AiGraph for Meeting Insights Relevance

Bipasha Sen
bise@microsoft.com
Microsoft

Prakash Pandey
prakpand@microsoft.com
Microsoft

Rajeev Gupta
rajgup@microsoft.com
Microsoft

Vipin Vangala
vipinv@microsoft.com
Microsoft

ABSTRACT

In this paper we present *AiGraph*, an enterprise knowledge graph, representing details about how an employee communicates through emails, meetings, and documents. By representing all her communication in the form of a graph, we are able to extract complex insights which are computationally expensive in silo'ed applications. We consider a recommendation application – *Meeting Insights* – to show power of AiGraph. This application *recommends* related emails and documents for a given meeting. There are a number of ways in which AiGraph can improve the Meeting Insights – most signifcantly, it can improve the relevance of the system by providing better candidate emails; and features for a ranker to rank these candidates. In this paper we describe various ways to improve relevance of Meeting Insights using AiGraph.

KEYWORDS

AiGraph, Meeting Insights, Recommendations, knowledge graph, graph representation, machine learning, pagerank

1 INTRODUCTION

According to DMR, each day accounts for 269 billion email exchanges with an average professional receiving 121 emails per day [1]. As of 2019, Statista reports Outlook as the third most important email client accounting for 9% of the worldwide traffic [3]. Major companies have shifted their efforts to intelligently ingesting this data and predicting user behavior to power intelligent applications like suggestion of relevant emails and files, sending reminders, interactive cards, etc., to present important information in a structured format to the user. This recommended information is either extracted from the email itself or the data is queried at run-time to power these applications, leading to delays, recency, and relevance issues. In this paper, we consider one such application called *Meeting Insights*, where relevant emails and files are suggested for a given meeting. Specifically, the meeting subject, body content, people involved, meeting time, etc., are used as context to query the system and recommendations are provided. This requires a number of queries to the massive data store in run-time to find the most relevant candidates. Typically, these parameterized queries are handcrafted to retrieve set of candidates (L1) which are then passed through a ranker (L2) to score and rank the retrieved candidates. These retrieved candidates are limited by the set of handcrafted queries which leads to a trade off between the latency and the performance.

AiGraph is a shallow knowledge-graph representation of all the communication done by a single user. Thus, it has relationships between emails sent and received by a user, documents created/updated/accessed by the user, meetings scheduled or attended by the user, etc. A separate graph is created for each user with individual emails, contacts, meetings, documents, keywords/topics, etc., as nodes. Emails and contacts have edges corresponding to the email sender and recipients, meetings and contacts has edges for the meeting organizer and attendees, a keyword node has edges to all the emails and meetings for whom that's an *important* keyword. There are 12 different types of edges in AiGraph. Instead of using a number of queries to get the L1 candidates, We employ AiGraph as a source of (L1) candidates to get relevant emails for a given meeting. We show that by using interconnection between various communication artifacts corresponding a user we can improve coverage, density, and relevance of the Meeting Insights recommendations.

2 PIPELINE

AiGraph is represented by a set of nodes and relations. Each day, the Outlook data is mined for each user to construct the AI Graph. Entities like emails, meetings and documents form the core nodes of the graph. Nodes like attachments, contacts, topics, acronyms, etc., are derived from these core nodes. The nodes are represented by important metadata such as node-ids and topics (obtained by processing the content of the email, meetings and documents) etc. These nodes are connected by edges representing meaningful implicit semantic relations such as *EmailSender*, *DocumentAuthor*, *DocumentModifier*, *MeetingOrganizer*, etc. The paths and the semantic relations between different nodes are exploited to obtain related email and document nodes for a given meeting node. This pipeline of candidate generation (L1) for a given meeting is given by a set of sequential components, as we describe below.

2.1 Graph Search

Graph search is a function of maximizing the number of *relevant candidates* by obtaining a sub-graph for a given node. To obtain the sub-graph, the node corresponding to the meeting is used as seed node which is then used to fetch rest of the nodes of the sub-graph by hopping through the edges. A hop-length of 3 along with Breadth First Search (BFS) approach is used retrieve the sub graph. The nodes in the sub-graph are ranked based on the structural and implicit semantic relationships using a graph walk algorithm. AI Graph supports multiple graph walk algorithms namely HITS[2], weighted HITS[4] and PageRank[5]. The ranked list of relevant graph nodes is then returned by the function.

2.2 Meeting Insights V1 Adapter

Meeting Insights V1 Adapater is a plugin on AiGraph to process the list of candidate nodes returned by the Graph Search function and obtain a subset of the candidates such that the following criterion is met,

$$(M_p \wedge E_p) \wedge (M_t \wedge E_t) \tag{1}$$

where M_p denotes meeting participants, E_p denotes candidate (email) participants, M_t denotes topics of the meeting and E_t denotes the topics of the candidate (email). Thus, an email is a candidate for a meeting if it shares at-least one contact node and one topic node with the meeting.

These candidate nodes are ranked on the score of the graph walk algorithm obtained from the Graph Search function. The V1 Adapter validates the performance of AiGraph by maintaining the relevance of the candidates against the production (L1) candidates. The relevance of the adapter is improved by the introduction of a ranker mentioned in the next subsection.

