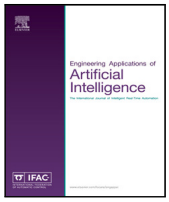
Engineering Applications of Artificial Intelligence 128 (2024) 107450

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Transformer models for mining intents and predicting activities from emails in knowledge-intensive processes

Faria Khandaker [a], Arik Senderovich [b,*], Junda Zhao [c], Eldan Cohen [c], Eric Yu [a], Sebastian Carbajales [d], Allen Chan [d]

[a] Faculty of Information, University of Toronto, Canada
[b] School of Information Technology, York University, Canada
[c] Department of Mechanical and Industrial Engineering, University of Toronto, Canada
[d] Business Automation, IBM, Canada

ARTICLE INFO

Keywords:
Process mining
Knowledge-intensive processes
Intent recognition in emails
Pre-trained transformers
Zero-shot/few-shot learning

ABSTRACT

Process mining is an interdisciplinary field that combines Artificial Intelligence and Business Process Management to extract insights from historical event data. Knowledge-intensive processes, which predominantly involve knowledge work, are often inadequately monitored by process-aware information systems. Consequently, the event data necessary for applying process mining techniques are frequently unavailable. Emails are widely used in knowledge-intensive processes for scheduling meetings, sharing documents, and reporting on the completion of outstanding tasks, which makes them suitable candidates for replacing event logs as the primary data source. In this work, we focus on the task of extracting the set of next recommended activities from incoming emails. Yet, we face two major challenges. Firstly, emails do not express process information explicitly but rather contain subtext that implies what the next best actions would be. Secondly, email data lacks domain-specific labels that would enable the use of machine learning. We overcome these limitations by utilizing an email taxonomy to represent user intents, thus bridging the gap between textual information and process semantics, as well as leveraging pre-trained transformer models applied in zero-shot and few-shot settings that require little to no labeled email data. An evaluation of our method on real-world unlabeled email communications demonstrates its effectiveness in recognizing intents and extracting activities.

1. Introduction

Business processes govern important aspects of our lives: from the trucks that supply our cities with food to the physician appointments that we book over the phone. Processes can be defined as sets of actions taken to complete specific goals and are considered to be the arterial system within organizational supply networks (Dumas et al., 2013). Process mining is a set of techniques that bring together Artificial Intelligence and Business Process Management to extract valuable insights from historical event data (Van Der Aalst, 2016). Applying process mining methods to analyze and improve processes requires an abundance of event data in the form of logs (aka event logs) extracted from process-aware information systems (PAIS) and their databases (Van Der Aalst, 2016).

Some processes are highly structured, with every activity and event explicitly documented and monitored by PAIS, e.g., in production and supply chain settings. Other processes, known as knowledge-intensive processes (KiPs), involve more knowledge work (as their name implies), are more flexible, and can be extremely dynamic (Dustdar et al., 2005). Such processes are encountered in healthcare, software development, and engineering projects. In KiPs different stakeholders may follow highly varying activity orderings, depending on their business goals and intents.

While the existence of event logs is common for structured processes, much of the knowledge work is performed outside PAIS (Di Ciccio et al., 2015). This has led to the realizations that in order to mine KiPs one would require a different input data that would contain information about knowledge work, and that email data is a natural candidate to fill this role (Di Ciccio and Mecella, 2013). Moreover, unlike structured processes where email data can be used to supplement the analysis of event logs, in KiPs email data is one of the few data sources available (Di Ciccio et al., 2015).

In this paper, we consider the task of recommending a set of potential future activities that should be followed upon the receipt of an email. The task can be viewed as prescriptive process mining, which

* Corresponding author.
E-mail address: sariks@yorku.ca (A. Senderovich).

https://doi.org/10.1016/j.engappai.2023.107450
Received 31 December 2022; Received in revised form 23 October 2023; Accepted 3 November 2023
Available online 19 November 2023

aims at predicting the next-best activity in response to the current state of the business process (Huber et al., 2015; Park and van der Aalst, 2022). In the KiP setting, once an email is received, our solution will produce a set of recommended activities, such as sending an email response, scheduling a meeting, or returning a call.

The problem of extracting business process information, including activities, from email data has been studied in the past. van der Aalst et al. presented the EmailAnalyzer, which looks at email participants and tags in the subject to create process mining logs to perform process discovery (van der Aalst and Nikolov, 2007). In Di Ciccio and Mecella (2013), the authors proposed an end-to-end solution dubbed 'MailOfMine' for extracting declarative models from email data using process and activity vocabulary provided by domain experts. Stuit and Wortmann proposed the Email Interaction Miner to produce Interaction Structure diagrams (Stuit and Wortmann, 2012). Others have attempted to extract business processes through investigating intent recognition (Wang et al., 2019), task recognition (Lin et al., 2018; Chambers et al., 2020), and topic modeling (Chambers et al., 2020; Elleuch et al., 2020c; Jlailaty et al., 2019) from the contents of the email body. However, these methods exhibit at least one of two major limitations.

The *primary limitation* of existing research is that it fails to consider the business context when analyzing emails and activities. Current methods focus on extracting process information directly from the data, such as identifying activities and decisions. For instance, in the study referenced as Di Ciccio and Mecella (2013), the approach overlooks the underlying business purpose behind the textual content by directly associating emails with activities. We posit that emails and actions are not exclusively related to business context. Personal emails and non-business meetings can be initiated based on personal correspondence. Similarly, the actions we take in response to emails are not strictly limited to business activities, as we respond to messages from friends and family. In our work, we introduce a novel perspective by considering user business intents expressed in emails as semantic representations of their goals. This perspective helps bridge the gap between emails and activities.

Specifically, we argue that the distinguishing factor between a business-related email or activity and a non-business one is the *intent of the sender*. The recipient comprehends the business intent expressed in the email, interprets it, and takes appropriate actions accordingly. Intent, in essence, refers to the underlying business subtext within the email conversation. Moreover, intents possess universality across various business processes and organizations, remaining consistent regardless of the specific task at hand. Therefore, we propose that intents can serve multiple purposes beyond recommending the most suitable set of activities to pursue. To comprehend the business purpose of emails and extract relevant business activities, we employ the concept of business intent from Carvalho and Cohen (2004) as an intermediate layer of knowledge derived from the data.

The *second limitation* shared by some of the works on email-based process mining is the often unrealistic assumption of possessing an abundance of readily-available labeled email data. The 'MailOfMine' approach proposed by Di Ciccio and Mecella (2013), for instance, assumes the existence of labels when extracting declarative process models from emails. Furthermore, most of the related work employ supervised learning models trained on historical email threads to predict the next activity. This design is not transferable to the real world: manually labeling a large amount of incoming and outgoing emails for training machine learning models is expensive and not flexible. In addition, privacy laws and regulations may hinder companies from using their existing emails as a training corpus for machine learning models (Stuit and Wortmann, 2012). To provide a solution that does not require large amounts of labeled data, we employ pre-trained transformer models in combination with zero-shot and few-shot learning (Brown et al., 2020). These settings require very little (few-shot) or no supervision (zero-shot). In fact, the transformers that we employ were trained on large, publicly available datasets from seemingly unrelated domains.

In order to evaluate our solution, we obtained a collection of email exchanges from knowledge workers in a large organization, and conducted an extensive fields study that comprised two surveys. The first survey is part of our solution pipeline: we asked users to map intents to activities, thus creating a connection between business semantics and actions. This link between intents and activities in a knowledge-based solution ties business semantics and specific user actions. In the second survey, domain experts labeled email sentences by assigning them to several possible business intents (chosen from a finite set of intents based on the intent taxonomy by Cohen et al., 2004), which provided ground-truth for our ML-driven intent extraction. While the former survey is part of our methodology, the latter survey is part of the evaluation. To summarize, the main contributions of this paper are:

1. We present an end-to-end solution for mining business process information from email data. Unlike previous approaches we apply a solution based on an email intent taxonomy to bridge the semantic gap between emails, activities, and business process semantics.
2. The solution is based on zero-shot and few-shot learning techniques to extract user intents from emails and requires little to no data labeling.
3. We present a knowledge-driven mapping of intents to tasks.
4. We provide an extensive evaluation of our solution using real-world email data that we labeled using the results of the field study.

In particular, our evaluation shows that: (i) pre-trained transformer models can classify intents even without training data (zero-shot learning); (ii) using very small amounts of labeled examples (between 3–7 examples) can lead to significant improvement for both intent and task mapping in terms of the accuracy (few-shot learning); (iii) our solution obtains a higher task mapping accuracy (F1 score of 0.79) compared to intent classification accuracy (F1 score of 0.55), since multiple user intents can lead to a similar set of activities. The results are even more promising when considering weaker notions of success.

The current paper is an extended and improved version of Khandaker et al. (2022). Specifically, we provide additional zero-shot and few-shot learning methods (see Section 4.2), in addition to the single zero-shot approach considered in the original paper. Moreover, we significantly extend the evaluation section considering more performance measures, having additional pre-trained models, and supply additional results on zero-shot and few-shot settings (Section 5). Lastly, we introduce a detailed discussion of the results, the limitations of our approach, and promising directions for future work (Section 6), which were missing in the preliminary version of the paper.

The remainder of the paper is organized as follows: Section 2 provides some background on the preliminaries that we employ in the implementation of our approach, followed by a literature overview. Section 3 formalizes the problem we are solving, while Section 4 outlines our solution, providing details on its key components. In Section 5, we discuss the evaluation metrics used in our experiments and present the main findings. In Section 6 we discuss our results, the limitations of our approach, and offer future research directions. Section 7 concludes our work.

2. Background

In this section, we first provide an overview of the preliminaries that our solution requires, namely language models and email intent taxonomies. Then, we present an overview of the relevant literature on email mining in business process analysis and in machine learning, and identify the existing gaps in the state-of-the-art.

