TP-Detector: Detecting Turning Points in the Engineering Process of Large-scale Projects

Qi Wu[1], Wenhan Chao[1], Xian Zhou[2] and Zhunchen Luo[2*]
[1] School of Computer Science and Engineering, Beihang University
{wuqismile, chaowenhan}@buaa.edu.cn
[2] Information Research Center of Military Science, PLA Academy of Military Science
zhouxian@sjtu.edu.cn, zhunchenluo@gmail.com

Abstract

This paper introduces a novel task of detecting turning points in the engineering process of large-scale projects, wherein the turning points signify significant transitions occurring between phases. Given the complexities involving diverse critical events and limited comprehension in individual news reports, we approach the problem by treating the sequence of related news streams as a window with multiple instances. To capture the evolution of changes effectively, we adopt a deep Multiple Instance Learning (MIL) framework and employ the multiple instance ranking loss to discern the transition patterns exhibited in the turning point window. To facilitate comprehensive evaluation of the task, we curate a dataset comprising 80 large-scale projects. Extensive experiments consistently demonstrate the effectiveness of our proposed approach on the constructed dataset compared to baseline methods. We deployed the proposed model [1] and provided a demonstration video[2] to illustrate its functionality. The code and dataset are available on GitHub[3].

1 Introduction

Large-scale projects are intricate and extensive endeavors requiring substantial resources, effort, and coordination to achieve specific objectives, often involving multiple stakeholders and phases with a significant impact on organizations, communities, and society. They encompass diverse fields, such as aerospace engineering, water resources facilities, and transportation infrastructure, and hold paramount importance in driving economic growth, enhancing infrastructure, and addressing societal needs, while also fostering innovation and sustainability. Typically, their successful execution requires careful planning, collaboration, and a long-term vision to maximize their positive impact on communities and society at large.

In order to tackle the intricacies and difficulties of carrying out large-scale projects, the engineering process employs a systematic and structured approach to design, plan, and execute complex endeavors efficiently and effectively (Martin, 2000; Gilb, 2005). The life cycle of large-scale projects consists of several phases that cover the entire journey from initiation to closure (Beitz et al., 1996; Bennett, 2003) and each phase dedicats to accomplishing different objectives. Figure 1 depicts the seven phases involved in NASA's Insight mission engineering process, wherein each phase comprises a series of subtasks or key events represented by the gray diamonds, which can occur simultaneously or have interdependencies.

During the engineering process of large-scale projects, there are significant moments that deserve attention, such as when the project reaches a new milestone, e.g., transitioning from the conceptual phase to on-ground implementation on the ground (as shown in the green diamond h of Figure 1). The moments or events which bring about critical changes in direction, course, or outcome are referred to *turning points* in this paper, signifying transitions occurring between adjacent phases. The identification of these turning points provides stakeholders, government agencies, and the general public with valuable insights, empowering them to navigate challenges, capitalize on opportunities, and effectively adjust their strategies in response to changing circumstances. For example, an analyst can assess Boeing's development status and offer design suggestions for products at Airbus.

However, detecting turning points in the engineering processes of large-scale projects, particularly ongoing projects, is a non-trivial undertaking. First, limited public data sharing and potential lack

*Corresponding Author
[1] http://43.138.60.114:7080/
[2] https://youtu.be/FH3av84I-Kg
[3] https://github.com/smile577/tpd

Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 177–185
December 6-10, 2023

