Temporal Summarization of Knowledge Evolutionary Trend

1. SUMMARIZATION FRAMEWORK

The major motivation of our work is to find a concise and intuitive summarization of a given research topic. More concretely, we want to select a set of representative terms that best describe the hot spots and major achievements along the development of the research topic, meanwhile, capture the temporal evolutionary pattern within the research area. We formulated the problem as a graph partitioning task. At a high level, the proposed summarization framework consists of three stages.

- **Document retrieval.** First, given a research topic q needs to be summarized, we find a documents collection $D = \{d_t\}^T$ that can represent the development of the research topic, where d_t denote the documents at time t.

- **Dynamic concept graph construction.** Second, we extract knowledge concept terms mentioned in each document and construct a graph G, where each term w_i at time slide t is associated with a node n_i^t. We explicitly model the semantic relationship and evolving patterns with the graph by associating feature scores on the nodes and edges.

- **Trend partitioning.** Third, we utilize a message passing algorithm on the constructed dynamic concept graph to select a set of terms based on their *authority*,

burstiness and network structures, and further partition the graph into evolutionary trends.

1.1 Document Retrieval

To summarize a given research topic t (i.e., user's query), we need to first map the topic to a set of documents. A straightforward idea is to use some traditional information retrieval technique to find the relevant documents. However, such an approach is not appropriate for the dynamic summarization task due to the following reasons. First, we want to summarize the development of the topic over time, if we use such an approach, we won't be able to trace how a knowledge concept evolve from and emerge into some other related concepts. Second, it has obvious bias that most retrieved documents will contain the queried words. Third, if a query contains a rare used term or a new term, it will face the problem of data sparseness.

To overcome the disadvantages above, we use a community-base document retrieval method. We first find a core research community (a group of experts) related to the topic, and then aggregate all the members' research work as the document collection to summarize. Such an approach is very intuitive since these researchers are leading the development the research area. To avoid some authoritative experts' work dominated in the documents and introduce potential bias, we normalize each member's contribution averagely.

1.2 Dynamic Concept Graph Construction

After the initial retrieval of related document collection, we extract knowledge concepts mentioned in each documents. In this work, we simply use wikipedia titles as a vocabulary to extract terms. With the extracted terms of each document over time, we construct a dynamic concept graph $G = \{V, E_r, E_e\}$ to model the semantic relationship and evolutionary pattern between terms uniformly. For a term w_i, we create a node n_i^t for each time slide t. There are two types of edges in the graph, E_r are representative edges denotes how well-suited for a term to act as a representative of another term at a given time. E_e are evolving edges connecting the same terms within two adjacent time slides indicating to what extent the meaning of the term has shifted.
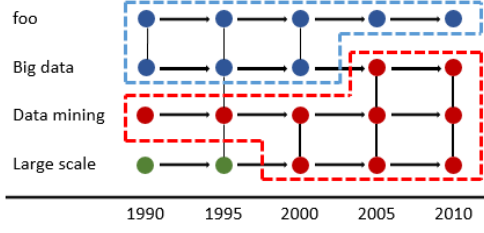
- **Representativeness:** To capture the semantic relationship between terms, we connect terms co-occurred in the documents at the same time slide with representative edges. Formally, the weight of the directed edge (n_i^t, n_j^t) is defined based on mutual information, indicates how appropriate it is to use term w_i to represent w_j at time t. We calculate mutual information between term w_i and term w_j at the given time slide t as:

$$\begin{aligned}
I(n_i^t, n_j^t) = {} & p(w_i \in d_t, w_j \in d_t) log \frac{p(w_i \in d_t, w_j \in d_t)}{p(w_i \in d_t)p(w_j \in d_t)} \\
& + p(w_i \in d_t, w_j \notin d_t) log \frac{p(w_i \in d_t, w_j \notin d_t)}{p(w_i \in d_t)p(w_j \notin d_t)} \\
& + p(w_i \notin d_t, w_j \in d_t) log \frac{p(w_i \notin d_t, w_j \in d_t)}{p(w_i \notin d_t)p(w_j \in d_t)} \\
& + p(w_i \notin d_t, w_j \notin d_t) log \frac{p(w_i \notin d_t, w_j \notin d_t)}{p(w_i \notin d_t)p(w_j \notin d_t)}
\end{aligned} \tag{1}$$

where d_t denotes all the retrieved documents at time t. The representativeness of term w_i to term w_j at time t is then defined as normalized mutual information

$$s(n_i^t, n_j^t) = \frac{I(n_i^t, n_j^t)}{I(n_i^t, I_i^t)}$$

- **Meaning shift:** Nodes corresponding to the term w_i within two adjacent time slides $-1t$ and t are connected by an evolving edge (n_i^{t-1}, n_i^t), indicates that the term is evolved from itself at the last time slide. The weight of the evolving edge indicates whether the meaning of the term is staying the same or shifted from time $t-1$ to t. Formally the weight is defined based on Jaccard coeffient.

$$s(n_i^{t-1}, n_i^t) = \frac{|NB(n_i^{t-1}) \cap NB(n_i^t)|}{|NB(n_i^{t-1}) \cup NB(n_i^t)|}$$

- **Burstiness:** We model burstiness by assuming the arrival of terms as an unknown binomial distribution, and use χ^2 tests to check for significant association between words and time periods. We calculate the contingency table as below and $\chi^2 = \frac{(ad-bc)^2 n}{(a+b)(c+d)(a+c)(b+d)}$.

