Alaska: Intelligent Email Assistant

Maha Tufail Agro
21010227

Muhammad Arslan Manzoor
21010240

Muhammad Umar Salman
21010241

{maha.agro, muhammad.arslan, umar.salman}@mbzuai.ac.ae

Abstract

Emails have become a part of our daily lives either for business communication or for personal use. Users can send emails to multiple individuals, making it easier to communicate information to large groups of people. Sending emails are not only a fast, cheap and efficient way to communicate but also provide an effective way to transmit electronic data, schedule meetings and set priority-based tasks to complete etc. Common voice assisted software's such as Amazon's Alexa, Apple's Siri and Google's Assistant have the ability to answer questions like "What is the weather like today?" and do tasks such as "Set my alarm for 9:00am". However, we have not yet seen the ease of sending emails through the aforementioned voice-activated assistants. In this paper, we design a product called 'Alaska' that will primarily take raw speech from a user and convert it to a structured email which can be further tweaked for minor changes and then sent to the recipient. The intelligent assistant makes use of pattern matching, text classification and question answering to interact with what the user says to the system. The system in response either carries out the instruction or asks a following question to clarify what it has been asked of.

Keywords: Intelligent Systems, Email Assistant, Speech Recognition

1. Introduction

In the last few decades, technology enhancements have revolutionized the way of communicating effectively. The Internet has played a vital role in communication, making it easily accessible for everyone worldwide [1]. Email is the most formal and professional way to communicate reliably and share confidential information.

Drafting an email or providing instant responses is usually not feasible for most people with their busy schedules. Providing convenience and saving time are the two important characteristics of any Artificial Intelligence (AI) based system [2]. The AI-based systems exploit Natural Language Processing and Speech Recognition to interact and understand human voice and language by extracting meaningful features such as keywords and expressions [3].

State-of-the-art voice assistants such as Amazon's Alexa, Google's Assistant, Apple's Siri are more general-purpose assistants which manage to carry out multiple tasks which are clearly shown in Table 1. The aforementioned products are extraordinarily efficient in recognizing a users voice, answering their queries and performing general-purpose tasks as simple as setting the alarm on your mobile device [4]. However, they all lack in providing more task-specific functionalities i.e., sending an email to one or many recipients with the main text of the email body via voice instruction. Thus, our product's intention is to create an AI-based task-specific tool that offers more features and flexibility to users by allowing them to send an email via speech. Our main objective is to develop a product that can securely, time-efficiently and effectively send emails with a performance as good as those of state-of-the-art voice-assistants.

Most of the existing voice-based email systems are designed using rule-based or pattern matching algorithms [5, 6]. The proposed system, on the other hand makes use of advanced deep learning methods such as text classification and question answering along with effective use of state-of-the-art speech recognition models to recognize what has been said more precisely. The QA system within our product pipeline intelligently maps and fetches the most relevant excerpts from the speech input.

We aim to create a more interactive, versatile, flexible tool for the ease of the user. Our proposed product collects raw speech from the user and through a Speech-to-Text model convert the user's speech into text. This unstructured textual representation of voice can contain noise or other information for an assistant to help distinguish various components of the email. The classifier module classifies the unstructured text into either a 'Email' or a 'Command'. Based on the class predicted the text is passed onto the QA system or text classification model. Finally, after the email is properly extracted into it's various components,

Features	Siri	Cortana	Alexa	Google Assistant
Play music	✓	✗	✓	✓
Set alarm	✓	✓	✓	✓
Search the web	✓	✓	✓	✓
Make calls	✓	✓	✓	✗
Look up directions	✓	✗	✗	✓
Read emails	✓	✓	✓	✗
Send emails	✗	✗	✗	✗

Table 1. Comparision of Existing Voice Assistants.

the text is post-processed and cleaned for formality and is ready to send to the recipient. Details on the entire pipeline are further discussed in detail in Section 3
The proposed approach consists of following contributions for .