2.1. Preliminaries

Our methodology mainly builds on two fields: (1) pre-trained language models in NLP (Brown et al., 2020; Radford et al., 2019; Brown et al., 2020), and (2) email taxonomies for intent representation (Di Ciccio and Mecella, 2013; Elleuch et al., 2020c; Cohen et al., 2004). Below, we provide a detailed review of the techniques that we borrowed from the two fields.

2.1.1. Pre-trained language models and learning settings

Language models allow for computers to ingest and analyze natural language to perform human-like tasks such as question-answering, text summarization, inference and sentiment analysis (Radford et al., 2019), to name a few. Language models like BERT (Devlin et al., 2018), BART (Lewis et al., 2019), and GPT-3 (Brown et al., 2020) that have been pre-trained on large, general datasets can be utilized to improve performance in a different domain or task (Alibadi et al., 2019). Pre-trained models can be used in different settings including fine-tuning, few-shot learning, and zero-shot learning (Brown et al., 2020):

- Supervised fine-tuning (SFT): In SFT, the pre-trained model is adjusted through an additional training period using large amount of labeled data from the target domain (typically thousands to hundreds of thousands of data points). SFT performs very well on the common benchmarks, yet it requires ample data from the desired end-task (Brown et al., 2020).
- Few-shot learning (FS): In FS, the model is given several (typically between 10 and 100) labeled examples from the target domain (Radford et al., 2019). We consider two settings: (1) FS without fine tuning, where one is only allowed to use the labeled examples at inference time without updating the original weights of the pre-trained model (Brown et al., 2020); (2) FS based on specialized fine-tuning techniques that only require small number of labeled examples (e.g., Tunstall et al., 2022; Tam et al., 2021; Mahabadi et al., 2022).

  Clearly, FS requires some labeled examples to work. However, the required amount is reasonably low. One can relatively easily obtain labels for 10 to 100 emails, whereas in standard supervised learning or in SFT, one would require thousands and even hundreds of thousands of labeled examples.
- Zero-shot learning (ZS): Zero-shot learning infers the relationship between a data point and a label without any prior supervision (the name comes from the fact that these methods observe zero training examples) (Brown et al., 2020). The labels in the test set are completely different from those seen by the model during training.

  ZS is the most convenient and flexible (no need for labeled examples) but it is also the most challenging setting. In NLP, ZS has been used in text classification (Sappadla et al., 2016), hate speech detection (Chiu and Alexander, 2021; Pamungkas et al., 2021), and improving spoken language understanding in dialogue agents (Williams, 2019).

In this work, we focus on zero-shot and few-shot approaches that require little to no labeled data.

2.2. Taxonomy and recipient intent

Taxonomy refers to the ordered arrangement of groups and categories. We draw inspiration from Speech Act Theory (Austin, 1965; Searle and Searle, 1969), to identify request and commitment utterances in email communication. Speech Act Theory is based on the study of the philosophy of language and attempts to account for functional meaning of utterances. A primary canon of the Speech Act Theory is that an individual's utterances are also followed by actions, which we find correlated with our view of intents as triggers of activities.

Several taxonomies have been proposed to reliably find ways to identify speech acts within task-based email conversations. Khosravi and Wilks (1999) recognize three classes of requests in email messages, Request-Action, Request-Information, and Request-Permission, leading to the determination of ten possible speech-act categories that combined these acts. Unfortunately, the cue-phrase-based rules they define for identifying requests are specific to the computer support domain from which their email corpus was drawn (Lampert et al., 2008).

Leuski (2004) proposed another speech-act-inspired taxonomy used to categorize email messages and to distinguish the roles of different email authors. The taxonomy focuses on requests: seeking information, advice, action or a meeting. A weakness of Leuski's work is the lack of any detail in the category definitions; the only information provided is a single example sentence or 43 phrases for each category. This makes manual annotation or computational classification nearly impossible.

The taxonomy that we adopt in this work was proposed by Cohen et al. (2004). Specifically, it consists of separate sub-taxonomies for verbs and nouns which describe words commonly used in email exchanges (see Fig. 1). These include categories for request and commitment content, such as Request, Commit, Propose, Amend and Refuse for verbs and activity, information and meeting for nouns. We employ the taxonomy to label sentences with possible intents by creating intent sentences that combine nouns and verbs describing objects and corresponding actions, respectively. Our intent taxonomy is a special case of speech acts information (Elleuch et al., 2020b). The latter is much more elaborate, and would require finding information about actors (email senders and recipients).

2.3. Literature review

In this part, we provide an overview of the most related works in two areas: email mining in business process management (BPM) and process mining (Section 2.3.1), and email intent recognition that employs machine learning techniques in a broader context, independently of the application domain (Section 2.3.2).

2.3.1. Email mining in business process management

The use of emails within organizations, the email storing habits, the types of correspondences which take place between employees, the content of work email, and more, have been widely studied in Dabbish et al. (2005) and Cohen et al. (2004). A survey on existing email mining approaches for business processes can be found in Elleuch et al. (2023).

The MailofMine framework by DiCiccio et al. was one of early works that presented an end-to-end solution for extracting declarative business processes from email data using process relevant vocabulary supplied by domain experts (Di Ciccio and Mecella, 2013). With recent advancements in NLP and Deep Learning technologies, many researchers have built more robust solutions for intent and task recognition. Specifically, Jlailaty et al. proposes the extraction of business activities from emails through use of labeled sentences, process model repositories, cosine similarity, word embeddings and clustering methods (Jlailaty et al., 2019).

Recently Chambers et al. extended the aforementioned solutions by using unsupervised machine learning algorithms to extract actions from emails, cluster the actions into cases, cluster cases as process traces and model the processes (Chambers et al., 2020). Elleuch et al. also take an unsupervised approach to discovering activities in emails (Elleuch, 2021), but through the use of custom developed algorithms which uses the context of the words in an email to derive the idea of a concept pattern that is specific to those emails (Elleuch et al., 2020c,b,d). Moreover, in Elleuch et al. (2020a), the authors propose a meta model for creating enriched event logs from emails. These works heavily rely on the analysis of datasets that contain business process jargon, while our solution is domain independent and light in terms of the data requirements due to our use of zero-shot and few-shot learning

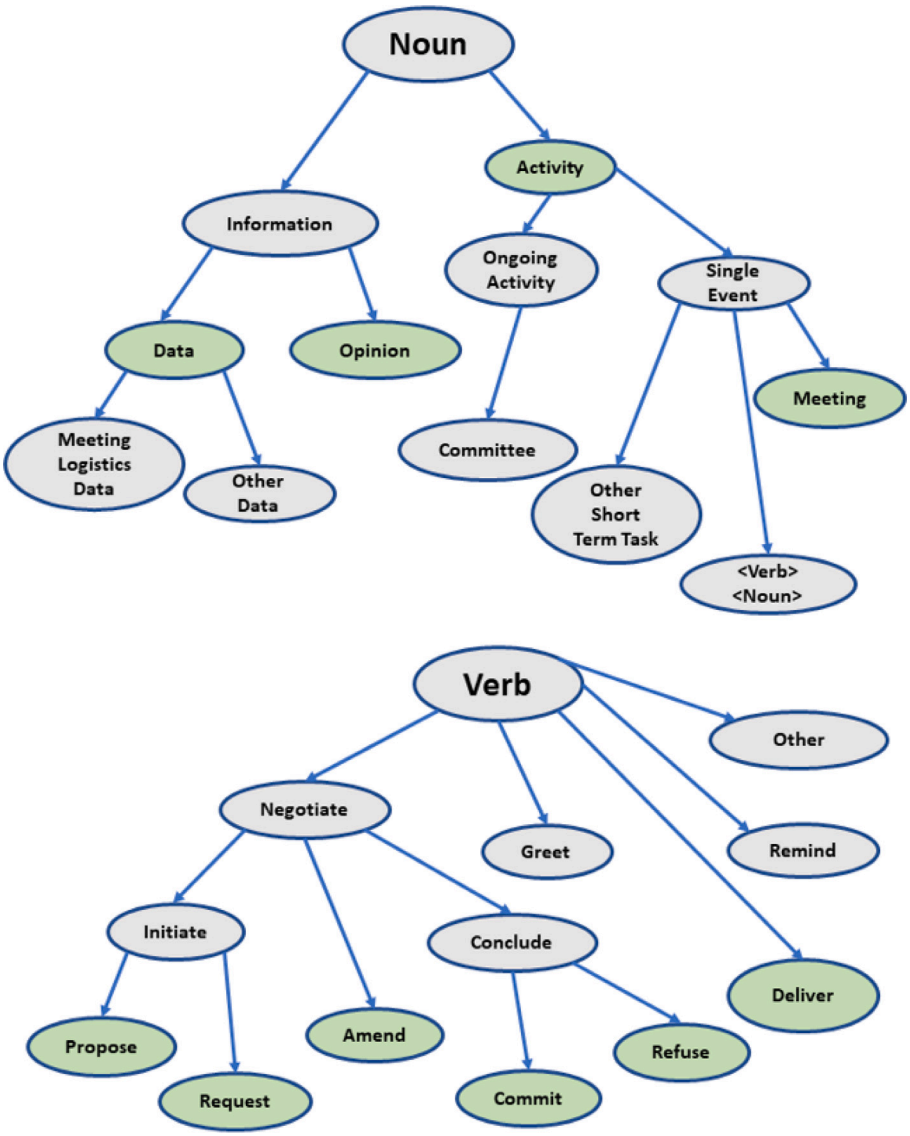


Fig. 1. The diagram of the nouns (top) and the verbs (bottom) which make up the taxonomy detailed in Cohen et al. (2004). The green ellipses indicate those nouns and verbs which were used to create our candidate intent labels.

approaches as well as our use of generalized labels for detecting intents and tasks. In addition, most approaches in Elleuch et al. (2023) use 'traditional' supervised learning and clustering methods, which require labeling and more manual tuning of algorithmic parameters between different business processes, while our work is the first approach to use light (zero-shot or few-shot) learning.

