Abstract-2 Entity Recommendation for Daily Digital Tasks

Background

When we are performing a digital task, wouldn't it be great if our computers can make a list of suggestions for the task? For example, when we write a paper in LaTeX, which is a digital task, we perform task-related actions, such as e-mailing the version of the paper to the co-authors, drawing explanatory figures, and reading/citing the most relevant papers. To increase the efficiency of the digital task, we should be able to get suggestions, such as people, drawing tools, and related documents, as shown in Figure 1.

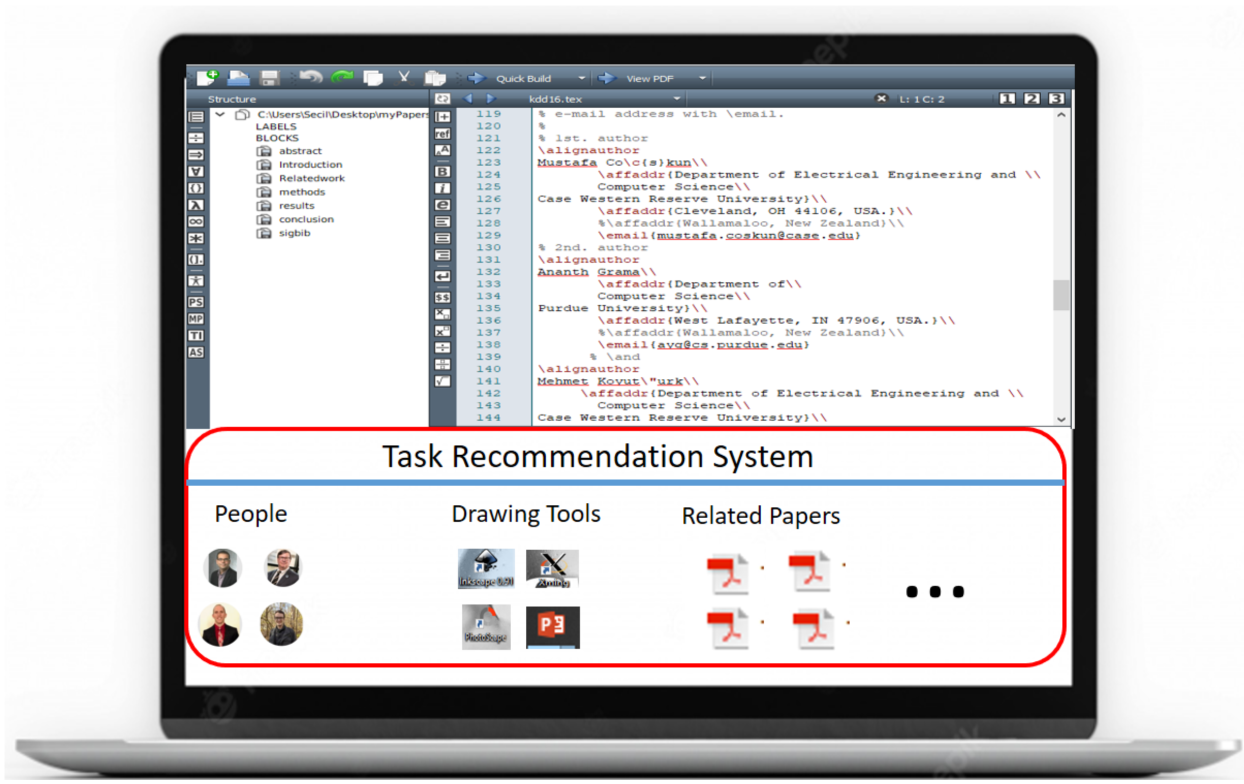

Figure-1: Task Recommendation System

Recommendation Systems (RS) have been extensively studied in academia and industries, see [1-10]. However, the traditional recommendation systems cannot be directly used for task recommendations due to their following drawbacks in this context:

