftlaw.
    com[4]>, "‘kurt@theoec.org[5]’" <kurt@theoec.org[5]>
```

Improving upon the current state of the datasets released as part of the work presented in this dissertation is another essential task. Section 7.5 highlights the failure of the SBERT model to recognize attachments. One way to improve this performance is by increasing the annotated attachment count in CEREC. We identify using named entity (NE) extraction for enriching CEREC as a possible solution. The email threads annotated with relations in Chapter

5 and the SEED dataset could be used to train a NE extraction model to add attachment mentions to CEREC. The addition should enhance the attachment recognition capabilities of models trained on CEREC. Furthermore, we can also use techniques like bootstrapping to improve the quality of annotations in CEREC.

The second dataset that we would like to improve is EMailQA. Currently, the questions in the dataset focus only on six users. Our primary direction in improving EMailQA is to incorporate all 131 users present in CEREC. This inclusion would increase the number of questions in both EMailQA-Full and EmailQA-Coref, thereby increasing the complexity of the task. The second direction we would like to pursue is to add questions that do not focus on attachments. Alrashed et al. (2018) in their work show that the top reason for an email revisit is to lookup 'instructions to perform a certain task.' Adding questions that focus on that topic would introduce another level of complexity to the dataset. The approach used by Yang et al. (2018) would be an excellent foundation to start upon in this direction. Finally, we would also like to explore work done on converting SPARQL queries to natural language questions. This would enable us to incorporate paraphrased versions of the template questions thereby increasing the complexity of the QA task.

The final short-to-medium term idea that we would like to work upon is to introduce Entity Linking to the knowledge extraction pipeline. Traditionally, Entity Linking is considered as a combination of two tasks: Named Entity Recognition and Disambiguation. The addition of world knowledge or common knowledge by linking entities to larger knowledge bases has proved beneficial to many machine learning tasks (Marino et al., 2016; Annervaz et al., 2018; Ostendorff et al., 2019; Li et al., 2019). The primary motivation behind the addition of world knowledge is to induce an inference or deduction process similar to how a human would perform the same task. For example, Marino et al. (2016) show how descriptive information about an animal can assist in correctly identifying the animal in an image.

We propose linking entities found during the coreference resolution phase with the corresponding entities in knowledge graphs like Wikidata or DBPedia. We hypothesize that incorporating additional knowledge captured in these large knowledge graphs improves performance in downstream tasks like email classification, question answering, or summarization. Entity linking would also help explore the addition of questions to EMailQA that focus on a broader context or a context beyond the email conversations. The addition of a single triple using the predicate `sameAs`[1] and the nodes representing *Kenneth Lay* in both *KG-Coref* and *DBPedia* would introduce a plethora of additional knowledge to the created graphs. Figure 8.1 shows a section of DBPedia knowledge graph for the entity *Kenneth Lay*. The information available in DBPedia is extensive and unlikely to be present in the Enron Email Corpus.

8.2.2 Medium-to-long term ideas

One of the two long-term ideas that we would like to pursue is the application of 'Game Theory' to perform coreference resolution. Recently, there has been an increasing interest in using game theory as a different tool to solve various NLP problems. The work of Saxena et al. (2020) was vital in motivating us to weigh this idea thoroughly. We propose framing coreference resolution as a multiplayer coalition game where each mention is a player. The game's goal is to form coalitions of players that maximize the total utility achieved by everyone. The final set of coalitions are the coreference clusters in the document. One of the crucial elements of this formulation is defining the utility function. On this front, we identify simplifying coreference resolution as a clustering task and evaluating previous works (Dhamal et al., 2012; Huo et al., 2017; Hassine et al., 2017; Bure and Staroverova, 2019; Sulistyo et al., 2019; Afsar et al., 2019; Georgakopoulos et al., 2020) that explore using colation games for clustering.

[1] `https://www.w3.org/TR/owl-ref/#sameAs-def`

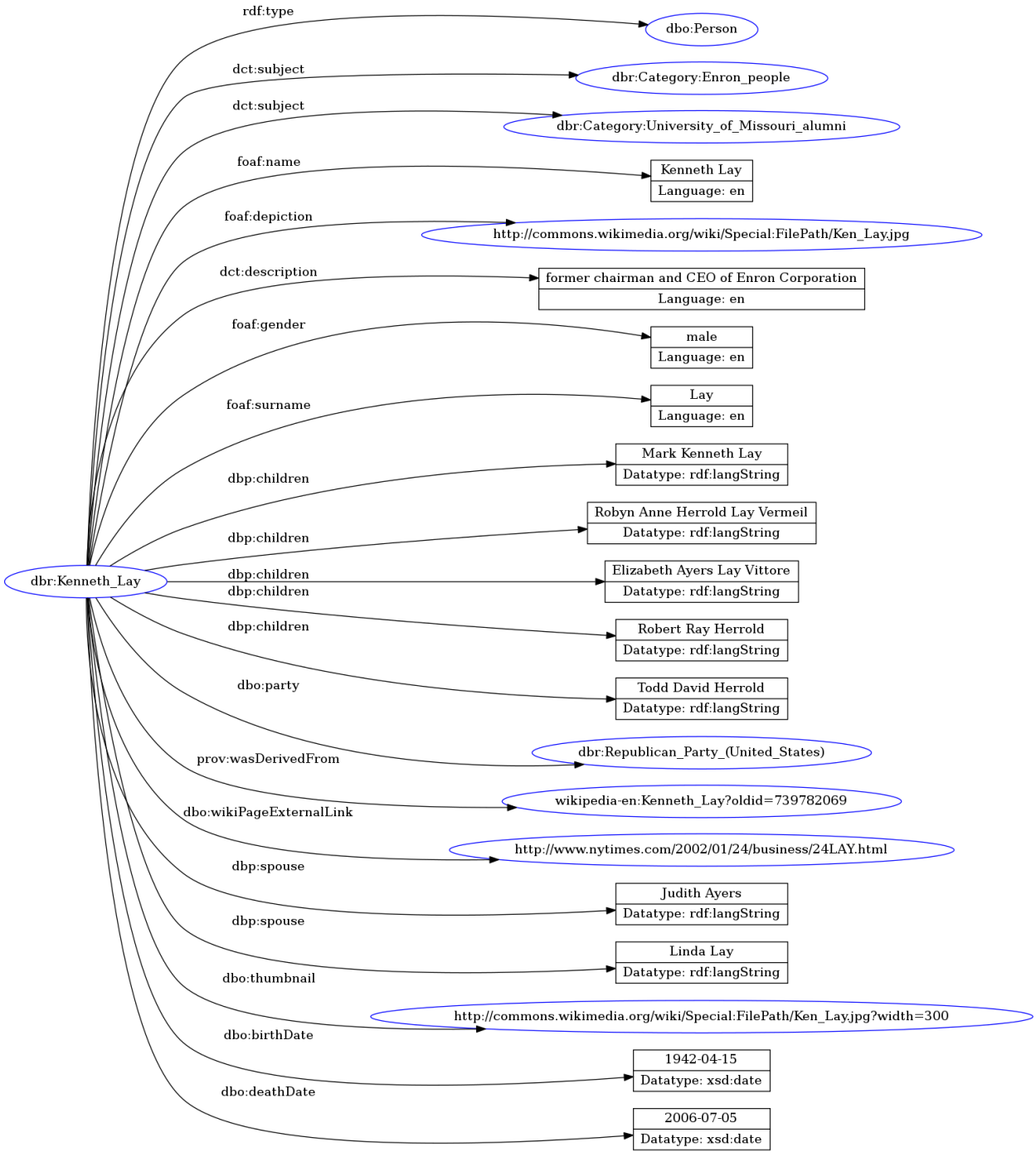

Figure 8.1. Example of a DBPedia knowledge graph section for an entity found in the Enron Corpus

The second long-term idea that we believe would significantly improve the knowledge extraction pipeline is the addition of end-to-end *Event coreference resolution*. We can follow an approach similar to the one taken for entity coreference resolution. Annotating a small corpus with event mentions and the corresponding coreference chain can help evaluate the current state-of-the-art models and help identify the limitations of the current models. Furthermore, improvement in the entity coreference resolution task can facilitate evaluating joint models for event coreference resolution.

Lastly, we would like to explore other downstream tasks like email classification, intent recognition, or summarization. It would be interesting to quantify the impact of various components in the pipeline on different downstream tasks and to evaluate using knowledge graphs on the same tasks.

8.3 Conclusions

Email communication is the exchange of messages by two or more people over the internet via email. Amidst the tremendous increase in social media usage, emails still play a vital role as a communication medium today. Past works have focused on various problems in emails using private email message datasets. These works provide meaningful insight, analysis, and solutions to various email tasks. However, the datasets, being private, result in a lack of reproducibility and comparability. Also, the focus of research in email processing has been primarily on email messages, with work on email conversations directed towards thread reconstruction or social network analysis. These tasks fail to capture the entity interactions in an email conversation.

Off-the-shelf document processing pipelines are capable of processing a myriad of document formats extracting different forms of knowledge. Such a pipeline for extracting knowledge from email conversations can shift the research focus from extracting the knowledge to

consuming the extracted knowledge. However, creating such a pipeline is challenging due to the absence of trained models for various email conversation tasks.

This dissertation implemented a relevant and realistic case of the knowledge extraction pipeline. It examined two novel tasks for email conversations, created five new annotated datasets and two new benchmarks. In doing so, it significantly contributes to the lack of publicly available datasets and benchmarks for email conversation tasks. As described earlier in this chapter, there are many open issues. But, we deem that this dissertation illustrates how knowledge can be extracted using a pipeline. Additionally, it also shows how the extracted knowledge benefits question answering.

APPENDIX A

ADDITIONAL EXPERIMENT RESULTS

Tables A.1 and A.2 show `PHASE II` experiments results for all runs and folds. A detailed description of these experiments can be found in 4.8.

Table A.1. `PHASE II` experiment results for all runs on SD, CD and OD.

Model	Run	MUC			B^3			CEAFE			Avg.
		P	R	F1	P	R	F1	P	R	F1	**F1**
ExSeed											