2.3.2. Email and intent mining using machine learning

Early works employed machine learning to extract action items from emails. Corsten et al. were the first to use machine learning to extract 'to-do lists' from emails: the SmartMail tool that they propose first extracts linguistic features from email sentences such as Parts-Of-Speech tags, punctuation, n-grams, etc. Then, it uses a trained SVM model to distinguish between tasks and non-task sentences (Corston-Oliver et al., 2004). In Ulrich (2008), the author proposes the use of supervised machine learning for email text summarization. Lin et al. (2018) learn tasks from emails using a deep learning approach that was trained on the Avocado dataset. The training of their model required a manual email annotation phase by the authors. In our work, we use the task list that was considered in Lin et al. (2018) (Table 3), and also employ deep learning as part of our pipeline. However, the main differences are that we perform intent extraction as an intermediate step prior to task identification, and our method utilizes pre-trained models in zero-shot and few-shot settings that require little to no labeled email data.

Recent literature on email intent recognition apply machine learning approaches to predict recipient and sender intent. Wang et al. employed both traditional ML models (e.g., SVM) and deep learning models to identify user intent from emails (Wang et al., 2019). Alibadi et al. tested six different types of language models including Word2Vec, ELMo, BERT, Deep Averaging Network Based Sentence Encoder, and Transformer Based Sentence Encoder to obtain word embeddings and trained a neural network of their own to classify email sentences with either a 'to-do' intent or a 'to-read' intent (Alibadi et al., 2019). Unlike our solution, both methods required a significant manual annotation effort, and only considered email to intent mapping without extracting the associated tasks. Moreover, it is worth mentioning that our ground-truth labels were obtained from domain experts who were involved in the email exchanges. In contrast, the annotation in Lin et al. (2018), Wang et al. (2019) and Alibadi et al. (2019) was not performed by the stakeholders who participated in the email exchanges. This mismatch may result in biased and less accurate labeling of the email exchanges.

A recent study leveraged the use of 'weak labels' which are created from email interactions through specific labeling functions (Shu et al.,

2020). This was done to overcome the problem of limited annotated data, which our paper also works to address. However, in Shu et al. (2020) the authors proposed specifically curated functions to create weak labels whereas we take advantage of pre-trained language models.

A noticeable body of literature considered intent recognition in settings with limited labeled data in a range of domains including spoken language understanding and question answering (Williams, 2019; Burnyshev et al., 2021; Yin et al., 2019; Ruan et al., 2020; Zhang et al., 2022; Chen et al., 2016). However, these approaches did not consider emails, and were not used to mine business process information from the data.

3. Problem formulation

In this paper, we are aiming to understand which tasks should be performed based on the intents behind an email exchange. Note that we consider intent to be an intermediary knowledge layer, since we assume that intents are the hidden meaning that explains the sender's goals, which can then be connected to future potential actions of the recipients.

Below, we formulate the problem that we subsequently solve using the proposed pipeline (Section 4). Note that we analyze emails at the *sentence level granularity*, assuming that sentences are independent, similarly to existing email mining techniques (see Corston-Oliver et al., 2004; Wang et al., 2019). We aim to relax this assumption in future work.

Formally, let $\mathcal{S}$ be a universe of sentences that can appear in emails and let $\mathcal{I}$ be the universe of possible sender intents, e.g., as they are represented in the Cohen et al. taxonomy (Carvalho and Cohen, 2004). Let $\mathcal{T}$ be the universe of possible tasks (or process activities) that can be performed by the email recipients (or other stakeholders). The universe of multisets over $\mathcal{I}$ is denoted by $\mathcal{B}(\mathcal{I})$.

Our overarching goal is to map an email to a set of tasks. Therefore, we define the main problem that we solve as follows.

Problem 1 (*Email Task Mapping*)**.** Given an email, represented by a set of sentences $S \subseteq \mathcal{S}$, find a mapping of the sentences to a set of tasks $T \subseteq \mathcal{T}$.

In this work, we break the problem into two sub-problems: (1) given an email containing sentences $S \subseteq \mathcal{S}$, we wish to map every sentence $s \in S$ to a set of intents $I_s \subseteq \mathcal{I}$; and (2) map the (multiset) union of the discovered sentence intents, $I = \biguplus_{s \in S} I_s$, to a set of tasks $T \subseteq \mathcal{T}$. In other words, we assume that sentences can contain multiple intents (multiset I), which can then be mapped to multiple tasks (T).[1] In essence, we wish to solve a dual matching problem: *classify* sentences into intents, and then *map* multi-sets of intents to actionable tasks. The two problems are formalized as follows.

Problem 2 (*Intent Classification*)**.** Given a set of email sentences S, find a function ϕ that classifies sentences in S into a set of intents, i.e., $\phi : \mathcal{S} \rightarrow 2^{\mathcal{I}}$

Problem 3 (*Task Mapping*)**.** Given a multiset of intents I, find a function θ that maps I into a set of tasks, i.e., $\theta : \mathcal{B}(\mathcal{I}) \rightarrow 2^{\mathcal{T}}$.

For example, the sentence 'Let us schedule a meeting on Monday at 10 AM to discuss the document you have sent' can be classified onto the verb and noun labels of "proposing" and "meeting" respectively (see Fig. 1). Then, these two intents can be then mapped onto an activity of 'Set-up Appointment'. Clearly, both intent classification and task mapping are in essence multi-label classification problems, where sentences are mapped to multiple intents, and then multiple intents are mapped to multiple tasks. Solving Problems 2 and 3 leads to the solution of Problem 1.

[1] We switch from sentence-level intent sets to email-level multisets of intents to account for repeating intents in the email.

Table 1
Intent Labels used to classify the email sentences adapted from the Cohen et al. (2004) taxonomy.

#	Intent label
1	A meeting is being committed
2	A meeting is being proposed
3	A meeting is being refused
4	A meeting is being requested
5	An activity is being committed
6	An activity is being proposed
7	An activity is being refused
8	An activity is being requested
9	An opinion is being amended
10	An opinion is being delivered
11	An opinion is being requested
12	Data is being amended
13	Data is being delivered
14	Data is being requested

4. Solution for intent classification and task mapping

Fig. 2 presents an overview of our solution to Problem 1. When an email message enters the inbox, the body of the message undergoes preprocessing through the *Email Processor* module where it is cleaned (e.g., stripped of URLs, bullet points, special punctuation marks, etc.), split into sentences and gets tokenized. These sentences are then fed into the *Intent Classifier*, and each sentence is classified into a set of intents, thus providing ϕ. The intents are then sent into the *Task Mapper* module to be mapped to specific tasks, thus, in essence, providing θ.

The Email Processor employs standard NLP preprocessing steps such as cleaning for punctuation and handling special symbols. Passive information such as a copy-pasted memos are removed from the email bodies to only keep sentences which are relevant for intents and tasks. In the end, a set of tokenized sentences goes into the intent classification component. Since Email Processor is a standard application of existing methods, in the remainder of the section, we only provide details on the main two components of our solution, namely the Intent Classifier and the Task Mapper.

4.1. Intent classifier

In this part, we solve Problem 2, by first defining the set of possible email intents, and then constructing ϕ by leveraging pre-trained transformer models. We start by creating a universe of intents, $\mathcal{I}$, from a well-known intent taxonomy. In this work, we used the Cohen et al. taxonomy (Cohen et al., 2004), which is a pair of trees that describes the categories of verbs and nouns frequently used in organizational emails (see Fig. 1). To create $\mathcal{I}$, we flattened the taxonomy tree into the intent labels in Table 1 as our current approach does not support hierarchical information.

To add context of how the noun and the verb should be understood together (example: deliver can refer to the giving of birth as well as the giving of an arbitrary object) the candidate intent labels are used to create present tense sentences. For example, instead of 'Request Activity', the sentence 'An activity is being requested' is used as the classification label.

Based on the universe of intent labels, we consider two zero-shot learning approaches and two few-shot learning approaches for mapping sentences to intents, with all approaches leveraging transformer language models (Vaswani et al., 2017) that were pre-trained on large scale, publicly available text datasets. Note that all zero-shot and few-shot approaches we consider return an ordered set of intent labels (e.g., according to the probability of how well each intent matches the input sequence). We can therefore obtain more than one intent per-email by selecting the most likely intents predicted by the models. Specifically, given an email represented by a set of sentences $S \subseteq \mathcal{S}$, the zero-shot or few-shot transformer-based approaches act as the main

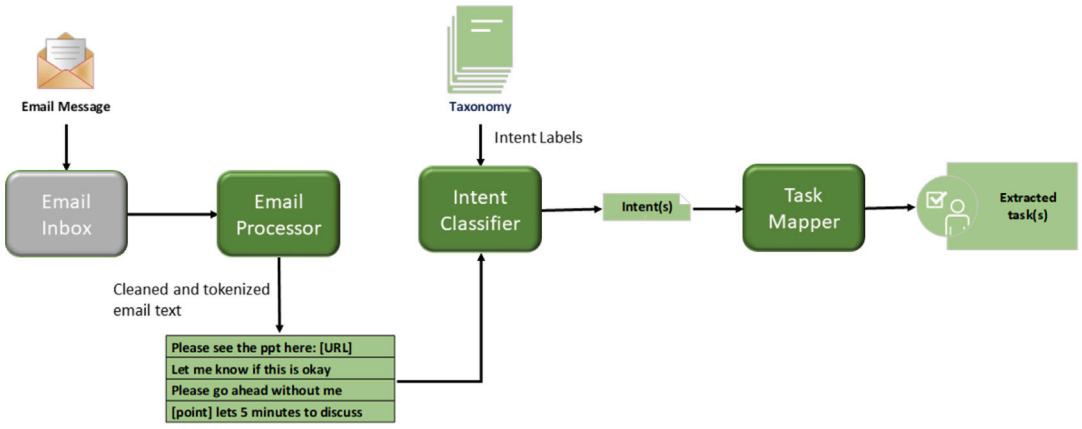


Fig. 2. The Email Intent Classification and Task Mapping Solution.

building block of ϕ, producing sets of intents for a given sentence, namely $\phi(s) = I_s \subseteq I, \forall s \in S$.

