System Design, Evaluation and Applications of Domain Term Extraction from Engineering Videos

Jiaxi Li*
University of Illinois Urbana-Champaign
jiaxili3@illinois.edu

Ninghan Zhong
University of Illinois Urbana-Champaign
ninghan2@illinois.edu

Rob Kooper
University of Illinois Urbana-Champaign
kooper@illinois.edu

Lawrence Angrave
University of Illinois Urbana-Champaign
angrave@illinois.edu

Abstract

Understanding the meanings of domain-specific terms is essential to academic success in college-level STEM courses. However, it can be challenging for students to obtain correct spellings and precise definitions of domain-specific terms from lecture videos, given the limited lecture time, rarity of the terms, and possibly confusing pronunciations. To provide accurate speech-to-text transcription, and enable students to search for domain-specific terms and obtain term definitions in real-time, we designed, implemented, and evaluated the PhraseHinter tool, a text analytics pipeline that efficiently extracts domain-specific terms from engineering educational videos. The tool is lightweight and adaptable to online instruction platforms.

In our approach, a series of key scenes are initially extracted from a lecture video using a novel scene detection algorithm. The algorithm employs a support vector machine to classify image differences based on pixel, face, and text similarity information [2]. A domain corpus is built by using the optical character recognition (OCR) technique to extract text from the scenes. A sequence of text-cleaning algorithms is applied to the domain corpus to filter out invalid characters, punctuation, and stop words. Frequent phrases are identified using standard text mining algorithms including PrefixSpan [15]. Using the TF–IDF metric [16], we compare the cleaned corpus to the background corpus to determine domain-specific terms and phrases.

The proposed PhraseHinter tool has been successfully integrated into ClassTranscribe [4, 11, 3, 19, 2, 10], a web-based video lecture platform, for multiple purposes: 1) Improve the Microsoft Azure Speech-to-Text accuracy by preparing a list of domain-specific terms with high confidence of occurrence in the audio, 2) Provide the input for the glossary tool, another text analytics service in ClassTranscribe that automatically generates the explanation for the domain-specific terms, and, currently in progress, 3) Provide search capability in order to locate the moments in the video when a domain-specific term is visually presented.

In this paper, we evaluate the performance and accuracy of the PhraseHinter system based on a representative corpus of videos from different engineering disciplines with domain-specific terms and phrases correctly pre-identified. We share the evaluation dataset to the education community for further research. In addition, we present the source code and provide guidance for instructors who would like to adopt the tool.

1 Introduction

Domain-specific terms refer to words or phrases that are primarily used within the context of a specific field of study. Domain-specific terms often involve abstract and sophisticated meanings e.g., “hybridization” in chemistry or “backpropagation” in machine learning. A large amount of domain-specific terms are introduced and taught in college-level STEM lectures and textbooks, which serve as the building blocks for advanced courses. Consequently, understanding the meanings of those terms plays an important role in achieving success in the curriculum. Yet, due to the varied learning outcomes from students’ high schools, college freshmen without sufficient prerequisites may face obstacles when learning domain-specific terms because not all of them are covered in the classroom due to the limited lecture time and the scope of the course.

Computer-based learning platforms can provide new learning tools and features that help mitigate this problem. For example, a web-based dictionary where every student is able to access and edit content would be a valuable platform for students to share, learn, and consolidate knowledge in domain-specific terms. However, in order to implement and adopt these features into online education, the first step is to generate a glossary of domain-specific terms for each course, because instructors may not have provided one for each course they teach. Today, advances in Text Mining and related algorithms make it possible to extract and present domain-specific terms and definitions in an efficient and accurate manner.

In this paper, we introduce “PhraseHinter”, an automated text analytic service that extracts domain-specific terms from the text-based course modality including text presented visually, which, based on empirical experiments, achieves a reasonable accuracy within a reasonable time frame. The PhraseHinter service is open source and is available at https://github.com/classtranscribe/pyapi. We present how the tool was integrated into a web-based learning system and present three features that were created using the service. These include more accurate closed captioning, a glossary tool, and a query system for domain-specific terms.

The paper is organized as follows. In section 2, we introduce the background of our work including the guiding methodologies and the integration of the service into an existing learning platform. In section 3, we present a formal description and motivation of the problem. We discuss the challenges encountered during the research and related works. In section 4, we present the system in detail using a data-flow diagram and explanations of the text processing procedures. In section 5, we report on the accuracy and processing time requirement of the service. In section 6, we discuss the applications of the tool within ClassTranscribe. We conclude the paper with an overview of the contributions of the paper and discuss the remaining challenges.

