Building Topic/Trend Detection System based on Slow Intelligence

Chia-Chun Shih
Institute for Information Industry
Taiwan
chiachun@iii.org.tw

Ting-Chun Peng
Institute for Information Industry
Taiwan
markpeng@iii.org.tw

***Abstract*—It becomes an interesting research topic to detect trend in the Internet era, where millions of data are posted online everyday. As social media, for example, blogs, forums, and micro-blogs, are prevailing, many offline events are discussed online. The discussion data, which reflects what people are interested in, is useful for detecting trend. This research proposes a design of online topic/trend detection system with the advantages of Slow Intelligence. Unlike traditional Topic Detection and Tracking (TDT) tasks, which source data from offline news articles, the proposed system attempts to collect and analyze huge amount of up-to-date data from many heterogeneous websites on Internet. The Internet data complicates the system in two aspects: 1) it needs careful resource allocation to collect huge amount of up-to-date data based on limited computing resources; 2) it needs mechanisms to automatically or semi-automatically adapt data processing algorithms to handle varieties of data. This research adopts Slow Intelligence, which provides a framework for systems with insufficient computing resources to gradually adapt to environments, to handle these complexities.**

I. Introduction

It becomes an interesting research topic to detect trend in the Internet era, where millions of data are posted online everyday. As social media, for example, blogs, forums, and micro-blogs, are prevailing, almost all offline events are discussed online. The discussion data, which reflects what people are interested in, is useful for detecting trend. For example, HP Labs have demonstrated that social media can be effective indicators for predicting movie revenues [2].

Information Retrieval researchers have been working on Topic Detecting and Tracking (TDT) tasks for decades [1]. Traditional TDT tasks, which source data from on a stream of offline news stories, have been studied for decades; however, the Internet brings new challenges on designing online trend detection systems. The Internet-based online topic/trend detection systems need to collect and process huge amount of up-to-date data from many heterogeneous websites, which bring complexities:

- It needs careful resource allocation to collect huge amount of up-to-date data based on limited computing resources.
- It needs mechanisms to automatically or semi-automatically adapt data processing algorithms to handle varieties of data.

This research adopts Slow Intelligence [6], which provides a framework for systems with insufficient computing resources to gradually adapt to environments, to handle these complexities. Slow Intelligence System (SIS) solves problems through iterated processes involving enumeration, propagation, adaptation, elimination, and concentration. Through the process, computing resources are gradually concentrated on prospect solutions.

In this paper, we propose a design of online topic/trend detection system with the advantages of Slow Intelligence. Four complexities of designing online topic/trend detection systems are identified, along with corresponding Slow Intelligence solutions. The remainder of this paper is organized as follows. Section II shows the design of topic/trend detection system without Slow Intelligence. Section III identifies four complexities of designing trend detection systems, along with corresponding Slow Intelligence solutions. Conclusions are finally drawn in Section IV.

II. Topic/Trend Detection System

The objective of the proposed online topic/trend detection system is to detect current hot topics and to predict future hot topics based on data collected from the Internet. Since it is unlikely to collect all data on the Internet, the system requires users to provide their information needs, including their concerned keywords and their concerned websites. Furthermore, because hot topics change quickly, the system requires periodical updates in hourly or daily intervals.

The system first collects latest data from Internet based on users' information needs by *Crawler & Extractor*, then adopt TDT techniques to discover current hot topics by *Topic Extractor*, and finally apply trend estimation algorithms [3] to predict hot topics by *Trend Detector*.

- ***Crawler & Extractor*** The responsibility of *Crawler* is to collect web pages from Internet. *Crawler* needs to be selective, that is, only collect web pages that satisfy predefined requirements. We plan to implement a focused crawler [4], which selectively collects web pages that are relevant to a pre-defined set of topics. The responsibility of *Extractor* is to extract information from web pages. Since a web page is mixed with information and noisy content (advertisements, navigation stuff, and so on), information extraction algorithms [5] are applied to extract desired information of web pages. The extraction algorithms rely on visual/textual/HTML-syntax features, such as CSS/tag attributes, visual alignments and layouts, for learning the template of web pages and further extracting desired information from identified informative blocks [5, 9, 12, 13]. Information extracted in a single web page is combined into *text documents*, and stored in *Web data DB*. Figure 1 illustrates the workflow of this component.