In what follows, we provide the details on the zero-shot approaches (Section 4.1.1), and the few-shot approaches (Section 4.1.2) we consider for intent classification. Lastly, in Section 4.1.3 we describe the nine pre-trained transformer models that we employ.

4.1.1. Zero-shot learning approaches

In Zero-Shot Text Classification, the classes used in query-time have not been seen before during training-time and are only represented by class labels that are sequences of words (Sappadla et al., 2016). In our setting, the class labels are the universe of intents I, and the classifier is not exposed to these class labels during training-time. We consider two zero-shot learning approaches as follows.

Similarity-based zero-shot learning. Similarity-based zero-shot approaches for text classification can embed both the texts and the class labels into the same embedding space and compute the similarity between them in this embedding (Sappadla et al., 2016). We consider a transformer model that was pre-trained on a large-scale text corpus and use it to embed both the email sentences as well as our universe of intents and for each email sentence selects the intent(s) that have the smallest distance in the embedded space, utilizing either the cosine or euclidean distance metrics. To derive a fixed size sentence embedding from the transformer model, we used a mean pooling operation.[2]

Entailment-based zero-shot learning. Entailment-based approaches cast the problem of zero-shot text classification as a textual entailment problem (Yin et al., 2019). Textual entailment models get a *textual premise* and a *textual hypothesis* and learn to determine whether the premise entails the hypothesis. These models can be pre-trained on existing large-scale textual entailment datasets and later used for zero-shot classification. Assuming an input sentence $s \in S$ and a candidate class label from our intent universe $i \in I$ (that was not seen by the model during training), we use the input sentence s as the premise and set the hypothesis to be "This example is i". The result is interpreted as the probability that sentence s belongs to class i.[3]

4.1.2. Few-shot learning approaches

In few shot learning approaches, we have a small number of labeled examples for each class (in this work, we consider less than 10 examples per class) that can be provided by domain experts. Few-shot text classification employs pre-trained large-scale language models and can rapidly generalize to new tasks using only a small number of examples with supervised information (Wang et al., 2020).

Similarity-based few-shot learning. In similarity-based zero-shot classification we use the distance between sentences and labels in the embedding space. Instead, in similarity-based few-shot classification we assume a small number of labeled examples and use the distance in the embedding space between an input sentence and the small number of labeled examples to determine the class labels for the input sentence. To determine the class labels for an input sentence, we employ ML-KNN, a k-nearest neighbors approach with multiple labels (Zhang and Zhou, 2007) that selects the class labels based on their frequency in the k labeled examples that are the nearest ones in the embedding space.[4] In our experiments, we set the number of neighbors $k = 3$.

Few-shot learning based on fine-tuning pre-trained models. A different approach for few-shot learning relies on fine-tuning pre-trained model using a small number of labeled examples. In a recent work Tunstall et al. (2022) proposed SetFit, an approach for few-shot classification that is based on two steps: (1) fine-tuning a sentence transformer in a contrastive, Siamese manner based on pairs of labeled examples; (2) training a classification head using the encoded training examples generated by the fine-tuned sentence transformer in the first step.[5]

4.1.3. Pre-trained transformer language models

The zero-shot and few-shot classification approaches in Sections 4.1.1 and 4.1.2 rely on large-scale pre-trained transformer models. In our experiments, we have selected a set of popular Transformer-based architectures including the well-known BERT, its distilled version DistilBERT, as well as the more recent architectures RoBERTa, BART, XLM-RoBERTa, and DeBERTa and obtain pre-trained models from the HuggingFace model repository.[6]

1. Bidirectional Encoder Representations from Transformers (BERT), a Transformer model pre-trained on a large corpus of English data in a self-supervised fashion on two main objectives, namely masked language modeling and next sentence prediction (Devlin et al., 2018).
2. DistilBERT is a distilled version of the BERT model, i.e., it is a smaller and faster version of BERT trained in a self-supervised fashion, using the BERT base model as a teacher (Sanh et al., 2019).
3. RoBERTa (Liu et al., 2019), a robustly optimized version of BERT where the masking pattern applied to the training dynamically changes during the training (Williams et al., 2018).

[2] We used the implementation in the SentenceTransformers library (Reimers and Gurevych, 2019).

[3] We used the implementation of entailment-based zero-shot classification in the Hugging Face library (Wolf et al., 2019).

[4] We use the implementation in `scikit-multilearn` (Szymaĺ et al., 2019).

[5] We use the official implementation in https://github.com/huggingface/setfit.

[6] https://huggingface.co/models

Table 2
Pre-trained transformer models considered in the work. Sim: Similarity, Ent: Entailment, SF: SetFit.

#	Short name	URI
Zero-shot and Few-shot Similarity-based approaches:		
1	BERT	`bert-base-uncased`
2	DistilBert	`distilbert-base-uncased`
3	BART	`facebook/bart-large`
4	RoBERTa	`roberta-large`
5	XLM-R	`xlm-roberta-large`
6	DeBERTa	`microsoft/deberta-large`
Zero-shot Entailment-based approach:		
7	BERT-M	`yoshitomo-matsubara/bert-large-uncased-mnli`
8	DistilBert-M	`typeform/distilbert-base-uncased-mnli`
9	BART-M	`facebook/bart-large-mnli`
10	RoBERTa-M	`roberta-large-mnli`
11	XLM-R-M	`vicgalle/xlm-roberta-large-xnli-anli`
12	DeBERTa-M	`microsoft/deberta-large-mnli`
Few-shot SetFit approach:		
13	Paraphrase	`sentence-transformers/paraphrase-mpnet-base-v2`
14	SentRoBERTa	`sentence-transformers/all-roberta-large-v1`

4. XLM-RoBERTa, a multilingual version of RoBERTa trained using more than two terabytes of filtered CommonCrawl data containing 100 languages (Conneau et al., 2020).
5. DeBERTa, a recent model that improves the BERT and RoBERTa models using two novel techniques, namely a novel disentangled attention mechanism and an enhanced mask decoder (He et al., 2021).

For our similarity-based zero-shot and few-shot approaches, we use mean pooling operation to obtain fixed size embedding that can be used to compute similarity. For the entailment-based zero-shot approach, we used the same set of Transformer-based architectures and select pre-trained models that were fine-tuned on the Multi-Genre Natural Language Inference (MNLI) dataset, a collection of 433,000 sentence pairs annotated with textual entailment information. Finally, to support the SetFit few-shot approach, we have included the models Paraphrase and SentRoBERTa that were used in the original work (Tunstall et al., 2022). Both models are based on the Sentence Transformer model that use siamese and triplet network structures to derive semantically meaningful sentence embeddings (Reimers and Gurevych, 2019).

1. Paraphrase-MPNet is a Sentence Transformer model trained using MPNet (Song et al., 2020) on English paraphrase corpus (Reimers and Gurevych, 2020).
2. SentRoBERTa is a Sentence Transformer trained by fine-tuning RoBERTa (Liu et al., 2019) on 1B pairs of similar sentences.

Table 2 summarizes the details of the 14 pre-trained models used in our experiments, including a short name we will use to refer to each model in this work, the unique identifier for each model on the Hugging Face model repository, and the approaches that utilize each model.

4.2. Task mapper

In this part, we provide our solution to Problem 3, which is based on a labeling provided by domain experts. Specifically, we only use domain knowledge, not ML, for mapping intents to activities. This knowledge is derived from a field study that involves domain experts. Once intents are matched to activities, a mapping between sentences and activities is created. This then leads to recommending the next action.

Technically, we construct an intent-to-task function θ by conducting a survey among the experts that mapped individual intents to sets of possible tasks, and then aggregating the results of the field study to create a single task set per email. Once the intent set has been extracted, it needs to be further processed to be useful for the recipient. Denote $I_s = \phi(s)$ the set of extracted intents from sentence $s \in S$ and let $I = \biguplus_{s \in S} I_s$ be the multiset of intents appearing in the email represented by S. In this work, we consider the set of email tasks proposed in Lin et al. (2018) (see Table 3 for details). For example, intents can be mapped to possible tasks such as 'Reply-YesNo', 'Reply-Ack', 'Reply-Other'; each slightly differ from each other as mentioned in Lin et al. (2018).

To create the mapping, we conducted a field study for obtaining the ground truth for intent classification and a mapping between intents and activities. Below, we provide details on the field study.

4.2.1. Field study

Field studies are a common practice in various fields of computer science, e.g., information systems and software development (c.f., El Emam and Madhavji, 1995). Our field study involved six knowledge workers who participated in the email exchanges; thus, we considered those workers to be domain experts. The exchange included 64 emails in two email chains (having 8 and 56 emails each, respectively); the total number of sentences that were considered in the 64 emails was 214. The field study comprised two surveys:

1. Mapping intents to tasks: The experts had to label intents from our taxonomy (Table 1) to a set of actions proposed in Table 3. Specifically, in order to create rules which define what tasks are most likely to be associated with our intent labels, we asked the participants to fill out a survey where each of our intent labels were shown and the participants were required to select all tasks which they deemed relevant for said intent. Processing the results of the survey included assigning each task with a weight based on how many participants selected it to be relevant for the intent in question. For example, if all of our domain experts selected the task "Reply Other" for the intent label "An opinion is being delivered", then that task would have a weight of 1, which is the highest weight possible according to our criteria. The weights of each task per email body were summed together and ordered.
2. Mapping sentences to intents: The second part, of mapping sentences to intents, serves us as ground-truth when assessing the effectiveness of the intent classification component. The participants had to link sentences to the two most representative intents in Table 1. In our experiments, we used $K = 2$, since experiments conducted by Wang et al. (2019) showed that emails typically contain 1–2 intents.

