Novel approach for tagging of discourse segments in help-desk emails

No Author Specified

***Abstract*—The volume of email that help-desks receive every day is very high and often queries are repeated. Any kind of automation in processing of emails requires good understanding of the emails. In the current work we propose a schema for tagging author composed sentences in help-desk emails by the intent of the author. We have created a corpus taking email data from two help-desks and annotated them at sentence level. We have achieved significant accuracies in learning to automatically tag the sentences using n-gram features and some hand-picked lexical features. At every stage right from choice of schema to choice of features, we have tried to be domain independent or keep domain related information as a separate component. Automation of Tagging of Discourse Segments (TODS) in email, we propose is a significant step towards finding the discourse parse of emails.**

I. Introduction

Email is widely in use and is one of the most popular means of business communication. Not surprisingly, lot of research goes on into improving the productivity and enhancing the email experience. Depending on the application, the information extracted from emails would vary and the more the information extracted, more and more applications can be built. Help-desks in organizations receive huge volumes of email containing varied queries, yet often repeated. Automation of processing of these help-desk emails would be of significant value to organizations by enhancing their customer support experience. Automatic redirection of email to the appropriate person, automatic composition of response to email, automatic content generation by identifying the query in the email, accessing organization database via email to retrieve the data are some of the possible tasks that ease the burden of handling emails.

Email research has been an active field and the tasks which are researched on are spam detection, email speech act recognition, extracting information and knowledge and its management, social network analysis, message grouping, enhanced experience with context sensitive information, email segmentation into zones and topics, email-leak detection, recipient prediction, task tracking etc.

We set out with the goal of finding the discourse parse of an email. Discourse parsing of email text is a new and complex task, especially given the fact that most of the research on emails till now was at message-level. We divided the task of identifying discourse parse into two stages a) tagging of discourse segments (TODS) b) identifying relations between the discourse segments. Tagging of discourse segments is to identify the intent of the author behind a segment. We propose 6 classes of segments in email namely greet, background, goal, concern, query, address. In trying to see how TODS will be of use in reaching the discourse parse, we draw our motivation from the analogy we see with dependency parsing of sentences. Parts Of Speech(POS) tagging of words in a sentence is a significant step in getting to dependency parse of sentence, where the relations between the words are identified. Researchers have gone through a phase where schema to POS-tag words might have been under debate and once there was a wide-spread agreement on how to tag words, they moved on to identify relations between these words and came up with parses of sentences. Similarly, discourse research is at a stage where all email researchers are trying to agree upon tag-sets and ways to tag discourse segments. Once TODS acheives good accuracies, it will enable finding discourse parse of emails.

In the current work we propose a schema convenient to tag and learn the intent of the content composed by author in the email (please note that there can be system generated content like disclaimer information which we are not processing). This would in turn facilitate in responding to a help-desk email by finding out sentences carrying concern and query in the email and other relevant sentences which are setting the context for the query. So, we propose 6 classes of sentences in email namely greet, background, goal, concern, query, address. These broad classes can help one narrow down to the exact query and extract it for further processing. Such a classification can also allow the query to be categorized based on content and sent to an appropriate human agent in a call center setting. This kind of segmentation of email based on author intent we think will be a very useful sub-stage towards identifying the discourse parse of the email. Discourse parse has lot of applications like summarization, question-answering, similarity measurement.

Fig.1. is an example of a typical help-desk email with a software company and the tagging of discourse segments we propose and Fig. 2. is the discourse parse of the same example email. The numbering used for discourse segments in the discourse parse can be seen in Fig.1.

II. Related Work

Research in identifying intentions in emails has its roots in the Speech Act Theory [7] and [8]. [9] applied text classification learning methods to email and suggested that many useful task-tracking tools can be built by identifying speech-acts in emails. Later they applied preprocessing techniques and improved the accuracies using n-gram selection methods [1].

Hi,	=> greet	
I am an ABC Coordinator for XYZ Mobile otherwise known as PQR @Hyderabad.	=> background	(1)
We are trying to use the GSA (global software assets).	=> goal	(2)
We do not have access to it from our domain–INHYDPQR.	=> concern	(3)
Is there any way we can access the site from our domain?	=> query	(4)
Please let me know if it is technically feasible.	=> query	(5)
Thanks,	=> greet	
Sender-Name	=> address	
Department-Name	=> address	
Location	=> address	
Office-Phone-Number	=> address	

Fig. 1. Example email within a software company

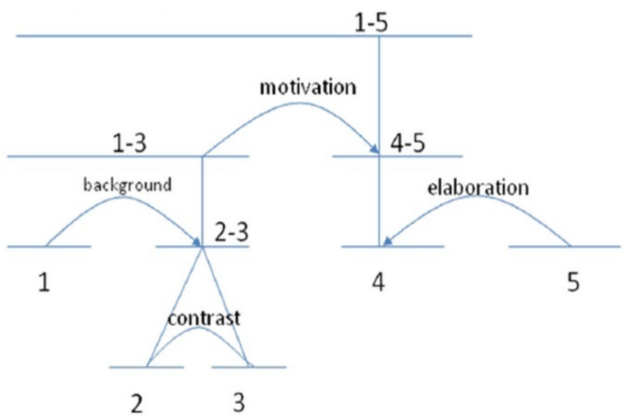

Fig. 2. Discourse parse (RST representation).