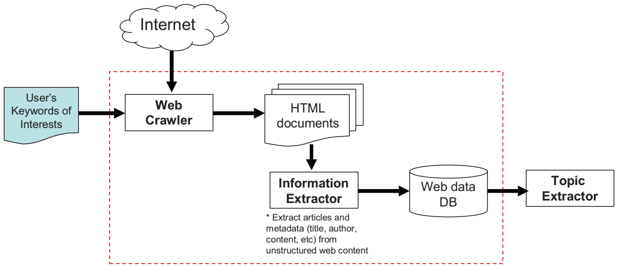
Internet
User's Keywords of Interests
Web Crawler
HTML documents
Information Extractor
Web data DB
Topic Extractor
* Extract articles and metadata (title, author, content, etc) from unstructured web content

Figure 1: Crawler & Extractor

- ***Topic Extractor*** The responsibility of *Topic Extractor* is to detect hot topics from a set of *text documents*. The process of topic detection can be divided into the following steps, which adopt some state-of-the-art techniques: 1) *topic word extraction*: TF-IDF [10] scheme is applied to measure the importance of terms in a given text document and generates top-N topic word candidates for each text document. 2) *topic word clustering*: single-pass clustering [1, 11], a popular topic detection approach, is adopted to cluster related documents into associated topic groups. The centroid topic word of cluster with highest weighting score is treated as the representative name of each generated cluster, which represents an extracted "topic". 3) *extract hot topics*: hot topics derive from hot events in a particular timeline [1, 8]. We apply the Aging Theory [7] to model the life cycle of an event in a topic, which assumes that an event has a life cycle with four stages: birth, growth, decay and death. To measure the hotness/popularity of each topic, we summarize the energy of related events by considering pervasiveness and topicality of topic words in the events [8] combined with the level of user participation using social data. Figure 2 illustrates the workflow of this component.

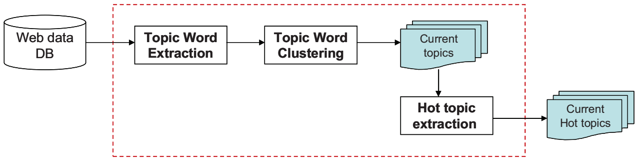

Figure 2: *Topic Extractor*

- ***Trend Detector*** The responsibility of *trend detector* is to detect trends (future hot topics) based on currently available data. Figure 3 illustrates the workflow of this component.

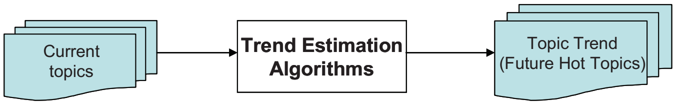

Figure 3: Trend Detector

III. Topic/Trend Detection System with Slow Intelligence.

In this section, we bring Slow Intelligence into the topic/trend detection system proposed in Section 2. The system needs to collect and process huge amount of up-to-date data from various heterogeneous websites on Internet, hence brings complexities:

1. The system needs computing resources to collect huge amount of data, while computing resources are usually limited. Users may input their interests of data to restrict the range of data collection. However, it is unlikely for users to update their interests consistently. The system need to be smart enough to know what users are currently concerned, and automatically adjust the range of data collection.

2. There are many kinds of computation methods for estimating trends [3]. Without extensive experiments, we have no idea how each computation methods performs, not even say the best parameter settings for each computation methods. However, we can utilize historical data, including predicted future hot topics and actual current hot topics, to find the best method/parameter combinations.

3. The system needs to revisit websites to collect up-to-date data in hourly or daily intervals to update data. As the number of interested websites increases, it needs sophisticatedly scheduling data collection tasks. The arrangement of schedule needs to consider many factors, for example, blocking policy of each website and data amount to be collected in each website. Because these factors may change over time, the scheduler needs to be smart enough to adapt to contextual change.

4. Any changes in web pages may disrupt *Extractors*. It is easy to fix the problem by human beings if there are only a few disrupted *Extractors*. However, it needs automatic or semi-automatic repair mechanism if the system is monitoring many websites. The repair mechanism needs to detect errors of *Extractors*, find alternatives, and choose the best solution from alternatives to fix the disrupted *Extractors*.

