Slide 1: Hello, everyone. Today I'm going to talk about a paper called "Email as Spectroscopy: Automated Discovery of Community Structure within Organizations". This paper is one of the most exciting works I've read recently. I will try my best to present this work well, and I hope all of you can enjoy this work.

Slide 2: There are mainly five parts of my presentation today. First, I will introduce the paper's background and the authors' information about this paper. Second, I will discuss the problem we want to solve. Next, I will illustrate their methodology and solution and discuss the experiment results. The last part is the conclusion.

Slide 3: This paper was published in the journal called the information society, volume 21 in 2005 issue 2. The paper was received on March 16th, 2004 and got accepted on June 30th, 2004. This work was done by HP labs and had three authors. Joshua and Dennis are all full-time researchers at HP labs. Bernardo Huberman was a professor at Stanford University. He was the senior fellow and director of the HP labs. Currently, he serves as Vice President of Next-Gen Systems at CableLabs, which mainly focuses on applying AI techniques to networking, and started a successful effort in quantum communications.

Slide 4: Now, I will introduce the motivation behind this work. In addition to the formal architecture of an organization or a company, we always have some informal connection networks. For example, at McGill, you may be affiliated with the school of computer science, but you may also work with students or professors from the EE/ECE departments.

Formally, we define these informal connection networks as the communities of practice, which are the informal networks of collaboration that naturally grow and coalesce within the organization. Usually, these communities can be used to uncover the reality of how people find information and execute their tasks. Obviously, these informal communities co-exist with the formal structure of an organization.

After finding these communities of practice, we can use them for many purposes, such as resolving the conflicting goals of the institution to which they belong, solving problems in more efficient ways, furthering the interests of their members, and enhancing the productivity of the formal organization.

Slide 5: As we mentioned, there many values of the communities of practice. Therefore, we desire a fast and accurate way to identify them.

Previous works solved this problem by gathering data through interviews and surveys. Then they constructed links and the communities of practice by manual inspection. This method is accurate but time-consuming and labor-intensive.

Slide 6: In this paper, the authors introduced an automated algorithm to identify the communities of practice through email logs. They only used Email data to construct a network

of correspondences and then discovered the communities by partitioning this network with their proposed algorithm.

Moreover, constructing the network of correspondences only needs the names of senders and receivers, which minimizes many privacy issues.

Slide 7: First, I would like to give you an overview of the method in this paper and then go to the details of each step.

There are two principal steps of the proposed method. The first step is how we construct the network of correspondences, and the second step is to use the algorithm we will introduce later to find the communities embedded in the graph.

Slide 8: I will explain the first step on this slide. The idea behind constructing the network of correspondences is very intuitive. The vertices of the graph are people, namely the senders and receivers who appeared in the email logs. For edges, we connect two vertices if the number of emails passed between them is larger than or equal to the threshold. In this paper, the authors set the threshold to 30.

Slide 9: Next, I will elaborate on the algorithm for partitioning the graph we constructed in the first step into several communities of practice. The idea is straightforward. First, we remove a "certain" edge from the graph so that a giant connected component will be partitioned into two separate connected components. Then, we keep removing edges until we find some communities of practice.

Unfortunately, some ambiguities need to be clarified. The first question is how should we define a community in a graph? And how can we identify the edges to be removed? Finally, when should we stop partitioning further? In the following slides, I will explain how each ambiguity is clarified.

Slide 10: First, how should we define a community in a graph? Formally speaking, we say a graph has a community structure if it consists of subsets of vertices, with many edges connecting vertices of the same subset but few edges lying between subsets.

For example, as the figure shown on the right, we have two subsets of vertices. One is denoted with square vertices, and another one is denoted with circle vertices. There are many edges connecting vertices of the same subset. In contrast, there is only one edge connecting these two subsets, which is the edge connecting vertex A and B.

Slide 11: Next, how can we identify the edges to be removed? To solve this question, we need to define two types of edges. As we mentioned in the previous slide, there are many edges connecting vertices in the same subset, while few edges are between two subsets. Thus, we define the edges connecting two subsets as inter-community edges. Respectively, the edges connecting vertices in the same subset are called intra-community edges. In the figure shown

on the right, the edge connecting A and B is an inter-community edge. All the other edges are intra-community edges.

Slide 12: Furthermore, we need a quantitative way to identify the inter-community edge so that we can remove them in the following steps. In this paper, the authors adopted a technique of calculating the betweenness of each edge, and they used Freeman's definition of betweenness.

So, The betweenness of an edge is defined as the number of shortest paths that traverse it. The inter-community edges link many vertices in different communities and have a high betweenness. In contrast, the betweenness of the intra-community edges is relatively low.

Slide 13: For faster calculating the betweenness for all edges, there is an algorithm proposed by Brandes.

We first choose one vertex, which is called the center. And for all other vertices, we find the shortest paths connecting them and the center. After that, we calculate the betweenness of each edge based on these shortest paths and add them to a running total.