A. The current Recommendation Systems (RS) are designed for fixed entity-wise recommendations, not evolving task-wise recommendations. Specifically, a person and an item are two fixed entities, and known items are recommended to known persons. On the other hand, tasks are evolving processes, even a performer of a task sometimes may not know how the tasks will evolve.
B. The current RSs assume there is a “guild-by-association” [2] relationship between items and users, i.e., if some users who are similar to a given user buy certain items, then due to the similarities, the user could buy these certain items. Task recommendation, on the other hand, cannot be related by guild-by-association. For example, we have two tasks:

write paper-A and paper-B. paper-A and paper-B can be written by completely different co-authors, with completely different tools, etc. Thus, tasks are self-evolving processes, we cannot relate tasks with guild-by-association.

C. In RSs, since there is a hidden “guild-by-association”, a Machine Learning (ML) model is trained to learn to encapsulate the similarity association across different entities (people, items or applications). However, in task recommendation, the ML model cannot be learnt from other people’s digital daily tasks, since each task of an individual is a self-evolving process and unique which cannot be solved via a trained generalized ML model. Thus, a user-specific technique is necessary for task recommendation problems.

The aim of this invention disclosure is to develop a task recommendation system for daily digital tasks. To enable this, we propose a novel approach that determines what currently performed task is based on Gaussian Mixture Model (GMM). Subsequently, we integrate the GMM with a random walk-based approach to present the final task-related tools and applications to the user.

Summary

This invention disclosure proposes techniques that are used to recommend tools, applications, documents, and people, for daily digital task recommendation. These daily digital task recommendations have use cases, such as searching for appropriate tools online and organizing files on computers based on their usage.

The proposed techniques present two novel approaches:

1. We propose to monitor each individual’s workspace, such as laptop, PC, and smartphone and determine what currently (ongoing) performed task is by using unsupervised Gaussian Mixture Models (GMM). This GMM-based approach takes snapshots of the user workspace and a broad category of the ongoing task as input. Then, the GMM approach outputs a confidence score that shows what is a narrow category of ongoing tasks, such as whether it is a patent, paper, project, etc.
2. We then propose a technique that integrates formerly performed tasks with GMM confidence scores on a graph so that ongoing task-related recommendations can be made by using random walk-based approaches.

Detailed Description

The daily digital tasks vary from user to user, i.e., tasks are user-specific. Thus, this disclosure proposes a user-centric approach that is processed and learned for each individual from his/her own data. To enable this user-centric approach, we propose to monitor the user’s digital workspace in consecutive time intervals, such as every 2 seconds, we can take a screenshot of the workspace.

After collecting such user-specific data, the invention disclosure proposes two novel approaches for digital task recommendation:

- Unsupervised Task Classification
- Task Informed Search

Details of Main Novelties:

A. <u>Unsupervised Task Classification:</u>

Intuitively, when a user performs various tasks on his/her computer, we can categorize these tasks into “super-tasks”. For example, document writing is a super-task, which might include patent, paper, and project writing subtasks; creating video content is another super-task, which might include creating YouTube, Instagram, and Facebook video subtasks. Clearly, we first need to determine these subtasks to suggest applications and tools for them. For the sake of argument, let us call “super-task” as “superclass” and “sub-task” as “subclass”, see Figure 2.

When we take screenshots of workspace, laptop, or PC, every 2 seconds, our user-specific data will be time-stamped images. With this data, determining superclass, such as document writing, is easy, i.e., if any text editing tool is used, that is the implication of superclass document writing. However, in this invention disclosure, we are particularly interested in determining subclasses: what is the document, is it a patent or paper? By only determining subclass, we can make accurate task recommendations. As an analogy, in an animal image, we aim to determine the “breeds” of the animal.