1. The system avails Google's Speech-to-Text module for effective voice recognition and generation of text.
2. Text classification which allows to differentiate between general commands given to the system from email text.
3. Fine-tuning spanBERT [7] QA model to extract and provides relevant answers to fundamental questions regarding the layout and content of the email
4. Post-processing and cleaning of the email to send it in a presentable and professional manner.
5. Creating a Web-platform which hosts this service for people to use the product on a daily-basis for intelligent email assistance.

2. Related Work

2.1. Conversational Question Answering

Conversational question answering (CQA) is a subdomain of QA systems. Traditional QA systems engage with the user in a single-turn: *user ask, system respond* setting, which is counter intuitive to how most real-world conversations take place. CQA provides a more natural, multi-turn *system ask, user respond* setting, where the user interacts with the system repeatedly and the system asks various follow-up questions to clarify the user's needs. CQA systems are broadly categorized into two categories: sequential knowledge-base question-answering (KB-QA) systems and conversational machine reading comprehension (CMRC) systems [8].

Sequential KB-QA systems are an extension to KB-QA systems, they iteratively query a structured KB such as Freebase [9], DBpedia [10], NELL [11], Wikidata [1], etc to model complex sequential question answering. Dynamic Neural Semantic Parsing DynSP [12] tackles the problem of sequential QA by formulating a weakly supervised reward-guided approach. The model is trained on SequentialQA, a dataset collected from decomposing WikiTableQuestions[2] entries into simpler sequential questions. Additionally, the authors develop a semantic parse language to map natural language into a logical form. However, DynSP fails when required to parse complex questions, especially utterances which exhibit ellipse phenomenon and need coreference resolution.

Dialog-to-Action (D2A) [13] solves the previous problem by deriving the logical form of utterances dynamically based on the current question q_i and the previous questions $q_{1:i-1}$. The task of generating a logical form is considered action generation. Consequently, the model runs into the problems of error propagation and unsupported actions, where error from previous actions is propagated to subsequent actions and examples where the model fails to derive the correct logical form. Multi-task Semantic Parsing (MaSP) [14] solves the coreference and error propagation problem by introducing a pointer-based semantic parsing model that jointly learns entity detection. The model leverages contextual information for entity type prediction and coreference resolution, which subsequently alleviates error propagation.

CMRC was introduced by [15] as an advancement over traditional MRC models which lack the conversational aspect. CMRC models include the following key functions: history selection and history modeling, encoding the given and context into embeddings, reasoning via a neural model to arrive at an answer vector, decoding the answer vector to natural language representation. Concretely, given a context, the question-answer pairs as history, and the current question Q_i, the model predicts the correct answer A_i.

Contextualized attentention based deep network (SDNet) [16], open retireval conversational question answering (ORConvQA) [17], and bi-directionational attention flow (BIDAF++) [18] are CMRC models that include history in current utterance by prepending to it the latest K rounds of conversational history turns. SDNet [16] uses both self-attention and inter-attention to extract relevant context from text passages. BiDAF++ [18] predicts the correct answer span by utilizing a layer with bi-directional attention flow followed by a biLSTM layer. ORConvQA [17] is a transformer-based end-to-end system that consists of passage retriever, passage ranker, and passage reader modules. However, these models do not account for topic return or topic shift. History Answer Modeling (HAM) [19] performs history selection based on relevancy using a weighting process.