2 Background

2.1 Universal Design for Learning

Universal Design for Learning (UDL) refers to a set of guidelines or principles that addresses the development of flexible learning environments that are suitable for all stu-

dents [17]. In other words, UDL seeks accessibility and inclusiveness when implementing educational frameworks. A UDL-guided course should provide students with multiple learning pathways and modalities toward academic success. For instance, students could absorb the course content by attending traditional lectures, watching lectures through online platforms, reading course notes and lecture transcriptions, or a combination of all methods. Under the UDL guidelines, a strong emphasis is placed on inclusiveness. Educational technologies should be able to deliver widely accessible contents that benefit all students, regardless of their backgrounds and physical conditions [12]. Adhering to the principles under UDL, the PhraseHinter system was designed to assist course glossary creation and improve automatic speech-to-text transcriptions. These modalities provide alternative learning options for all students and are particularly valuable for students who are Deaf or Hard-of-Hearing (DHH).

2.2 ClassTranscribe

ClassTranscribe is a web-based video learning platform designed to offer accessible educational content, developed at the University of Illinois. ClassTranscribe has been previously described in ASEE Conferences [11, 3, 19, 2, 10]. Designed to provide accessible education to engineering college students, ClassTranscribe is equipped with user-friendly features such as automatic transcriptions and digital book generation [10]. ClassTranscribe also allows students to easily search through the automatically generated transcriptions and fix transcription errors [19]. ClassTranscribe has been used as an educational platform for Computer Science and other engineering disciplines. Its usage covers multiple large-enrollment classes with over 300 students, and its videos have been watched by over 6,200 students.

One feature of ClassTranscribe is automatic speech-to-text transcription, which allows students to understand the oral content of the lectures more efficiently. Further, to be an effective learning modality, the transcriptions need to be accurate, which is challenging in engineering classes that include domain-specific terms. Thus, a robust speech-to-text transcription system that can precisely capture the domain-specific terms for a wide range of engineering disciplines is valuable.

2.3 SceneDetection

SceneDetection is a tool that has been implemented in ClassTranscribe, which enables efficient identification and removal of similar and repetitive frames from a lecture video. It compresses a lecture video into a few unique scenes that include the equivalent visual content to the full lecture video. Though SceneDetection primarily processes pixel differences from one frame to the next, to further improve the accuracy, an Optical Character Recognition (OCR) step is included to extract text data from visual images. The detailed algorithm design, evaluation, and implementation of the SceneDetection system are described in [2]. This OCR text data is used as the initial source of the PhraseHinter.

3 Problem Description

3.1 Motivation and Goals

With the goal of providing students with web-based interactive tools to effectively fulfill the knowledge gap in domain-specific terms for college-level STEM education, our first task was to build a text analytic service that efficiently discovered domain-specific terms from text-based lecture modalities. Domain-specific terms refer to terms that are primarily used in a specific academic context and rarely used in general English. In our approach, domain-specific terms refer to words e.g., “binomial” and phrases e.g., “inner product”. As a computational performance target, the service should be able to process within a minute the text extracted from a 50-minute lecture.

3.2 Challenges

The task of domain-specific term extraction is non-trivial. First, to ensure students an effective learning experience, the domain-term extraction system should not add significant additional latency into the lecture video processing pipeline. Second, the domain term identification should reach an acceptable accuracy, allowing students to read the correct words. Further, the domain-term extraction system should be robust enough to adapt across all engineering disciplines. Lastly, in STEM subjects, domain-specific terms do not only imply single words but also technical phrases, e.g., “support vector machine” in machine learning and “forward kinematics” in robotics. These phrases occupy a large proportion of the domain-specific terms in engineering disciplines and are non-trivial to identify. Thus, to ensure helpful domain term extractions, the system should identify not only single keywords but also technical phrases for completeness.

3.3 Related Works

Early works on domain-specific term extraction utilized rule-based approaches, in which the structural characteristics of the domain-specific term were first identified. Then, strings that matched these characteristics were extracted [8, 6]. These approaches were focused on a specified language or domain. Later works developed statistical and distributional approaches. In [14], term co-occurrence patterns were analyzed to extract terms that referred to specialized concepts. In [9], an unsupervised domain-term extraction method was proposed, based on Term Frequency and Inverse Document Frequency (TF-IDF, Section 4.3), a common weighting technique for keyword extractions. If the TF-IDF value of a term is greater than a specified threshold, the term is considered domain-specific. More recent methods employed machine learning approaches. In [18], a set of seed terms were first extracted using rule-based approaches. Then, the seed terms were used to train an LSTM-CRF model to extract extended terms from technical documents. In [13], domain-term identification was considered as a binary classification, where each word was classified as either a domain-term or not. A deep neural network with a pre-trained word-embedding layer was used to perform the classification. Existing methods of domain-specific term extraction focused on the natural language processing tasks. To the best of our knowledge, few have considered processing time requirements and did not evaluate their approaches within an educational context.

