EMCODIST: A Context-based Search Tool for Email Archives

1st Santhilata Kuppili Venkata
Digital Archiving Department
The National Archives
London, UK
0000-0003-2406-073X

2nd Stephanie Decker
School of Management and Department of Economic History
University of Bristol and University of Gothenburg
Bristol, UK, and Gothenburg, Sweden
0000-0003-0547-9594

3rd David A Kirsch
Robert H. Smith School of Business and College of Information Studies
University of Maryland
College Park, MD USA
0000-0002-2143-8306

4th Adam Nix
Birmingham Business School
University of Birmingham
Birmingham, UK
0000-0003-1539-8130

***Abstract*—Preservation of emails poses particular challenges to future discovery as alternative historical sources. Emails represent communications between individuals and contain a wealth of information when viewed as an organisation-wide collection. Existing search tools can extract named entities and keyword searches but are less effective when it comes to extracting patterns and contextual information across multiple custodians. To address this, we present EMCODIST, a discovery tool for searching the contextual information across emails using attention-based models of Natural Language Processing (NLP). The EMCODIST aims to steer end-users to personalise their searches towards a concept. In this paper, we explain the definition of the 'context' for emails which is also suitable for object-oriented computational modelling. The tool is evaluated based on the relevancy of the emails extracted.**

***Index Terms*—Contextualisation, Email archives processing, Content analysis, Natural Language Processing**

I. Introduction

Emails will likely become records of significant interest to future historians, since they replaced letters, and to some extent phone calls, in the late 1990s. For over two decades, organizations and individuals have used it as a primary communication channel, the traces of which have increasingly become part of archival workflows. While many in the archival profession have been wrangling with the vital pre-conditions for providing access to these collections (e.g., privacy), another important consideration is how end-users will actually engage with them.

Unlike other born-digital collections, email archives pose unique challenges to make them searchable. This is often overlooked because emails are currently only rarely accessible due to privacy restrictions. Yet even where they are accessible (e.g., Enron Email Corpus[1], US gubernatorial email [5]), providing access to researchers in a meaningful way can be challenging due to the inherently networked nature of email conversations.

[1]https://www.cs.cmu.edu/ enron/

A. Why is the Context of Email Important?

Unlike many other types of digital data, emails represent the conversations between two or more people, often in real-time, and in organizational settings. They take place in parallel to other forms of communications such as meetings and phone calls. The immediacy of email messages as a historical source provides authenticity and adds a personal touch to the historical events.

Also, email tracks the development of a conversation over time through the email thread in a way that older forms of information exchange did not, as well as generating a network of recipients and senders. This complex nature of email means that search and discovery in such a dataset are not straightforward, but can also benefit from the additional information about the timing and communications networks in terms of developing a discovery tool.

The very existence of email archives is encouraging users to expect more from these collections. Users want to explore the network properties of collections to search for the patterns across multi-custodian archives, but users also want to understand specific constraints and decisions, why someone acted one way rather than the opposite. For these types of questions, detailed readings, interpretations and context of email are necessary. Thus, a collection like the Enron dataset only becomes viable as a historical source when placed within its proper context.

B. Existing Search tools for Email Archives

So far, many tools have focused on the preservation of emails rather than providing discovery of the content of email archive collections. Existing tools are helpful to redact sensitive personal data and afford a broad understanding of the nature of a collection. The exception here is tools such

as EPADD[2] and RATOM[3], which make an effort to provide keyword search and extract the metadata information of email archives. They are comprehensive tools for preservation and support access. However, they are not sufficient to retrieve contextual information considering the entire email corpus as a single repository. The existing tools use Natural Language Processing (NLP) to extract named-entity recognition, topic modelling to enable name-based searches and keyword searches. They are useful to search known keywords or establish topic models but provide no functionality to explore and "deep-dive" into particular topics that are not associated with clear and previously known keywords, nor do they facilitate the exploration of the interactions of individuals around specific issues.

Also, since emails are informal, complex and loosely structured texts, keyword search alone is often not sufficient to establish context. People do not write with keywords, and the terms that we use to describe our search interest does not necessarily match with how individuals were writing about them in their emails at the time. A keyword may be used in some contexts and a colloquial description of the keyword may be used at other times. As a result, we need a discovery tool that extracts the meaning of whole sentences and paragraphs.

Finally, even though models used for searching data are quick and straightforward, they are often of limited use due to the volume of results they return. They can be excellent for finding popular or highly used terms and phrases, but frequently return word matches that are not relevant, making search results less relevant. Relevant emails might be buried in a list of thousands of results. Adding contextual information to such searches renders results more meaningful.

To the best of our knowledge, our research is one of the first to consider the issue of content discovery facilitated by AI tools for digital archives. In this research, we aim to focus beyond the digital preservation to a futuristic context-sensitive discovery in large digital collections. This can be achieved by combining the information extracted (using Natural language processing and AI) to the email network. The solution needs to combine different analytical approaches to achieve more relevant search results. Our intention is to create a tool that would improve discovery for archival users to identify relevant content from large email archives for further investigation.

C. Assisting Users with Adaptive Search

This section explains how to help users find the information they are looking for. While emails provide an effective means of communication, they quickly generate huge digital heaps to make search impossible by manual methods. Even though metadata such as the sender, list of recipients, cc, bcc, sent date (time) etc. provide a structure, it is the unstructured text of the subject, email body and the contents of attachments that are important for context-based search. Since email represent conversations and discussions between real people, there could be multiple ways of expressing and engaging with anyone concept. While it requires automated methods to tackle the scale, AI methodologies (especially the support of NLP) can help identify the central meaning of the text.