In this paper, we adopt Slow Intelligence [6] to handle these complexities. Slow Intelligence provides a framework for systems with insufficient computing resources to gradually adapt to environments, to handle these complexities. Slow Intelligence System solves problems through iterated processes involving enumeration, propagation, adaptation, elimination, and concentration. Through the process, computing resources are gradually concentrated on prospect solutions.

In the remaining of this section, we introduce four Slow Intelligence subsystems. Each subsystem accordingly targets a complexity discusses in this section.

A. Slow Intelligence Subsystem 1

Figure 4 illustrates the design of Slow Intelligence subsystem (the area with dotted border) to help restrict the range of data collection, following the structure of SIS -- enumeration, propagation, adaptation, elimination, and concentration.

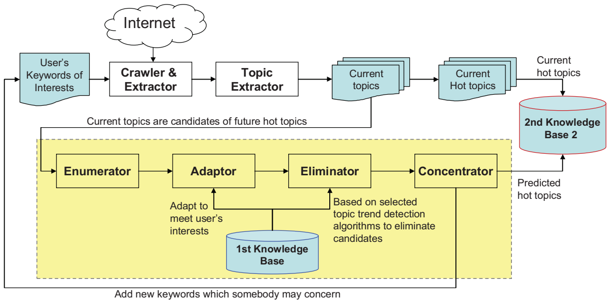
Internet
User's Keywords of Interests
Crawler & Extractor
Topic Extractor
Current topics
Current Hot topics
Current hot topics
2nd Knowledge Base 2
Current topics are candidates of future hot topics
Enumerator
Adaptor
Eliminator
Concentrator
Predicted hot topics
Adapt to meet user's interests
Based on selected topic trend detection algorithms to eliminate candidates
1st Knowledge Base
Add new keywords which somebody may concern

Figure 4: Slow Intelligence Subsystem 1

- **Enumerator**: The enumerator generates candidates of future hot topics. The enumerator takes current topics as candidates of future hot topics.
- **Adaptor**: The adaptor adapts candidates to meet users' interests listed in profile. For example, if a user is from a government department, the user is more likely to prefer economical topics rather than gossiping topics.
- **Eliminator**: The eliminator evaluates candidates based on the selected trend detection algorithms, and eliminates candidates with low probabilities to become hot topics. (We'll discuss how trend detection algorithms are selected and adapted in the next subsystem.)
- **Concentrator**: Only few candidates are selected as potential future hot topics. Keywords of these topics are fed into *Crawler*, so that *Crawler* can concentrate on collecting data related future hot topics that users may be interested in.

Two knowledge bases are involved in this subsystem:

- The first knowledge base is a typical SIS knowledge base, which provides essential domain knowledge for SIS. In this subsystem, essential knowledge includes users' profiles and available algorithms for detecting trend.
- The second knowledge base stores historical records, including actual hot topics and predicted hot topics. Historical records are helpful for selecting and adapting trend detection algorithms.

B. Slow Intelligence Subsystem 2

Figure 5 illustrates the design of Slow Intelligence subsystem (the area with dotted border) to help select and adapt trend detection algorithms for estimating trends.

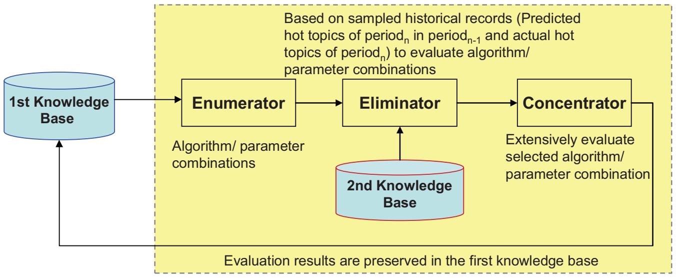
Based on sampled historical records (Predicted hot topics of period$_n$ in period$_{n-1}$ and actual hot topics of period$_n$) to evaluate algorithm/ parameter combinations
1st Knowledge Base
Enumerator
Eliminator
Concentrator
Algorithm/ parameter combinations
2nd Knowledge Base
Extensively evaluate selected algorithm/ parameter combination
Evaluation results are preserved in the first knowledge base