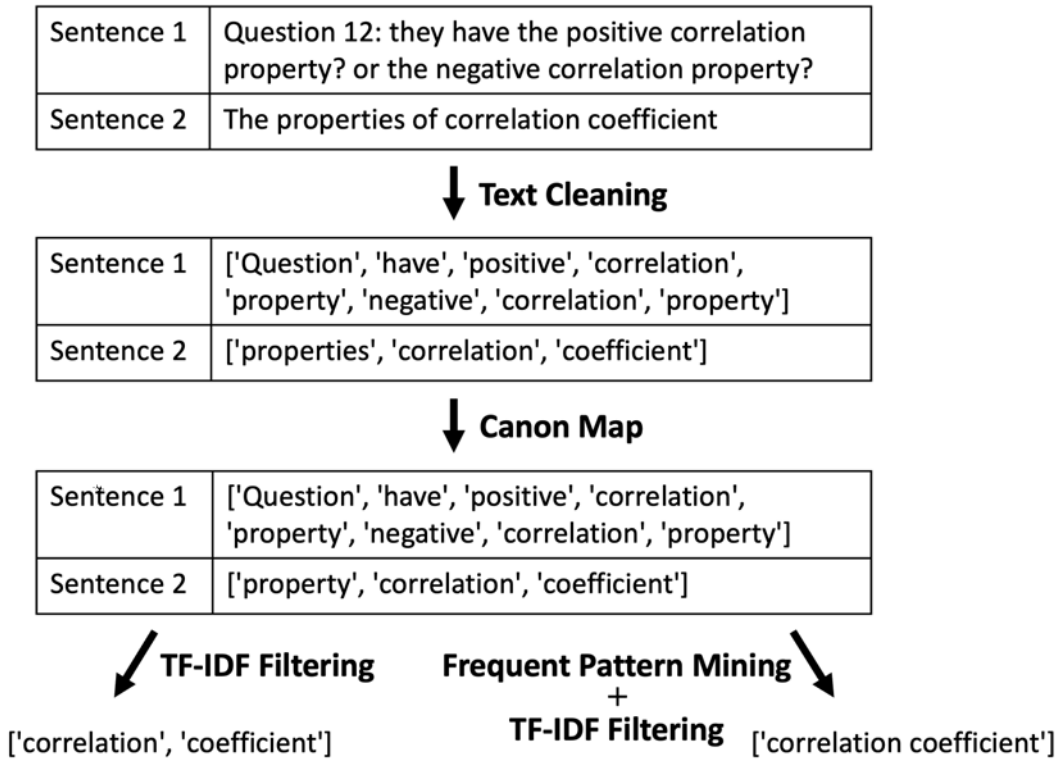

Figure 1: PhraseHinter System Overview.

4 System Design

Figure 1 provides an overview of the data-flow of the PhraseHinter system. The system was implemented in Python and used the Natural Language Toolkit (NLTK) [5] and the PrefixScan library [15].

4.1 Text Cleaning

The raw text input from the text-based lecture modality contains punctuation and invalid characters due to OCR errors. Further, high frequency (common) stop words (e.g., a, an, the) are not domain-related. Consequently, the following words or characters were removed to provide a more representative and meaningful data set for later processing.

- Punctuation
- Characters that are neither alphabets nor numbers
- Stop words
- Words with less than 2 characters
- Words that only contain digits

As indicated by Figure 1, punctuation, digits, and stop words were removed from both sentences after text-cleaning.

4.2 Canonical Word Map

The same word can be written in different styles, such as in the plural form, uppercase, titlecase, lowercase, etc. Without accurate matching and conversion, the dataset may not precisely represent the frequency of each word. Therefore, a canonical word map was created to record the number of occurrences of each word independent of style. Words written in other styles were converted to the style that has the most number of occurrences in the dataset, which was assumed to be the most representative style of the word. For example, in Figure 1, "property" and "properties" were referring to the same word. Since "property" appeared twice, "properties" was converted to "property" in the output.

4.3 TF-IDF Filtering

TF-IDF, or term frequency–inverse document frequency, is a statistical index that measures how a word is relevant to a document or a text corpus [16]. We compared the TF-IDF value of the words inside the course-specific dataset to those inside the Brown Corpus [7], a background corpus that collected English text from everyday language use. Words that were important in the course-specific dataset but unimportant in the background corpus were determined to be domain-specific. For example, as shown in Figure 1, words that were often used in daily life e.g., "question", "have", and "property" were removed from the final output.