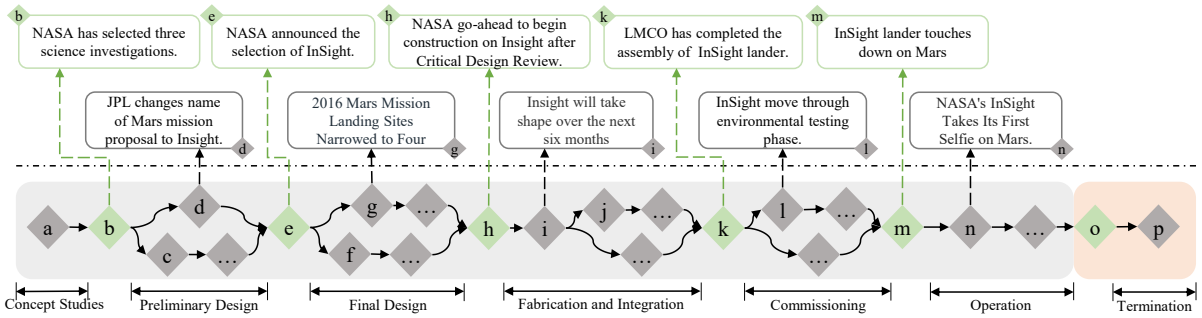

Figure 1: Illustration of the engineering process of NASA's Insight mission. The gray diamonds represent subtasks or events within a phase, whereas the green diamonds represent turning points that occur during the transition between phases.

of standardized protocols may obstruct access for those not directly engaged. Second, listing all possible turning point events for various projects is challenging. Additionally, the turning points may encompass multiple critical events, and the same event could be part of a phase or being in a transition under different circumstances. Nevertheless, publicly available news serves as a conveniently obtainable and reasonably trustworthy data source. In light of this, we've discovered that discerning the occurrence of a turning point within a series of interconnected news articles can be accomplished by comparing and analyzing the significant events. To accomplish this, we view a sequence of related news streams as a window and resort to the Multiple Instance Learning (MIL) (Andrews et al., 2002; Sultani et al., 2018) framework to obtain window-level labels, indicating the presence or absence of a turning point within the window. In order to avoid biased understanding of individual reports, the instances of a window comprise several news to form a relatively comprehensive grasp of the events that occurred.

Following the insights, this paper delves into the intricate task of detecting turning points in the engineering process of large-scale projects by utilizing multiple instance learning techniques. To facilitate this study, we collect 80 large-scale projects to construct the turning point detection dataset. We structure the sequence of related news streams into a window with multiple instances through a deep MIL framework and identify turning points via convolutional transformer encoder (Dosovitskiy et al., 2020; Li et al., 2022). We then employ the multiple instance ranking loss to push the positive instances and negative instances far apart in terms of the extent of change or shift that occurs between phases. Additionally, we develop a website for detection and visualization with our deployed model, providing users with transition timeline and news related to single or multiple large-scale projects.

2 Related Work

Turning Point Detection In the large-scale projects, specific methods for detecting turning points have not yet been established. However, similar concepts exist in other domains such as time series change-point detection, as well as turning point detection in video and text data. In the field of time series data change-point detection(Aminikhanghahi and Cook, 2017; Truong et al., 2020), the focus is primarily on identifying fluctuations in time series data, such as those in financial stock markets(Grillenzoni, 2012; Tang et al., 2019) and weather temperatures(Banesh et al., 2019). These methods commonly detect changes based on fluctuations along specific dimensions in low-dimensional spaces. However, these techniques may not be directly applicable to high-dimensional data such as video and text. In case of video sequences, Chang et al. (Liu et al., 2019a) introduced the use of the Two Clocks theory (Lotker, 2016) to detect a key event in narrative works, aiming to identify multiple turning points in cartoon movie stories. In the text sequences, Papalampidi et al. (Papalampidi et al., 2019) proposed the task of identifying turning points in movie screenplays to analyze narrative structures. They defined turning points in screenplays and developed an end-to-end neural network model for recognition.

Multiple Instance Learning Multiple Instance Learning (MIL) is a form of weakly supervised learning where training instances are organized into sets, called bags(Maron and Lozano-Pérez, 1997; Herrera et al., 2016). Only the label for the bag is provided. Due to this characteristic, MIL has found extensive applications in domains with large amounts of weakly labeled data(Quellec et al.,