JM2	1	82.3	61.3	70.28	66.2	44.7	53.39	60.9	30.6	40.79	54.85
	2	82.6	62.9	71.5	65.7	48.3	55.7	64.3	29.4	40.37	55.85
	3	82.6	62.5	71.17	69.1	46.6	55.69	61.6	31.8	42.01	56.29
	4	83.3	62.1	71.2	69.7	44.3	54.24	58.2	31.7	41.12	55.52
	5	81.2	64.5	71.94	64.8	49.6	56.23	64.6	31.8	42.7	**56.95**
JM2+S	1	82.3	61.3	70.28	68.8	50.2	58.14	48.8	56.5	52.4	60.27
	2	82.6	62.9	71.5	68.5	53.4	60.05	51.6	53.7	52.71	61.42
	3	82.6	62.5	71.17	71.1	51.9	60.03	49.2	56.5	52.65	61.28
	4	83.3	62.1	71.2	71.5	50.2	59	47	58.7	52.21	60.80
	5	81.2	64.5	71.94	66.7	54.3	59.88	54.5	55.1	54.83	**62.21**
CEREC											
JM2	1	88.4	60.8	72.06	74.4	41.8	53.57	65.2	32.5	43.47	56.36
	2	84.2	64.9	73.33	65.9	48	55.57	67.7	32.3	43.79	**57.56**
	3	86.1	63.8	73.37	69.1	45.5	54.92	66	32.6	43.65	57.31
JM2+S	1	88.4	60.8	72.06	75.9	45.4	56.83	55.3	49.7	52.41	60.43
	2	84.2	64.9	73.33	67.6	51.3	58.3	59.62	49.4	54.1	61.93
	3	86.1	63.8	73.37	70.7	50	58.61	55.5	55	55.31	**62.43**
OntoNotes											
SBERT	1	85.1	75	79.76	76.1	63.3	69.18	72.3	57.8	64.28	71.07
	2	84.3	76.7	80.37	74.2	65.3	69.46	72.6	58.6	64.91	71.58
	3	83.2	78	80.56	73.2	67.3	70.18	71.5	60.1	65.36	**72.03**
JM2	1	84.1	78.5	81.21	73.8	67.9	70.77	72.8	61.9	66.97	72.98
	2	84.1	79.7	81.9	76.2	70.4	73.22	73.7	65.4	69.34	**74.82**
	3	84.8	78.9	81.78	77.1	68.7	72.69	72.7	65.4	68.89	74.45
JM2+S	1	84.1	78.5	81.21	69.9	69.3	69.67	51.2	68.1	58.52	69.8
	2	84.1	79.7	81.9	72.6	71.5	72.11	55.4	70.6	62.13	**72.05**
	3	84.8	78.9	81.78	72.8	70	71.41	52	70.7	59.97	71.05

Table A.2. `PHASE II` experiment results for all folds on LD.

Model	Fold	MUC			B^3			CEAFE			Avg.
		P	R	F1	P	R	F1	P	R	F1	**F1**
SBERT	1	90	82.2	85.95	72.7	51.6	60.42	65.3	18.2	28.48	58.28
	2	92	87.3	89.62	79	57.4	66.49	67	15.7	25.48	60.53
	3	89.2	87.5	88.41	73	59.7	65.71	61.2	17.8	27.68	60.6
	4	90.4	86.2	88.3	78.4	57.6	66.46	66.4	17.6	27.86	60.87
	5	88.2	87.2	87.74	72	63.6	67.57	66.7	20.3	31.24	62.18
	6	87.7	86.4	87.08	62.8	56.1	59.28	64.5	16.9	26.82	57.72
	7	91.6	86.9	89.22	77.6	62.6	69.33	67.6	18.1	28.57	62.37
	8	90.1	87	88.58	67.8	61.8	64.73	64.3	17.1	27.13	60.14
	9	89.2	86.3	87.78	78.2	56.1	65.33	63	14.6	23.81	58.97
	10	86.2	86.8	86.56	66.5	59.2	62.69	56.5	16.5	25.6	58.28
JM2	1	89.1	82.4	85.63	66.8	54.9	60.3	66.4	17.7	27.97	57.96
	2	90.5	87.7	89.09	76.8	59.3	66.95	65.4	16.8	26.8	60.94
	3	90.7	85.8	88.22	73.9	53.2	61.87	62.8	17.3	27.2	59.09
	4	89.7	86.6	88.17	76.7	59.6	67.09	69.1	17.9	28.55	61.27
	5	88.3	87.4	87.89	75	63.6	68.89	69.5	20.8	32.11	62.96
	6	87.3	85.9	86.67	68.9	53.8	60.49	65.4	19	29.52	58.89
	7	91.2	87.5	89.32	76.9	63.6	69.64	67.4	19.5	30.31	63.09
	8	88	87.3	87.88	65.4	60.4	62.83	61.9	17.8	27.69	59.46
	9	89.8	86.1	87.97	78.7	53.9	63.99	64.3	15.9	25.61	59.19
	10	86.9	88.2	87.58	64.2	60.3	62.21	63.1	17.9	27.97	59.25
JM2+S	1	89.1	82.4	85.63	69.4	69	69.23	62.7	66	64.33	73.06
	2	90.5	87.7	89.09	77.1	75.6	76.39	64.3	70.4	67.26	77.58
	3	90.7	85.8	88.22	72.7	68	70.35	53.2	73.7	61.85	73.47
	4	89.7	86.6	88.17	76.2	74.7	75.5	62.8	68.2	65.43	76.36
	5	88.3	87.4	87.89	74.1	76.7	75.43	61.8	68.7	65.13	76.15
	6	87.3	85.9	86.67	71.6	68.4	70.01	66.8	68.1	67.5	74.72
	7	91.2	87.5	89.32	76.5	75.4	76.01	58.3	69.7	63.53	76.28
	8	88	87.3	87.88	67.1	71.5	69.27	58.4	69.9	61.06	72.73
	9	89.8	86.1	87.97	78.3	73.8	76	62.7	77.2	69.25	77.74
	10	86.9	88.2	87.58	66.3	71.4	68.78	64.7	61.2	62.94	73.1

APPENDIX B

SAMPLE EMAILQA QUESTIONS

Table B.1. Examples of questions with complexity level 3.

Template	Original Question	Updated Question
T3	Did sullo, sharon e send the Bankruptcies . ppt to tribolet, michael?	Did sharon send the draft of the Bankruptcy Presentation that Rick Buy will present to the Management Committee on Monday morning to michael?
	Answer: **Yes**	
	Did keohane, peter send the 'memo . bowen . debt funding to Enron Corp . doc' to bowen jr., raymond?	Did peter send the memo to ray bowen?
	Answer: **Yes**	
	Did janice r moore send the 'ENRON - AM . pdf' to edward sacks?	Did janice send the revised cover sheet that includes everything plus credit terms to ed sacks?
	Answer: **Yes**	
	Did thapar, raj send the 'Info Request _ Ene _ Jan 11 _ 021 . xls' to wilson, shona?	Did raj send the names of EWS lead persons name to shona?
	Answer: **Yes**	
	Did rachel mcmahon send the 'Sept 11 to Governor re contracts . doc' to jack pigott?	Did mcmahon send the revised version of CDWR power contracts to jack?
	Answer: **Yes**	
	Did elliott, lexi send the 'UT undergrad school summary . xls' to causey, richard?	Did lexi send the brief campus update , including all previous and future events for the fall recruiting season to causey?
	Answer: **Yes**	

Table B.2. Examples of questions with complexity level 2.

Template	Original Question	Updated Question
T1	When did ben coes send the 'ISDA . ppt'?	When did ben send the copy of our Power Point presentation?
	Answer: **Monday , March 12 , 2001 11 : 28 AM**	
	When did lauren goldblatt send the 'corp _ resp _ _ 20 _ Sept _ [2] . doc'?	When did lauren send the fact sheet as well as the revised principles?
	Answer: **10 / 25 / 2000 05 : 44 PM**	
	When did katie kaplan send the 'Staff MMP . pdf - MMP Notice . pdf'?	When did katie send the description of issues , resources , budget and a straw man proposal for decision - making procedures?
	Answer: **Tuesday , March 06 , 2001 6 : 45 PM**	
T2	Did vicki.sharp@enron.com receive the 'corp _ resp _ _ 20 _ Sept _ [2] . doc'?	Did vicki receive the fact sheet as well as the revised principles?
	Answer: **Yes**	
	Did tribolet, michael receive the 'New Distressed CP V3 . xls'?	Did michael receive the detail to support the # of bankruptcies by business unit?
	Answer: **Yes**	
T3	Did drumheller, robert b. send the 'bndes letter (10 - 30 - 00) c . doc' to orlando gonzalez?	Did drumheller, robert b. send the first draft of the BNDES letter to orlando?
	Answer: **Yes**	
	Did stephen littlechild send the list of items we recently discussed in our Trading status meeting to mona petrochko?	Did littlechild send the list of items we recently discussed in our Trading status meeting to mona?
	Answer: **No**	
	Did mike d smith send 'sce 011005 . xls' to sandra mccubbin?	Did mds send 'sce 011005 . xls' to sandi mccubbin?
	Answer: **No**	

Table B.3. Examples of questions with complexity level 1.

Template	Original Question	Updated Question
T1	When did ruiz, annie send the 'SDG & E Reply re Wetzel (v1) . DOC'?	When did ruiz, annie send the updated information requested by ALJ Wetzell in his Ruling?
	Answer: **Friday , August 17 , 2001 8 : 31 AM**	
	When did mday send the 'X28655 . DOC'?	When did mday send the settlement sheet on the one cent surcharge?
	Answer: **Friday , October 19 , 2001 3 : 46 PM**	
T2	Did tribolet, michael receive the 'PG & E PX Credit Calculation . doc'?	Did tribolet, michael receive the summary of PG & E ' s notes on how they calculate the PX Credit?
	Answer: **Yes**	
	Did ken_pietrelli@ocli.com receive the 'Estate Org Chart'?	Did pietrelli receive the 'Estate Org Chart'?
	Answer: **No**	
T3	Did drumheller, robert b. send the 'bndes letter (10 - 30 - 00) c . doc' to john.novak@enron.com?	Did drumheller, robert b. send the first draft of the BNDES letter to john.novak@enron.com?
	Answer: **Yes**	
	Did rivera, nancy a. send the 'Closing Memo (11 - 1 - 01) . doc' to brett.r.wiggs@enron.com?	Did rivera, nancy a. send the memo we promised that describes the major Eletrobolt pending issues to brett.r.wiggs@enron.com?
	Answer: **Yes**	
T4	Who sent the 'finalagenda . doc'?	Who sent the agenda?
	Answer: **liz o' sullivan**	
	Who sent 'Cmpltnss & AccrcyJS . doc'?	Who sent the list of items we recently discussed in our Trading status meeting?
	Answer: **tom bauer**	
T5	What is anshuman srivastav's email address?	What is anshuman's email address?
	Answer: **anshuman.srivastav@enron.com**	
	What is jane allen's email address?	What is jane's email address?
	Answer: **jane.allen@enron.com**	

APPENDIX C

SPARQL TEMPLATES

C.1 SimpleQuery