[1] https://www.wikidata.org/wiki/Wikidata:Main Page

[2] https://github.com/ppasupat/WikiTableQuestions

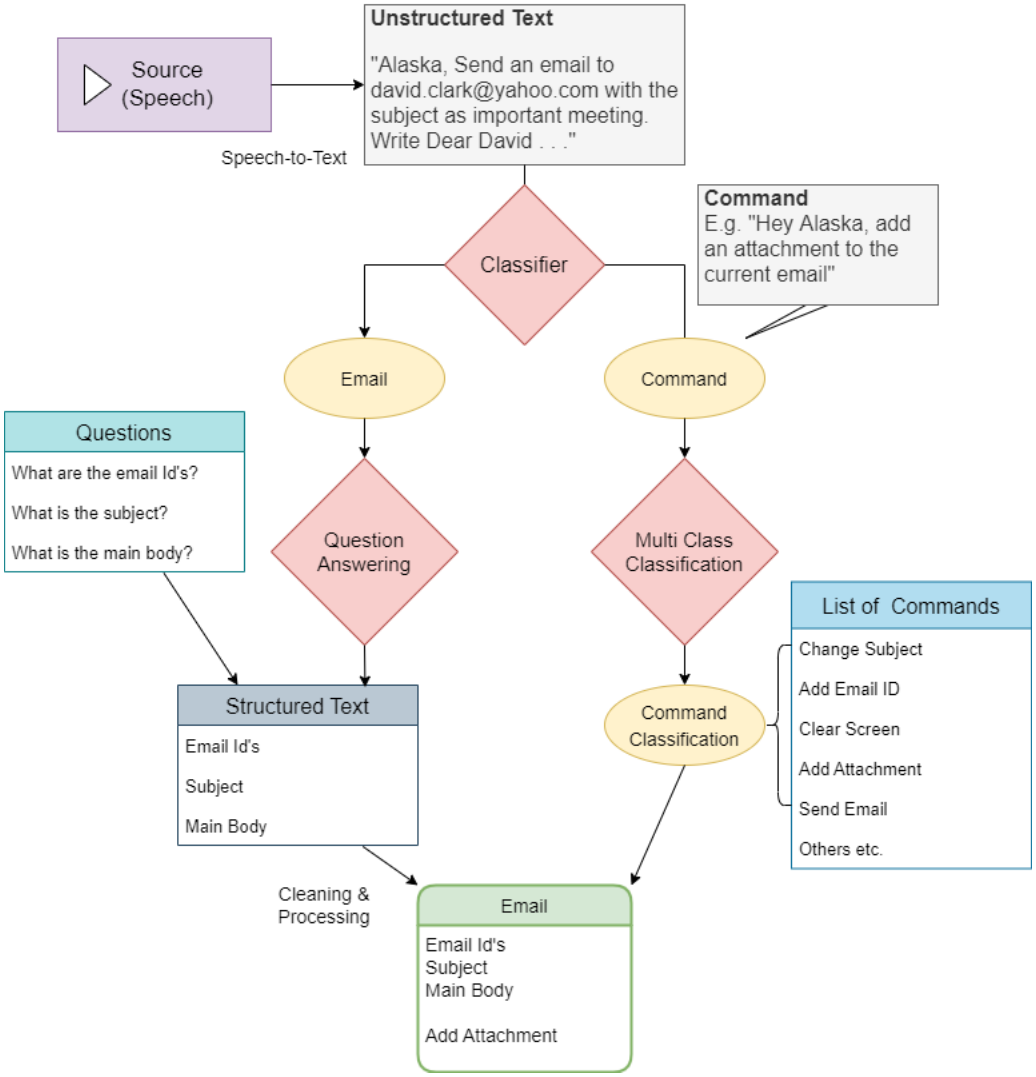

Figure 1. Alaska's architecture.

2.2. SpanBERT

SpanBERT [7] is a modification of BERT [20] designed to predict spans of text. SpanBERT modifies BERT [20] in several ways: first, it changes the masked language model (MLM) objective to mask spans of words instead of individual tokens. Given a sequence X, the model samples a set of spans Y such that Y is 15% of X. Second, they introduce the span boundary objective (SBO). SBO predicts a span of text given the tokens at the boundary of the span. Formally, each token x_i in a span is represented using the output encodings of the two boundary tokens x_{s-1} and x_{e+1}, as well as the positional encodings of the token x_i, p_{i-s+1}. Third, they discard BERT's [20] next-sentence-prediction since the model processes a single segment of text during the training step instead of processing two. This approach removes noise from the MLM and ensures that the model can exploit longer full-length context. The SpanBERT [7] model significantly outperforms the existing baseline on question answering tasks, coreference resolution tasks and relation extraction tasks.