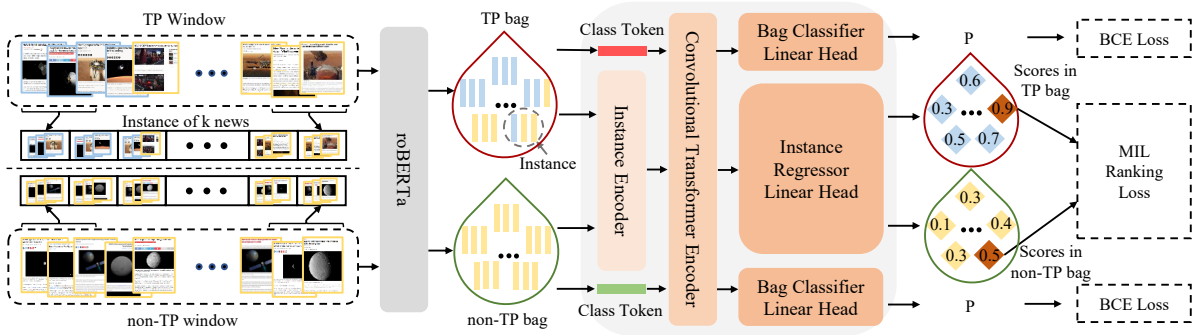

Figure 2: An overview of multiple instance learning framework for turning point detection of large-scale projects.

2017; Tian et al., 2020), such as video classification (Ding et al., 2013; Sultani et al., 2018; Li et al., 2022), image classification(Sudharshan et al., 2019; Li et al., 2021; Yang et al., 2023), and text classification(He and Wang, 2009; Liu et al., 2018), among others. In the context of weakly supervised video classification, typically only video-level category labels are provided. Sultani et al. (Sultani et al., 2018) proposed the MIL ranking model. They utilized the MIL ranking model to compute the highest-scoring instance within the bag for video classification. Li et al. (Li et al., 2022) introduced a MIL ranking model based on Transformer for video classification. In the realm of text classification, He et al.(He and Wang, 2009) proposed a KNN algorithm-based multi-instance Chinese text classifier. Liu et al.(Liu et al., 2018) introduced Selective Multi-Instance Transfer Learning to address the issue of knowledge-safe transfer in multi-instance learning for text classification.

3 TP-Detector

3.1 Task Definition

Our task is defined as follows: We want to detect TP in an input news stream, where TP is typically associated with changes in sequences. To detect changes of sequences within a specific range, we partition the news stream into windows and check for the presence of TP within these windows. Given a window W on the news stream, window W contains multiple news texts, denoted as $W = \{x_1, x_2, ..., x_n\}$. The window W has two classes: TP window W_{tp} and non-TP window W_{ntp}. If there is a TP within the window, meaning that events within the window across different phases, it is classified as a TP window; otherwise, it's a non-TP window. The outputs include Y and Y_{tp}. Y represents whether window W is a TP window. If $Y = 1$, it's a TP window; if $Y = 0$, it means it's not. Y_{tp} represents the evidence within W that is most likely to be a TP, and it's only output when $Y = 1$.

3.2 Model Overview

We propose a multi-instance learning model, as shown in Figure 2. The model takes two windows from one news stream as input, namely TP window and non-TP window. It starts by employing a pre-trained language model (Liu et al., 2019b) to represent the textual features. Next, the features of continuous k news within each window are organized as instances. These instances are then processed by an Instance Encoder, which consists of multiple layers of 1D convolutions, to extract their feature representations. At this point, the window is treated as a bag containing multiple instances. Subsequently, a Transformer encoder with convolutional layers (Dosovitskiy et al., 2020; Li et al., 2022) is employed to attend to the feature representations of both the bag and the instances within the bag. This process helps the model improve its understanding of the features within the bag and its constituent instances. Finally, two linear heads assign scores to the bags and instances. The decision of whether the input window is a TP window is made based on the scores of the bags and their instances.

3.3 Multiple Instance Learning

Due to the diversity of turning point events in large-scale project engineering processes, it is impractical to exhaustively enumerate all possible turning points. However, we have found that by comparing and analyzing known instances of turning points, it is possible to learn the general patterns of turning points and detect unknown turning point events. Turning point events are typically sparse

in the news streams, with the majority of information being non-TP related. To effectively discover critical turning point events, we introduce a multi-instance learning framework (Andrews et al., 2002; Sultani et al., 2018). Multi-instance learning is a weakly supervised method where the data unit is a bag. Taking binary classification as an example, a bag contains multiple instances. If at least one instance in a bag is a turning point instance, the bag is considered a turning point bag. Otherwise, it is a non-TP bag. This approach enables us to detect turning points even when they are sparsely distributed within the news streams.