From our initial engagement with researchers interested in looking at email, some issues were raised that existing tools cannot fully address, such as,

- How can we search across multiple custodians for conceptually related information to develop explanations?
- How can we identify particular events and the range of actors involved?
- How can we scour the email collections to find the sequence of exchanges that led to an important decision?

In practice, it can be difficult to find answers to these questions with existing search modalities, and in our project, we sought to develop a discovery tool that seeks to find some possible ways for users to find better information for their queries. We propose the prototype tool called **EMCODIST**, to allow users a context-rich search of organizational email corpora[4] [9]. The EMCODIST focuses on e-discovery to extract chains of emails beyond keyword search and facilitates a context-based search of user queries across multiple custodians, which is particularly relevant to organisational email collections. Later sections of this paper describe the construction and how to use the tool for discovery. However, this paper does not cover the issues related to privacy, sensitivity and redaction of Personal Identifiable Information (PII) during the search.

II. Background

This section gives a background account of what does a 'context' mean for digital collection and who are the users and what their expectations (or requirements) are for whom this tool should be developed. Along with the context, we give a brief background on the recent developments of NLP to use in AI tools.

A. The Context in Digital Collections

The meaning of the 'context' for an archive may vary according to the usage of that record. While for some documents, the context can be defined with only metadata (such as the author of the document, date of the document modified, document length etc.), others may need the information retrieved from the contents of the document. For example, the contextual information of the court judgment records includes the name(s) of the judges, type of the court(s) and other legal information from the contents of the record along with the metadata (date, author, other embedded digital objects etc.) to identify the record.

In information retrieval, defining the context of a digital record (object) depends on digital curation[5] and reuse of that object. While the curation focus on metadata, the reuse concentrates on the extraction of the meaning through metadata [10]. The context represents some kind of background to

[2]https://library.stanford.edu/projects/epadd

[3]https://ratom.web.unc.edu/

[4]https://orghist.com/ahrc-project-historicizing-the-dot-com-bubble/

[5]Curation is the processes used for preservation of digital documents

understand the patterns of behaviour and social & cultural meanings and all known properties associated with it and all operations [3], [4], [21]. Mayer & Rauber [16] introduced a semi-automatic approach using various dimensions of the context such as the time of the object creation and modification, the type represented by the object, people involved and content type, genre, acronyms used etc. to support users' analysis and interpretive processes. We reuse their usecase with emails as the base for our research.

B. The End-Users

The need for a search tool depends on the end-users requirements for the tool. In a prior study of user-base for digital objects, Talboom and Underdown [20] classify users as (i) traditional readers, (ii) researchers, who are keen on emulating the records in their original setting and (iii) users of 'big data' who are aware of technological advancements. While the first type read the content of the documents just as they would with the paper-based documents, the third type may want to perform some computational analysis of meta-data and other semantic features [12]. The second type of users who are mentioned as 'digitally curious' would want to search the collections and make lateral connections to the information available on other digital media. In the context of email collections, the importance of network patterns [1], timing norms [6], and the information embedded in the conversations aid users in their analysis to make connections with the information available from other resources.

C. NLP for the Tool Development

Natural language conversation is one of the most challenging artificial intelligence problems, which needs interpretation of the language, the reasoning behind the phrase utilisation, parts of speech tagging and, ultimately, replicating common sense. Besides the above, in our context, to understand emails we need to perform state-of-art approaches of NLP such as named-entity recognition (NER), semantic role labelling to extract the concepts that people are using in their email messages.

The earlier works with the natural language mainly employed rule-based, learning-based and language parsers [15], [17], [23]. Since they were developed for restricted vocabulary applications, learning algorithms and rules do not apply to larger systems that need a more extensive language base. The growth of digital media such as Facebook and Twitter introduced the challenges of short text conversations and volume to NLP [13], [18]. Short text conversations are the task-at-hand kind of conversations that make sense only when the complete conversation thread is read. The neural-based encoder-decoder architectures [2], [19] dynamically changed the machine translation to use in NLP. Especially, Bahdanau et al. [2] introduced an attention mechanism that allows the decoder to dynamically select and combine different parts of the input. Attention is a method for making both biological and artificial neural systems more flexible. It means this method has improved the performance of understanding the central concept of a sentence, especially for long sequences. While encoder-decoder architectures and attention mechanisms have boosted the capabilities of NLP, Vaswani et al. [22] proposed the *Transformer* to overcome the bottlenecks of sequential encoding steps used previously. The wide applicability of the Transformers models such as Bidirectional Encoder Representations from Transformers (BERT) [8] and the availability of the tools for downstream tasks made them a popular product in the NLP world. We make use of the attention mechanism in developing our tool.

III. Search Tool Environment

Equipped with the background on related issues, this section describes the environment we used to develop the tool. Originally, this project has acquired access to the email dataset of an organisation called 'AvocadoIT' (anonymised). Even though the following algorithms and processes are developed for the AvocadoIT dataset, we used the publicly available, organisation-wide email dataset from Enron in this paper due to access restrictions for the former. While the AvocadoIT set comprises attachments, which are processed alongside the emails by EMCODIST, there are no attachments in the publicly available Enron dataset.

A. Description of the Dataset

The Enron dataset has a total of 495554 emails belonging to 150 custodians within the organisation. We used at least 50 concepts (topics of interest) from Enron data for our work. Most of our evaluation was focused on the top two managers of the organization. This dataset has no attachments tagged along with the emails. While attachments often reveal interesting stories, we ignored them here due to their unavailability. However, the methods developed are suitable to datasets with attachments.