2.3. Digital Voice Assistants

In recent years digital voice assistants have gained prominence due to increasingly sophisticated technology incorporated in hand-held devices. They are also a popular choice for users who prefer to use voice commands to operate their phones. Siri [3], the digital assistant found on apple phones performs various tasks on behalf of the user, such as setting an alarm, reminding the user of specific tasks and web searching etc. While Siri, in theory, can open, read and send emails. It does so with less precision, often not distin-

[3] https://www.apple.com/siri/

Hey Alaska send an email to david.clarke@yahoo.com with the subject as meeting tomorrow at nine. In the email say dear david hope you are doing well I am keeping the meeting for tomorrow's project alignment and proposal submission at nine am. Looking forward to seeing you there best regards Mark.
1. Who are the email recipients? 2. What is the subject for the email? 3. What is the main body?

Table 2. An example of an unstructured email voiced out by a user and the questions that our QA model will ask to structure the email

guishing the correct recipient or incorrectly formatting the email. Moreover, it still requires manual intervention from the user which is not always convenient. Other voice assistants such as Microsoft's Cortana [4], Amazon's Alexa [5], and Google's Assistant[6] suffer from similar problems when dealing with email.

3. Methods

To model our digital assistant product, we have designed an architecture which can be seen in Figure 1. The aim of the product is to assist individuals to send emails via speech along with other features that email sending has to offer. The input to the model or the source is raw speech from the user which is converted into text using a Speech-to-Text converter. The Speech-to-Text model used for this product is Google Cloud's Speech-to-Text [7] model. The Google Cloud's Speech-to-Text model achieves state-of-the-art accuracy on standard datasets as it makes use of extremely advanced deep learning neural network algorithms for automatic speech recognition (ASR). Using Google Cloud's Speech-to-Text model we convert the raw speech from the user and convert it into unstructured text data. This text data from here needs to be converted into a structured form primarily comprising of 3 main components: email recipients, subject and main text body. Currently, the unstructured data includes trigger words like 'Hey Alaska' or has other words that user says while voicing out his email for e.g. 'Send an email to . . .' etc.

However, before converting our unstructured text into a structured email form we first need to classify whether the speech input is the actual email or just some command given to our product to carry out a task such as adding an attachment. For this we train binary classifier whose input is an unstructured text and output are two classes 'Email' and 'Command'. The class 'Email' refers to text that includes content such as email recipients, subject and the main body, whereas 'Command' refers to a text that includes a command that needs to be carried out such as adding an attachment or removing a recipient etc. In the scenario where the unstructured text is classified as an 'Email' from the user, it is then passed to a QA model which asks 3 fundamental questions from our unstructured text shown in Table 2. Here we borrow the idea of span-prediction whose concept originated from QA for machine reading comprehension (MRC). This specific type of QA model attempts to extract the correct answer of a given question from a single span within the given text. In our case, we will first fine-tune SpanBERT [7] on our emails dataset such that it extracts the answers of our 3 questions from the unstructured dataset to provide us with a structured format of the email the user wishes to send.

However, just before we send the email, there is a need to clean and process our structured data. Here we will use pattern matching to structure the email into a proper format. For example, if we instruct the Google Cloud's Speech-to-Text model via speech to "Send an email to david.clark@yahoo.com", the expected output would be "david dot clark at the rate yahoo dot com". Another example would be that the unstructured text coming in will give the main body of the email a passage-type look, we would have to begin a new line after the salutation and add a new line right before the users signature. There are other numerous examples of pattern matching that will be needed to ensure a satisfying and enjoyable user experience.