4.4 Sequential Pattern Mining

Instead of generating only single words, OCR retained the sentence structure and contextual relationship between individual words, which enabled the Sequential Pattern Mining algorithms[1] to discover multi-word phrases. Sequential Pattern Mining is a data mining process that extracts important sequential patterns from the dataset and was used to extract frequent phrases from the dataset.

Considering the processing performance requirements, a PrefixSpan algorithm [15] was selected to extract frequent phrases because of its minimal processing time requirement and efficient memory utilization. We specified the minimum support to be 2 instances, which removed all phrases that only appeared once in the text and built a bag-of-words model for the remaining phrases. TF-IDF filtering was applied to the model to remove the common phrases e.g., "different from" and incomplete phrases e.g., "is a". For example, as shown in Figure 1, the multi-word phrase "correlation coefficient" was extracted using the PrefixSpan algorithm.

The Sequential Pattern Mining is the final processing step of the PhraseHinter. An example output of the PhraseHinter after processing a Bioengineering video is "convective, stagnant, gas, initial concentration, concentration, surface, evaporating, liquid, convective stream, stagnant liquid, mass transfer, solid, moisture, diffusivity (100 more phrases not shown)."

Recall (%)	Full Corpus Processing Time (seconds)	Single Lecture Time (seconds)
71.36	1.55	0.14

Table 1: Performance metrics of the PhraseHinter service; Recall accuracy and processing time requirements of the PhraseHinter system for all 10 lectures and one 50 minute lecture.

5 Evaluation

5.1 Dataset

We tested the performance of the PhraseHinter system using a dataset of (N = 10) lecture videos from multiple STEM disciplines offered at the University of Illinois. The total length of the videos in the dataset was 5 hours and 30 minutes. A testing dataset with 3,928 lines of text that accompanied the lecture videos was generated using the SceneDetection system, which served as the source input for the PhraseHinter service. With the help of students with related engineering experience, 398 valid domain-specific terms were identified from the testing dataset.

5.2 Recall Accuracy

Our approach was able to achieve high accuracy (71%) by successfully extracting 284 target domain-specific terms as indicated in Table 1.

Recall, the ratio of the number of domain-specific terms correctly extracted to the number of domain-specific terms that appeared in the videos, measures the robustness in completely discovering all target domain-specific terms in the testing set. It is an important accuracy metric because we wish to present as many domain-specific terms as possible. Providing too many terms, though distracting, is unlikely to significantly impact learning outcomes. Students could simply ignore the extra terms that they were already familiar with. However, missing a term in the learning environment may cause students to assume that the term is not relevant to the context and hence skip the important content.

5.3 Time Requirement

The service was tested using a laptop with an Intel Core i7-6920HQ CPU. We chose this testing configuration because we want to ensure that our system has acceptable latency in general hardware environments, especially in systems without powerful GPUs. As shown in Table 1, it required only 1.55 seconds for our system to process the text of all 10 lecture videos in the dataset (around 0.14 seconds for a 50-minutes lecture). This was within our desired processing time.

6 Applications

6.1 Improving Speech-to-Text Accuracy

Domain-specific terms can improve the accuracy of some automated speech-to-text services, including the Microsoft Azure Speech-To-Text Service. Most online learning

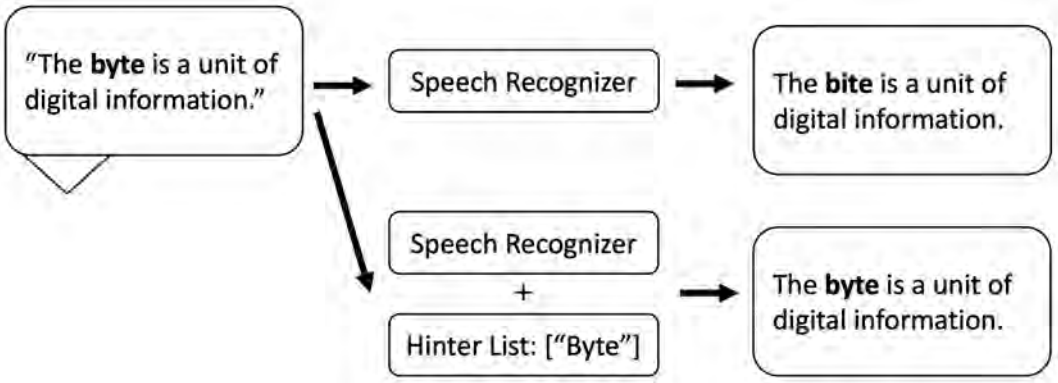

Figure 2: Improving Speech-to-Text Accuracy.