B. Email Data Modeling

The object-oriented approach is used to define the email contextual object for computational modeling. The context-base is a database of context objects. In our definition, the context for email is the tuple of three interdependent 'entities':

$$< who,\ what,\ when >$$

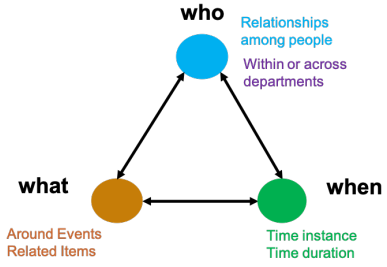

Fig. 1. The context of emails

The entities are uniquely identifiable things such as persons, organizations, and places that can be extracted from the

content/subject or address of an email characterized by their types, attributes, and relationships to other entities. The entity ***who*** represents the people who are connected with a particular event ***what*** during the time instance ***when*** (Fig. 1).

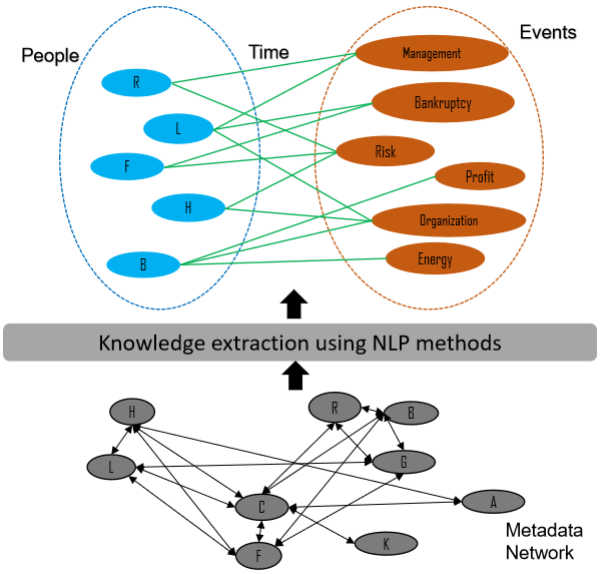

Fig. 2. The metadata to context conversion for information extraction

The relation between the person (*who*) to an event (*what*) is a many-to-many relation, while the time entity (the *when*) represents a period. It is shown in Fig. 2 that the metadata network of emails is combined with the knowledge extracted from emails (from the body and attachments) to create a knowledge base of contextual objects, the <*who, what, when*> triplets. The knowledge abstraction is the processing of information about various people involved, incidents and interesting developments happening within the organisation. A Python implementation of the Context graph is listed in Listing 1.

Listing 1. Context Graph Code Listing

```
class Context:
    def __init__(self, person, event, time): ...
    #interface
    def extractEvent(person): ...
    def findPerson(event): ...
    def findColleagues(person): ...
    def getEdgeWeight(person, event): ...
    def getTimePeriod(event): ...

class ContextGraph:
    def __init__(self, contexts): ...
    #interface
    def getContextGraph(): ...
    def searchGraph(): ...
    def resetEdgeWeight(): ...
```

Usually, emails are represented as graph structures with metadata properties [7], [11], [14]. The nodes on the metadata network of Fig. 2 are the properties of interest connected together. The edges represent the strength of relationship by graph metrics such as degree distribution, diameter, average distance, compactness between nodes, etc. We then implement a knowledge extraction layer over the network graph to get the contextual information of the events from the contents and connection between the timeline and events. We follow a three-step process for knowledge extraction:

- *Metadata network:* Each email is assigned with a unique identifier (id), a primary key to locate an email. Other metadata is constructed into a person-to-person network and person-to-id relation.
- *Event identification:* Techniques of NLP are used to detect names and events from email's subject, body-content and attachments. The events, names (of the people in the email body) and the dates are then grouped to generate a repository of topics/events.
- *The context graph:* The events are connected to the sender/recipient to create a bipartite graph[6] to optimise search time. Each edge between a person to the event is the time when the mail was sent. The person, time and event together represent a contextual object. If there are more emails sent or received from a person describing a particular event, the connection becomes a period during which the person was associated with the event. This gives an idea for how long a person was associated with the event. The edge weight is the number of mails sent/received by a person mentioning the event.

The search for the context is instigated when the user query the context-base using the person or event individually or in a combination mentioning the time. There are two distinct advantages of modelling the email data into contextual objects. (i) In the real-life, the concepts or events are connected to people. The contextual objects can reflect the connections through the combinations of <*person, event, time*>. (ii) Any search algorithm with metadata alone would suffer with the increase in the number of emails. Since the contextual objects are connected with people to events, the increase in the number of emails only alter the edge characteristics (weight and time) of the person-event graph. Thus the graph search algorithm can easily handle the increase in the volume of emails.

C. Sample User Queries on Enron Dataset

Just like any other e-discovery facilities to archival collections, users of EMCODIST can submit their queries as plain English text. Following are the sample queries to Enron email dataset:

1) User query: "The implications on business due to presidential elections": This query is an event led one that looks into the parallel stories around the event 'presidential elections'. The search tool is expected to search for the 'event' of 'elections' and 'business stories' around the time of elections. The user expects a result set to all relevant email threads

[6]In the mathematical field of graph theory, a bipartite graph is a graph whose vertices can be divided into two disjoint and independent sets.

among multiple senders and receivers. The users might want to look at:

- emails that refer to conversation threads about the implications on business,
- examines the set of senders and recipients of these emails to understand who is involved in the conversation, dates referred in the query,
- what other contemporary events and intentions such as political developments that could affect the business.

2) User query: "Customer product price negotiation": The above query is a subjective query. Users expect to see the emails relevant to *price negotiation* of *productA, productB*. They may be interested to see who are the people involved in the *price negotiation* over the length of the email dataset with no specified date.

