Supplemental Material of EventThread2

I. Details of Sequence Segmentation Algorithm

We clarify the details of sequence segmentation, which is the final step in the progression analysis process. In this step, the mean-sequence S is split into segments to identify latent stages to further derive the segmentation results of each individual sequences. To this end, we employ a version of the content vector segmentation algorithm [1] (originally introduced to segment document into sections) and adapted it to perform the task of event sequence segmentation.

The goal of the algorithm is to maximize the sum of the coherence of each segment k denoted as $\phi(k)$. Conceptually, it aims to determine a segmentation which keeps events that are expected to co-occur within the same stage, while separating events not expected to be in the same context. Formally, $\phi(k)$ is given by:

$$\phi(k) = \sum_{i \in p_k} \bar{v}_i^k \cdot c_k^*$$

where $\bar{v}_i^k$ is the vector representation of the i-th event in k-th segment of the aggregate sequence(i.e., p_k). c_k^* is the estimated content vector that captures the overall distribution of the underlying events inside the k-th latent stage within the embedded vector space, which is calculated based on the maximum likelihood:

$$c_k^* = \arg\max_c \log \sum_i P(c_k|\bar{v}_i^k)$$

where $P(c_k|\bar{v}_i^k)$ estimates the content vector c_k given the observation of all event vectors $\bar{v}_i^k$ in the k-th segment.

A greedy algorithm has been used to approximately, but efficiently, solve the above optimization problem. In each iteration, it splits S or a subsequence of S into two parts at the place which maximizes the sum of $\phi(k)$ over all k segments. The algorithm stops automatically when the best available split results in a total score increase that is smaller than a threshold. We refer to the final number of segments after the greedy algorithm completes as K, and the time complexity is $O(n^2K)$. Finally, we unpack S into individual sequences inheriting the stage information from S. Note that if an event in the individual sequence is aligned with multiple elements on the mean sequence, it will follow the element with highest similarity and be assigned to the corresponding stage.

II. Comparative Analysis of EventThread2 versus EventThread

To further validate the stage analysis results, we compared the stages produced by ET^{2}with the results produced by a competing stage analysis method, the tensor-based algorithm proposed in EventThread [2]. To allow a clear comparison, we used both algorithms to analyze the same academic career dataset described in the academic career path case.

As described in the case study, ET^{2}extracted 13 stages from this dataset. Applying the tensor-based model for the *thread view* to these results, the system identified three threads as shown in Fig. 1(1). Each of these threads signifies a career path of a group of scholars, and nodes on each thread segment indicate events with highest occurrence probability in the corresponding stage. The competing algorithm, meanwhile, uses fixed duration segments and does not support the time warping alignment process used in ET^2. As a result, we had to pre-define the number and length of stages. To allow direct comparison, we configured the system with 13 fixed width stages, each of 1.77 years so that in total the stages spanned the full 23 years represented in the dataset. The resulting threads are shown in Fig. 1(2).

When the stages were segmented through the ET^2 progression analysis algorithm proposed in this paper, we found a number of representative events that allow for meaningful interpretations of the stages(as labeled in Fig. 1). Based on these interpretations, it was possible to identify three semantically relevant high-level phases.

In contrast, when the stages were defined according to fixed-width time intervals using the previously published method, stages were harder to interpret and, at times, misleading. For example, the first stage (defined by the older algorithm) combined both acquiring bachelors and masters degree. However, earning both degrees in the period of time represented by one stage (i.e., less than 2 years) is unusual. This occurred because scholars had different initial states when their sequences began. Moreover, we found titles changed irregularly from stage 4 to 9 (Fig. 1(d)) when using the older algorithm due to different progression rate of the scholars' career path, making it hard to reveal their true promotion trajectory. Both of these problems are overcome in part by the more flexible staging algorithm in ET^2, which can account for differences in the speed of progression between sequences.