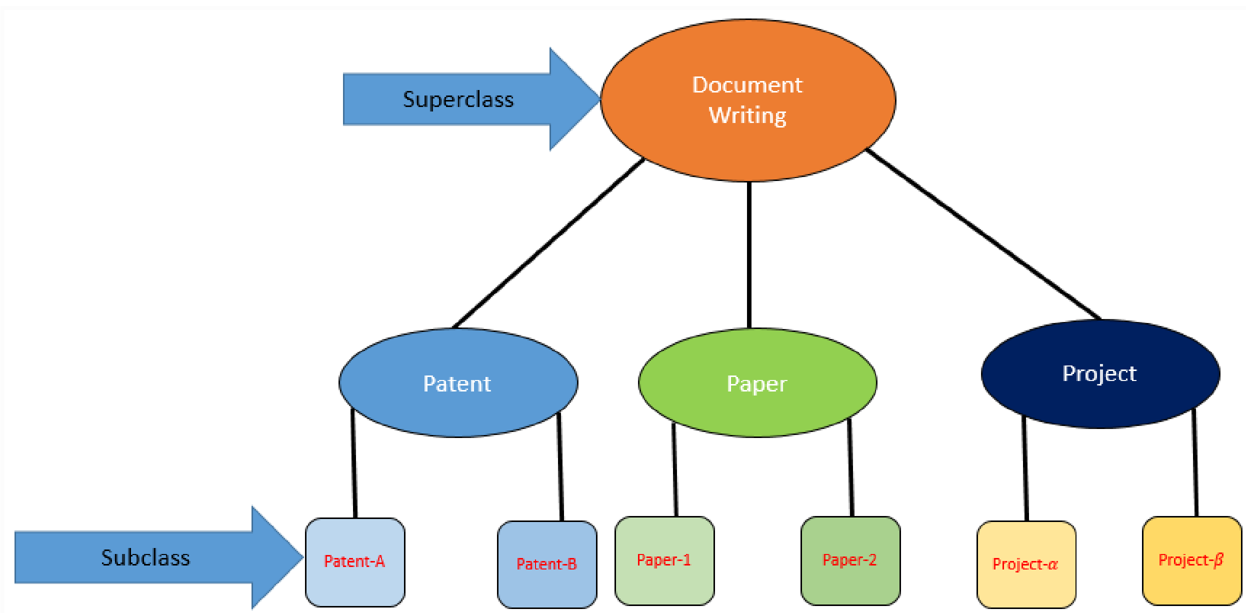

Figure-2: Document Wring is a superclass while writing Patent-A is a subclass. The first step is to **determine subclasses** from the user’s screenshots to determine what applications and tools will be recommended to the user.