Finally, when an email dataset is queried for the conceptual information, the contextual objects can return the relevant information from one or more threads across multiple custodians. A user query can be parsed for events to query contextbase to get all the information which is impossible to obtain using metadata alone.

IV. The Working of EMCODIST

Keeping in view the sensitivity and privacy of the data, We have developed the tool as a desktop application against contemporary cloud-based products.

EMCODIST offers two kinds of search over an email collection. The first type of search is targeted at users who are new to a particular email collection. An algorithm developed with the help of attention models (BERT embedding models) [8] interprets the overall meaning of the query and provides themes of emails that align with the general meaning of the query. This type of search allows users to gain high-level insights into the collections which would, in turn, encourage them to refine their search using the second type of search by formulating queries more specific to an event or concept, with a greater knowledge of the context of who was involved and when. In this way, EMCODIST iteratively aids users to gain a complete picture of a concept they are searching for.

As every email collection is unique, the search algorithm needs to be trained on each collection to best support the range of possible users. Another aspect considered in the development of EMCODIST is to help and prioritize the sender/recipient's limited attention and weaving through multiple concepts within a single mail. Thus making a person related to multiple events. The search algorithm ranks emails based on the relevance and allows users to sequentially select from the sorted list.

A. Model 1

The first model is a phrase search model that matches the phrases in the query to the phrases content of emails. This model is an improvement over the basic keyword search and is suitable for expert users. The keywords and phrases from the query are identified with the help of NLP and the model returns all emails and threads that contain the phrase (as a group of words). Since it performs the search on the whole phrase as a single unit, it offers a linguistically better search than a basic keyword search. However, the phrase may not extract the conceptual meaning. When the search query consists of a single phrase, this model performs similar to keyword search. This model is suitable for advanced users who have some idea about the dataset and want to search for specific phrases. A high level process diagram of Model 1 is presented in Fig. 3.

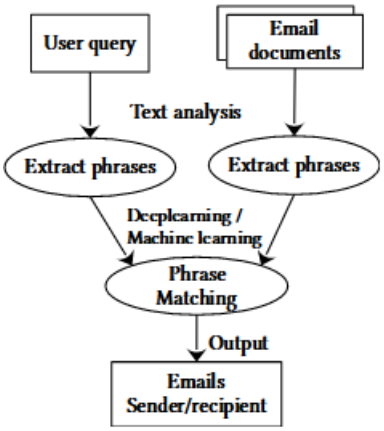

Fig. 3. Process diagram of Model1

B. Model 2

Model 2 matches the conceptual meaning of user queries to the main topics discussed in the corpus. A high level process diagram of Model 2 is presented in Fig. 4. This model makes use of embeddings offered by NLP and pre-training BERT[7] and technology[8] to understand the meaning of the words in context to their neighbouring words in an email's content (contexts). We used 'cosine similarity' to calculate the nearness in this version. Since this model works on similarity, it is essential to set up an ideal threshold to cut-off non-relevant email(threads). A too high threshold may not return any results when there are no matching emails. On the other hand, a too low threshold would return a large number of emails remotely related to the query. The ideal threshold depends on the user requirements and the quality of the contents in the email collection. The algorithm to search and adapt to human input is listed in Algorithm 1.

C. Overall Architecture of the Tool

The architecture used for the development of the tool is shown in Fig. 5. Email collections are pre-processed using NLP (python packages: Spacy, NLTK) to extract knowledge and contextual objects are created based on the contents of the emails and indexed to be searched.

The query submission Interface allows users to formulate/reconstruct their queries and select appropriate additional information to support and improve search capabilities. It also

[7]https://blog.google/products/search/search-language-understanding-bert/
[8]https://github.com/google-research/bert

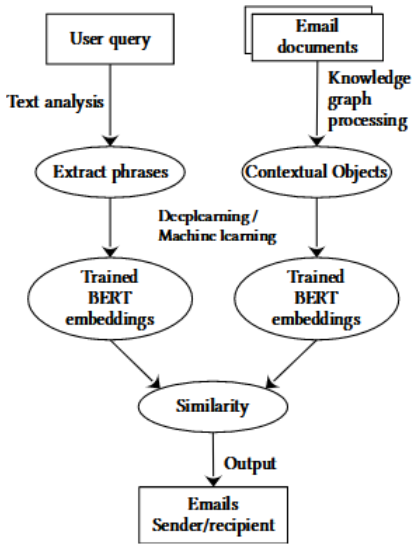

Fig. 4. Process diagram of Model2

Algorithm 1 The context-search algorithm (Model 2)

Require: context-graph with edges set to 0.
Ensure: context-graph not empty
Ensure: threshold greater than zero.
  for Each iteration **do** ▷ user input adaptivity
    email-list ← *empty*
    $X \leftarrow query + userinputs$
    $E(X) \leftarrow embeddings$ of X
    $E(Y) \leftarrow event(context)\ embeddings$
    $match \leftarrow$ E(X) similarity to E(Y)
    if match $\geq$ threshold **then**
      Add person-event edge to email-list
      Add 1 to person-event edge weight
      Get person's other connected events
      for Each connected event **e** in the sorted order **do**
        if embeddings(**e**) $\geq$ threshold **then**
          add edge to email-list
          add 1 to person-**e** edge weight
        end if
      end for
    end if
    Return email-list
  end for

offers users to select *Model 1* or *Model 2* for phrase search or content search.