```
Question Template: When did X send A?
Query Template:
PREFIX nco: <http://www.semanticdesktop.org/ontologies/2007/03/22/nco#>
PREFIX nmo: <http://www.semanticdesktop.org/ontologies/2007/03/22/nmo#>
PREFIX nfo: <http://www.semanticdesktop.org/ontologies/2007/03/22/nfo#>
select ?date where {
    VALUES (?nametype) {
        (nco:fullname)
        (nco:nickname)
        (nco:emailAddress)
    }
    ?emailmessage nmo:messageFrom ?person .
    ?person ?nametype "X" .
    ?emailmessage nmo:sentDate ?date .
    ?emailmessage nmo:hasAttachment ?attachment .
    ?attachment nfo:fileName "A" .
}

Question Template: Did X receive A?
Query Template:
PREFIX nco: <http://www.semanticdesktop.org/ontologies/2007/03/22/nco#>
PREFIX nmo: <http://www.semanticdesktop.org/ontologies/2007/03/22/nmo#>
PREFIX nfo: <http://www.semanticdesktop.org/ontologies/2007/03/22/nfo#>
```

```
ask where {
    VALUES (?nametype) {
        (nco:fullname)
        (nco:nickname)
        (nco:emailAddress)
    }
    VALUES (?recipienttype) {
        (nmo:emailTo)
        (nmo:emailCc)
        (nmo:emailBcc)
    }
    ?emailmessage ?recipienttype ?person .
    ?person ?nametype "X" .
    ?emailmessage nmo:hasAttachment ?attachment .
    ?a nfo:fileName "A" .
}
```

Question Template: Did X send A to Z?

Query Template:

```
PREFIX nco: <http://www.semanticdesktop.org/ontologies/2007/03/22/nco#>
PREFIX nmo: <http://www.semanticdesktop.org/ontologies/2007/03/22/nmo#>
PREFIX nfo: <http://www.semanticdesktop.org/ontologies/2007/03/22/nfo#>
ask where {
    VALUES (?nametype) {
        (nco:fullname)
        (nco:nickname)
        (nco:emailAddress)
    }
```

```
    VALUES (?recipienttype) {
        (nmo:emailTo)
        (nmo:emailCc)
        (nmo:emailBcc)
    }
    ?emailmessage ?recipienttype ?recipient .
    ?recipient ?nametype "Y" .
    ?emailmessage nmo:messageFrom ?sender .
    ?sender ?name "X" .
    ?emailmessage nmo:hasAttachment ?attachment .
    ?attachment nfo:fileName "A" .
}
```

Question Template: Who sent A?

Query Template:

```
PREFIX nco: <http://www.semanticdesktop.org/ontologies/2007/03/22/nco#>
PREFIX nmo: <http://www.semanticdesktop.org/ontologies/2007/03/22/nmo#>
PREFIX nfo: <http://www.semanticdesktop.org/ontologies/2007/03/22/nfo#>
select ?sendername where {
    VALUES (?nametype) {
        (nco:fullname)
        (nco:nickname)
        (nco:emailAddress)
    }
    ?emailmessage nmo:messageFrom ?sender .
    ?sender ?nametype ?sendername .
    ?emailmessage nmo:hasAttachment ?attachment .
    ?attachment nfo:fileName "A" .
```

```
}
```

Question Template: What is X's email address?

Query Template:

```
PREFIX nco: <http://www.semanticdesktop.org/ontologies/2007/03/22/nco#>
PREFIX nmo: <http://www.semanticdesktop.org/ontologies/2007/03/22/nmo#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
select ?emailaddress where {
    VALUES (?nametype) {
        (nco:fullname)
        (nco:nickname)
    }
    ?person rdf:type nco:PersonContact .
    ?person nco:hasPersonName ?personname .
    ?personname ?nametype "X" .
    ?person nco:hasEmailAddress ?personemailaddress .
    ?personemailaddress nco:emailAddress ?emailaddress .
}
```

C.2 CorefQuery

Question Template: When did X send A?

Query Template:

```
PREFIX nco: <http://www.semanticdesktop.org/ontologies/2007/03/22/nco#>
PREFIX nmo: <http://www.semanticdesktop.org/ontologies/2007/03/22/nmo#>
PREFIX nfo: <http://www.semanticdesktop.org/ontologies/2007/03/22/nfo#>
PREFIX kefec: <http://example.org/kefec/ontology#>
select ?date where {
```

```
VALUES (?contactmedium) {
    (nco:hasPersonName)
    (nco:hasEmailAddress)
}
?email nmo:messageFrom ?sender .
?person ?contactmedium ?sender .
?email nmo:sentDate ?date .
?email nmo:hasAttachment ?attach .
{
    VALUES (?nametype) {
        (nco:fullname)
        (nco:nickname)
        (nco:emailAddress)
    }
    ?sender ?nametype "X" .
}
UNION
{
    ?person kefec:referredBy ?ref .
    ?ref kefec:referringText "X" .
}
{
    ?attach nfo:fileName "A" .
}
UNION
{
    ?attach kefec:referredBy ?ref1 .
    ?ref1 kefec:referringText "A" .
```

```
    }
}

Question Template: Did X receive A?
Query Template:
PREFIX nco: <http://www.semanticdesktop.org/ontologies/2007/03/22/nco#>
PREFIX nmo: <http://www.semanticdesktop.org/ontologies/2007/03/22/nmo#>
PREFIX nfo: <http://www.semanticdesktop.org/ontologies/2007/03/22/nfo#>
PREFIX kefec: <http://example.org/kefec/ontology#>
ask where {
    VALUES (?recipienttype) {
        (nmo:emailTo)
        (nmo:emailCc)
        (nmo:emailBcc)
    }
    VALUES (?contactmedium) {
        (nco:hasPersonName)
        (nco:hasEmailAddress)
    }
    ?email ?recipienttype ?recipient .
    ?person ?contactmedium ?recipient .
    {
        VALUES (?nametype) {
            (nco:fullname)
            (nco:nickname)
            (nco:emailAddress)
        }
        ?sender ?nametype "X" .
```

```
    }
    UNION
    {
        ?person kefec:referredBy ?ref .
        ?ref kefec:referringText "X" .
    }
    ?email nmo:hasAttachment ?attach .
    {
        ?attach nfo:fileName "A" .
    }
    UNION
    {
        ?attach kefec:referredBy ?ref1 .
        ?ref1 kefec:referringText "A" .
    }
}
```

Question Template: Did X send A to Z?

Query Template:

```
PREFIX nco: <http://www.semanticdesktop.org/ontologies/2007/03/22/nco#>
PREFIX nmo: <http://www.semanticdesktop.org/ontologies/2007/03/22/nmo#>
PREFIX nfo: <http://www.semanticdesktop.org/ontologies/2007/03/22/nfo#>
PREFIX kefec: <http://example.org/kefec/ontology#>
ask where {
    VALUES (?recipienttype) {
        (nmo:emailTo)
        (nmo:emailCc)
        (nmo:emailBcc)
```

```
}
VALUES (?rcontactmedium) {
    (nco:hasPersonName)
    (nco:hasEmailAddress)
}
?email ?recipienttype ?recipient .
?rperson ?rcontactmedium ?recipient .
{
    VALUES (?recipientnametype) {
        (nco:fullname)
        (nco:nickname)
        (nco:emailAddress)
    }
    ?recipient ?recipientnametype "Y" .
}
UNION
{
    ?rperson kefec:referredBy ?ref .
    ?ref kefec:referringText "Y" .
}
VALUES (?scontactmedium) {
    (nco:hasPersonName)
    (nco:hasEmailAddress)
}
?email nmo:messageFrom ?sender .
?sperson ?scontactmedium ?sender .
{
    VALUES (?sendertype) {
```

```
            (nco:fullname)
            (nco:nickname)
            (nco:emailAddress)
        }
        ?sender ?sendertype "X" .
    }
    UNION
    {
        ?sperson kefec:referredBy ?ref1 .
        ?ref1 kefec:referringText "X"
    }
    ?email nmo:hasAttachment ?attach .
    {
        ?attach nfo:fileName "A" .
    }
    UNION
    {
        ?attach kefec:referredBy ?ref2 .
        ?ref2 kefec:referringText "A" .
    }
}
```

Question Template: Who sent A?

Query Template:

```
PREFIX nco: <http://www.semanticdesktop.org/ontologies/2007/03/22/nco#>
PREFIX nmo: <http://www.semanticdesktop.org/ontologies/2007/03/22/nmo#>
PREFIX nfo: <http://www.semanticdesktop.org/ontologies/2007/03/22/nfo#>
PREFIX kefec: <http://example.org/kefec/ontology#>
```

```
select ?person where {
    VALUES (?nametype) {
        (nco:fullname)
        (nco:nickname)
        (nco:emailAddress)
    }
    ?email nmo:messageFrom ?sender .
    ?sender ?nametype ?person .
    ?email nmo:hasAttachment ?attach .
    {
        ?attach nfo:fileName "A" .
    }
    UNION
    {
        ?attach kefec:referredBy ?ref .
        ?ref1 kefec:referringText "A" .
    }
}
```

Question Template: What is X's email address?

Query Template:

```
PREFIX nco: <http://www.semanticdesktop.org/ontologies/2007/03/22/nco#>
PREFIX nmo: <http://www.semanticdesktop.org/ontologies/2007/03/22/nmo#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX kefec: <http://example.org/kefec/ontology#>
select ?email where {
    ?person rdf:type nco:PersonContact .
    {
```

```
        VALUES (?nametype) {
            (nco:fullname)
            (nco:nickname)
        }
        ?person nco:hasPersonName ?pname .
        ?pname ?nametype "X" .
    }
    UNION
    {
        ?person kefec:referredBy ?ref .
        ?ref kefec:referringText "X" .
    }
    ?person nco:hasEmailAddress ?pemail .
    ?pemail nco:emailAddress ?email .
}
```

APPENDIX D

RELATION EXTRACTION EXAMPLE

Example 24. Example showing the output of the relation extraction process on a sample email message.