Subsequently, we change the center to another vertex and repeat until all vertices have been at the center once.

As a result, the running total for each edge is equal to exactly twice the exact betweenness of that edge.

Slide 14: Last question, when should we stop partitioning further? Generally speaking, We should stop removing edges when we cannot further meaningfully subdivide the communities.

For example, after removing the edge connecting A and B, we cannot meaningfully partition the graph further.

Keeping this in mind, we should discuss two different cases.

Structurally, a component of 5 or fewer vertices cannot consist of two viable communities. As we can see from the example shown on the right, the square vertices and circle vertices form two communities, respectively. Imagine that if you only have five vertices, you cannot form two communities.

Therefore, the first stopping criterion is: If at any time we remove an edge from our graph and separate a component of size smaller than 6, we can identify it as a community.

Slide 15: The second case is that Components of size equal to or greater than six can also be individual communities, like the community formed by the circle vertices in this figure (指一下).

To derive the stopping criterion for this case, we should notice one important thing first. Generally, the single edge connecting a leaf vertex to the rest of a graph of N vertices has a betweenness of N – 1 since it contains the shortest path from X to all N – 1 other vertices. For example, the edge connecting X and Y, in this case, has betweenness 5 because the edge is included in all shortest paths from X to all the other vertices.

Consequently, the stopping criterion for components of size equal to or greater than six is, therefore, that the highest betweenness of any edge in the component be equal to or less than N – 1.

Slide 16: Obviously, after removing an edge, we should recalculate the betweenness for all edges. Therefore, the order of removal of edges affects which edges are going to be removed.

In the beginning, there will be many inter-community edges with high betweennesses, or even some of them have the same values of betweenness. The choice of which to remove is arbitrary but dictates which edges will be removed later and will result in different sets of communities at the end.

Fortunately, this is not a problem. Even better, we can take advantage of this arbitrariness by repeatedly partitioning the graph into many different sets of communities and then comparing the different sets and aggregating the result into a final list of communities.

In other words, we can choose different inter-community edges to start the partitioning process and get different sets of communities at the end of the proposed algorithm. We finally integrated all the helpful information.

However, you may ask a question. Why this arbitrariness is important and useful?

Slide 17: To explain this question clearly, let me give you an example. Consider the placement of two people called John and Sarah in communities.

If John appears within the same community in all 50 sets, it is clear that John definitely belongs to that community. The order of edge removal had no effect on him.

However, if Sarah appears in one community in some sets in another (or even several others) in other sets, the order of edge removal did affect her. We should consider that she has some affiliation with those two (or more) communities.

If we only considered one community structure, Sarah would have been placed in one community rather than several communities, and we would have lost information about her role in the other community (communities).

Slide 18: To be more vivid, I show an example through a graph in this slide. Clearly, the graph consists of two communities, one on the left, including vertex A, and another on the right, including C.

In this case, BC initially has the highest betweenness among its edges, and AB's betweenness is also high.

Therefore, there are two different situations.

If we choose to remove BC first, AB becomes an intra-community edge with low betweenness, which will never be removed, and vertex B will eventually be placed in a community with vertex A.

If we removed AB first, BC would have been rendered intra-community, and vertex B would end up in the community with C.

Slide 19: Formally, we randomly select one edge from all edges with high betweenness to start the proposed algorithm and repeat this procedure n times.

After repeating the proposed algorithm with a random start for n times, we obtain n community structures imposed on the graph. We can then compare the different structures and identify communities.

For example, after imposing 50 structures on our graph, we might find: a community of people A, B, C, and D in 25 of the 50 structures; a community of people A, B, C, D, and E in another 20; and one of the people A, B, C, D, E and F in the remaining 5. We can write this result as the following way.

Here we signify that A, B, C, and D form a well-defined community, E is related to this community but also to some other(s), and F is only slightly, possibly erroneously, related to it.

Slide 20: 解释一下这个 pseudo-code。如果有人问 modified Brandes Algorithm 是什么的话，这个东西其实就是为了 randomly select start edge，并不是算好了之后从高的里面抽。而是在用 brandes algorithm 计算 betweenness 的时候，只随机选 m 个点作为 center。算出来之后，依然取 betweenness 最高的 edge 开始 remove。

Slide 21: 念 Slides

Slide 22: 念 Slides
一些 Discussion：这个验证结果的时候，16 个人 interview 的 sample size 有点儿小。至少 50 个或者 100 个比较好。另外一个问题就是为什么要 share 这篇 paper，一个 motivation 就是说邮件的 spam detection。NLP 的做法读邮件内容有 privacy issue，如果能只通过的交流记录就做出 spam detection 是更好的。另外就是在 web3 里面的 fraud detection，我们

有很多的交易记录形成的 graph，我们可不可以通过类似的分析来找到可疑的 edge，也就是可疑的交易记录？

Slide 23: Here’re the references for today’s presentation. Please feel free to check them.

Slide 24: That’s all of today’s presentation. If you have any questions about the paper or comments on my presentation, please feel free to discuss them with me.