-	W	$\overline{W}$
t	a	b
$< t$	c	d

We define burstiness of term w_i at time t as $b_i^t = \chi^2$ value.

- **Authority:** We simply defined the authority of a term at a given time by $u_i^t = df_t(w_i)/|d_t|$, where $df_t(w_i)$ indicates the document frequency of w_i at time t, and $|d_t|$ is the number of documents at time t.

1.3 Trend Partitioning

In order to give a concise and intuitive summarization, we need to partition the dynamic concept graph into evolutionary trends, and select a set of representative terms best describe the development of the research topic. Here, we use a massage passing algorithm to perform the selection and partition. The method is analogous to affinity propagation(AP) algorithm proposed in Frey et al. and Tang et al, the difference is that our method is modified to fit the dynamic setting. The AP algorithm performs clustering by identifying exemplars. Essentially it solves the following optimization problem

$$c^* = argmin(-\sum S(i, c_i))$$

where $C = (c_i)$ is the mapping between nodes and exemplars, $S(i, c_i)$ indicates the similarity between i and its exemplar. $S(i, i)$ is the penalty for i to being exemplar of itself.

In the algorithm, we introduce three sets of variables $\{r_{ij}^t\}$, $\{a_{ij}^t\}$ and $\{e_i^{t-1,t}\}$. Where r_{ij}^t indicates the how well-suited term w_j is to serve as an exemplar of term w_i (i.e. w_j covers the meaning of w_i) at time slide t. a_{ij} denotes the availabilities of term w_j to serve as an exemplar of w_i at time t. $\{e_i^{t-1,t}\}$ indicates the equivalence between the term w_i at two adjacent time slides which captures to what extent the meaning of term w_i has shifted from $t-1$ to t. All the r_{ij}^t, a_{ij}^t and $e_i^{t-1,t}$ are set to 0 initially, their values are updated iteratively with following rules until convergence.

$$r_{ij}^t = (1-\lambda)\rho_{ij}^t + \lambda r_{ij}^t \tag{2}$$
$$a_{ij}^t = (1-\lambda)\alpha_{ij}^t + \lambda a_{ij}^t \tag{3}$$

where λ is a damping factor, ρ_{ij}^t and α_{ij}^t are messages passing from n_i^t to n_j^t. The massage passing rules are defined as follows:

$$\rho_{ij}^t = s(n_i^t, n_j^t) - \max_{k \in NB(n_j^t)} \{s(n_i^t, n_k^t) + a_{ik}^t\} \tag{4}$$

$$\alpha_{ii}^t = \sum_{k \in NB(n_i^t)} \max\{0, r_{ij}^t\} \tag{5}$$

$$\alpha_{ij}^t = min\{0, r_{jj}^t + \sum_{k \in NB(n_j^t)} \max\{0, r_{ik}^t\}\} \tag{6}$$

where $NB(n_j^t)$ denotes the neighboring nodes of node n_j^t, $s(n_i^t, n_j^t)$ is w_i's representativeness of w_j at time t. $s(n_i^t, n_i^t)$ is set to $s(n_i^t, n_i^t) = \beta b_i^t + (1-\beta)u_i^t$ indicates that the nodes with high authority and burstiness are preferred to be choose as exemplars, where β is a parameter controlling the tradeoff between timeliness and importance.

2. REFERENCES

Input: dynamic concept graph $G = \{V, E_r, E_e\}$;
Output: a set of representative nodes V_r and the par;
Initialize all $r_{ij}^t \leftarrow \mathbf{0}$;
repeat
    foreach *edge (n_i^t)* **do**
        //Initialization;
        $L \leftarrow$ initialization list;
        Factor graph $FG \leftarrow BuildFactorGraph(L)$;
        // Learn the parameter θ for factor graph model;
        repeat
            foreach $v_i \in order$ **do**
                Update the messages of v_i by Eqs. **??** and
                ??;
            end
        until *(all messages μ do not change)*;
        foreach $\theta_i \in \theta$ **do**
            Calculate gradient ∇_i according to Eq. **??**;
            Update $\theta^{new} = \theta^{old} + \eta \cdot \nabla_i$;
        end
    end
until *converge*;

Algorithm 1: Message passing algorithm on dynamic concept graph.