Factors which may impact survey responses such as response rate and response order effects were addressed using standard techniques (Krosnick, 1999).

Table 3
Table of recipient tasks and their descriptions borrowed from the Lin et al. (2018).

#	Task labels	Descriptions
1	Reply-YesNo	Short Yes-no replies to questions
2	Reply-Ack	Acknowledgement such as 'got it'
3	Reply-Other	Email replies without investigating some resources
4	Investigate	Gathering additional information to address issue
5	Send New-Email	Sending an email that is not a part of the current thread
6	Set-up Appointment	Set up or cancel appointments
7	Approve Request	Approve requests
8	Share content	Share contents such as an attachment
9	No Action	No action is required for this email

4.2.2. Task mapping

In this part, we describe how we leveraged the results of the field study to construct θ, thus creating the Task Mapper component of our solution. From the first survey, we obtained a weighted list of tasks mapped to each intent in $\mathcal{I}$. Denote $\omega(i,t), i \in \mathcal{I}, t \in \mathcal{T}$ the weight of task t for intent i. The weighted mapping $\omega(i,t)$ is a sentence-level intent to task mapping. In the remainder, we aggregate sentence-level intent to task weights into a single email-based intent-to-task mapping.

Recall that based on the results of the Intent Classifier ϕ for each sentence $s \in S$, we can compute the email-level multiset of intents, I, with the multiplicity of each element corresponding to the number of times it appeared in the email. Let $\lambda(i), i \in I$ be the multiplicity of i in the multiset and let $\bar{I}$ be the set of distinct intents in I. Further, let $\omega(I,t)$ be the aggregated weight of a task for an intent set $I \subseteq \mathcal{I}$, which we propose to compute as,

$$\omega(I,t) = \sum_{i \in \bar{I}} \lambda(i)\omega(i,t). \tag{1}$$

The intent weights emphasize intents that appear in more sentences by inflating their weights with respect to the original individual mappings. To create a final set of tasks for the intent set I, we return the top-2 tasks with the highest associated weights. Clearly, this is a design choice that can be further examined in future work.

5. Evaluation

In this part, we present the empirical evaluation of our solution, demonstrating its relevance and ability to detect intents and map these intents to tasks. In Section 5.1, we present the data that we used for zero-shot and few-shot learning. Then, in Section 5.2 we provide an overview of the multi-label measures of success that we employed to assess our pipeline. And lastly, in Section 5.4, we outline the main results of our empirical evaluation.

5.1. The dataset

One of the motivations for our work is the lack of labeled email data (in real-world applications). However, in order to evaluate the accuracy of the proposed pipeline (Section 4), we require a dataset that contains the ground-truth. Yet, since all the datasets we used were unlabeled, we had to perform a labeling procedure. To this end, we conducted the second part of our field study (see Section 4.2 for details), during which participants of two email chains labeled all the sentences with the two most representative intents from our list of flattened intent labels (see Table 1). The results of this annotation were used as test data in our experiments. Table 4 provides a set of descriptive statistics that summarize the data.

Table 4
Descriptive Statistics of our Dataset.

Statistic	Value
Total number of emails	66
Total number of sentences	241
Average sentence length (in words)	13.39
Word count	3228
Number of unique words	800 (case insensitive)

5.2. Evaluation measures

We analyzed and evaluated the results using common supervised learning metrics for multi-label classification. Below, we provide details on the three measures that we chose, namely F1 score, Jaccard similarity, and 1-Accuracy. In order to formally define these measures, we must first introduce the following notation.

Let $\mathcal{X}$ be the universe of labels (e.g., $\mathcal{X} = \mathcal{I}$ for intents and $\mathcal{X} = \mathcal{T}$ for tasks). We consider a sample of N items (e.g., N sentences), each needs to be classified into a subset of $\mathcal{X}$. Denote $X_i \subseteq \mathcal{X}$ the ground-truth class set for item $i \in \{1, \dots, N\}$ (a set, since we are in a multi-label classification setting), and let $Y_i \subseteq \mathcal{X}$ denote the labels predicted by a classifier.

F1 score. In order to define the F1 score in a multi-label setting, we must first define precision and recall and then aggregate them together in a meaningful way. In this work, we consider the micro-average precision and recall, which yield a micro-average F1 score (Pillai et al., 2012).

Let $tp(x), fp(x), fn(x)$ be the number of true positives, false positives, and false negatives for a label $x \in \mathcal{X}$. Those metrics are computed over all N examples in the dataset. For example, $tp(x)$ is the number of times label x was predicted to be relevant for the N items, out of the number of times it was in their ground-truth label set.

Then the micro averaged precision and recall are defined as follows:

$$microP = \frac{\sum_{x \in \mathcal{X}} tp(x)}{\sum_{x \in \mathcal{X}} fp(x) + tp(x)}, \qquad microR = \frac{\sum_{x \in \mathcal{X}} tp(x)}{\sum_{x \in \mathcal{X}} fn(x) + tp(x)}, \tag{2}$$

respectively. Subsequently, the micro-average F1 score is defined as the harmonic average of $microP$ and $microR$:

$$F1 = 2\frac{microP \times microR}{microP + microR}.$$

Jaccard similarity. Given two sets X and Y, the Jaccard similarity measure is defined as the ratio between the size of the intersection between X and Y (which corresponds to their similarity) and the union of X and Y, which normalizes the similarity to be at most 1 (Gouk et al., 2016), i.e.,

$$Jaccard(X,Y) = \frac{|X \cap Y|}{|X \cup Y|}.$$

The average Jaccard similarity score that we report in our results is calculated as

$$Jaccard = \frac{1}{N}\sum_{i=1}^{N} Jaccard(X_i, Y_i).$$

1-Accuracy. Jaccard similarity is also known as the *Accuracy* measure in the multi-label classification literature, since a full score is obtained only when there is a perfect match (Ganda and Buch, 2018; Brighi et al., 2021). We relax the notion of accuracy in Jaccard similarity by introducing the 1-Accuracy measure. Specifically, in 1-Accuracy, we

Table 5
Results for zero-shot classification.

Approach	Embedding	F1 score	Jaccard score	1-Accuracy
Entailment-based	BERT-M	0.264	0.190	0.486
	DistilBERT-M	0.266	0.192	0.491
	BART-M	0.325	0.243	0.570
	RoBERTa-M	0.371	0.274	**0.664**
	XLM-R-M	0.306	0.227	0.542
	DeBERTa-M	**0.383**	**0.296**	0.645
	average	0.319	0.237	0.566
Similarity-based	BERT	0.250	0.179	0.463
	DistilBERT	0.178	0.128	0.327
	BART	0.224	0.156	0.430
	RoBERTa	0.161	0.111	0.313
	XLM-R	0.154	0.103	0.308
	DeBERTa	0.315	0.221	0.598
	average	0.214	0.150	0.407
	Random	0.121	0.084	0.234

propose to count a success to be the appearance of at least one element of an origin set X in target set Y. Let $\mathbb{I}_{X,Y}$ be the indicator that the intersection between two sets X and Y is nonempty, i.e., $|X \cap Y| > 0$. Then, the average 1-Accuracy score is defined as

$$\frac{1}{N}\sum_{i=1}^{N}\mathbb{I}_{X_i,Y_i}.$$

Clearly, 1-Accuracy provides a weaker notion of success compared to the Jaccard similarity score, since in the latter, the size of the intersection matters.

5.3. Random baseline

To provide a simple baseline, we have included a random classifier that predicts, uniformly at random, two intents for each sentence. We then evaluate and report its performance according to the evaluation metrics to support the interpretation of the results of our zero-shot and few-shot approaches.

5.4. Results

In the experiment, we applied each of the methods mentioned in Section 4.2, and then used the results of the field studies (mentioned in Sections 4.2 and 5.1) as ground truth to measure the accuracy of intent classification and task matching. Below, we report the main findings of the experiment.

5.4.1. Intent classification

We evaluate the performance of the two zero-shot and two few-shot classification approaches for intent identification. The ground truth intent labels obtained from the field study are compared against the predicted labels to compute the evaluation metrics for each of the approaches.

Results for zero-shot approaches. Table 5 shows the performance for the two zero-shot approaches described in Section 4.1.1, similarity-based and entailment-based. Each of the approaches was tested for a range of applicable pre-trained embeddings (Section 4.1.3). In addition, we report the average metrics for each of the approaches over all embeddings.

Table 5 shows that the entailment-based approach performs, on average, better than the similarity-based approach. In fact, the embedding with the worst performance in the entailment-based approach (BERT-M) still outperforms all but one of the embeddings in the similarity-based approach.

When comparing the different embeddings used in the entailment-based approach, we found that DeBERTa outperformed all other approaches in terms of F1 score and Jaccard score, while obtaining the second-best performance in terms of 1-Accuracy score. In the similarity-based approach, we find that DeBERTa significantly outperformed all other models across all evaluation metrics.

Finally, we note that the best configuration significantly outperforms the random baseline. Further, even the worse performing approach (XLM-R) still clearly outperforms the random baseline.

Table 6
Result for few-shot classification using five labeled examples per class.

Approach	Embedding	F1 score	Jaccard score	1-Accuracy
SetFit	SentRoBERTa	0.478	0.383	0.763
	Paraphrase	**0.537**	**0.434**	**0.848**
	average	0.508	0.408	0.806
Similarity-based	BERT	0.401	0.306	0.688
	DistilBERT	0.391	0.299	0.668
	BART	0.363	0.270	0.640
	RoBERTa	0.370	0.279	0.640
	XLM-R	0.352	0.267	0.608
	DeBERTa	0.366	0.275	0.640
	average	0.374	0.283	0.647
	Random	0.121	0.084	0.234

Table 7
F1 score for few-shot classification for different number of labeled examples.