In the scenario where a command is sent to the product such as 'Hey Alaska, add an attachment to the current email', our binary classifier from the unstructured text classifies it as a 'Command'. This command is then passed onto a multi-class classification model which aims to predict the command text into pre-defined commands whose examples can also be seen in Figure 1. Supposing the classifier predicts the 'Add Attachment' command from the unstructured text, this instruction will be then sent to the platform to carry out that task by opening the attachment prompt. Finally, when the user has constructed his entire email and wishes to send it he can instruct the product to do so after viewing the email for any mistakes.

4. Data Exploration and Processing

For our product we will use the Enron email dataset [8]. The Enron email corpus contains approximately 500,000

[4]https://www.microsoft.com/en-us/cortana
[5]https://www.alexa.com/
[6]https://assistant.google.com/
[7]https://cloud.google.com/speech-to-text
[8]https://www.cs.cmu.edu/ ./enron/

Message-ID	Date	From	To	Subject	CC	BCC	Content
<30866805.	7/13/2001 19:47	frozenset({	frozenset({	RTO Orders - Grid South, SE Trans,			The Southeast RTO orders are
<26879196.	7/12/2001 11:36	frozenset({	frozenset({	More UC/CSU Info			--------------------- Forwarded
<15339082.	7/10/2001 0:47	frozenset({	frozenset({	California Update 07.09.01			The Bond Legislation The Der
<5265469.1	7/6/2001 20:45	frozenset({	frozenset({	Davis & Company -- incompetence			FYI --------------------- Forward
<13706905.	7/6/2001 20:44	frozenset({	frozenset({	Link to DWR contract info			FYI --------------------- Forward
<20784115.	6/22/2001 1:04	frozenset({	frozenset({	CPUC Proposed Decision Modifyin			In the same Decision that the
<16989362.	6/21/2001 21:29	frozenset({	frozenset({	CPUC Decis	Harry Kingerski <Harry		The following points highlight

Figure 2. Enron email dataset snapshot.

emails generated by employees of the Enron Corporation. This dataset was collected and prepared by the CALO Project(A Cognitive Assistant that Learns and Organizes). It contains data from about 150 users, mostly senior management of Enron, organized into folders. This data was originally made public, and posted to the web, by the Federal Energy Regulatory Commission after its investigation of Enron's collapse. Figure 2 shows a snapshot of the email data available.

We will primarily be using this data to fine-tune our spanBERT [7] model to accurately answer which spans in the unstructured email refer to the recipients, subject and main body. Looking at the data in Figure 2 we will only be requiring the fields 'To', 'Subject' and 'Content'. We will further use python's Faker library [9] to generate random number of recipients for an email and to sign off with various names so that the model doesn't overfit on the limited users that sent or received emails from Enron Corporation. Furthermore, within all the fields there will be a need to clean the data. For e.g. when the content's include tags like 'Forwarded to —–' and 'Replied by —–' and the recipients are not in proper format as can be seen from the Figure 2. Finally, when the data is clean for all fields we will need to augment them to examples of voice instructions so that they properly match what is said by a user when they ask a digital assistant to send an email. These examples will be taken from survey from individuals who will record how they would voice an instruction to send an email.

5. Future Work

For the next phase of our project we intend to begin working on preparing and cleaning the email dataset for our models. This requires some pre-processing and augmentation to cleanly extract the email recipients, subject and main body from our dataset. Having extracted these elements we will then survey a group of people on how they would send an email via voice. These recording will then be merged with the extracted elements to include extra and trigger words that individuals use while recording their email instructions such as 'Hey Alaska, Send an email ...'.

Furthermore, we will manually select the possible commands for our model which will align with the various features we wish to keep on the platform. Having done that we will hand-build a small dataset to train a multi-class classification model which predicts the commands in our product feature set.