```
Original Email Thread Excerpt:
"Taft, Sheldon A." <SATaft@vssp.com> on 08/24/2000 10:47:18 AM
To: "'bmerola@enron.com'" <bmerola@enron.com>, "'pmikuls@wpsr.com'"
<pmikuls@wpsr.com>, "'bkorandovich@newenergy.com'"
<bkorandovich@newenergy.com>, "'mayer@taftlaw.com'" <mayer@taftlaw.com>,
"'kurt@theoec.org'" <kurt@theoec.org>
cc: "Petricoff, M. Howard" <MHPetricoff@vssp.com>
Subject: FW: Tech. Req. for Single & 3 Phase, 8-15-00.DOC

Here are the Technical Requirements proposed by the utilities at the August
23 PUCO Workshop on Interconnection. Please have your technical people
review these and share with us any issues or problems that marketers would
have with them. We will need to identify these issues and problems and to
propose alternatives before the next workshop meeting on August 30.

-----Original Message-----
From: Colbert, Paul [mailto:pcolbert@Cinergy.com]
Sent: Thursday, August 24, 2000 10:40 AM
To: Taft, Sheldon A.
Subject: Tech. Req. for Single & 3 Phase, 8-15-00.DOC

 <<Tech. Req. for Single & 3 Phase, 8-15-00.DOC>> Here it is. Thank you.
```

From the law offices of Vorys, Sater, Seymour and Pease LLP.

CONFIDENTIALITY NOTICE: This e-mail message is intended only for the person or entity to which it is addressed and may contain confidential and/or privileged material. Any unauthorized review, use, disclosure or distribution is prohibited. If you are not the intended recipient, please contact the sender by reply e-mail and destroy all copies of the original message. If you are the intended recipient but do not wish to receive communications through this medium, please so advise the sender immediately.

- Tech. Req. for Single & 3 Phase, 8-15-00.DOC

JSON Representation:


```
{
    "4": {
        "header-text": [
            "\" Taft , Sheldon A . \" < SATaft @ vssp . com > on 08 / 24 / 2000 10
                 : 47 : 18 AM",
            "To : \" ' bmerola @ enron . com ' \" < bmerola @ enron . com > , \" '
                 pmikuls @ wpsr . com ' \" < pmikuls @ wpsr . com > , \" '
                bkorandovich @ newenergy . com ' \" < bkorandovich @ newenergy .
                com > , \" ' mayer @ taftlaw . com ' \" < mayer @ taftlaw . com > ,
                 \" ' kurt @ theoec . org ' \" < kurt @ theoec . org >",
            "cc : \" Petricoff , M . Howard \" < MHPetricoff @ vssp . com >",
            "Subject : FW : Tech . Req . for Single & 3 Phase , 8 - 15 - 00 . DOC"
        ],
        "to": [
```

```
        "bmerola@enron.com",
        "pmikuls@wpsr.com",
        "bkorandovich@newenergy.com",
        "mayer@taftlaw.com",
        "kurt@theoec.org"
    ],
    "from": [
        "Taft, Sheldon A.",
        "SATaft@vssp.com"
    ],
    "cc": [
        "Petricoff, M. Howard",
        "MHPetricoff@vssp.com"
    ],
    "date-time": "08 / 24 / 2000 10 : 47 : 18 AM",
    "subject": "Tech . Req . for Single & 3 Phase , 8 - 15 - 00 . DOC"
    "body-text": [
        "Here are the Technical Requirements proposed by the utilities at the
            August 23 PUCO Workshop on Interconnection .",
        "Please have your technical people review these and share with us any
            issues or problems that marketers would have with them .",
        "We will need to identify these issues and problems and to propose
            alternatives before the next workshop meeting on August 30 ."
    ]
    "reply-to": "5"
},
"5": {
    "header-text": [
```

```
    "- - - - - Original Message - - - - -",
    "From : Colbert , Paul [ mailto : pcolbert @ Cinergy . com ]",
    "Sent : Thursday , August 24 , 2000 10 : 40 AM",
    "To : Taft , Sheldon A .",
    "Subject : Tech . Req . for Single & 3 Phase , 8 - 15 - 00 . DOC"
],
"to": [
    "Taft, Sheldon A."
],
"from": [
    "Colbert, Paul",
    "pcolbert@Cinergy.com"
],
"datetime": "Thursday , August 24 , 2000 10 : 40 AM",
"subject": "Tech. Req. for Single & 3 Phase, 8-15-00.DOC"
"body-text": [
    "< < Tech . Req . for Single & 3 Phase , 8 - 15 - 00 . DOC > >",
    "Here it is .",
    "Thank you .",
    "From the law offices of Vorys , Sater , Seymour and Pease LLP .",
    "- Tech . Req . for Single & 3 Phase , 8 - 15 - 00 . DOC"
],
"attachments": [
    "Tech. Req. for Single & 3 Phase, 8-15-00 . DOC"
]
"footer-text": [
```