Approach	Embedding	3 Examples	5 Examples	7 Examples
SetFit	SentRoBERTa	0.430	0.478	0.519
	Paraphrase	**0.468**	**0.537**	**0.554**
	average	0.449	0.508	0.537
Similarity-based	BERT	0.367	0.401	0.410
	DistilBERT	0.395	0.391	0.410
	BART	0.332	0.363	0.362
	RoBERTa	0.319	0.370	0.366
	XLM-R	0.315	0.352	0.362
	DeBERTa	0.341	0.366	0.341
	average	0.345	0.374	0.375

Results for few-shot approaches. Table 6 shows the performance of few-shot classification using 5 labeled examples per class for the two approaches described in Section 4.1.2, namely the similarity-based KNN classifier, and the SetFit classifier that is based on contrastive fine-tuning and training of a classification head. The similarity-based KNN approach was tested for the selected list of pre-trained embeddings (Section 4.1.3) and the SetFit approach was tested using the embeddings used in the original work (namely, SentRoBERTa and Paraphrase). We selected the labeled examples randomly based on the ground-truth labels from the field-study and performed the evaluation on the remaining data points. To reduce bias due to a specific selection of labeled examples, we repeated the analysis five times based on five different randomly selected sets of labeled examples and report the average evaluation metrics.

Table 6 shows that both approaches seem to obtain better performance, on average, compared to zero-shot approaches. When comparing the few-shot approaches, SetFit has a significant advantage over similarity-based approach. Moreover, SetFit based on fine-tuning Paraphrase embedding significantly outperforms the best similarity-based embedding in each of the evaluation metrics. All few-shot approaches, including the worst performing approach (Similarity-based XLM-R), exhibit significant improvement over the random baseline.

To analyze the impact of the number of labeled examples, Table 7, Table 8, and Table 9, show the F1 score, the Jaccard score, and the 1-Accuracy, respectively, for varying number of labeled examples per class in $\{3, 5, 7\}$. For the SetFit approach, we can clearly see significant improvement across all metrics for each additional two examples. For similarity-based few-shot learning the improvement that we observe is, on average, more limited and in some cases additional labeled examples can actually degrade the performance.

Table 8
Jaccard score for few-shot classification for different number of labeled examples.

Approach	Embedding	3 Examples	5 Examples	7 Examples
SetFit	SentRoBERTa	0.343	0.383	0.427
	Paraphrase	**0.377**	**0.434**	**0.455**
	average	0.360	0.408	0.441
Similarity-based	BERT	0.276	0.306	0.315
	DistilBERT	0.305	0.299	0.319
	BART	0.251	0.270	0.269
	RoBERTa	0.241	0.279	0.284
	XLM-R	0.231	0.267	0.277
	DeBERTa	0.257	0.275	0.255
	average	0.260	0.283	0.286

Table 9
1-Accuracy for few-shot classification for different number of labeled examples.

Approach	Embedding	3 Examples	5 Examples	7 Examples
SetFit	SentRoBERTa	0.693	0.763	0.795
	Paraphrase	**0.743**	**0.848**	**0.851**
	average	0.718	0.806	0.823
Similarity-based	BERT	0.639	0.688	0.696
	DistilBERT	0.665	0.668	0.686
	BART	0.573	0.640	0.638
	RoBERTa	0.555	0.640	0.611
	XLM-R	0.567	0.608	0.616
	DeBERTa	0.593	0.640	0.600
	average	0.599	0.647	0.641

5.4.2. Task mapping

As explained in Section 4.2, the Task Mapper connects the intents to a set of tasks for each email. In this section, we evaluate the accuracy of the tasks mapped from the classified intents by comparing them to the tasks mapped from the ground-truth intents. We are particularly interested in investigating whether higher intent classification accuracy leads to higher accuracy of task mapping. Before the evaluation, we aggregate individual sentence intents to email level intents as described in Section 4.1.

Task mapping for zero-shot intent classification. Table 10 shows the results for task mapping based on the zero-shot intent classification approaches. We can clearly see that the entailment-based approaches that have obtained significantly higher task classification accuracy also lead to higher accuracy of the mapped tasks. For the entailment-based approach, XLM-R-M was the best performing model, followed by DeBERTa-M and BART-M. For the similarity based approach, the best performing model across metrics is BART.

We detect an improvement in extracting tasks compared to the accuracy scores for the corresponding identified intents (see Table 10). Our results demonstrate that task mapping can often 'hide' the noise that stems from intent classification as there is a potential for many (intents) to one (task) relationship. In other words, emails with multiple intents can often lead to similar recommended tasks. In contrast, we note that a limited improvement in intent classification accuracy does not necessarily entails higher accuracy in task mapping and can depend on the concrete mistakes of the classifiers. For example, we note that similarity-based zero-shot approaches tend to perform slightly worse compared to the random baseline despite obtaining better performance on intent classification.

Task mapping for few-shot intent classification. Table 11 shows the results for task mapping based on the few-shot intent classification approaches. SetFit, that obtained higher intent classification accuracy compared to the similarity-based approach, similarly obtain higher accuracy of mapped tasks in terms of F1 score and Jaccard score. For 1-Accuracy, all approaches performed very well and obtained between 0.945–0.969. We observe that the accuracy of mapped tasks is higher than the accuracy of intent classification, similar to our observation for zero-shot approaches. While the single best-performing few-shot method (SetFit with Paraphrase embeddings) obtains comparable performance to the single best performing zero-shot approach (entailment-based with XLM-R-M embeddings) in terms of task mapping accuracy, on average the few-shot approaches significantly outperform the zero-shot approaches. Finally, all few-shot approaches significantly outperform the random baseline.

Table 12, Table 13, and Table 14 show the F1 score, the Jaccard score and the 1-Accuracy, respectively, for varying number of labeled examples per class in $\{3, 5, 7\}$. We note that the impact of adding more examples is relatively small for the SetFit approaches, and can even decrease performance in the similarity-based approaches (see Section 6 for a discussion of these results).

Table 10
Results for task mapping based on zero-shot intent classification.

Intent Classification	Embedding	F1 score	Jaccard score	1-Accuracy
Entailment-based	BERT-M	0.659	0.576	0.909
	DistilBERT-M	0.500	0.389	0.833
	BART-M	0.727	0.646	**0.970**
	RoBERTa-M	0.697	0.611	0.955
	XLM-R-M	**0.795**	**0.737**	**0.970**
	DeBERTa-M	0.735	0.662	0.955
	average	0.686	0.604	0.932
Similarity-based	BERT	0.492	0.379	0.833
	DistilBERT	0.530	0.409	0.894
	BART	0.576	0.455	0.939
	RoBERTa	0.280	0.227	0.439
	XLM-R	0.356	0.263	0.636
	DeBERTa	0.500	0.414	0.758
	average	0.456	0.358	0.750
	Random	0.576	0.465	0.909

Table 11
Task mapping based on few-shot intent classification (5 examples per class).

Approach	Embedding	F1 score	Jaccard score	1-Accuracy
SetFit	SentRoBERTa	0.734	0.664	0.945
	Paraphrase	**0.791**	**0.738**	0.948
	average	0.762	0.701	0.946
Similarity-based	BERT	0.733	0.657	0.959
	DistilBERT	0.729	0.655	0.952
	BART	0.721	0.644	0.952
	RoBERTa	0.723	0.644	0.958
	XLM-R	0.717	0.636	0.959
	DeBERTa	0.732	0.653	**0.969**
	average	0.726	0.648	0.958
	Random	0.576	0.465	0.909

Table 12
Task mapping F1 score for few-shot classification.

Approach	Embedding	3 Examples	5 Examples	7 Examples
SetFit	SentRoBERTa	0.751	0.734	0.760
	Paraphrase	**0.767**	**0.791**	**0.795**
	average	0.759	0.762	0.778
Similarity-based	BERT	0.725	0.733	0.727
	DistilBERT	0.744	0.729	0.729
	BART	0.701	0.721	0.703
	RoBERTa	0.711	0.723	0.687
	XLM-R	0.660	0.717	0.703
	DeBERTa	0.724	0.732	0.690
	average	0.711	0.726	0.707

6. Discussion, limitations, and future work

The section is divided into two main parts: 1. Discussion, where we present insights from our empirical evaluation, and 2. Limitations

Table 13
Task mapping Jaccard score for few-shot classification.

Approach	Embedding	3 Examples	5 Examples	7 Examples
SetFit	SentRoBERTa	0.684	0.664	0.696
	Paraphrase	**0.705**	**0.738**	**0.740**
	average	0.694	0.701	0.718
Similarity-based	BERT	0.653	0.657	0.654
	DistilBERT	0.671	0.655	0.652
	BART	0.625	0.644	0.617
	RoBERTa	0.628	0.644	0.597
	XLM-R	0.570	0.636	0.622
	DeBERTa	0.656	0.653	0.603
	average	0.634	0.648	0.624

Table 14
Task mapping 1-Accuracy for few-shot classification.

Approach	Embedding	3 Examples	5 Examples	7 Examples
SetFit	SentRoBERTa	0.955	0.945	0.950
	Paraphrase	0.952	0.948	**0.961**
	average	0.953	0.946	0.956
Similarity-based	BERT	0.942	0.959	0.946
	DistilBERT	**0.964**	0.952	**0.961**
	BART	0.931	0.952	**0.961**
	RoBERTa	0.961	0.958	0.957
	XLM-R	0.929	0.959	0.947
	DeBERTa	0.928	**0.969**	0.953
	average	0.943	0.958	0.954

and Future Work, where we discuss the design choices that restrict the generalizability of our solution and suggest potential enhancements for the proposed pipeline.

6.1. Discussion

The evaluation focused on assessing the performance of two crucial components in our approach: the Intent Classifier and the Task Mapper. Below, we delve into the insights and lessons learned from our empirical evaluation.