Initially, both models provide results (email threads) based on the contextual object mappings extracted from the emails (as explained in section IV). The edge weights are calculated as the sum of the number of emails sent or received by a person on a given event. When users add more search terms or other information, the weights on the edges changed. The edge weights of the contextual object graph are used to fine-tune by learning user preferences. In this version, it is assumed that users may want to search the query afresh every time. Hence the system returns to its original state of object mapping with the start of every new search query. We would like to observe whether to retain the fine-tuning information across users in our next versions.

Finally, the output of the system consists of (i) a list of email threads, (ii) timeline plots and (iii) word clouds to help users to fine-tune the tool and adjust their search criteria.

D. Evaluation

To assist users to extract more meaningful insights into the email collections, EMCODIST presents the following types of outcomes to the user query.

- *A set of emails*: A restricted number of the most appropriate emails are listed to users who would like to read through the mails and examine for more details personally.
- *A word cloud*: A word cloud to have a quick look at the specific and important words that appeared in the resultant email list.
- *A timeline graph*: A scatter plot of emails on the timeline to give an insight about when the concept/event/phrase appeared.

E. Evaluation Methods

Since the search tool and models depend on how satisfied users are with the results, the evaluation is designed to test (i) the number of emails relevant to the query submitted, (ii) satisfaction with the rank order of the results and (iii) ability to deliver results for a quick glance as well as the list of emails.

However, it is a common understanding that emails have to be presented after redacting any PII. Hence, we could not design a quantifiable method to measure the readability of the redacted email up to user satisfaction. In some cases, email may be too heavily redacted to allow effective discovery (e.g., EPADD's publicly available email datasets). Also, to our knowledge, EMCODIST is the only tool of its kind. Hence we could not compare EMCODIST with any other tool.

F. Number of Relevant Emails

This method of evaluation examines the results provided by the models and their contextual similarity to the query submitted. A sample evaluation on a set of eleven queries is presented in Fig. 6. The Y-axis represents the queries and fraction of the emails in the resultset relevant to the query. We have chosen three categories of queries. While the Model 1 supposed to return all the mails that have an exact phrase matching, we set of threshold greater than or equal to 0.75 of similarity between the query and the email contents. The relevance for Model 1 is calculated as,

$$\text{Fraction of relevance} = \frac{\text{\# emails appropriate for the query}}{\text{total \# mails returned}} \quad (1)$$

The relevance for Model 2 is calculated as,

$$\text{Fraction of relevance} = \frac{\text{\# emails above threshold}}{\text{total \# mails returned}} \quad (2)$$