[10] has done zoning of email into different sections and at multiple granularities like 3-zone or 9-zone. Recently, zone segmentation was used to significantly improve accuracy in identifying emails containing requests for action [2] . Zone segmentation is a promising step towards analyzing content in the zones at utterance level.

Work on capturing speech-acts in emails was mostly at the message-level i.e. for example, whether an email contains a request [2]. Identifying requests at utterance-level would mean finding exact sentences where the request is expressed and even related sentences where some auxiliary information related to request is expressed. [11] has analyzed requests and commitments at utterance level and conducted annotation experiments to show that they can be defined and identified with high precision.

A lot of work has been done on automation of triaging and automatically responding to help-desk emails [17]. The approach in these systems is mostly keyword based. They assume the whole email as a query and the key words in the email are used to compare with previous emails using some similarity metrics and the earlier responses to similar emails are used to compose the response to the email at hand. This approach does not involve identifying or understanding structure of email and is mostly at message level. Tagging of Discourse Segment can significantly enhance the task of identifying question and answer pairs by locating the query in the email precisely.

TABLE I
CLASS WISE DISTRIBUTION OF DATA

Greet	Background	Goal	Concern	Query	Address
23.87%	23.81%	2.64%	11.90%	17.24%	23.17%

Current work is an attempt at identifying speech acts in an email at utterance level. Automatically identifying email speech acts in sentences of an email have not been attempted. Utterance-level speech-act recognition is a complex task compared to message-level identification especially when there can be multiple requests in an email because there can be confusion in identifying which sentence is setting up context for which query. Typically every query has a context and background and some utterances provide that context. [12] has analyzed this problem elaborately and mentioned it as *Locus Ambiguity*. To overcome this issue of Locus Ambiguity, current work is taking only emails where there is single line of argument leading to one query or request.

III. CORPUS AND ANNOTATION SCHEME

A. Corpus

Most corpora that are used for email research are not open due to privacy reasons. The popular publicly available corpora are Enron corpus [5] and BC3 corpus [6]. Since we are analyzing a helpdesk scenario, both Enron corpus and BC3 corpus are not appropriate for the analysis. So, we choose the emails of a) postgraduate admissions helpdesk and b) IT support helpdesk of our university. 231 emails which were short and contained around 10-15 sentences were chosen and were annotated. They had 1705 sentences which were annotated with one of the 6 classes. Annotation was done by us.

Issues that came up during annotation were the following:

Choice of unit: The emails were manually segmented. The units could be phrases or full sentences. We chose a sentence as a unit because we did not find a uniform way to decide on when to spilt the sentence into clauses to annotate. The study on further splitting could be taken up in future.

Multiple classes in the same sentence: There is a possibility that there could be two or more classes in the same sentence. For example, a sentence like "*We want to use ABC software, but we are not able to access it from our domain -HYDPQR*" has both the goal and concern in it. So, the annotator was asked to annotate with the important class out of the two in case of such a clash. The order of importance was chosen as Query > Concern > Goal > Background > Address > Greet. This is purely on the basis of importance of some classes of information for processing of email at the help-desk. In a help-desk scenario Query is the most important information for automation and then comes Concern and Goal. Greet is given least importance since it contains least amount of information.

B. Annotation schema

DAMSL [13] is the most popular schema which is the base for most of the annotation schemas which emerged out of it

TABLE II
CATEGORY DEFINITIONS

Class and definition	Example(s)
Greet: Greetings and Pleasantries	Hi Sir, Respected Sir/Maam, best regards, eagerly waiting for your response
Background: Context setting by the author. Mainly informative.	I have written the PGEE entrance exam on feb-19th 2009. My name is ABCD and I work at the Hyderabad center.
Goal: What is the user trying to achieve.	I want to know when I will get my result.
Concern: Any issue faced by the author.	I am not able to login to the server with the given details. I paid the DD but did not receive any acknowledgement yet.
Query: The request being made	Can you please tell me when I can get the result of the exam. Please tell me process for applying for PhD at your college.
Address: Description/Signature of author	Abhishek. Application Number: 328839 Mobile: 9490055922

and are used to annotate the speech acts in an email. However, it was defined for online conversations encoding the state and very high number of speech act categories. Other taxonomies like [14] [15] too were either with very high number of speech act categories or were designed for the speech acts at message level.