Figure 5: Slow Intelligence Subsystem 2

- **Enumerator**: The first knowledge base provides the enumerator knowledge about algorithms for estimating trends, including available algorithms and known best parameter settings for algorithms. The enumerator generates candidates of algorithm/parameter combinations for further evaluation.
- **Eliminator**: Historical records (including actual hot topics and predicted hot topics), which are stored in the second knowledge base, are useful for measuring performance of candidates. A sample of historical records is used to evaluate performance of candidates. Poorly-performed candidates are eliminated.
- **Concentrator**: Extensive evaluations are conducted on candidates, and the evaluation results are preserved in the first knowledge base.

C. Slow Intelligence Subsystem 3

In practice, the crawling schedule for each data source varies depending on characteristics of each website, for example, blocking policy of each website and data amount to be collected in each website.

In the cases of collecting data from websites with severe blocking policies, the crawling behavior and schedule of crawlers should be designed as similar as humans' access behavior to prevent being blocked. For instance, a target website may block accesses which send more than one requests in two second; therefore it would be smarter for crawlers to set a two-to-three-second interval between requests.

Furthermore, contradictory conditions exist. For instance, in the cases of collecting data from websites with frequent updates, *Crawler* needs to visit these websites more often than others, which may also cause *Crawler* being blocked.

Notably, characteristics of websites may change over time. However, it is impossible to manually configure efficient schedules for large-scale systems involving many targeting websites. A knowledge base with basic scheduling rules can aid the system to find acceptable scheduling plans in short-term cycles. Feedbacks from *Crawler* can further enrich the knowledge base for developing scheduling rules in long-term cycles.

Figure 6 illustrates the design of Slow Intelligence subsystem to help *Crawler* arrange schedules to efficiently collect up-to-date data while preventing being blocked by websites.

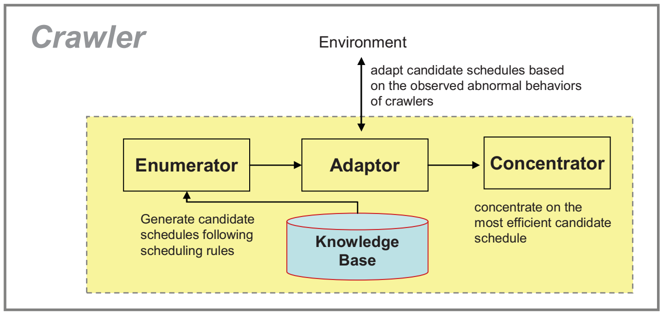
Crawler
Environment
adapt candidate schedules based on the observed abnormal behaviors of crawlers
Enumerator
Adaptor
Concentrator
Generate candidate schedules following scheduling rules
Knowledge Base
concentrate on the most efficient candidate schedule

Figure 6: Slow Intelligence Subsystem 3

- **Enumerator**: The enumerator generates candidate schedules of data collection following scheduling rules in the knowledge base. The scheduling rules include general rules, learned heuristic rules, and user-defined rules.
- **Adaptor**: The adaptor adapts candidate schedules based on the observed abnormal behaviors of crawlers, for example, being blocked by websites or unexpected delay.
- **Concentrator**: The concentrator concentrates on the most efficient candidate schedule. The efficiency of candidate schedule is evaluated based on the estimated time to complete the task.

D. Slow Intelligence Subsystem 4

Extractor applies extraction algorithms, which use parameterized features to learn the template of web pages, to extract desired information. Once web pages change template, adequate features and optimal parameters have to be investigated and revised.

Figure 7 illustrates the design of Slow Intelligence subsystem to help *Extractor* adapt to changes in web pages.