```
            "CONFIDENTIALITY NOTICE : This e - mail message is intended only for
               the person or entity to which it is addressed and may contain
               confidential and / or privileged material .",
            "Any unauthorized review , use , disclosure or distribution is
               prohibited .",
            "If you are not the intended recipient , please contact the sender by
               reply e - mail and destroy all copies of the original message .",
            "If you are the intended recipient but do not wish to receive
               communications through this medium , please so advise the sender
               immediately .",
            "- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
               - - - -"
        ]
    }
}
```

APPENDIX E

TACRED RELATIONS

Table E.1 lists all 41 relations present in the TACRED corpus along with if they are part of the relation set considered for this dissertation. The table also contains an example[1] for each relation. In every example, the entities between which the relation holds have been italicized.

Table E.1. TACRED relation details

Relation Name	**Example**	**Selected**
per:date_of_birth	the Jan. 1 anniversary of *Williams*' death, and on his *Sept. 17* birthday	No
per:titles	*Thomas Edward Lehman* is an American professional *golfer*	Yes
org:top_members/employees	*US CDC* director *Julie Gerberding*	Yes
org:country_of_headquaters	*CSS* is a *Brazilian* rock band from São Paulo.	Yes
per:parents	*Hank* lived in Georgiana in the mid 1930's with his mother, *Lillie Williams*	Yes
per:age	*Michael Jackson* died June 25 2009 at the age of *50*	No
per:countries_of_residence	*Finland*'s *Nurmi* won his nine golds in the 1920's	Yes
per:children	*Williams*' son, *Hank Williams Jr*	Yes
org:alternate_names	*Harkat-ul-Mujahideen*, or *Movement of Holy Warriors*	Yes
per:charges	President *Abdurrahaman Wahid*, a step that could lead to his impeachment over alleged involvement in two *corruption* scandals	Yes
per:cities_of_residence	The *Gore* family resides in *Nashville*, Tennessee	Yes
per:origin	*Del Ponte*, a *Swiss* citizen	Yes

[1]Additional information for each relation can be found at `https://tac.nist.gov/2012/KBP/task_guidelines/TAC_KBP_Slots_V2.4.pdf`

Table E.1 continued

Relation Name	**Example**	**Selected**
per:origin	*Del Ponte*, a *Swiss* citizen	Yes
org:founded_by	Dismayed by the lack of marksmanship shown by their troops, Union veterans *Col. William C. Church* and *Gen. George Wingate* formed the *National Rifle Association* in 1871	Yes
per:employee_of	Gen. *David Petraeus*, *U.S.* Army general, said...	Yes
per:siblings	Unable to get a post in American hospitals, *Blackwell*, her sister *Emily* and friend Dr Marie Zakrzewaska started their own hospital, the New York Infirmary for Indigent Women and Children.	Yes
per:alternate_names	*Rudolf Walter Wanderone, Jr.* was an American professional pocket billiards player, best known as "*Minnesota Fats*"	Yes
org:website	Perhaps the best source available is the *National Rifle Association* (*ww.nra.org*)	Yes
per:religion	Afghan-trained *Muslim* firebrand *Abubakar Abdurajak Janjalani*	Yes
per:stateorprovince_of_death	*Khan* was killed in the *North West Frontier Province*	No
org:parents	the *Africa Export/Import Bank*, a subsidiary of the *African Development Bank* based in Cairo, Egypt	Yes
org:subsidiaries	The *Treasury Department*'s *Bureau of Engraving and Printing* introduced the new design,	Yes
per:other_family	*Williams*' son, Hank Williams Jr., and grandson, *Hank Williams III*	Yes
per:stateorprovinces_of_residence	The *Gore* family resides in Nashville, *Tennessee*	Yes

Table E.1 continued

Relation Name	**Example**	**Selected**
org:members	With *Bulgaria*, *Romania* and *Slovenia NATO* members since 2004 and *Albania* and *Croatia* since April of this year...	Yes
per:cause_of_death	Author *Marilyn French*, 79, passed away of *heart failure* on May 2, 2009, in New York City	No
org:member_of	*Supreme Court* justice *Samuel Alito*	Yes
org:number_of_employees/members	The *CDC* employs nearly *15,000* people in the United States and more than 54 foreign countries.	Yes
per:country_of_birth	*Al-Zarqawi*, the *Jordanian*-born militant...	Yes
org:shareholders	In a rare public statement, *Standard Life Investments*, a top-ten shareholder, signaled that it would be voting against the discretionary payments at *Shell*'s annual meeting this month.	Yes
org:stateorprovince_of_headquarters	*Abu Sayyaf*, headquartered in the southern province of *Basilan*	Yes
per:city_of_death	*Smith* died in *Wilkes-Barre* General Hospital	No
per:city_of_birth	*Williams* lived in Georgiana in the mid 1930's with his mother, Lillie, and his sister, Irene, after his birth in *Mount Olive West*	No
per:spouses	*Williams*' wife at the time of his death, *Billie Jean Jones*	Yes
org:city_of_headquarters	*CSS* is a Brazilian rock band from *São Paulo*.	Yes
per:date_of_death	But *Williams* never played the Opry again. At age 29, while on the way to a concert in Canton, Ohio, he was found dead in the back seat of his Cadillac on *New Year's Day 1953*.	No

Table E.1 continued

Relation Name	**Example**	**Selected**
per:schools_attended	*He* attended the *University of Minnesota*, graduating with a degree in Business/Accounting and turned professional in 1982	Yes
org:political/religious_affiliation	The *Hungarian Reformed Church* is a Reformed Church in the *Calvinist* tradition.	No
per:country_of_death	*Khan* was killed in North West Frontier Province... NWFP is located in *Pakistan*	No
org:founded	Dismayed by the lack of marksmanship shown by their troops, Union veterans Col. William C. Church and Gen. George Wingate formed the *National Rifle Association* in *1871*.	Yes
per:stateorprovince_of_birth	*Harper* was born in Toronto in April 1959. Toronto, *Ontario* is a beautiful city.	No
org:dissolved	It failed to prevent the outbreak of the Second World War, at the end of which the *League* itself was officially disbanded in *1946*.	Yes

APPENDIX F

DOCRED RELATIONS

Table F.1. DocRED relation details

Relation Name	Description	Wikidata ID
country	sovereign state of this item; don't use on humans	P17
father	male parent of the subject.	P22
mother	female parent of the subject.	P25
spouse	the subject has the object as their spouse (husband, wife, partner, etc.).	P26
country of citizenship	the object is a country that recognizes the subject as its citizen	P27
continent	continent of which the subject is a part	P30
capital	primary city of a country, state or other type of administrative territorial entity	P36
child	subject has the object in their family as their offspring son or daughter (independently of their age)	P40
educated at	educational institution attended by the subject	P69
employer	person or organization for which the subject works or worked	P108
founded by	founder or co-founder of this organization, religion or place	P112
owned by	owner of the subject	P127
located in the administrative territorial entity	the item is located on the territory of the following administrative entity	P131
contains administrative territorial entity	(list of) direct subdivisions of an administrative territorial entity	P150
headquarters location	specific location where an organization's headquarters is or has been situated	P159
manufacturer	manufacturer or producer of this product	P176

Table F.1 continued

Relation Name	**Description**	**Wikidata ID**
production company	company that produced this film, audio or performing arts work	P272
location	location of the item, physical object or event is within.	P276
subsidiary	subsidiary of a company or organization, opposite of parent company	P355
member of	organization or club to which the subject belongs	P463
chairperson	presiding member of an organization, group or body	P488
country of origin	country of origin of the creative work or subject item	P495
residence	the place where the person is, or has been, resident	P551
start time	indicates the time an item begins to exist or a statement starts being valid	P580
end time	indicates the time an item ceases to exist or a statement stops being valid	P582
participant	person, group of people or organization (object) that actively takes/took part in the event (subject).	P710
location of formation	location where a group or organization was formed	P740
parent organization	parent organization of an organisation	P749
work location	location where persons were active	P937
product or material produced	material or product produced by a government agency, business, industry, facility, or process	P1056
participant of	event a person or an organization was a participant in	P1344
capital of	country, state, department, canton or other administrative division of which the municipality is the governmental seat	P1376
sibling	the subject has the object as their sibling (brother, sister, etc.).	P3373

REFERENCES

Abadi, D. (2003). Comparing domain-specific and non-domain-specific anaphora resolution techniques. *Cambridge University MPhil Dissertation*.

Afsar, M. M., R. T. Crump, and B. H. Far (2019). Energy-efficient coalition formation in sensor networks: a game-theoretic approach. In *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*, pp. 1–6. IEEE.

Agarwal, A., A. Omuya, A. Harnly, and O. Rambow (2012). A comprehensive gold standard for the enron organizational hierarchy. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 161–165.

Ailon, N., Z. S. Karnin, E. Liberty, and Y. Maarek (2013). Threading machine generated email. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pp. 405–414.

Aktaş, B., T. Scheffler, and M. Stede (2018). Anaphora resolution for twitter conversations: An exploratory study. In *Proceedings of the First Workshop on Computational Models of Reference, Anaphora and Coreference*, pp. 1–10.

Aktaş, B., V. Solopova, A. Kohnert, and M. Stede (2020, November). Adapting coreference resolution to Twitter conversations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, Online, pp. 2454–2460. Association for Computational Linguistics.

Alkhereyf, S. and O. Rambow (2017, August). Work hard, play hard: Email classification on the avocado and Enron corpora. In *Proceedings of TextGraphs-11: the Workshop on Graph-based Methods for Natural Language Processing*, Vancouver, Canada, pp. 57–65. Association for Computational Linguistics.

Alonso-Maturana, R., E. Alvarado-Cortes, S. López-Sola, M. O. Martínez-Losa, and P. Hermoso-González (2018). La rioja turismo: The construction and exploitation of a queryable tourism knowledge graph. In *International Conference on Web Engineering*, pp. 213–220. Springer.

Alrashed, T., A. H. Awadallah, and S. Dumais (2018). The lifetime of email messages: A large-scale analysis of email revisitation. In *Proceedings of the 2018 Conference on Human Information Interaction amp; Retrieval*, CHIIR '18, New York, NY, USA, pp. 120–129. Association for Computing Machinery.

Andersen, P. M., P. J. Hayes, S. P. Weinstein, A. K. Huettner, L. M. Schmandt, and I. Nirenburg (1992). Automatic extraction of facts from press releases to generate news stories. In *Third Conference on Applied Natural Language Processing*, pp. 170–177.