We accept two windows, W_{tp} and W_{ntp}, as inputs, each containing multiple news articles. We use the pre-trained language model RoBERTa(Liu et al., 2019b) to extract feature representations for each news within the windows. The output of the feature representations for the TP window are denoted as

$$W_e = \{x_i^e | x_i^e = f_e(x_i), x_i \in W\}$$

where f_e is RoBERTa, and W_e is the output feature representations set.

We combine the continuous k feature representations from the above output into one instance. To obtain the feature representation of each instance, we use Instance Encoder to fuse the above feature representations. The Instance Encoder consists of multiple layers of 1D Convolutional and multiple layers of max-pooling. The formulation can be expressed as follows:

$$I_i = f_{ie}(x_i^e, ..., x_{i+k-1}^e)$$

where f_{ie} is the Instance Encoder and I_i represents the feature representation of an instance.

To ensure that critical transition information is not missed due to data segmentation and to increase the information density within each bag based on the sparse nature of news stream data, we form instances using an overlapping approach. Specifically, we iterate through the window with a step of 1 news article to generate instances. Thus, we represent the input W as one bag:

$$B = \{I_i | i \in \{1, 2, ..., n - k + 1\}\}$$

where B represents the multi-instance bag.

3.4 Turning Point Detection via Convolutional Transformer Encoder

To enhance the understanding of feature representations for instances within the bag, we utilized the Convolutional Transformer Encoder (CTE) (Dosovitskiy et al., 2020; Li et al., 2022). The 1D convolution within the CTE enables information interaction among instances within the bag. Then, we employ the self-attention mechanism in the Transformer to enhance the understanding of feature representations for instances. Finally, two linear layers are used to classify the enhanced feature representations of both the bag and its internal instances.

We input the bag's class information and the feature representations of bag instances into the CTE , which can be represented as follows:

$$B_{cte} = CTE(ClassToken||B)$$

where $||$ represents concatenation.

The output of the CTE is then separately fed into two Linear Heads: the Instance Regressor Linear Head (IRLH) and the Bag Classifier Linear Head (BCLH) (Li et al., 2022) as follows.

$$B_{ir} = IRLH(B_{cte}[1, 2, ..., n - k + 1])$$

$$p = BCLH(B_{cte}[0])$$

where $B_{cte}[i]$ represents the i-th element of B_{cte}, B_{ir} contains the scores of all instances in the bag, and p is the bag's class prediction value.

During the prediction phase, to reduce the fluctuation of instance prediction scores, we use the bag's class prediction score for calibration: $B_p = B_{ir} * p$, where B_p represents the final prediction scores of all instances in the bag. We select the maximum value in B_p as the prediction value for the bag. If the prediction value is greater than the threshold, the bag is classified as a TP bag; otherwise, it is classified as a non-TP bag.

3.5 Optimization via MIL Ranking Loss

To guide the multi-instance learning and achieve end-to-end TP detection, we introduce the MIL Ranking Loss (Sultani et al., 2018) to optimize the learning process of our model. This loss function helps us effectively train the model to distinguish between TP and non-TP instances within the bags.

To identify the category of a bag, we compare the input bags in such a way that the highest predicted score of instances in TP bags is greater than all instances in non-TP bags. The instance with the highest score in the non-TP bag is most similar to the TP instances in TP bags, which can lead to false positives. To distinguish between true positives and false positives, we aim to maximize the separation

between them. Therefore, the MIL Ranking Loss function can be formulated as:

$$l(B_{tp}, B_{ntp}) = max(0, 1 - \max_{I_i \in B_{tp}} f(I_i) + \max_{I_i \in B_{ntp}} f(I_i))$$

Where B_{tp} and B_{ntp} represent the TP bags and non-TP bags, respectively, and $f(I_i)$ denotes the final prediction score of instance I_i.