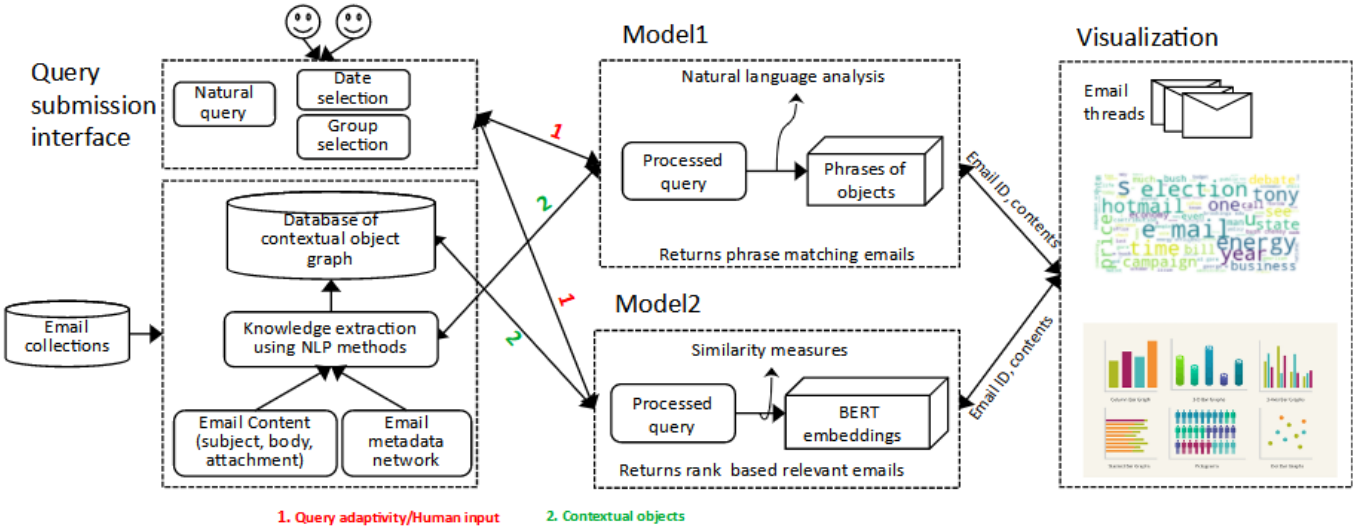

Fig. 5. Overall architecture of the tool

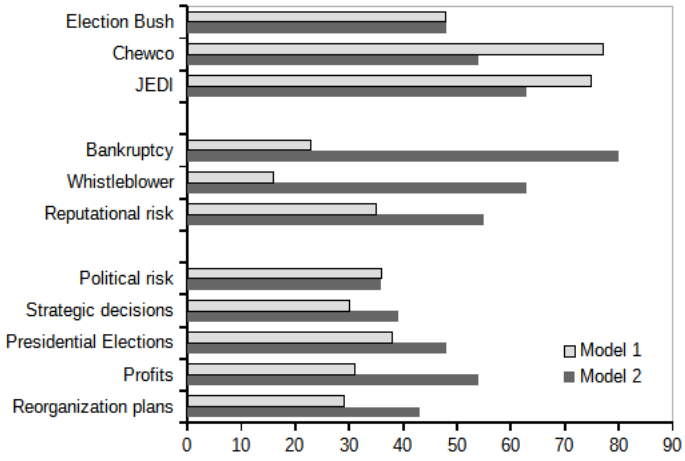

Fig. 6. Fraction of emails for the sample queries submitted

1) Queries with specific keywords: : For the queries that contain words such as 'JEDI', 'Chewco', 'Election Bush', Model 1 performed well. These queries have a good number of appropriate results with the help of the extra input provided by the user. The extra information includes the (i) choosing an appropriate group of mails to search the result from, (ii) a probable start date and end date and (iii) the proper noun such as the name of a company/person. For example, the conversations about the elections where Mr Bush was a candidate were well picked up by Model 1. These queries showed the adaptability of the tool to utilise a user's knowledge to add to the context. The users are well knowledged about the keywords specific to the email corpus and the context in which they were used.

2) Generic queries: : For generic queries such as 'bankruptcy', 'whistleblower', and 'reputational risk', Model 2 returned a larger number of relevant emails based on the conceptual similarity. Even though there was no mention of these words in the conversations, their general concept was similar to the central meaning of the query. We expected this because BERT embeddings are designed to facilitate such 'interpretation'.

3) Queries with multiple meanings: : The third set of queries are the queries with multiple meanings such as 'profits', 'reorganisation plans', 'political risk' etc. The result set returned has many topics belonging to a variety of risks or profits. For example, the query with 'reorganisation' returned emails containing reorganisation of the structure of the official encounters to reorganisation of the office floor plans. For this type of queries, we suggest that user should run the Model 2 first to get an idea of the corpus followed by Model 1 to a focused search. For the query 'Presidential elections', Model 2 gave almost the exact emails as for 'Election Bush'. However, Model 1 could not make the connections of 'presidential election' with the keyword 'Bush'. In all, for content-based queries, Model 2 performed reasonably well. We summarise the characteristics of both models in Table I.

G. Satisfaction with the Rank Order

Model 1 returns all those emails that match the important phrase submitted in the query. Hence the rank order only depends on the sorted date order of the emails. However, Model 2 which works matching the central idea of the query to the email subject/body/attachment content, can rank them in the order of the highest relevance.

H. Scalability of models

Model 1 works on the individual emails from the collections in sequential order. Even though Model 1 has less computational time for the search function, the scalability hampers with the volume of email documents. Whereas Model 2 works on the pre-processed contextual objects. Once the context database is created, search time can go down as the model

TABLE I
MODEL COMPARISON

Concept	Model 1	Model 2
Technique	NLP Phrase matching	BERT embeddings for document Classification and knowledge graph
Query type	Simple phrases	Can handle complex sentences
Scalability	Works well with small volumes of data	Can handle medium to large data volumes
Processing	Sequential processing of emails	Processing depends on the contextual objects
Memory requirements	Volume of data corpus	Volume of data corpus + word embeddings generated by BERT
Target users	Users equipped with specific search phrase	New users to the email corpus
Fairness & Bias	No bias observed	Can induce bias
Accountability & errors	Since this model looks for exact words, some of the resulting emails may not be relevant at all.	Even though best efforts are made to understand the context, often words with multiple meanings may be found
Working with higher models of BERT	Not applicable	Can provide more accurate results with higher versions

learns about the relationship between contextual objects for each query.

V. DISCUSSION

A. The Challenges

Following are two challenges we face with the tool at the moment:

1) The EMCODIST tool works on the content of the emails to find patterns or themes. However, most of the time emails are closed and a large portion of the data is redacted. It could affect the success of the tool to extract relevant information.
2) Like any other AI algorithm, this tool requires initial training on a dataset and human input to give some initial perspective of the data. However, when the tool is deployed to end-users, the tool will enable them to personalise their search and send more specific queries.

Using natural language search terms is something that users commonly do when they use web searches such as Google. The expectations are for a high standard of results, but the difficulty remains for an outsider to understand how their conceptual queries (e.g., developing strategy, policy formulation, approaches to project management) translate into the terminology used within organizations, as well as the departments and individuals involved, and the timing of such processes. Just like AI tools require training on specific data sets, most likely users coming to a new collection with little knowledge of its content will proceed from exploratory general searches (natural language queries) towards more focused searches (keywords identified from reading previous results) as their expert knowledge of a collection grows. The architecture of EMCODIST seeks to facilitate both types of discovery and should allow access to some prior searches to enhance early familiarisation.

VI. CONCLUSION & FUTURE SCOPE

In our project, we have focused on how to provide access to email archives to the users who want to use emails for content discovery. We have developed a prototype tool, EMCODIST to assist users with context-based search from emails. This paper described the first version of the prototype that provides a list of relevant email threads from organisation-wide email archives. Two models were designed to explore the contents of the emails with the help of attention models (BERT) and deep learning. We have defined 'context' suitable for computational modeling. A context-graph is built and modified with each iteration of the user search by improving the edge weights. Also, since the context graph represents the links between persons and events over a period, only the time instance change with the increase in the number of emails. This arrangement makes the structure of the context-graph independent of the increase in the number of emails and hence a scalable solution.

Overall, this paper provided insights into the development of an intelligent system that makes use of AI with a human in the loop to provide a context-based search facility for email archives. At present, the tool starts a fresh search every time there is a new query. It means, the edge weights on the context graph are reset to zero to make the search of each query independent of its previous knowledge. In future, we would like to make use of learnings from the past queries to aide the current search by considering probabilistic edge weights.

ACKNOWLEDGMENT

The EMCODIST Tool is developed as a part of the AHRC US-UK partnership development funded project (AH/T013060/1), 'Historicizing the Dot.com Bubble and Contextualizing Email Archives'.