Annervaz, K., S. B. R. Chowdhury, and A. Dukkipati (2018). Learning beyond datasets: Knowledge graph augmented neural networks for natural language processing. *arXiv preprint arXiv:1802.05930*.

Arasu, A. and H. Garcia-Molina (2003). Extracting structured data from web pages. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pp. 337–348.

Awad, W. A. and S. M. ELseuofi (2011). Machine learning methods for spam e-mail classification. *International Journal of Computer Science and Information Technology 3*(1), 173–184.

Balaneshin-kordan, S. and A. Kotov (2016). Sequential query expansion using concept graph. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pp. 155–164.

Bamman, D., O. Lewke, and A. Mansoor (2020, May). An annotated dataset of coreference in English literature. In *Proceedings of the 12th Language Resources and Evaluation Conference*, Marseille, France, pp. 44–54. European Language Resources Association.

Barhom, S., V. Shwartz, A. Eirew, M. Bugert, N. Reimers, and I. Dagan (2019). Revisiting joint modeling of cross-document entity and event coreference resolution. *arXiv preprint arXiv:1906.01753*.

Barley, S. R., D. E. Meyerson, and S. Grodal (2011). E-mail as a source and symbol of stress. *Organization Science 22*(4), 887–906.

Belleau, F., M.-A. Nolin, N. Tourigny, P. Rigault, and J. Morissette (2008). Bio2rdf: towards a mashup to build bioinformatics knowledge systems. *Journal of biomedical informatics 41*(5), 706–716.

Bellotti, V., N. Ducheneaut, M. Howard, and I. Smith (2003). Taking email to task: the design and evaluation of a task management centered email tool. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 345–352.

Bendersky, M., X. Wang, M. Najork, D. Metzler, et al. (2021). Search and discovery in personal email collections. *Foundations and Trends® in Information Retrieval 15*(1), 1–133.

Bergman, M. (2019). *A Common Sense View of Knowledge Graphs*. `https://www.mkbergman.com/2244/a-common-sense-view-of-knowledge-graphs`.

Beseiso, M., A. R. Ahmad, and R. Ismail (2012). A new architecture for email knowledge extraction. *International Journal of Web & Semantic Technology 3*(3), 1.

Bollacker, K., C. Evans, P. Paritosh, T. Sturge, and J. Taylor (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 1247–1250.

Boufaden, N., W. Elazmeh, Y. Ma, S. Matwin, N. El-Kadri, and N. Japkowicz (2005). Peep-an information extraction base approach for privacy protection in email. In *CEAS*.

Bowen, D. A., J. Wang, K. Holland, B. Bartholow, and S. A. Sumner (2020). Conversational topics of social media messages associated with state-level mental distress rates. *Journal of mental health 29*(2), 234–241.

Brickley, D., R. V. Guha, and B. McBride (2014). Rdf schema 1.1. *W3C recommendation 25*, 2004–2014.

Brutlag, J. D. and C. Meek (2000). Challenges of the email domain for text classification. In *ICML*, Volume 2000, pp. 103–110.

Bure, V. M. and K. Y. Staroverova (2019). Applying cooperative games with coalition structure for data clustering. *Automation and Remote Control 80*(8), 1541–1551.

Carenini, G., R. T. Ng, and X. Zhou (2007). Summarizing email conversations with clue words. In *Proceedings of the 16th international conference on World Wide Web*, pp. 91–100.

Chang, S. (2018). Scaling knowledge access and retrieval at airbnb. airbnb medium blog.

Chen, H., Z. Fan, H. Lu, A. Yuille, and S. Rong (2018, October-November). PreCo: A large-scale dataset in preschool vocabulary for coreference resolution. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, pp. 172–181. Association for Computational Linguistics.

Chen, Y.-H. and J. D. Choi (2016). Character identification on multiparty conversation: Identifying mentions of characters in tv shows. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 90–100.

Chinchor, N. A. (1998). Overview of muc-7/met-2. Technical report, SCIENCE APPLICATIONS INTERNATIONAL CORP SAN DIEGO CA.

Clark, K. and C. D. Manning (2015). Entity-centric coreference resolution with model stacking. 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. 1405–1415.

Cohen, A. D., S. Rosenman, and Y. Goldberg (2020). Relation extraction as two-way span-prediction. *arXiv preprint arXiv:2010.04829*.

Cohen, W. W. (1995). Fast effective rule induction. In *Machine learning proceedings 1995*, pp. 115–123. Elsevier.

Cohen, W. W. et al. (1996). Learning rules that classify e-mail. In *AAAI spring symposium on machine learning in information access*, Volume 18, pp. 25. Stanford, CA.

Cohen, W. W., V. R. Carvalho, and T. M. Mitchell (2004). Learning to classify email into "speech acts". In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 309–316.

Corston-Oliver, S., E. Ringger, M. Gamon, and R. Campbell (2004). Task-focused summarization of email. In *Text Summarization Branches Out*, pp. 43–50.

Cowan-Sharp, J. (2009). A study of topic and topic change in conversational threads. Technical report, NAVAL POSTGRADUATE SCHOOL MONTEREY CA DEPT OF COMPUTER SCIENCE.

Cowie, J. and W. Lehnert (1996). Information extraction. *Communications of the ACM 39*(1), 80–91.

Craswell, N., A. P. De Vries, and I. Soboroff (2005). Overview of the trec 2005 enterprise track. In *Trec*, Volume 5, pp. 1–7.

Crispin, M. and K. Murchison (2008, June). Internet message access protocol - sort and thread extensions. RFC 5256, RFC Editor.

Culotta, A., R. Bekkerman, and A. McCallum (2005). Extracting social networks and contact information from email and the web. Technical report, MASSACHUSETTS UNIV AMHERST DEPT OF COMPUTER SCIENCE.

da Silva, A. R. A. G. (2016). Email classification: a case study.

Dabbish, L. A. and R. E. Kraut (2006). Email overload at work: An analysis of factors associated with email strain. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pp. 431–440.

Dawson, F. and T. Howes (1998, September). vcard mime directory profile. RFC 2426, RFC Editor.

Decker, S. and M. Frank (2004). The social semantic desktop. *Digital Enterprise Research Institute, DERI Technical Report May 2*(7).

Dehghani, M., A. Shakery, M. Asadpour, and A. Koushkestani (2013). A learning approach for email conversation thread reconstruction. *Journal of Information Science 39*(6), 846–863.

DeJong, G. (1979). Prediction and substantiation: A new approach to natural language processing. *Cognitive Science 3*(3), 251–273.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2019, June). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota, pp. 4171–4186. Association for Computational Linguistics.

Dhamal, S., S. Bhat, K. Anoop, and V. R. Embar (2012). Pattern clustering using cooperative game theory. *arXiv preprint arXiv:1201.0461*.

Diehl, C. P., L. Getoor, and G. Namata (2006). Name reference resolution in organizational email archives. In *Proceedings of the 2006 SIAM International Conference on Data Mining*, pp. 70–81. SIAM.

Diehl, C. P., G. Namata, and L. Getoor (2007). Relationship identification for social network discovery. In *AAAI*, Volume 22, pp. 546–552.

Doddington, G. R., A. Mitchell, M. A. Przybocki, L. A. Ramshaw, S. M. Strassel, and R. M. Weischedel (2004). The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, Volume 2, pp. 1. Lisbon.

Dong, X. L. (2019). Building a broad knowledge graph for products. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pp. 25–25. IEEE.

Douglas Oard, William Webber, D. K. S. G. (2015). Avocado research email collection. *Philadelphia: Linguistic Data Consortium*.

Dredze, M., J. Blitzer, and F. Pereira (2006). " sorry, i forgot the attachment": Email attachment prediction. In *CEAS*.

Eberts, M. and A. Ulges (2021, April). An end-to-end model for entity-level relation extraction using multi-instance learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Online, pp. 3650–3660. Association for Computational Linguistics.

Elsayed, T. and D. W. Oard (2006). Modeling identity in archival collections of email: A preliminary study. In *CEAS*, pp. 95–103.

Erera, S. and D. Carmel (2008). Conversation detection in email systems. In *European Conference on Information Retrieval*, pp. 498–505. Springer.

Färber, M. (2019). The Microsoft Academic Knowledge Graph: A Linked Data Source with 8 Billion Triples of Scholarly Data. In *Proceedings of the 18th International Semantic Web Conference*, ISWC'19, pp. 113–129.

Fleiss, J., B. Levin, and M. Paik (2003). The measurement of interrater agreement. In *Statistical Methods for Rates and Proportions, Third Edition*, pp. 598 – 626. John Wiley Sons, Inc.

Futia, G. and A. Vetrò (2020). On the integration of knowledge graphs into deep learning models for a more comprehensible ai—three challenges for future research. *Information 11*(2), 122.

Gandhi, N., A. Field, and Y. Tsvetkov (2021). Improving span representation for domain-adapted coreference resolution. *arXiv preprint arXiv:2109.09811*.

Georgakopoulos, P., L. Kanaris, T. Akhtar, A. Kokkinis, S. Stavrou, and I. Politis (2020). Coalition formation games for coordinated service in realistic small cell propagation topologies. *IEEE Access 8*, 186789–186804.

Ghosh, D., A. R. Fabbri, and S. Muresan (2018). Sarcasm analysis using conversation context. *Computational Linguistics 44*(4), 755–792.

Goldstein, J., A. Kwasinksi, P. R. Kingsbury, R. E. Sabin, and A. McDowell (2006). Annotating subsets of the enron email corpus. In *CEAS*.

Gonçalves, R. S., M. Horridge, R. Li, Y. Liu, M. A. Musen, C. I. Nyulas, E. Obamos, D. Shrouty, and D. Temple (2019). Use of owl and semantic web technologies at pinterest. In *International Semantic Web Conference*, pp. 418–435. Springer.

Gong, L. (2003). San francisco, ca (us) united states us 2003.01671. 67a1 (12) patent application publication c (10) pub. *No.: US 167167*, A1.

Grbovic, M., G. Halawi, Z. Karnin, and Y. Maarek (2014). How many folders do you really need?: Classifying email into a handful of categories. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pp. 869–878.

Grishman, R. and B. Sundheim (1996). Message understanding conference-6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Group, T. G. and L. R. Group (2012). Enough Already! Stop Bad Email. `http://www.yourthoughtpartner.com/email-perception-study/`. [Online; accessed 22-September-2021].

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition 5*(2), 199–220.

Gupta, A., R. Sharda, R. Greve, and M. Kamath (2004). An exploratory analysis of email processing strategies. In *Proceedings of 22 nd Annual Decision Sciences Meeting, Boston*.

Gupta, R., R. Kondapally, and S. Guha (2019). Large-scale information extraction from emails with data constraints. In *International Conference on Big Data Analytics*, pp. 124–139. Springer.

Hachenberg, C. and T. Gottron (2013). Locality sensitive hashing for scalable structural classification and clustering of web documents. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pp. 359–368.

Hamad, F., I. Liu, and X. Zhang (2018). Food discovery with uber eats: Building a query understanding engine. *Uber Engineering*.

Hassine, N. B., P. Minet, M.-A. Koulali, M. Erradi, D. Marinca, and D. Barth (2017). Coalition game for video content clustering in content delivery networks. In *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pp. 407–413. IEEE.