In general, TP occurs within a relatively short period; therefore, most of the instances in the window are non-TP instances. In other words, in a TP bag, only a few instance scores are close to 1, while the rest of the scores are close to 0. To address this sparsity, we introduce a sparsity constraint. Additionally, we include a binary cross-entropy loss for bag classification. Therefore, our ranking loss formula is as follows:

$$l(B_{tp}, B_{ntp}) = max(0, 1 - \max_{I_i \in B_{tp}} f(I_i) + \max_{I_i \in B_{ntp}} f(I_i)) + \lambda_1 \sum_{I_i \in B_{tp}} f(I_i) + \lambda_2 BCE(p, Y)$$

4 Experiments

4.1 Dataset

Due to the unavailability of a dataset for our task, we collected dataset to train and evaluate our model. In order to make the development status of large-scale projects accessible to everyone interested, we aimed to collect data through publicly available channels rather than relying on proprietary sources. Given the accessibility, real-time nature, and authenticity of news, we chose to gather data from publicly available news sources as the primary data source for our dataset. Our data collection process involved several steps. Firstly, we gathered available projects as candidates. Next, we utilized the Google Search API to search for relevant news, resulting in a news stream. To facilitate the labeling process, we provided a clear definition of the phases in large-scale projects, as outlined in Table 3. This served as a reference point for annotators to label each window of news articles according to whether it represented a turning point or not.

Due to the diversity of large-scale technology projects, in order to make the dataset more clear and standardized, we categorize the projects in the dataset into three main types: Deep Space Exploration Projects, Large Ground Infrastructure Projects, and Aeronautic and Marine Engineering Projects. Deep Space Exploration (DSE) Projects: These projects are aimed at conducting deep space scientific research and exploring new technologies. They include celestial exploration and space astronomy research, among others. Large Ground Infrastructure (LGI) Projects: These projects involve the development, maintenance, or improvement of critical large-scale ground infrastructure. This category includes projects related to transportation networks, power systems, nuclear test facilities, network and computing facilities, and ground-based observatories, among others. Aeronautic and Marine Engineering (AME) Projects: These projects are focused on the design, development, and implementation of aeronautic and marine engineering solutions. This category includes projects related to aircraft and ship design, manufacturing, and usage, among others.

The partial statistics of our dataset are presented in Table 4. In total, we collected data from 80 different projects. On average, there was a time span of 53 days between successive news articles within a news stream, indicating that the data within news streams was sparse. To address this issue and to ensure that turning point windows were not missed, we adopted an overlapping window segmentation strategy. For more detailed information about the dataset, please refer to Appendix B.

We allocated a total of 64 projects to the training set, 8 to the development set, and another 8 to the test set. Within these sets, there were 654, 83, and 85 turning point windows, and 3564, 460, and 461 non-turning point windows, respectively.

4.2 Baseline

We employed DSVDD(Ruff et al., 2018) as our baseline model and RoBERTa(Liu et al., 2019b) as our strong baseline model.

To validate the effectiveness of each proposed component, we excluded them one by one from our model: (1) w/o InCoder: Excluding the Instance Encoder module. (2) w/o CTE: Excluding the Transformers module with convolution.

Experiment Settings: We followed the hyperparameters of RoBERTa-Base (125M parameters) and initialized our model using public pre-trained checkpoints. We set the learning rate to 8e-4 and the batch size to 8. For the Baseline model DSVDD, we set the learning rate to 8e-4 and the batch size to 64. For the strong Baseline model RoBERTa, we set the learning rate to 8e-4 and the batch size to 16.

Table 1: Experimental Results on the constructed dataset.

Model	Overall Results			DSE Projects			LGI Projects			AME Projects		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Bin. Classifier	0.18	0.05	0.08	0.28	0.05	0.09	0.40	0.18	0.25	0.18	0.05	0.07
DSVDD	0.24	**0.80**	0.36	0.34	0.79	0.48	0.22	**0.82**	0.35	0.13	0.60	0.21
RoBERTa	0.60	0.67	0.63	0.58	0.67	0.63	0.43	0.72	0.54	0.32	0.62	0.42
TP-Detector	**0.70**	0.77	**0.74**	**0.80**	0.75	**0.77**	**0.61**	0.68	**0.64**	**0.34**	**0.83**	**0.48**
w/o InCoder	0.67	0.75	0.71	0.60	**0.83**	0.69	0.60	0.67	0.60	0.33	0.79	0.47
w/o CTE	0.69	0.76	0.72	0.74	0.69	0.71	0.51	0.78	0.61	0.26	0.81	0.39