[9] have classified email into a verb-noun pair speech act at the message level and not at the level of sentences. The Verbmobil schemas granularity is a misfit for the task. Background and Concern would come under inform class. It is important to separately identify these classes to reach the goal of query processing. Since our goal is a schema for emails, which helps understand email better especially in a help-desk environment, we came up with a new schema. We classified the sentences into 6 classes namely Greet, Background, Goal, Concern, Query, Address. If we are able to classify sentences into these 6 classes, identifying the query and accordingly doing further processing for tasks like auto-response, issue-based redirect, query to database etc can happen. So, we chose to annotate the sentences in emails into these 6 classes. Table II defines the classes and gives some examples for each class.

IV. APPROACH

The approach we took to learn the classes was to convert each training sentence into a feature vector and all the feature vectors of the training data and their respective classes information would be given to the SVM classifier to learn. We experimented with various features to learn the classes. Many interesting observations were made in this process. Observation in [16] that stemming of words and stop word removal actually reduces the accuracy of classification was reinforced. SVM was used as the classifier since it outperformed other classifiers in [16] during a similar task. The implementation of SVM classifier in libsvm was used. We employed 10 fold cross validation procedure.

The baseline accuracy using the lexical n-grams (up to trigrams) of the sentences as features to learn was 33.06%. Then we did preprocessing of sentences as done in [1], but also went on to generalise sentences by suppressing noun phrase information since we wanted the system to be domain independent. Then we used n-grams (up to trigrams) POS tags of the lexical items in the sentence and it boosted accuracy to 60.3% which was a significant jump given that the number of classes is 6 and baseline was only at 33.06%. We increased the number of features by incorporading more n-grams (of length 4 and 5), but this only reduced the accuracies. When we reduced the number of features by choosing only those n-grams which appeared atleast twice, the accuracies went up again. This gave rise to an important observation that "diluting" the features reduces the accuracy. At this stage, on data analysis we noticed that there were some lexical items in each class which are typical to that class and can help classify. So we built a rule-based component which would check for some lexical features in the input sentence and produce a "guess-class". For example, a sentence having a word "hi" is highly probable to belong to "greet" class. So the "guess-class" of such a sentence would be "greet". Such rules were written for each class. This "guess-class" information was used as one of the features for learning and it improved the classification accuracy to 68.3%.

We shifted all the characteristics which we used as rules earlier to produce a "guess-tag" as features in the feature vector. This gave our best classification accuracy of 74%. Not surprisingly, the classifier learnt the rules better and the hand-picked features which were unique to each class helped distinguish classes. The features which we chose are listed in Table III.

TABLE III
HANDPICKED FEATURES

Greet	sir, dear, ma'am, hi, hello, warm, regard(s), wish, thank, madam,sincerely,yours, truly, obediently, await, waiting, grateful
Query	please, pls, plz, pl, kindly,sentence starting with do/what/why/when/how/where/can/ would/could/should/let/may/ ,question mark(?) at the end
Address	Number of words starting with capital letter, Total number of words
Concern	but, not

V. ERROR ANALYSIS

Table IV shows the confusion matrix for classification with hand-picked features and pos n-grams (upto trigrams) as features. One can notice high classification accuracies of 92.52% and 98.73% for "Greet" and "Address" classes respectively. Though not very high, the accuracies for "Background" and "Query" classes are also good. "Concern" got misclassified as "Background". The confusion matrix shown in Table IV does not show "goal". The reason for this is that the data for goal is very less and hence misclassification was very high. So, for now we did analysis without "Goal", but we plan to work on that soon. Following are few observations looking at the confusion data:

Concern as Background: One can notice from the confusion matrix that the toughest classes to differentiate are Background and Concern. We see that domain information and context information will be important in classifying these two classes. For example, the sentence "*Internet has not been working on my system for last 3 days.*" looks like a concern. But consider the following email.