He, Q., B. Chen, and D. Agarwal (2016). Building the linkedin knowledge graph. *Engineering. linkedin. com*.

Heath, T. and E. Motta (2007). Revyu. com: a reviewing and rating site for the web of data. In *The Semantic Web*, pp. 895–902. Springer.

Henderson, M., P. Budzianowski, I. Casanueva, S. Coope, D. Gerz, G. Kumar, N. Mrkšić, G. Spithourakis, P.-H. Su, I. Vulić, and T.-H. Wen (2019, August). A repository of conversational datasets. In *Proceedings of the First Workshop on NLP for Conversational AI*, Florence, Italy, pp. 1–10. Association for Computational Linguistics.

Hogan, A., E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutierrez, J. E. L. Gayo, S. Kirrane, S. Neumaier, A. Polleres, R. Navigli, A.-C. N. Ngomo, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, and A. Zimmermann (2020). Knowledge graphs.

Honnibal, M. and I. Montani (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Huang, H., L. Heck, and H. Ji (2015). Leveraging deep neural networks and knowledge graphs for entity disambiguation. *arXiv preprint arXiv:1504.07678*.

Huang, Y. and T. M. Mitchell (2008). Exploring hierarchical user feedback in email clustering. In *Enhanced Messaging Workshop in Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI 2008)*.

Huo, Y., W. Dong, J. Qian, and T. Jing (2017). Coalition game-based secure and effective clustering communication in vehicular cyber-physical system (vcps). *Sensors 17*(3), 475.

Ide, N., C. Baker, C. Fellbaum, C. Fillmore, and R. Passonneau (2008). Masc: The manually annotated sub-corpus of american english. In *6th International Conference on Language Resources and Evaluation, LREC 2008*, pp. 2455–2460. European Language Resources Association (ELRA).

Jackson, T., R. Dawson, and D. Wilson (2002). Case study: evaluating the effect of email interruptions within the workplace. In *Conference on Empirical Assessment in Software Engineering, Keele University, EASE*, pp. 3–7. © EASE.

Ji, S., S. Pan, E. Cambria, P. Marttinen, and S. Y. Philip (2021). A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*.

Joshi, M., D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy (2019). Spanbert: Improving pre-training by representing and predicting spans. *arXiv preprint arXiv:1907.10529*.

Joshi, M., O. Levy, D. S. Weld, and L. Zettlemoyer (2019). Bert for coreference resolution: Baselines and analysis. *arXiv preprint arXiv:1908.09091*.

Joty, S. and T. Mohiuddin (2018). Modeling speech acts in asynchronous conversations: A neural-crf approach. *Computational Linguistics 44*(4), 859–894.

Jung, Y., K. Stratos, and L. P. Carloni (2015). Ln-annote: An alternative approach to information extraction from emails using locally-customized named-entity recognition. In *Proceedings of the 24th International Conference on World Wide Web*, pp. 538–548.

Kärle, E., U. Şimşek, O. Panasiuk, and D. Fensel (2018). Building an ecosystem for the tyrolean tourism knowledge graph. In *International Conference on Web Engineering*, pp. 260–267. Springer.

Khashabi, D., S. Min, T. Khot, A. Sabharwal, O. Tafjord, P. Clark, and H. Hajishirzi (2020, November). UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, Online, pp. 1896–1907. Association for Computational Linguistics.

Khosla, S. and C. Rose (2020, November). Using type information to improve entity coreference resolution. In *Proceedings of the First Workshop on Computational Approaches to Discourse*, Online, pp. 20–31. Association for Computational Linguistics.

Kingsley, E., F. F. Kopstein, and R. J. Seidel (1969). Graph theory as a metalanguage of communicable knowledge. Technical report, HUMAN RESOURCES RESEARCH ORGANIZATION ALEXANDRIA VA.

Kleinberg, S. (2018). 5 ways voice assistance is shaping consumer behavior. *Think with Google*.

Klimt, B. and Y. Yang (2004). The enron corpus: A new dataset for email classification research. In *European Conference on Machine Learning*, pp. 217–226. Springer.

Kushmerick, N. (1997). *Wrapper induction for information extraction*. University of Washington.

Lam, D. S. (2002). *Exploiting e-mail structure to improve summarization*. Ph. D. thesis, Massachusetts Institute of Technology.

Lee, K., L. He, M. Lewis, and L. Zettlemoyer (2017, September). End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, pp. 188–197. Association for Computational Linguistics.

Lee, K., L. He, and L. Zettlemoyer (2018, June). Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, New Orleans, Louisiana, pp. 687–692. Association for Computational Linguistics.

Lehmann, J., R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. Van Kleef, S. Auer, et al. (2015). Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web 6*(2), 167–195.

Levesque, H., E. Davis, and L. Morgenstern (2012). The winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*.

Lewis, D. D. and K. A. Knowles (1997). Threading electronic mail: A preliminary study. *Information processing & management 33*(2), 209–217.

Li, J., Y. Song, Z. Wei, and K.-F. Wong (2018). A joint model of conversational discourse and latent topics on microblogs. *Computational Linguistics 44*(4), 719–754.

Li, X., Y. Wang, D. Wang, W. Yuan, D. Peng, and Q. Mei (2019). Improving rare disease classification using imperfect knowledge graph. *BMC Medical Informatics and Decision Making 19*(5), 238.

Logan, R., N. F. Liu, M. E. Peters, M. Gardner, and S. Singh (2019, July). Barack's wife hillary: Using knowledge graphs for fact-aware language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, pp. 5962–5971. Association for Computational Linguistics.

Luan, Y., L. He, M. Ostendorf, and H. Hajishirzi (2018, October-November). Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, pp. 3219–3232. Association for Computational Linguistics.

Lyddy, F., F. Farina, J. Hanney, L. Farrell, and N. Kelly O'Neill (2014). An analysis of language in university students' text messages. *Journal of Computer-Mediated Communication 19*(3), 546–561.

Mahlawi, A. Q. and S. Sasi (2017). Structured data extraction from emails. In *2017 International Conference on Networks & Advances in Computational Technologies (NetACT)*, pp. 323–328. IEEE.

Manco, G., E. Masciari, M. Ruffolo, and A. Tagarelli (2002). Towards an adaptive mail classifier. In *Proc. of Italian Association for Artificial Intelligence Workshop*.

Marino, K., R. Salakhutdinov, and A. Gupta (2016). The more you know: Using knowledge graphs for image classification. *CoRR abs/1612.04844*.

Mehta, A. and S. Mehta (2020). Detecting conversational toxicity using neural networks. In *Proceedings of the International Conference on Innovative Computing & Communications (ICICC)*.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM 38*(11), 39–41.

Minkov, E., R. Balasubramanyan, W. W. Cohen, and M. L. Dep (2008). Activity-centric search in email. In *Enhanced Messaging Workshop, AAAI*.

Minkov, E., R. C. Wang, and W. Cohen (2005). Extracting personal names from email: Applying named entity recognition to informal text. In *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, pp. 443–450.

Montiel-Ponsoda, E., J. Gracia, and V. Rodríguez-Doncel (2018). Building the legal knowledge graph for smart compliance services in multilingual europe. In *CEUR workshop proc.*, Number ART-2018-105821.

Moosavi, N. S. and M. Strube (2016, August). Which coreference evaluation metric do you trust? a proposal for a link-based entity aware metric. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany, pp. 632–642. Association for Computational Linguistics.

Mujtaba, G., L. Shuib, R. G. Raj, N. Majeed, and M. A. Al-Garadi (2017). Email classification research trends: Review and open issues. *IEEE Access 5*, 9044–9064.

Muresan, S., E. Tzoukermann, and J. L. Klavans (2001). Combining linguistic and machine learning techniques for email summarization. In *Proceedings of the ACL 2001 Workshop on Computational Natural Language Learning (ConLL)*.

Navigli, R. and S. P. Ponzetto (2012). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence 193*, 217–250.

Nenkova, A. and A. Bagga (2004). Facilitating email thread access by extractive summary generation. *Recent advances in natural language processing III: selected papers from RANLP 2003*, 287–294.

Newman, P. S. and J. C. Blitzer (2003). Summarizing archived discussions: a beginning. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pp. 273–276.

Noy, N., Y. Gao, A. Jain, A. Narayanan, A. Patterson, and J. Taylor (2019). Industry-scale knowledge graphs: lessons and challenges. *Queue 17*(2), 48–75.

Ostendorff, M., P. Bourgonje, M. Berger, J. Moreno-Schneider, G. Rehm, and B. Gipp (2019). Enriching bert with knowledge graph embeddings for document classification. *arXiv preprint arXiv:1909.08402*.

Pant, K., T. Dadu, and R. Mamidi (2020). Bert-based ensembles for modeling disclosure and support in conversational social media text. In *AffCon@ AAAI*, pp. 130–139.

Park, S., A. X. Zhang, L. S. Murray, and D. R. Karger (2019). Opportunities for automating email processing: A need-finding study. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–12.

Patil, A. P., R. Subramanian, G. Karkal, K. Purushotham, J. Wadhwa, K. D. Reddy, and M. Sawood (2020, December). Optimized web-crawling of conversational data from social media and context-based filtering. In *Proceedings of the Workshop on Joint NLP Modelling for Conversational AI @ ICON 2020*, Patna, India, pp. 33–39. NLP Association of India (NLPAI).

Pennington, J., R. Socher, and C. Manning (2014, October). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp. 1532–1543. Association for Computational Linguistics.

Peroni, S., A. Dutton, T. Gray, and D. Shotton (2015). Setting our bibliographic references free: towards open citation data. *Journal of Documentation*.

Pittman, R., A. Srivastava, S. Hewavitharana, A. Kale, and S. Mansour (2017). Cracking the code on conversational commerce. ebay blog.

Pradhan, S., A. Moschitti, N. Xue, O. Uryupina, and Y. Zhang (2012). Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint Conference on EMNLP and CoNLL-Shared Task*, pp. 1–40. Association for Computational Linguistics.

Preotiuc-Pietro, D., S. Samangooei, T. Cohn, N. Gibbins, and M. Niranjan (2012). Trendminer: An architecture for real time analysis of social media text. In *Proceedings of the International AAAI Conference on Web and Social Media*, Volume 6.

Qamar, S., H. Mujtaba, H. Majeed, and M. O. Beg (2021). Relationship identification between conversational agents using emotion analysis. *Cognitive Computation 13*(3), 673–687.

Raffel, C., N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research 21*, 1–67.

Rambow, O., L. Shrestha, J. Chen, and C. Lauridsen (2004). Summarizing email threads. In *Proceedings of HLT-NAACL 2004: Short Papers*, HLT-NAACL-Short '04, USA, pp. 105–108. Association for Computational Linguistics.

Rebele, T., F. Suchanek, J. Hoffart, J. Biega, E. Kuzey, and G. Weikum (2016). Yago: A multilingual knowledge base from wikipedia, wordnet, and geonames. In *International semantic web conference*, pp. 177–185. Springer.