4.3 Evaluation

For the TP detection task, our goal is to minimize the possibility of overlooking TP windows and accept a certain level of false positives if necessary. Therefore, we aim to achieve a higher recall rate while maintaining the precision of the prediction results. Based on this objective, we observed significant improvements in our model compared to the baseline model DSVDD and the strong Baseline model RoBERTa, which demonstrates the effectiveness of our model in identifying TP windows. For Deep Space Exploration (DSE) projects, our prediction results perform well. There are two main reasons for this: (1) DSE projects are space exploration initiatives managed by organizations like NASA and ESA. These agencies adhere to strict standard procedures and rigorous review processes for their projects. Therefore, these projects generally follow well-defined engineering processes, making the detection of turning points relatively straightforward. (2) There is a relatively large number of DSE projects, and information about them is regularly and comprehensively released by these organizations. Media outlets also tend to focus more on these projects, resulting in a wealth of available information. The abundance of information allows the model to extract more features, which is advantageous for successful turning point detection. For Large Ground Infrastructure (LGI) and Aeronautic and Marine Engineering (AME) projects, our prediction results perform less well. There are several reasons for this: (1) LGI and AME projects are relatively underrepresented in the dataset due to their smaller numbers, which makes it challenging for the model to learn robust patterns specific to these categories. (2) LGI projects generally receive less attention from media organizations compared to DSE projects. Consequently, there is less news coverage for these types of projects, leading to a limited amount of available information. (3) In AME projects, there are multiple projects with multiple end targets. For instance, in aeronautic engineering projects, the goal is to deliver multiple identical aircraft. Because each aircraft follows the same fixed process from manufacturing to delivery, we consider the first moment of this process as the turning point. However, in such cases, it is challenging for the model to accurately distinguish between these instances.

For the ablation study, we can draw the following conclusions: (1) Instance Encoder: This serves as an encoder for the instances within a bag, with its primary function being to fuse feature representations of multiple news articles within the instances and generate a feature representation for each instance. (2) CTE (Convolutional Transformer Encoder): CTE is a Transformer encoder with convolutional components. Its role is to focus on and enhance the feature representations of both bags and the instances within them.

5 Demonstration

We provide our services in a web-based. We deployed the proposed model [4] and provided a demonstration video[5] to illustrate its functionality.

We offer two detection options: Single Project and Multiple Project. In the "Single Project" mode, the system performs TP detection on an individual project. The input entails a JSON file containing the news stream, while the output encompasses TP detection results image and the highest-scoring instance within TP windows. The explanation for a TP window is provided by showcasing the highest-scoring instance. On the other hand, the "Multiple Projects" mode enables simultaneous TP detection across multiple projects. Similar to the single project mode, the input comprises JSON files containing news streams for multiple projects, and

[4] http://43.138.60.114:7080/

[5] https://youtu.be/FH3av84I-Kg

Figure 3: A result example from our demo website.

Table 2: The instance with highest score in the third TP window of the Kepler space telescope project.

Date	News content
Oct. 30th, 2018	NASA's Kepler Space Telescope mission has officially ended. All good things must come to an end, on Earth and even in space. NASA announced on Tuesday that the Kepler mission - which has transformed how we understand planets outside of our solar system - is officially over. According to the space agency, Kepler has run out of fuel in space, ending its 9.5-year planet hunting mission.
Oct. 31st, 2018	NASA retires planet-hunting Kepler space telescope. NASA on Tuesday announced the demise of its elite planet-hunting telescope just a few months shy of its 10th anniversary. The Kepler space telescope that found thousands of planets beyond our solar system and boosted the search for worlds that might support life has run out of fuel.

the output is similar to the "Single Project" mode. Figure 3 illustrates the results of Multiple Project detection.

Table 2 presents evidence for the correctness of a detection result. It displays the highest-scored Instance in the third TP window of the prediction results for the Kepler Space Telescope project. This TP occurs between the Operation phase and the Termination phase. Each row in the table represents a news. The first news describes NASA's official retirement of the Kepler space telescope. The second similarly discusses a similar topic. In this instance, the explanation centers around the decision to retire Kepler due to fuel exhaustion, which cannot be replenished. This crucial event has caused the transition of Kepler from an operational state to a terminated state. This crucial event serves as the turning point that we are interested in identifying.