REFERENCES

[1] B. Aven (2015). The Paradox of Corrupt Networks: An Analysis of Organizational Crime at Enron. Organization Science 26(4): 980–996.
[2] D. Bahdanau, K. Cho, Y. Bengio. (2014.) Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.
[3] J. Beaudoin (2012). Context and Its Role in the Digital Preservation of Cultural Objects. D-Lib Magazine 18. DOI: 10.1045/november2012-beaudoin1
[4] H. Brocks, A. Kranstedt, G. Jaschke, et al. (2009). Modeling Context for Digital Preservation., pp. 197–226. DOI: 10.1007/978-3-642-04584-4_9.
[5] B. Bromiley, R. Christman, S. Page. 2015. "I Really Can't Wait to Archive This Exchange." In Appraisal and Acquisition: Innovative Practices for Archives and Special Collections, ed., Kate Theimer, 31–44. Rowman & Littlefield.
[6] H. Byun, D.A. Kirsch (2020). The Morning Inbox Problem: Email Reply Priorities. Academy of Management Discoveries.
[7] A. Chapanond, M.S. Krishnamoorthy, B. Yener (2005). Graph Theoretic and Spectral Analysis of Enron Email Data. Computational & Mathematical Organization Theory: 265–281. DOI: https://doi.org/10.1007/s10588-005-5381-4.
[8] J. Devlin, M. Chang, K. Lee, K. Toutanova (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". arXiv:1810.04805v2.

[9] S. Decker, D. Kirsch, S. Kuppili Venkata, A. Nix (2021). Finding light in dark archives: Using AI to connect context and content in email. Accepted for publication in the Journal of Knowledge, Culture and Communication, AI & Society (in press).
[10] I. Faniel, E. Yakel (2011). Significant Properties as Contextual Metadata. Journal of Library Metadata 11: 155–165. DOI: 10.1080/19386389.2011.629959
[11] X. Fu, S. Hong, N.S. Nikolov, et al. (2007) Visualization and analysis of email networks. In: 2007 6th International Asia-Pacific Symposium on Visualization, 2007, pp. 1–8. DOI: 10.1109/APVIS.2007.329302.
[12] J. Jett, T. Cole, C. Maden, (2016). The Hathi Trust Research Center workset ontology: A descriptive framework for non-consumptive research collections. Journal of Open Humanities Data 2. Ubiquity Press.
[13] Z. Ji, Z. Lu, and H. Li. (2014). An information retrieval approach to short text conversation. arXiv preprint arXiv:1408.6988
[14] M. Laclav'ik, S. Dlugolinsky, M. Seleng , et al. (2012). Emails as Graph: Relation Discovery in Email Archive. In: Proceedings of the 21st International Conference on World Wide Web, New York, NY, USA, 2012, pp. 841–846. WWW '12 Companion. Association for Computing Machinery. DOI: 10.1145/2187980.2188210.
[15] D. Litman, S. Singh, M. Kearns, and M. Walker. (2000). Njfun: a reinforcement learning spoken dialogue system. In Proceedings of the 2000 ANLP/NAACL Workshop on Conversational systems, pages 17–20. ACL.
[16] R. Mayer, A. Rauber (2009) Establishing Context of Digital Objects' Creation, Content and Usage. In DP'09, June 19, 2009
[17] T. Misu, K. Georgila, A. Leuski, and D. Traum. (2012). Reinforcement learning of question-answering dialogue policies for virtual museum guides. In SIGDIAL, pages 84–93. ACL.
[18] A. Ritter, C. Cherry, W. B. Dolan. (2011) Data-driven response generation in social media. In EMNLP, pages 583–593. Association for Computational Linguistics.
[19] I. Sutskever, O. Vinyals, Q. Le. (2014). Sequence to sequence learning with neural networks. In NIPS, pages 3104–3112.
[20] L. Talboom, D. Underdown (2019). Access is What we are Preserving: But for Whom? - Digital Preservation Coalition.
[21] S. Talja, H. Keso, P. Tarja (1999). The Production of 'Context' in Information Seeking Research: A Metatheoretical View. Information Processing & Management 35(6): 751–763
[22] A. Vaswani, N. Shazeer, N. Parmar, and J. Uszkoreit (2017). "Attention is all you need," arXiv preprint arXiv:1706.03762.
[23] J.D. Williams and S. Young. (2007) Partially observable markov decision processes for spoken dialog systems. Computer Speech & Language, 21(2):393–422

APPENDIX

A. EMCODIST Prototype

The user interface of EMCODIST is designed to allow users to aide the context search. Fig. 7 shows various components on the front-end. These labelled components are listed in Table II. The display of emails for a sample query 'Election Bush' is shown in Fig. 8.

TABLE II
NUMBER DESCRIPTION OF LABELLED USER INTERFACE

#	description
1	search option from topics
2	query input bar
3	time period input
4	simple search (Model 1)
5	advanced search (Model 2)
6	start search process button
7	Optional visualisation
8	Time line plot
9	Word cloud for important words