Recasens, M., L. Màrquez, E. Sapena, M. A. Martí, M. Taulé, V. Hoste, M. Poesio, and Y. Versley (2010). Semeval-2010 task 1: Coreference resolution in multiple languages. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pp. 1–8. Association for Computational Linguistics.

Richens, R. H. (1956). Preprogramming for mechanical translation. *Mech. Transl. Comput. Linguistics 3*(1), 20–25.

Sarawagi, S. (2002). Automation in information extraction and integration. In *Tutorial of The 28th International Conference on Very Large Data Bases (VLDB)*.

Saxena, A., M. Mangal, and G. Jain (2020, December). KeyGames: A game theoretic approach to automatic keyphrase extraction. In *Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, Spain (Online), pp. 2037–2048. International Committee on Computational Linguistics.

Schneider, E. W. (1973). Course modularization applied: The interface system and its implications for sequence control and data analysis.

Sharma, S., B. Santra, A. Jana, S. Tokala, N. Ganguly, and P. Goyal (2019, November). Incorporating domain knowledge into medical NLI using knowledge graphs. 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)*, Hong Kong, China, pp. 6092–6097. Association for Computational Linguistics.

Simoes, G., H. Galhardas, and L. Coheur (2004). Information extraction tasks: a survey.

Singhal, A. (2012). Introducing the knowledge graph: things, not strings. *Official google blog 5*.

Soboroff, I., A. P. de Vries, and N. Craswell (2006). Overview of the trec 2006 enterprise track. In *Trec*, Volume 6, pp. 1–20.

Song, Y. W. et al. (2019). User acceptance of an artificial intelligence (ai) virtual assistant: an extension of the technology acceptance model.

Soucek, R. and K. Moser (2010). Coping with information overload in email communication: Evaluation of a training intervention. *Computers in Human Behavior 26*(6), 1458–1466.

Stadler, C., J. Lehmann, K. Höffner, and S. Auer (2012). Linkedgeodata: A core for a web of spatial open data. *Semantic Web 3*(4), 333–354.

Sulistyo, S., S. Alam, and R. Adrian (2019). Coalitional game theoretical approach for vanet clustering to improve snr. *Journal of Computer Networks and Communications 2019*.

Sundheim, B. M. (1995). Overview of results of the MUC-6 evaluation. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*.

Team, S. (2017). Hey siri: An on-device dnn-powered voice trigger for apple's personal assistant. *Apple Machine Learning Journal 1*(6).

Terzopoulos, G. and M. Satratzemi (2020). Voice assistants and smart speakers in everyday life and in education. *Informatics in Education 19*(3), 473–490.

The Radicati Group, I. (2015). Email Statistics Report, 2015-2019. `http://www.radicati.com/wp/wp-content/uploads/2015/02/Email-Statistics-Report-2015-2019-Executive-Summary.pdf`. [Online; accessed 22-September-2021].

Timmapathini, H., A. Nayak, S. Mandadi, S. Sangada, V. Kesri, K. Ponnalagu, and V. G. Venkoparao (2021). Probing the spanbert architecture to interpret scientific domain adaptation challenges for coreference resolution. In A. P. B. Veyseh, F. Dernoncourt, T. H. Nguyen, W. Chang, and L. A. Celi (Eds.), *Proceedings of the Workshop on Scientific*

Document Understanding co-located with 35th AAAI Conference on Artificial Inteligence, SDU@AAAI 2021, Virtual Event, February 9, 2021, Volume 2831 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Toshniwal, S., S. Wiseman, A. Ettinger, K. Livescu, and K. Gimpel (2020, November). Learning to Ignore: Long Document Coreference with Bounded Memory Neural Networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online, pp. 8519–8526. Association for Computational Linguistics.

Tulshan, A. S. and S. N. Dhage (2018). Survey on virtual assistant: Google assistant, siri, cortana, alexa. In *International symposium on signal processing and intelligent recognition systems*, pp. 190–201. Springer.

Ulrich, J., G. Murray, and G. Carenini (2008). A publicly available annotated corpus for supervised email summarization. In *Proc. of aaai email-2008 workshop, chicago, usa*.

Uzuner, Ö., B. R. South, S. Shen, and S. L. DuVall (2011). 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association 18*(5), 552–556.

Van Hee, C., E. Lefever, and V. Hoste (2018). We usually don't like going to the dentist: Using common sense to detect irony on twitter. *Computational Linguistics 44*(4), 793–832.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin (2017). Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008.

Viégas, F. B., S. Golder, and J. Donath (2006). Visualizing email content: portraying relationships from conversational histories. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 979–988.

Vrandečić, D. and M. Krötzsch (2014, September). Wikidata: A free collaborative knowledgebase. *Commun. ACM 57*(10), 78–85.

Wambsganss, T., R. Winkler, P. Schmid, and M. Söllner (2020). Unleashing the potential of conversational agents for course evaluations: Empirical insights from a comparison with web surveys. Association for Information Systems.

Wang, X., M. Xu, N. Zheng, and M. Chen (2008). Email conversations reconstruction based on messages threading for multi-person. In *2008 International Workshop on Education Technology and Training & 2008 International Workshop on Geoscience and Remote Sensing*, Volume 1, pp. 676–680. IEEE.

Webster, K., M. Recasens, V. Axelrod, and J. Baldridge (2018). Mind the GAP: A balanced corpus of gendered ambiguous pronouns. *Transactions of the Association for Computational Linguistics 6*, 605–617.

Wendt, J. B., M. Bendersky, L. Garcia-Pueyo, V. Josifovski, B. Miklos, I. Krka, A. Saikia, J. Yang, M.-A. Cartright, and S. Ravi (2016). Hierarchical label propagation and discovery for machine generated email. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pp. 317–326.

Whittaker, S. and C. Sidner (1996). Email overload: exploring personal information management of email. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 276–283.

Wu, W., F. Wang, A. Yuan, F. Wu, and J. Li (2020, July). CorefQA: Coreference resolution as query-based span prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, pp. 6953–6963. Association for Computational Linguistics.

Xia, P. and B. Van Durme (2021). Moving on from ontonotes: Coreference resolution model transfer. *arXiv preprint arXiv:2104.08457*.

Yamada, I., A. Asai, H. Shindo, H. Takeda, and Y. Matsumoto (2020). Luke: Deep contextualized entity representations with entity-aware self-attention. *arXiv preprint arXiv:2010.01057*.

Yang, X., A. H. Awadallah, M. Khabsa, W. Wang, and M. Wang (2018). Characterizing and supporting question answering in human-to-human communication. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 345–354.

Yao, Y., D. Ye, P. Li, X. Han, Y. Lin, Z. Liu, Z. Liu, L. Huang, J. Zhou, and M. Sun (2019, July). DocRED: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, pp. 764–777. Association for Computational Linguistics.

Ye, D., Y. Lin, J. Du, Z. Liu, P. Li, M. Sun, and Z. Liu (2020, November). Coreferential Reasoning Learning for Language Representation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online, pp. 7170–7186. Association for Computational Linguistics.

Youn, S. and D. McLeod (2007). A comparative study for email classification. In *Advances and innovations in systems, computing sciences and software engineering*, pp. 387–391. Springer.

Zawinski, J. (1997). Message threading. `https://www.jwz.org/doc/threading.html`. [Online; accessed 14-October-2021].

Zhang, W., A. Ahmed, J. Yang, V. Josifovski, and A. J. Smola (2015). Annotating needles in the haystack without looking: Product information extraction from emails. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2257–2266.

Zhang, Y., V. Zhong, D. Chen, G. Angeli, and C. D. Manning (2017). 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.

Zhou, E. and J. D. Choi (2018). They exist! introducing plural mentions to coreference resolution and entity linking. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 24–34.

Zylich, B., A. Viola, B. Toggerson, L. Al-Hariri, and A. Lan (2020). Exploring automated question answering methods for teaching assistance. In *International Conference on Artificial Intelligence in Education*, pp. 610–622. Springer.

BIOGRAPHICAL SKETCH

Parag Dakle, son of Pravin and Sunita Dakle, was born in Pune, India. He completed his bachelor's degree in Computer Engineering from Pune University in May 2014 and joined UT Dallas in August 2016 to pursue his master's degree. He joined the PhD program at UT Dallas in August 2017. Parag worked with Dr. Dan Moldovan to research in the fields of knowledge extraction and email processing. He interned as an NLP Software Engineer at Lymba Corporation in Richardson, TX in Summer of 2021, 2020, 2019 and 2018. He also interned as a Software Engineer at Bottle Rocket in Dallas, TX in Summer 2017. Before joining UT Dallas, Parag worked in Pune, India as a Software Engineer at Great Software Laboratory from July 2014 to January 2016, and as a Full Stack Engineer at Muffin App from February 2016 to July 2016. He will join Fidelity Investments at their headquarters in Boston, MA as a Data Scientist in Spring 2022.

CURRICULUM VITAE

Parag Pravin Dakle

November 2021

Contact Information:

Department of Computer Science
The University of Texas at Dallas
800 W. Campbell Rd.
Richardson, TX 75080-3021, U.S.A.

Email: paragpravin.dakle@utdallas.edu

Educational History:

BS, Computer Engineering, University of Pune, 2014
MS, Computer Science, The University of Texas at Dallas, 2021
PhD, Computer Science, The University of Texas at Dallas, 2021

Knowledge Extraction from Email Conversations and its Application to Question Answering
PhD Dissertation
Department of Computer Science, The University of Texas at Dallas
Advisor: Dr. Dan I. Moldovan

Employment History:

Lymba Corporation, Richardson, TX
  Software Engineer Intern, June 2021 – August 2021 (Part-time)
  Software Engineer Intern, May 2020 – August 2020 (Part-time)
  Software Engineer Intern, June 2019 – August 2019
  Software Engineer Intern, May 2018 – August 2018
Bottle Rocket, Dallas, TX
  Software Engineer Intern, June 2017 – August 2017
Muffin App, Pune, India
  Full Stack Developer, February 2016 – July 2016
Great Software Laboratory, Pune, India
  Software Engineer, July 2014 – January 2014

Publications[1]:

1. Desai, Takshak, Parag Dakle, and Dan Moldovan. "Generating questions for reading comprehension using coherence relations." In Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications, pp. 1-10. 2018.

[1]Publications marked with an asterisk are relevant to this dissertation

2. Desai, Takshak, Parag Pravin Dakle, and Dan Moldovan. “Joint Learning of Syntactic Features helps Discourse Segmentation.” In Proceedings of The 12th Language Resources and Evaluation Conference, pp. 1073-1080. 2020.

3. * Dakle, Parag Pravin, Takshak Desai, and Dan Moldovan. “A study on entity resolution for email conversations.” In Proceedings of The 12th Language Resources and Evaluation Conference, pp. 65-73. 2020.

4. * Dakle, Parag Pravin, and Dan Moldovan. ”CEREC: A Corpus for Entity Resolution in Email Conversations.” In Proceedings of the 28th International Conference on Computational Linguistics, pp. 339-349. 2020.