6 Conclusion

To investigate phase transitions in engineering processes, we propose the Turning Point Detection task on large-scale projects. For this task, we introduce a deep multi-instance learning model. This model initially performs feature extraction on input windows using a pre-trained language model. Subsequently, it employs an Instance Encoder to capture instance-level features. Following this, a Convolutional Transformers Encoder is employed to detect phase transition features. Ultimately, a linear head is employed to provide prediction outcomes. We collected a new dataset specifically for this task. Extensive experiments demonstrated that our model outperforms strong baselines. We have deployed our model online to assist in detecting phase transitions in news streams. We hope that our work will contribute to further research in this emerging task and benefit relevant stakeholders. The dataset we have constructed currently only includes publicly available news reports. In the future, we aim to incorporate officially released information into the dataset as an essential supplement to enhance our understanding of the projects with more comprehensiveness.

Acknowledgements

This work was supported by National Natural Science Foundation of China (No. 61976221).

References

nsf21107 Research Infrastructure Guide (RIG) (December 2021) | NSF - National Science Foundation — nsf.gov. https://www.nsf.gov/publications/pub_summ.jsp?ods_key=nsf21107. [Accessed 07-08-2023].

Samaneh Aminikhanghahi and Diane J Cook. 2017. A survey of methods for time series change point detection. *Knowledge and information systems*, 51(2):339–367.

Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. 2002. Support vector machines for multiple-instance learning. *Advances in neural information processing systems*, 15.

D Banesh, M Petersen, J Wendelberger, J Ahrens, and B Hamann. 2019. Comparison of piecewise linear change point detection with traditional analytical methods for ocean and climate data. *Environmental Earth Sciences*, 78:1–16.

W Beitz, G Pahl, and K Grote. 1996. Engineering design: a systematic approach. *Mrs Bulletin*, 71.

F Lawrence Bennett. 2003. *The management of construction: a project life cycle approach*. Routledge.

Xinmiao Ding, Bing Li, Weiming Hu, Weihua Xiong, and Zhenchong Wang. 2013. Horror video scene recognition based on multi-view multi-instance learning. In *Computer Vision–ACCV 2012: 11th Asian Conference on Computer Vision, Daejeon, Korea, November 5-9, 2012, Revised Selected Papers, Part III 11*, pages 599–610. Springer.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Tom Gilb. 2005. *Competitive engineering: a handbook for systems engineering, requirements engineering, and software engineering using Planguage*. Elsevier.

Carlo Grillenzoni. 2012. Evaluation of recursive detection methods for turning points in financial time series. *Australian & New Zealand journal of statistics*, 54(3):325–342.

Wei He and Yu Wang. 2009. Text representation and classification based on multi-instance learning. In *2009 International Conference on Management Science and Engineering*, pages 34–39. IEEE.

Francisco Herrera, Sebastián Ventura, Rafael Bello, Chris Cornelis, Amelia Zafra, Dánel Sánchez-Tarragó, Sarah Vluymans, Francisco Herrera, Sebastián Ventura, Rafael Bello, et al. 2016. *Multiple instance learning*. Springer.

Steven R Hirshorn, Linda D Voss, and Linda K Bromley. 2017. Nasa systems engineering handbook. Technical report.

Bin Li, Yin Li, and Kevin W Eliceiri. 2021. Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14318–14328.

Shuo Li, Fang Liu, and Licheng Jiao. 2022. Self-training multi-sequence learning with transformer for weakly supervised video anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1395–1403.

Bo Liu, Yanshan Xiao, and Zhifeng Hao. 2018. A selective multiple instance transfer learning method for text categorization problems. *Knowledge-Based Systems*, 141:178–187.

Chang Liu, Mark Last, and Armin Shmilovici. 2019a. Identifying turning points in animated cartoons. *Expert Systems with Applications*, 123:246–255.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zvi Lotker. 2016. The tale of two clocks. In *Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 768–776.

Oded Maron and Tomás Lozano-Pérez. 1997. A framework for multiple-instance learning. *Advances in neural information processing systems*, 10.