Once our data is prepared we will begin implementing the various deep learning models that will be needed for this product. There are in total four deep learning algorithms in the entire pipeline namely: Speech-to-text model, Question Answering system, Binary Classification model and Multi-Class Classification model. After working on the accuracy of these models we will then look to create a website platform to host our product to showcase our efforts and progress of our product.

[9] https://faker.readthedocs.io/en/master/

References

[1] Marjorie D Kibby. Email forwardables: folklore in the age of the internet. *New Media & Society*, 7(6):770–790, 2005. 1

[2] Ming-Hui Huang and Roland T Rust. Artificial intelligence in service. *Journal of Service Research*, 21(2):155–172, 2018. 1

[3] Tianqi Wang, Chao-Kai Wen, Hanqing Wang, Feifei Gao, Tao Jiang, and Shi Jin. Deep learning for wireless physical layer: Opportunities and challenges. *China Communications*, 14(11):92–111, 2017. 1

[4] Veton Kepuska and Gamal Bohouta. Next-generation of virtual personal assistants (microsoft cortana, apple siri, amazon alexa and google home). In *2018 IEEE 8th annual computing and communication workshop and conference (CCWC)*, pages 99–103. IEEE, 2018. 1

[5] Rijwan Khan, Pawan Kumar Sharma, Sumit Raj, Sushil Kr Verma, and Sparsh Katiyar. Voice based e-mail system using artificial intelligence. *International Journal of Engineering and Advanced Technology (IJEAT)*, 9(3), 2020. 1

[6] Sharob Sinha, Shyanka Basak, Yajushi Dey, and Anupam Mondal. An educational chatbot for answering queries. In *Emerging Technology in Modelling and Graphics*, pages 55–60. Springer, 2020. 1

[7] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *Transactions*

of the Association for Computational Linguistics, 8:64–77, 2020. 2, 3, 4, 5

[8] Munazza Zaib, Wei Emma Zhang, Quan Z Sheng, Adnan Mahmood, and Yang Zhang. Conversational question answering: A survey. *arXiv preprint arXiv:2106.00874*, 2021. 2

[9] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, 2008. 2

[10] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195, 2015. 2

[11] Tom Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Bishan Yang, Justin Betteridge, Andrew Carlson, Bhavana Dalvi, Matt Gardner, Bryan Kisiel, et al. Never-ending learning. *Communications of the ACM*, 61(5):103–115, 2018. 2

[12] Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. Search-based neural structured learning for sequential question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1821–1831, 2017. 2

[13] Daya Guo, Duyu Tang, Nan Duan, Ming Zhou, and Jian Yin. Dialog-to-action: Conversational question answering over a large-scale knowledge base. *Advances in Neural Information Processing Systems*, 31, 2018. 2

[14] Tao Shen, Xiubo Geng, Tao Qin, Daya Guo, Duyu Tang, Nan Duan, Guodong Long, and Daxin Jiang. Multi-task learning for conversational question answering over a large-scale knowledge base. *arXiv preprint arXiv:1910.05069*, 2019. 2

[15] Chen Qu, Liu Yang, Minghui Qiu, Yongfeng Zhang, Cen Chen, W Bruce Croft, and Mohit Iyyer. Attentive history selection for conversational question answering. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1391–1400, 2019. 2

[16] Chenguang Zhu, Michael Zeng, and Xuedong Huang. Sdnet: Contextualized attention-based deep network for conversational question answering. *CoRR*, abs/1812.03593, 2018. 2

[17] Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W. Bruce Croft, and Mohit Iyyer. Open-retrieval conversational question answering. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020. 2

[18] Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. Quac : Question answering in context. *CoRR*, abs/1808.07036, 2018. 2

[19] Chen Qu, Liu Yang, Minghui Qiu, Yongfeng Zhang, Cen Chen, W. Bruce Croft, and Mohit Iyyer. Attentive history selection for conversational question answering. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, Nov 2019. 2

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. 3