2.3 Meeting Insights V2 Adapter

Meeting Insights V2 Adapter is an additional plugin on AiGraph to rank the Graph Search candidates on their relevancy. It is a model trained on handcrafted features computed on the sub-graph obtained from the Graph Search function. The ranking problem is treated as a classic classification problem where a model is trained to classify an email or document node as a relevant node. The model assigns a score to all the nodes obtained from the Graph Search function. The candidates are ranked on the score and top N candidates above a pre-determined threshold are returned.

Different models are trained offline on the data obtained by scraping user's Outlook emails, meetings and shared documents content. The model is trained on a fundamental assumption, for a given meeting, any email attachment to the meeting is considered a positive data-point, i.e., the candidate (email) corresponding to the attachments are positive data points whereas all the other returned candidates are negative data points. This assumption creates a huge imbalance between the positive and negative data points in the data-sets. The models are trained on two datasets: Eyes-on and Eyes-off. The Eyes-on data is obtained by scraping the data of the team members involved in the project. The Eyes-off data is obtained by extracting 3 months of MSIT data through Heron pipeline. Experiments on this dataset is performed on Aether.

Model training on Eyes-on data is done in iterations to deal with the class imbalance problem. The first model is trained on the aforementioned assumption. This model is then used to rank and fetch top 20 candidates for each meeting for seven in-house users. The users then self-annotate these candidates as negative/positive predictions. A dataset with 4630 datapoints, with 328 positive samples is obtained through this exercise which is then used to iterate and train different model versions.

The Eyes-off data is made of $73k$ positive datapoints and $43M$ negative datapoints. Techniques like random negative sampling and Positive-Unlabeled learning (PUL) is used to deal with the class imbalance.

(1) In random negative sampling, the unlabeled samples are first randomly divided into 400 mini-datasets. For each of the mini-datasets, $100k$ negative datapoints are randomly sampled and concatenated with the $73k$ positive datapoints to train a model. One model out of the 400 models trained on the mini-batches is chosen on the basis of the individual precision-recall score on an unseen test dataset.
(2) In PUL, the model is trained in iteration by obtaining and expanding a set of reliable negatives. The reliable negatives are obtained on the Heron dataset using the models trained on step 1. A new dataset is then formed by combining reliable negatives and positive datapoints to train a final model.

2.4 Meeting Insights $\mathbf{V_{next}}$ Adapter

Meeting Insights V_{next} Adapter is a proprietary function of merging the candidate nodes from V1 and V2 Adapters such that redundant candidates are eliminated.

3 RESULTS

Table 1 presents a comparison between the recall measure of the different Meeting Insights adapters. *MCand* represents the maximum number of candidates returned by the adapter. *FCand* represents the maximum number of candidates returned by the V2 Adapter. *C* denotes the coverage of the adapter where the value represents the number of meetings for which the adapter makes suggestions. All the experiments are performed on the same set of MSIT data while keeping the threshold for the V2 Adapter at 0.5.

Table 1: Recall of AI Graph (L1) Candidates On Offline Data

			V_{next}	
MCand	V1	FCand	V2-Eyes-On	V2-Eyes-Off
C	~178k	N/A	~182k	~186k
100	**29.3**%	N/A	N/A	N/A
20	27.4%	10	28.85%	**30.04**%
10	25.4%	5	26.7%	27.6%
5	22.7%	3	23.4%	23.12 %

The V_{next} Adapter with V2 trained on MSIT Eyes-off data improves the recall of the V1 Adapter by 10% at Top 20 candidates. It also obtains a higher recall score at Top 20 compared to the V1 adapter at Top 100. We also see that the coverage improves in both of the V_{next} configuration while maintaining high recall. Through flighting for the MSIT users, we have obtained an increase in the coverage and density for emails by 18% and 33%, respectively.

4 CONCLUSION

Relevant Candidates (L1) fetched by handcrafted queries by the Meeting Insights application in run-time limits the performance of the application in terms of coverage, density and relevance. AiGraph, an enterprise knowledge graph, as a source of relevant (L1) candidate improves the performance of the application by increasing the coverage, density and relevance of the application by more than 18, 33 and 10%, respectively. Further, the (L2) ranker can be retrained on the features computed in Meeting Insights V2 Adapter to make the ranker robust for the AiGraph (L1) candidates.

ACKNOWLEDGMENTS

To the Recommendation and AiGraph team for the collaboration.

REFERENCES

[1] Smith Craig. 2020. 90 Interesting Facts About Email | How Many Emails are Sent Per Day? https://expandedramblings.com/index.php/email-statistics/. [Online; accessed 1-July-2020].

[2] Takayasu H Takayasu M Deguchi T, Takahashi K. 2014. Hubs and Authorities in the World Trade Network Using a Weighted HITS Algorithm.

[3] Richter Flex. 2019. The World's Most Popular Email Clients. https://www.statista.com/chart/17570/most-popular-email-clients/. [Online; accessed 2-April-2019].

[4] J. M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM, 46(5):604–632, 1999.*

[5] Lawrence Page. 1997. Pagerank: bringing order to the web.