Intent classifier. The results of the Intent Classifier indicate that transformer models, even with limited or no supervision, can effectively identify intents in sentences. This is clearly demonstrated by the significant improvement of our approaches over the random baseline in intent classification, highlighting the benefit of pre-trained language models in learning settings with limited data. Concretely, the best performing zero-shot and few-shot models obtained F1 scores of 0.383 and 0.537, respectively, compared to an F1 score of 0.121 for the random baseline.

For zero-shot learning, entailment-based techniques that establish semantic correlations between embeddings and intents, represented by taxonomy-driven sentences, outperformed similarity-based techniques that solely relied on proximity in latent space. This result highlights the preference for semantic-based techniques over syntactic ones in zero-shot learning scenarios.

Our experiments find that few-shot approaches significantly outperform zero-shot approaches in intent classification. Although the allure of zero-shot learning lies in its independence from labeled examples, allocating resources to label a small subset of email exchanges between stakeholders can significantly enhance performance. In the few-shot setting, fine-tuning the model using SetFit has a significant advantage over the simple, similarity-based approach. Interestingly, our results indicate that the few-shot similarity-based approach is significantly less sensitive to the choice of architecture compared to the zero-shot similarity-based approach. Specifically, we observe relatively limited variation in intent classification performance across the different model architectures.

Task mapper. Shifting our focus to the Task Mapper, we observe improvements across all performance measures and settings (both zero-shot and few-shot learning) compared to the Intent Classifier. This success can be attributed to the fact that different intents may lead to similar tasks. This matching process helps mitigate some of the noise observed in the intent classification task. Overall, the results demonstrate a consistent trend in performance: the entailment-based approach outperformed the similarity-based approach in the zero-shot setting, SetFit fine-tuning outperformed the similarity-based approach in the few-shot setting, and few-shot, on average, outperformed zero-shot learning. As with intent classification, the few-shot similarity-based approach remains less sensitive to the choice of architecture and exhibit relatively limited variation in task mapping performance. We note that the advantage of our approaches over the random baseline carries over to task mapping in our zero-shot entailment-based approach, our few-shot SetFit approach, and our few-shot similarity-based approach.

6.2. Limitations and future work

In this part, we outline the main limitations of our solution based on the different components of our pipeline.

Email processor. In the input phase we treat emails as independent sentences without taking into account the position of these sentences in the body of the email. Moreover, we did not take into account the fact that emails within the same thread (or within the dataset in general) may depend on each other.

When processing the incoming emails, we ignored special symbols, bullet-point lists, and other components that differentiate emails (that are semi-structured objects) from free-flowing text. Further, the dataset that we used for our evaluation is limited in size. We considered two email threads having eight and 56 emails, respectively. This relatively small amount of data is sufficient when sentence and email independence assumptions are made (see representation design choices), but may turn out to be very limited (and unbalanced) if one assumes that emails in a thread are dependent.

As possible future steps, we aim to investigate approaches that can capture the dependencies between sentences and between emails in the same thread. In addition, we intend to take into account the special structure of emails (including position of sentences, and considering special symbols like '@'), which we believe may lead to improved performance of the two downstream components (Intent Classifier and Task Mapper). Beyond these improvements, we aim to extend the Email Processor with additional domain knowledge including the identity of the sender and recipients, their organizational hierarchy, etc. Lastly, we plan to collect additional data from inter-organizational email exchanges between knowledge workers to increase our sample size.

Intent and task representation. In Section 2.2 we discussed various taxonomies that we considered before choosing the email taxonomy presented in Cohen et al. (2004). Clearly, the taxonomy that we chose is not the only option and one can consider alternatives. Moreover, we employed a flattened version of the taxonomy proposed in Cohen et al. (2004), without taking into account the hierarchy between the different nouns and verbs (see Fig. 1). One could improve the pipeline by considering a combination of several taxonomies, which would allow a richer intent representation, together with a method to capture the hierarchical dependencies within Cohen et al. (2004).

For task representation, we used the tasks presented in Lin et al. (2018). As an extension, one can consider alternative sets of tasks, since the list that we provide in Table 3 does not include all possible tasks in knowledge-intensive processes. One can, for instance, use repositories of organizational process documentation and process models to extract additional activity names (see Leopold, 2013 for knowledge extraction methods).

Lastly, tasks are not the only business process artefacts that may be associated with emails and intents. For example, decision outcomes

and document status changes are important pieces of information that can be extracted from emails. In future work, we plan to mine these additional key artefacts that are often apparent in procedural models (like BPMN), but are missing from typical declarative and other non-procedural representations (with the latter being most useful in knowledge-intensive processes).

Intent classifier and task mapper. When constructing the two functions, ϕ and θ, we made several design choices. First, we selected the top two intents as representatives of intents in a sentence (limiting the number of intents per sentence). Similarly, we limited the number of tasks per email to two tasks per email intent. These two choices can be reasonably explained by prior literature (1–2 intents per email Wang et al., 2019) and by the desire to limit the number of tasks any tool that would build upon our approach would generate per email. However, the impact of these choices on the performance of the pipeline remains an open question. In future work, we wish to test the sensitivity of the approach to different limits on the number of intents per sentence and tasks per email.

Our current evaluation of the performance of our intent classifier and task mapper is relying on a relatively small data set. In future work, we plan to deploy our end-to-end pipeline within a partner organization and continually collect additional data and user feedback. Furthermore, adding a human-in-the-loop component into our pipeline where the users will be prompted to for feedback may provide an invaluable alternative to manual labeling of data that can help improve the performance of our models.

7. Conclusion

In this paper, we presented an end-to-end solution for extracting useful business process information on knowledge-intensive processes from unlabeled email data. Specifically, the approach takes an email as an input, preprocesses it using standard natural language processing approaches, parses it into sentences, and maps the sentences to user intents and subsequently to recommended tasks. Our solution is a novel end-to-end pipeline that utilizes a well-established taxonomy to extract intents as an intermediate knowledge layer, which is then mapped to future tasks. The pipeline employs pre-trained transformer models with little or no supervision to predict both intents (via the Intent Classifier) and tasks (via the Task Mapper).

Our experimental evaluation provides valuable insights into the accurate extraction of intents and tasks. Firstly, we demonstrate that precise intent extraction acts as a crucial foundation for achieving accurate task extraction. Moreover, our findings reveal the superiority of entailment-based approaches over similarity-based approaches in the zero-shot setting. Notably, we highlight the significant enhancements achievable through few-shot learning approaches, which leverage a small number of labeled examples, in terms of both intent and task extraction accuracy. By incorporating these key observations, our study underscores the importance of intent classification in facilitating task extraction and emphasizes the advantages of entailment-based techniques and few-shot learning methods for improved performance in both intent and task extraction tasks.

In future work, we aim to improve our pipeline (Fig. 2) by relaxing the assumption that emails (and sentences) are independent. This will require adjustments to the intent classification and task mapping components to be able to capture dependencies in email threads. In addition, we plan to investigate pre-training using email specific datasets, such as the well-known Enron corpus, to improve performance. Lastly, we plan to implement a human-in-the-loop feedback component that would collect labels during interactions of the pipeline with its users, thus enabling better fine-tuning and evaluation of the transformer models.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

Appendix. Zero-shot intent classification examples

In this appendix, we demonstrate the results for the two zero-shot intent classification approaches. We use two illustrative example sentences[7] and the DeBERTa model and present the top three labels returned by the two zero-shot approaches. Note that the zero-shot entailment-based approach ranks the labels based on their probability (denoted p) where higher values indicate better match, while the similarity-based approach ranks the labels based on cosine distance (denoted d) where lower values indicate better match.

Example 1: "Hello Alice, can you please review the attached document and let me know what you think?".

Zero-shot entailment-based approach:

1. 'an activity is being requested' ($p = 0.40$)
2. 'an opinion is being requested' ($p = 0.30$)
3. 'an opinion is being delivered' ($p = 0.09$)

Zero-shot similarity-based approach:

1. 'an opinion is being requested' ($d = 0.19$)
2. 'data is being requested' ($d = 0.24$)
3. 'meeting is being proposed' ($p = 0.25$)

Example 2: "Hi Bob, the updated spreadsheet is attached".

Zero-shot entailment-based approach:

1. 'data is being delivered' ($p = 0.32$)
2. 'data is being amended' ($p = 0.32$)
3. 'an activity is being committed' ($p = 0.14$)

Zero-shot similarity-based approach:

1. 'data is being requested' ($d = 0.21$)
2. 'an opinion is being requested' ($d = 0.21$)
3. 'a meeting is being proposed' ($d = 0.23$)

References

Alibadi, Z., Du, M., Vidal, J.M., 2019. Using pre-trained embeddings to detect the intent of an email. In: Proceedings of the 7th ACIS International Conference on Applied Computing and Information Technology. pp. 1–7.

Austin, J.L., 1965. How to do things with words the William James lectures delivered at Harvard University in 1955.

Brighi, M., Franco, A., Maio, D., 2021. Metric learning for multi-label classification. In: Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition. SSPR, Springer, pp. 24–33.

Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al., 2020. Language models are few-shot learners. arXiv preprint arXiv:2005.14165.

Burnyshev, P., Malykh, V., Bout, A., Artemova, E., Piontkovskaya, I., 2021. A single example can improve zero-shot data generation. arXiv preprint arXiv:2108.06991.

Carvalho, V.R., Cohen, W.W., 2004. Learning to extract signature and reply lines from email. In: Proceedings of the Conference on Email and Anti-Spam, Vol. 2004.

[7] As we are unable to share the dataset used for this study, we used two new sentences that are not part of the original dataset to illustrate the results of the zero-shot approaches.

Chambers, A.J., Stringfellow, A.M., Luo, B.B., Underwood, S.J., Allard, T.G., Johnston, I.A., Brockman, S., Shing, L., Wollaber, A., VanDam, C., 2020. Automated business process discovery from unstructured natural-language documents. In: International Conference on Business Process Management. Springer, pp. 232–243.