In addition, the grouping of threads over time was more informative when using the ET^2 stage analysis results. For example, we found the first thread showed string differences in stage 6, where scholars were more likely to have various types of publications than in the other two threads (see Fig. 1(e)). The three threads were then grouped together from stage 7 to stage 10, reflecting similar behaviors during the periods of the scholars' first promotion. The third thread then split from the others at stage 11, where a group of scholars showed a higher probability of conference publications than the others after being promoted to full professor (Fig. 1(f)).

References

[1] A. A. Alemi and P. Ginsparg. Text segmentation based on semantic word embeddings. *arXiv:1503.05543*, 2015.

[2] S. Guo, K. Xu, R. Zhao, D. Gotz, H. Zha, and N. Cao. Eventthread: Visual summarization and stage analysis of event sequence data. *IEEE TVCG*, 24(1):56–65, Jan 2018.

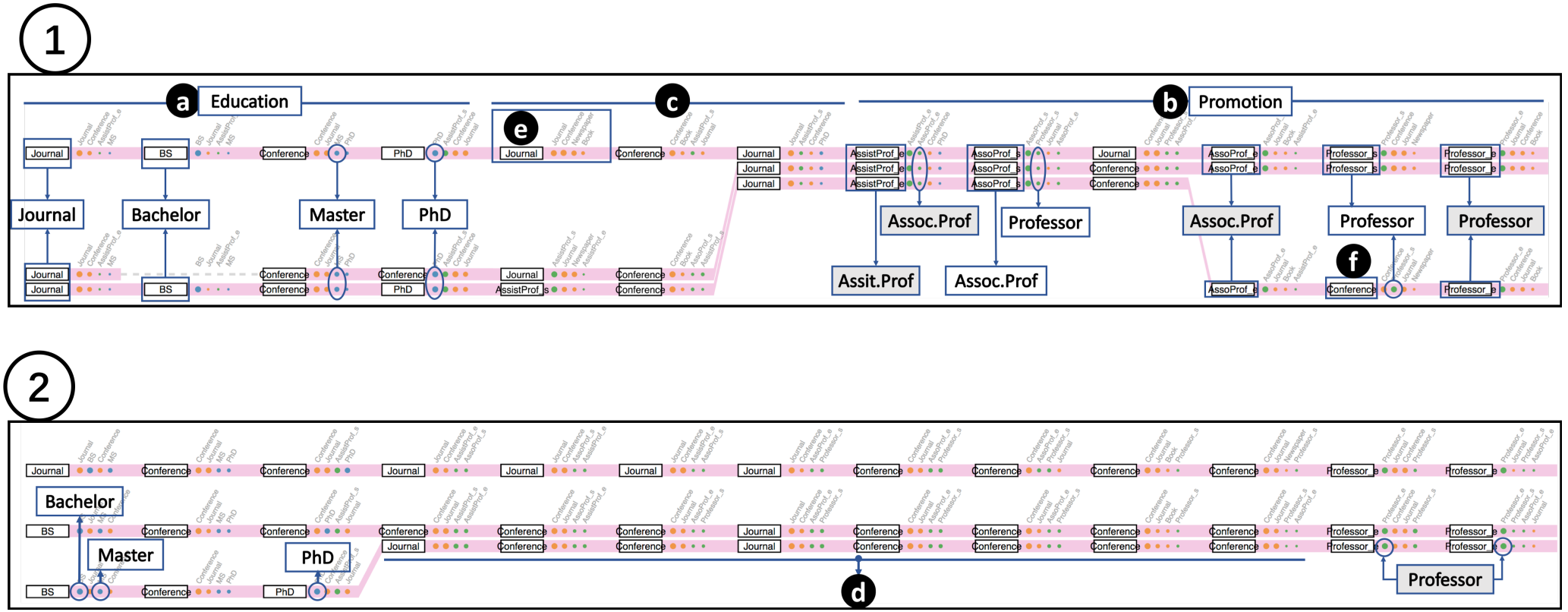

Fig. 1. The comparison of tensor analysis result under stages delivered by (1) the progression analysis result and (2) fixed-width time interval.