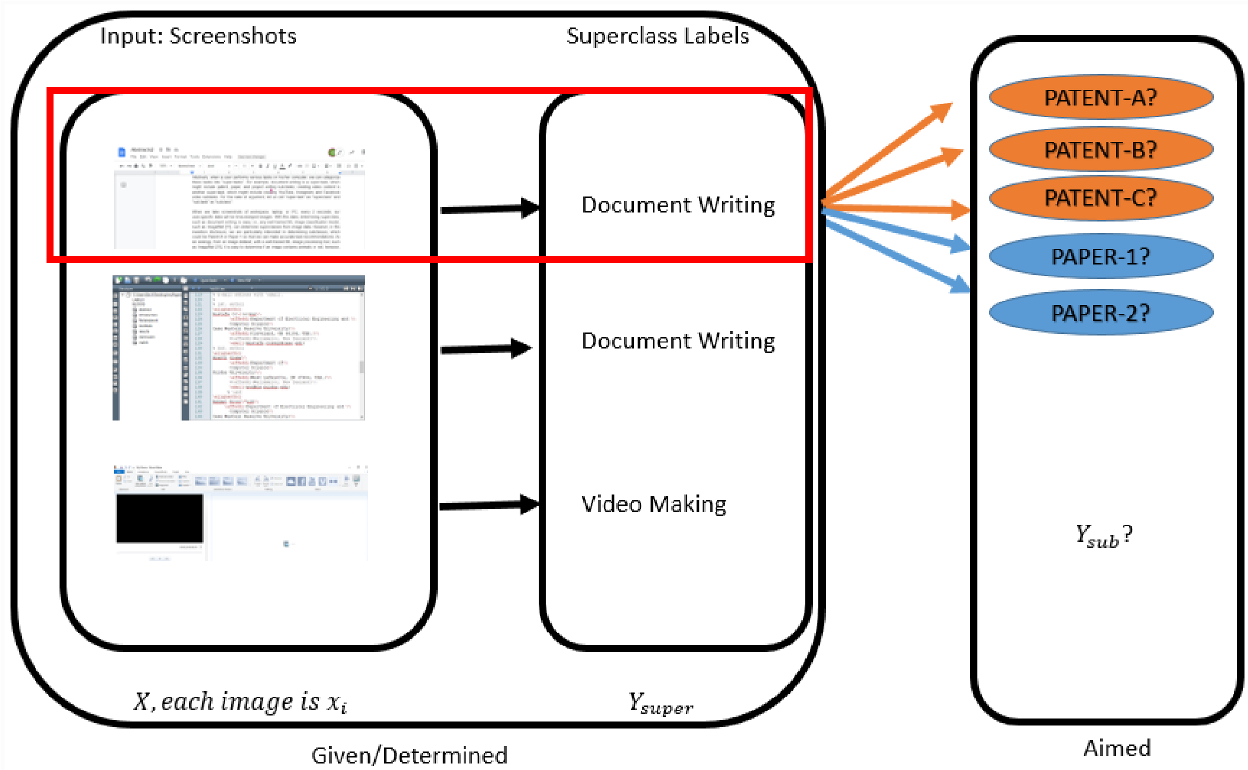

Figure-3: Given consecutive screenshots of images and superclass labels, we aim at finding subclass labels.

With these considerations in mind, as shown in Figure 3, we have input images, X, and superclass labels of images, Y_{super}, and want to determine subclass label of images, Y_{sub}.

How we are going to determine Y_{sub} :

- As shown in Figure 4, assume we observe a person's workspace from T1 to T2, by taking screenshots of his/her workspace.
- Assume from T1 to T2, he has performed 3 tasks: writing patent-paper-patent, which we want to determine without knowing what he is writing.
- Claim: each task revolves around certain subjects and specific words. Thus, in this disclosure, in one embodiment, we collect tf-idf weights of words, we attend a distribution of each subclass, task, in terms of its words tf-idfs as shown in Figure 4.
- Now the problem is turned into unsupervised distribution clustering. Thus, in one embodiment, we propose to use Gaussian Mixture Model /Expectation Maximization Algorithm to determine subclasses (Note we will write this algorithmic approach in more details).