TABLE IV
CONFUSION MATRIX

	Greet	Background	Address	Concern	Query
Greet	92.52	2.24	4.49	0	1
Background	0.74	84.9	8.91	2.23	0.99
Address	0.76	0.51	98.73	0	0
Concern	0.49	49.02	0.49	48.53	1.96
Query	0.34	14.24	2.03	0.68	82.71

Dear sir,
My name is ABC, and I am in XYZ department. Internet has not been working on my system for last 3 days. I have raised a support-ticket couple of days back but no action has been taken yet. This delay in handling support tickets is now very frequent. Request you to make a serious note of this and enquire into the situation to make sure that support-tickets are solved promptly.
Thanks,
ABC.
In the email above, the sentence "Internet has not been working on my system for last 3 days." is the background to the query's context. The concern is related to solving support-tickets on time. It can be deduced only when one notices the query and its context. This highlights the need for finding the discourse parse of the email. Only when we can identify which sentences are related to which ones, can we capture the information in the email precisely.

Background as Address: This was because in the middle of the email some text is written like an Address. E.g:
(greet) Hi Help-team,
(background) Following is my system configuration:
(background) Processor: ABC
(background) Memory:2GB
(background) Hard-disk:500GB
(query) Please let me know whether the new OS Bambino-3 can run on this system.
(address) Sender-Name
(address) Department-Name
The 3 lines in the email above where the configuration of the system is listed are similar to the address information if capitalization and sentence length are the only differentiating factors. Spatial features like location of sentence might help solve this issue.

VI. FUTURE DIRECTIONS

Doing zone segmentation [10] first and then classifying utterances is a logical next step to be attempted. Also to handle cases of multiple lines of arugument where there can be multiple queries and hence multiple contexts to explain each query. So classification in such a setting needs to be explored. After acheiving good accuracies of classification, we plan to use this classification information to identify the relations between sentences. With that information, the confusion between background and concern can also be addressed and concerns and queries can be identified with higher precision.

VII. CONCLUSION

Emails are highly unstructured documents and trying to understand the strucutre of them is a complex task. Helpdesks in organistations need to process huge volumes of email and automation of response would be of great value. Keeping that in mind, in this work we proposed a schema for utterance-level classification of author composed sentences in help-desk emails. Identifying the queries, concerns and context(background) of those the query is the task we have taken up. Though not comparable with other works because of difference in data used, we have achieved good classification accuracies using a combination of n-gram features and hand-picked features. Error analysis shows the limitations of current feature set and motivates for exploring new features which can classify the sentences better. We also proposed that automation of this classification is an important step towards finding the discourse parse of emails.

REFERENCES

[1] Vitor R. Carvalho, *Improving email speech act analysis via n-gram selection*, In Proceedings of the HLT/NAACL 2006 ACTS Workshop, 2006.
[2] Andrew Lampert and Robert Dale and Cecile Paris, *Detecting Emails Containing Requests for Action*.
[3] Jan Alexandersson, Bianka and Buschbeck-wolf Tsutomu and Tu Berlin and Michael Kipp, Stephan Koch, and Melanie and Elisabeth Maier, Norbert Reithinger, *Dialogue Acts in VERBMOBIL-2 Second Edition*, 1998.
[4] Andrew Lampert and Robert Dale and Ccile Paris, *Classifying Speech Acts using Verbal Response Modes*.
[5] Bryan Klimt and Yiming Yang, *Introducing the Enron Corpus*, 2004.
[6] J. Ulrich and G. Murray and G. Carenini, *A Publicly Available Annotated Corpus for Supervised Email Summarization*, AAAI08 EMAIL Workshop, 2008, Chicago, USA.
[7] J.L. Austin, Oxford University Press, *How to do things with words*, 1962.
[8] John R. Searle, Cambridge University Press, *Speech Acts: An Essay in the Philosophy of Language*, 1969.
[9] William Cohen Vitor, *Learning to Classify Email into "Speech Acts"*.
[10] Andrew Lampert and Csiro Ict Centre and Robert Dale, *Segmenting Email Message Text into Zones*.
[11] Andrew Lampert and Robert Dale and Ccile Paris, *The nature of requests and commitments in email messages*, In Proceedings of the AAAI Workshop on Enhanced Messaging, 2008.
[12] Andrew Lampert and Robert Dale and Ccile Paris *Requests and Commitments in Email are More Complex Than You Think: Eight Reasons to be Cautious*
[13] Mark G. Core and James F. Allen , *Coding Dialogs with the DAMSL Annotation Scheme*, 1997.
[14] Jade Goldstein and Roberta Evans Sabin , *Using Speech Acts to Categorize Email and Identify Email Genres*, HICSS, 2006.
[15] Anton Leuski, *Email is a stage: discovering people roles from email archives*,(SIGIR 04), 2004.
[16] Anthony Khoo and Yuval Marom and David Albrecht, *Experiments with Sentence Classification*.
[17] Rahul Malik and L. Venkata Subramaniam and Saroj Kaushik, *Automatically Selecting Answer Templates to Respond to Customer Emails.*,(IJCAI 07),1659-1664,2007.