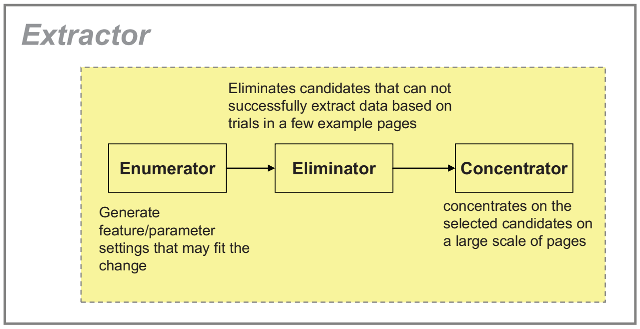
Extractor
Eliminates candidates that can not successfully extract data based on trials in a few example pages
Enumerator
Eliminator
Concentrator
Generate feature/parameter settings that may fit the change
concentrates on the selected candidates on a large scale of pages

Figure 7: Slow Intelligence Subsystem 4

- **Enumerator**: The enumerator generates feature/parameter settings that may fit the change.
- **Eliminator**: The eliminator eliminates candidates that can not successfully extract data based on trials in a few example pages.
- **Concentrator:** The concentrator concentrates on the selected candidates on a large scale of pages.

IV. CONCLUSIONS

Although Internet provides abundant data for topic/trend detection, it incurs challenges on designing online topic/trend detection system. An online trend detection system requires careful resource allocation and automatic algorithm adaptation to process huge size of heterogeneous data. This research adopts Slow Intelligence, which provides a framework for systems with insufficient computing resources to gradually adapt to environments, to response the challenges. Four Slow Intelligence subsystems are proposed, and each subsystem targets a challenge in designing online topic/trend detection systems. In the future, we plan to implement the topic/trend detection system to empirically verify the four proposed Slow Intelligence subsystems.

ACKNOWLEDGMENT

This study is conducted under the "Digital Convergence Service Open Platform" project of the Institute for Information Industry which is subsidized by the Ministry of Economy Affairs of the Republic of China.

REFERENCES

[1] Allan, J., Carbonell, J., Doddington, G., Yamron, J., and Yang, Y., "Topic detection and tracking pilot study: Final report," *Proceedings of the DARPA broadcast news transcription and understanding workshop*, 1998.

[2] Asur, S. and Huberman, B., "Predicting the Future With Social Media," *Arxiv preprint arXiv:1003.5699,* 2010.

[3] Bianchi, M., Boyle, M., and Hollingsworth, D., "A comparison of methods for trend estimation," *Applied Economics Letters,* vol. 6, pp. 103-109, 1999.

[4] Chakrabarti, S., Van den Berg, M., and Dom, B., "Focused crawling: a new approach to topic-specific Web resource discovery," *Computer Networks,* vol. 31, pp. 1623-1640, 1999.

[5] Chang, C. H., Kayed, M., Girgis, M. R., and Shaalan, K. F., "A survey of web information extraction systems," *IEEE transactions on knowledge and data engineering,* vol. 18, pp. 1411-1428, 2006.

[6] Chang, S. K., "A General Framework for Slow Intelligence Systems," *International Journal of Software Engineering and Knowledge Engineering,* vol. 20, pp. 1-15, 2010.

[7] Chen, C. C., Chen, Y. T., and Chen, M. C., "An aging theory for event life-cycle modeling," *IEEE Transactions on Systems, Man and Cybernetics, Part A,* vol. 37, pp. 237-248, 2007.

[8] Chen, K. Y., Luesukprasert, L., and Chou, S., "Hot topic extraction based on timeline analysis and multidimensional sentence modeling," *IEEE transactions on knowledge and data engineering,* vol. 19, pp. 1016-1025, 2007.

[9] Deng, C. A. I., Shipeng, Y. U., Ji-Rong, W., and Wei-Ying, M., "VIPS: a vision-based page segmentation algorithm."

[10] Salton, G. and Yang, C. S., "On the specification of term values in automatic indexing," *Journal of documentation,* vol. 29, pp. 351-372, 1973.

[11] Smith, D. A., "Detecting and browsing events in unstructured text," *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, 2002, pp. 73-80.

[12] Song, R., Liu, H., Wen, J. R., and Ma, W. Y., "Learning block importance models for web pages," *Proceedings of the 13th international conference on World Wide Web*, 2004, p. 211.

[13] Zhai, Y. and Liu, B., "Web data extraction based on partial tree alignment," *Proceedings of the 14th international conference on World Wide Web*, 2005, p. 85.