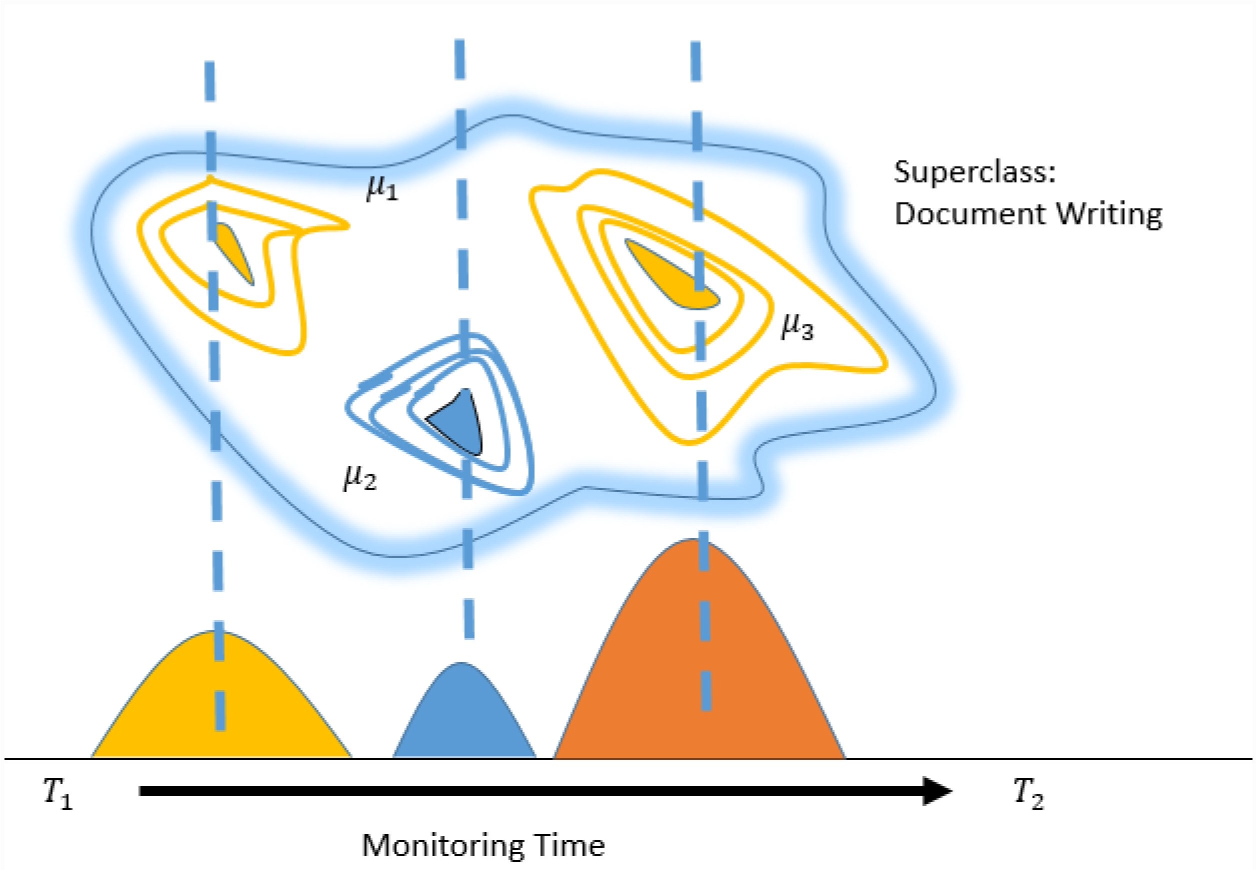

Figure-4: Distribution of tasks in terms of their tf-idf weights

B. Task Informed Search:

Note that step-A spits out the probability of the ongoing task belonging to a certain task, such as the ongoing task with 0.8 probability of a patent and 0.2 probability of a paper. We then utilize the former user interaction with the application, such as the user had interaction with person-1 five (5) times during the last patent writing in Figure 5. Now, we propose a simple association graph as shown in Figure 5 to determine what is going to be recommended for the ongoing task.