Chen, Y.-N., Hakkani-Tür, D., He, X., 2016. Zero-shot learning of intent embeddings for expansion by convolutional deep structured semantic models. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 6045–6049.

Chiu, K.-L., Alexander, R., 2021. Detecting hate speech with GPT-3. arXiv preprint arXiv:2103.12407.

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

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, É., Ott, M., Zettlemoyer, L., Stoyanov, V., 2020. Unsupervised cross-lingual representation learning at scale. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 8440–8451.

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

Dabbish, L.A., Kraut, R.E., Fussell, S.R., Kiesler, S.B., 2005. Understanding email use: predicting action on a message. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Di Ciccio, C., Marrella, A., Russo, A., 2015. Knowledge-intensive processes: characteristics, requirements and analysis of contemporary approaches. J. Data Semant. 4, 29–57.

Di Ciccio, C., Mecella, M., 2013. Mining artful processes from knowledge workers' emails. IEEE Internet Comput. 17 (5), 10–20.

Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A., et al., 2013. Fundamentals of Business Process Management, Vol. 1. Springer.

Dustdar, S., Hoffmann, T., Van der Aalst, W., 2005. Mining of ad-hoc business processes with TeamLog. Data Knowl. Eng. 55 (2), 129–158.

El Emam, K., Madhavji, N.H., 1995. A field study of requirements engineering practices in information systems development. In: Proceedings of 1995 IEEE International Symposium on Requirements Engineering. RE'95, IEEE, pp. 68–80.

Elleuch, M., 2021. Business Process Discovery from Emails, a First Step Towards Business Process Management in Less Structured Information Systems (Ph.D. thesis). Institut polytechnique de Paris.

Elleuch, M., Assy, N., Laga, N., Gaaloul, W., Ismaili, O.A., Benatallah, B., 2020a. A meta model for mining processes from email data. In: 2020 IEEE International Conference on Services Computing. SCC 2020, Beijing, China, November 7–11, 2020, IEEE, pp. 152–161. http://dx.doi.org/10.1109/SCC49832.2020.00028.

Elleuch, M., Ismaili, O.A., Laga, N., Assy, N., Gaaloul, W., 2020b. Discovery of activities' actor perspective from emails based on speech acts detection. In: van Dongen, B.F., Montali, M., Wynn, M.T. (Eds.), 2nd International Conference on Process Mining. ICPM 2020, Padua, Italy, October 4–9, 2020, IEEE, pp. 73–80. http://dx.doi.org/10.1109/ICPM49681.2020.00021.

Elleuch, M., Ismaili, O.A., Laga, N., Gaaloul, W., Benatallah, B., 2020c. Discovering activities from emails based on pattern discovery approach. In: International Conference on Business Process Management. Springer, pp. 88–104.

Elleuch, M., Laga, N., Ismaili, O.A., Gaaloul, W., 2020d. Discovering business processes and activities from messaging systems: State-of-the art. In: 29th IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises. WETICE 2020, Virtual Event, France, September 10–13, 2020, IEEE, pp. 137–142. http://dx.doi.org/10.1109/WETICE49692.2020.00035.

Elleuch, M., Laga, N., Ismaili, O.A., Gaaloul, W., 2023. Multi-perspective business process discovery from messaging systems: State-of-the art. Concurr. Comput. Pract. Exp. 35 (11), http://dx.doi.org/10.1002/cpe.6642.

Ganda, D., Buch, R., 2018. A survey on multi label classification. Recent Trends Program. Lang. 5 (1), 19–23.

Gouk, H., Pfahringer, B., Cree, M., 2016. Learning distance metrics for multi-label classification. In: Asian Conference on Machine Learning. PMLR, pp. 318–333.

He, P., Liu, X., Gao, J., Chen, W., 2021. Deberta: decoding-enhanced bert with disentangled attention. In: 9th International Conference on Learning Representations. ICLR 2021, Virtual Event, Austria, May 3–7, 2021.

Huber, S., Fietta, M., Hof, S., 2015. Next step recommendation and prediction based on process mining in adaptive case management. In: Proceedings of the 7th international conference on subject-oriented business process management. pp. 1–9.

Jlailaty, D., Grigori, D., Belhajjame, K., 2019. On the elicitation and annotation of business activities based on emails. In: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing. pp. 101–103.

Khandaker, F., Senderovich, A., Yu, E., Carbajales, S., Chan, A., 2022. Transformer models for activity mining in knowledge-intensive processes. In: International Conference on Business Process Management. Springer, pp. 13–24.

Khosravi, H., Wilks, Y., 1999. Routing email automatically by purpose not topic. Nat. Lang. Eng. 5 (3), 237–250.

Krosnick, J.A., 1999. Survey research. Annu. Rev. Psychol. 50 (1), 537–567.

Lampert, A., Dale, R., Paris, C., et al., 2008. The nature of requests and commitments in email messages. In: Proceedings of the AAAI Workshop on Enhanced Messaging. pp. 42–47.

Leopold, H., 2013. Business process management. In: Natural Language in Business Process Models. Springer, pp. 1–23.

Leuski, A., 2004. Email is a stage: discovering people roles from email archives. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 502–503.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L., 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461.

Lin, C.-C., Kang, D., Gamon, M., Pantel, P., 2018. Actionable email intent modeling with reparametrized rnns. In: Thirty-Second AAAI Conference on Artificial Intelligence.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V., 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

Mahabadi, R.K., Zettlemoyer, L., Henderson, J., Mathias, L., Saeidi, M., Stoyanov, V., Yazdani, M., 2022. Prompt-free and efficient few-shot learning with language models. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 3638–3652.

Pamungkas, E.W., Basile, V., Patti, V., 2021. A joint learning approach with knowledge injection for zero-shot cross-lingual hate speech detection. Inf. Process. Manage. 58 (4), 102544.

Park, G., van der Aalst, W.M., 2022. Action-oriented process mining: bridging the gap between insights and actions. Prog. Artif. Intell. 1–22.

Pillai, I., Fumera, G., Roli, F., 2012. F-measure optimisation in multi-label classifiers. In: Proceedings of the 21st International Conference on Pattern Recognition. ICPR2012, IEEE, pp. 2424–2427.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al., 2019. Language models are unsupervised multitask learners. OpenAI Blog 1 (8), 9.

Reimers, N., Gurevych, I., 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. EMNLP-IJCNLP, pp. 3982–3992.

Reimers, N., Gurevych, I., 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing. EMNLP, pp. 4512–4525.

Ruan, Y.-P., Ling, Z.-H., Gu, J.-C., Liu, Q., 2020. Fine-tuning bert for schema-guided zero-shot dialogue state tracking. arXiv preprint arXiv:2002.00181.

Sanh, V., Debut, L., Chaumond, J., Wolf, T., 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

Sappadla, P.V., Nam, J., Mencía, E.L., Fürnkranz, J., 2016. Using semantic similarity for multi-label zero-shot classification of text documents. In: ESANN.

Searle, J.R., Searle, J.R., 1969. Speech Acts: An Essay in the Philosophy of Language, Vol. 626. Cambridge University Press.

Shu, K., Mukherjee, S., Zheng, G., Awadallah, A.H., Shokouhi, M., Dumais, S., 2020. Learning with weak supervision for email intent detection. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1051–1060.

Song, K., Tan, X., Qin, T., Lu, J., Liu, T.-Y., 2020. Mpnet: Masked and permuted pre-training for language understanding. Adv. Neural Inf. Process. Syst. 33, 16857–16867.

Stuit, M., Wortmann, H., 2012. Discovery and analysis of e-mail-driven business processes. Inf. Syst. 37 (2), 142–168.

Szymaĺ, P., Kajdanowicz, T., et al., 2019. Scikit-multilearn: A Python library for multi-label classification. J. Mach. Learn. Res. 20 (6), 1–22.

Tam, D., Menon, R.R., Bansal, M., Srivastava, S., Raffel, C., 2021. Improving and simplifying pattern exploiting training. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 4980–4991.

Tunstall, L., Reimers, N., Jo, U.E.S., Bates, L., Korat, D., Wasserblat, M., Pereg, O., 2022. Efficient few-shot learning without prompts. arXiv preprint arXiv:2209.11055.

Ulrich, J., 2008. Supervised Machine Learning for Email Thread Summarization (Ph.D. thesis). University of British Columbia.

Van Der Aalst, W., 2016. Process Mining: Data Science in Action, Vol. 2. Springer.

van der Aalst, W.M., Nikolov, A., 2007. Emailanalyzer: an E-Mail Mining Plug-In for the Prom Framework. BPM Center Report BPM-07-16, BPMCenter. org.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. In: Advances in Neural Information Processing Systems. pp. 5998–6008.

Wang, W., Hosseini, S., Awadallah, A.H., Bennett, P.N., Quirk, C., 2019. Context-aware intent identification in email conversations. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 585–594.

Wang, Y., Yao, Q., Kwok, J.T., Ni, L.M., 2020. Generalizing from a few examples: A survey on few-shot learning. ACM Comput. Surv. 53 (3), 1–34.

Williams, K., 2019. Zero shot intent classification using long-short term memory networks. In: INTERSPEECH. pp. 844–848.

Williams, A., Nangia, N., Bowman, S.R., 2018. A broad-coverage challenge corpus for sentence understanding through inference. In: Proceedings of NAACL-HLT. pp. 1112–1122.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al., 2019. Huggingface's transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771.

Yin, W., Hay, J., Roth, D., 2019. Benchmarking zeroshot text classification: Datasets, evaluation and entailment approach. arXiv preprint arXiv:1909.00161.

Zhang, X., Cai, F., Hu, X., Zheng, J., Chen, H., 2022. A contrastive learning-based Task Adaptation model for few-shot intent recognition. Inf. Process. Manage. 59 (3), 102863.

Zhang, M.-L., Zhou, Z.-H., 2007. ML-KNN: A lazy learning approach to multi-label learning. Pattern Recognit. 40 (7), 2038–2048.