James N Martin. 2000. Processes for engineering a system: an overview of the ansi/eia 632 standard and its heritage. *Systems Engineering*, 3(1):1–26.

Tracy L Osborne. 2022. Nasa space flight program and project management handbook.

Pinelopi Papalampidi, Frank Keller, and Mirella Lapata. 2019. Movie plot analysis via turning point identification. *arXiv preprint arXiv:1908.10328*.

Gwenolé Quellec, Guy Cazuguel, Béatrice Cochener, and Mathieu Lamard. 2017. Multiple-instance learning for medical image and video analysis. *IEEE reviews in biomedical engineering*, 10:213–234.

NASA Procedural Requirement. 2018. 7120.8 a, nasa research and technology program and project management requirements".

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. 2018. Deep one-class classification. In *International conference on machine learning*, pages 4393–4402. PMLR.

PJ Sudharshan, Caroline Petitjean, Fabio Spanhol, Luiz Eduardo Oliveira, Laurent Heutte, and Paul Honeine. 2019. Multiple instance learning for histopathological breast cancer image classification. *Expert Systems with Applications*, 117:103–111.

Waqas Sultani, Chen Chen, and Mubarak Shah. 2018. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6479–6488.

Huimin Tang, Peiwu Dong, and Yong Shi. 2019. A new approach of integrating piecewise linear representation and weighted support vector machine for forecasting stock turning points. *Applied soft computing*, 78:685–696.

Yuan Tian, Wenning Hao, Dawei Jin, Gang Chen, and Ao Zou. 2020. A review of latest multi-instance learning. In *Proceedings of the 2020 4th International Conference on Computer Science and Artificial Intelligence*, pages 41–45.

Charles Truong, Laurent Oudre, and Nicolas Vayatis. 2020. Selective review of offline change point detection methods. *Signal Processing*, 167:107299.

Yang Yang, Yanlun Tu, Houchao Lei, and Wei Long. 2023. Hamil: Hierarchical aggregation-based multi-instance learning for microscopy image classification. *Pattern Recognition*, 136:109245.

A Large-scale Project Phases

Table 3: Phase definitions in the engineering process of large-scale projects.

Phase	Description
Conceptual Studies	Propose ideas or concepts; Assess the feasibility of ideas or concepts; Establish the requirements and objectives of the task
Preliminary Design	Establish baseline tasks; Design architecture; Determine required technologies; Establish a design solution; complete 'implementation' level of design
Final Design	Establish complete, validated detailed design; Complete all design specialty audits; Establish manufacturing processes and controls
Fabrication&Integration	Prepare production facilities; Manufacture products that meet specifications and acceptance standards; Assemble and integrate systems
Commissioning	Validate the system; System test and commissioning
Operation	Perform mission; Sustain system
Renewal/Pause	Improve/augment system; Suspend system operations
Termination	Implement decommission/disposal

Table 3 provides the definitions of task phases within the context of large-scale projects. These definitions have been developed by drawing upon the insights and guidelines provided by NASA(Hirshorn et al., 2017; Requirement, 2018; Osborne, 2022) and NSF(nsf) regarding phase definitions in projects. Our approach involved a comprehensive review and synthesis of the definitions from these reputable sources to formulate a clear and concise delineation of the various phases that characterize large-scale projects.

B Dataset Statistic

Table 4 presents the statistical information for the dataset. There are a total of 80 projects in the dataset, with 46 Deep Space Exploration Projects, 15 Large Ground Facilities Projects, and 19 Aeronautic and Marine Engineering Projects.

Table 4: Statistics of constructed dataset.

Project Type	All	DSE	LGI	AME
# of Project	80	46	15	19
Avg. # of phases	3.8	4.1	2.9	3.7
Avg. # of windows	66.3	62.2	60.4	81.1
Avg. # of TP windows	10.5	10.8	9.3	10.6
Avg. # of non-TP windows	55.8	51.4	51.1	70.4
Avg. # of news in window	5.0	5.0	4.3	5.4
Avg. # of news in phase	14.0	14.3	12.3	13.9
Avg. phase time span(M)	30	28	35	33
Avg. 2 news time span(D)	53	53	69	47