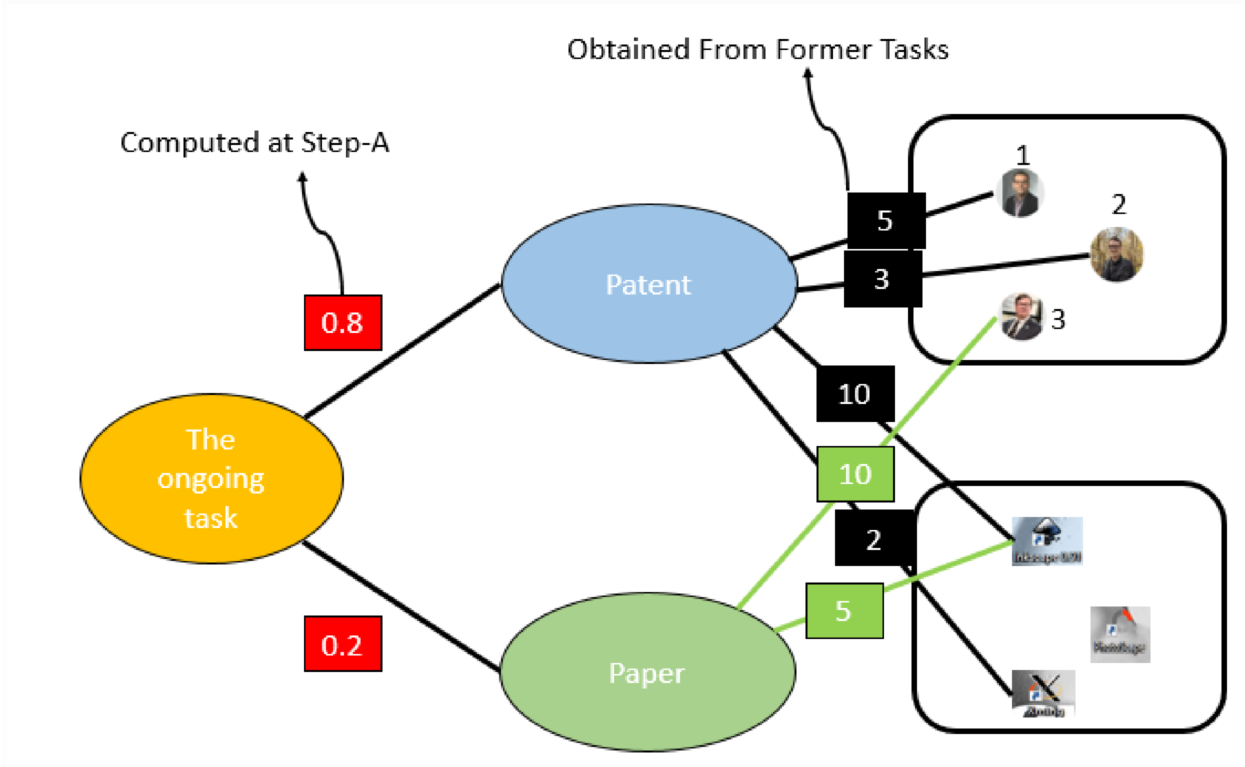

Figure-5: Task and tool/application/people relationship graph.

In one algorithmic embodiment, we can use random walk with restarts (RWR) [12], starting from the ongoing task node in Figure 5 and propagating over people, tools, and applications. More specifically, we reach person-1 with 0.8x5 = 4 hitting score; while reach person-3 with 0.2x10 =2 hitting score. Thus, we recommend person-1 before person-3.

References

[1] Aditya Grover and Jure Leskovec. “node2vec: Scalable feature learning for networks”. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. 2016, pp. 855–864.
[2] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. “Deepwalk: Online learning of social representations,” in Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. 2014, pp. 701–710.
[3] Jian Tang et al. “Line: Large-scale information network embedding”. In: Proceedings of the 24th international conference on the world wide web. 2015, pp. 1067–1077.
[4] Samuel Theodore Sandler. Real-time updates to item recommendation models based on matrix factorization. US Patent App. 15/633,696. Oct. 2017.
[5] Harald Steck. Techniques for recommending items to users. US Patent App. 16/664,761. May 2020.
[6] Aravind Sankar et al. Dynamic graph representation learning via attention networks. US Patent App. 17/273,275. Oct. 2021.

[7] Alberto Garcia Duran and Mathias Niepert. System and method for multi-modal graph-based personalization. US Patent App. 15/593,353. Aug. 2018.

[8] Palash Goyal, Sujit Rokka Chhetri, and Arquimedes Martinez Canedo. Capturing network dynamics using dynamic graph representation learning. US Patent App. 16/550,771. Mar. 2020.

[9] Yokila Arora et al. Personalized item recommendations through large-scale deep-embedding architecture. US Patent App. 16/777,555. Aug. 2021.

[10] Jurij Leskovec et al. Efficient convolutional network for recommender systems. US Patent App. 16/101,184. Sept.2019.

[11] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Communications of the ACM* 60, no. 6 (2017): 84-90.

[12] Coskun, Mustafa, Ananth Grama, and Mehmet Koyuturk. "Efficient processing of network proximity queries via chebyshev acceleration." In *Proceedings of the 22nd ACM SIGKDD International conference on knowledge discovery and data mining*, pp. 1515-1524. 2016.