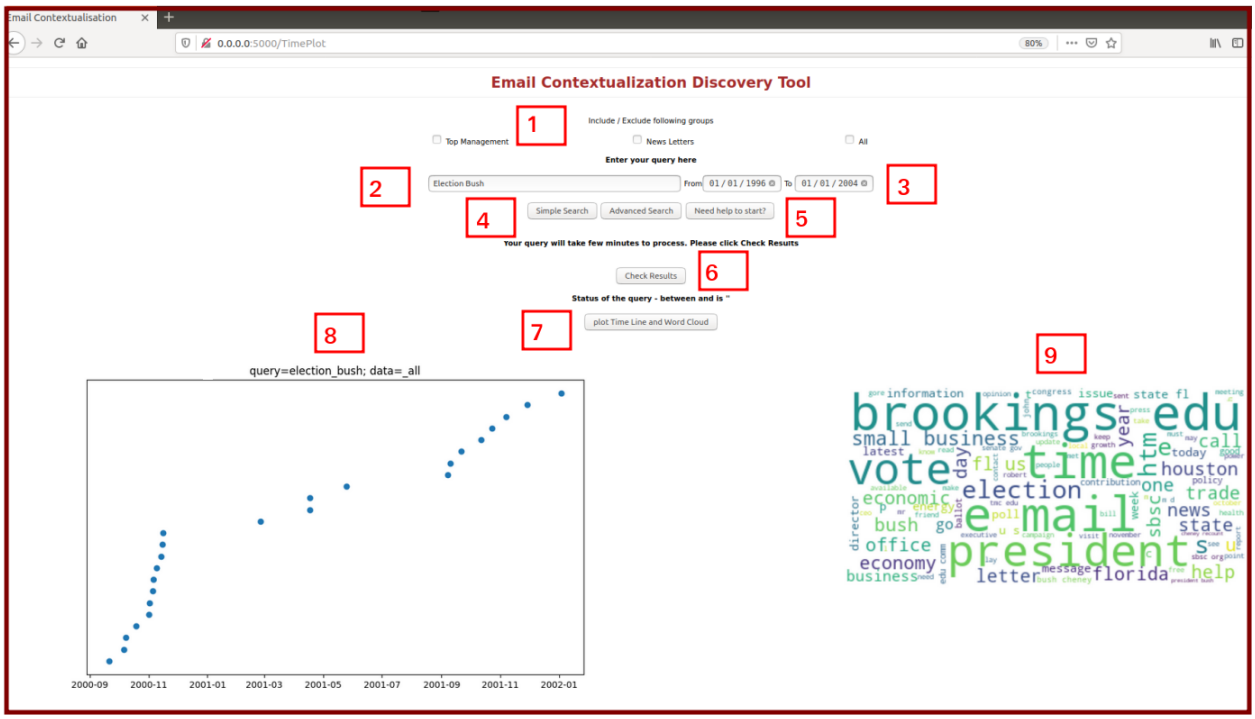

Fig. 7. Labelled User Interface

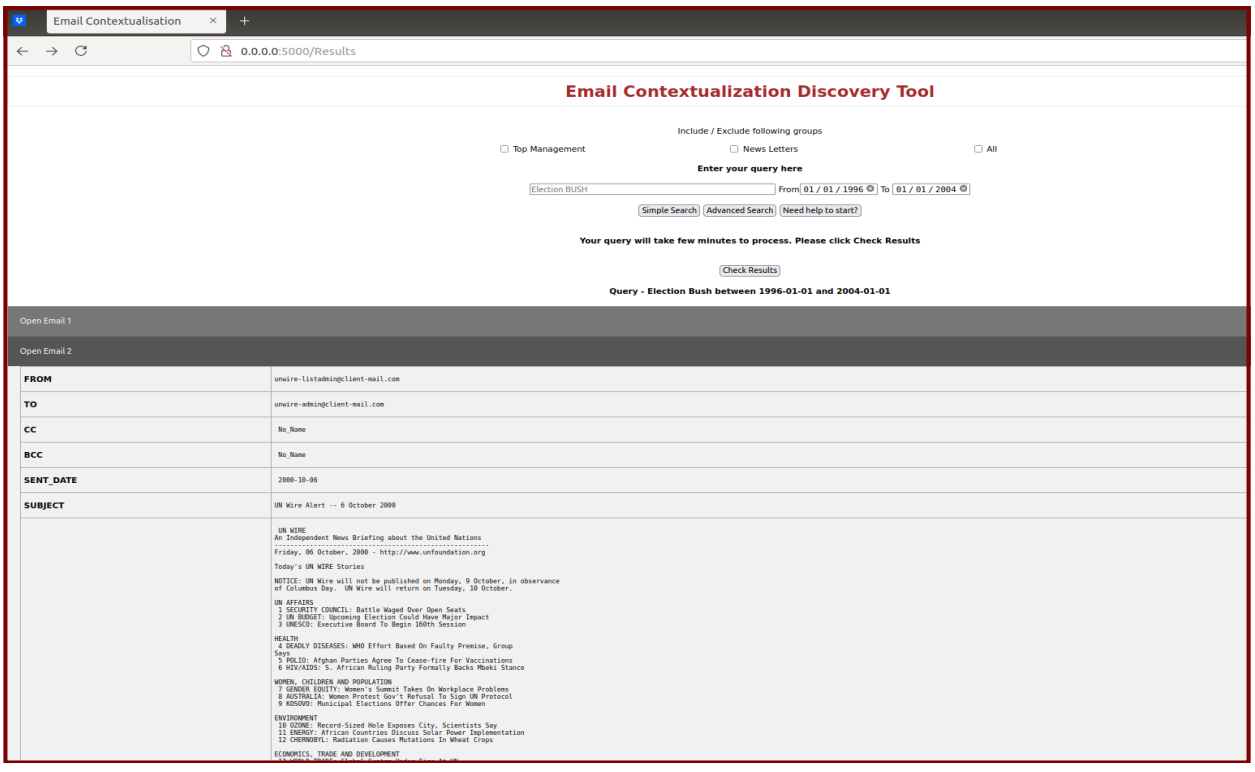

Fig. 8. Resulting emails for the search query ‘Election Bush’