platforms provided accessible closed captioning for DHH students by transcribing the lecture audio into text using speech recognition systems. However, speech recognizers have difficulty in accurately transcribing domain-specific terms. The technical terms may be absent or incorrectly transcribed into phonetically-similar words. The PhraseHinter tool can provide a list of potential phrases as an a priori set of hints to the speech recognizer to improve the likelihood of correctly identifying domain-specific terms. For example, the technical word "byte" (see Figure 2) might be transcribed as the common word "bite" by a speech recognizer. But after providing the domain hint "byte", the speech transcriber will increase the probability likelihood of selecting the word "byte" during the transcribing process. In the next section, we discuss a Glossary Tool, which is another application of the PhraseHinter service.

6.2 Glossary Tool

When students are confused about a domain-specific term, additional time and effort is required searching for meanings. To improve the learning experience, we created an automated workflow that efficiently collected definitions and explanations of the domain-specific terms from a variety of open-source sites including Wikipedia and WordNet. An interactive glossary tool was created and implemented in ClassTranscribe, which presented the definition, domain, affiliated course, and external link for each domain-specific term (see Figure 3). Both instructors and students can upvote or edit the information for each term. We propose that this new approach will encourage knowledge exploration in domain-related concepts and help learning outcomes. The detailed system design, data flow, and user experience of the glossary tool are beyond the scope of this paper and will be presented as a full paper at the main 2023 ASEE Conference. In the next section, we present text-based visual search; a third application of the PhraseHinter.

6.3 Domain-Specific Term Search

In addition to extracting the text from video frames, the SceneDetection system also recorded the timestamp for each scene. Therefore, it is possible to create a map between the domain-specific terms and visual time of their appearances in each video. This

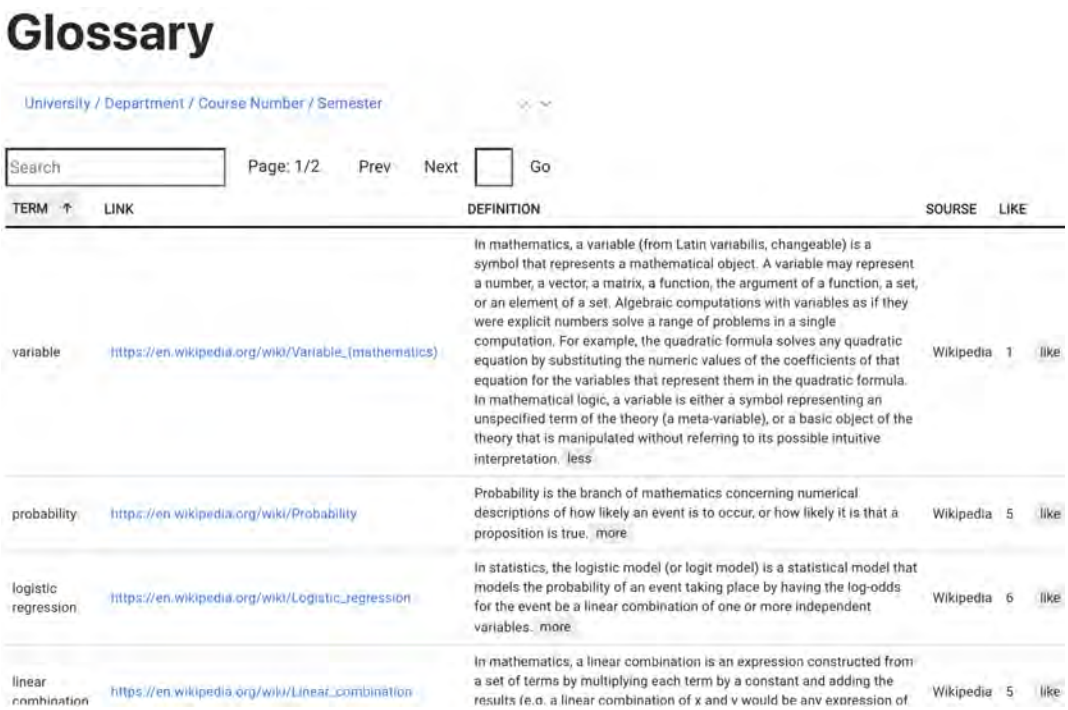

Figure 3: The Glossary Tool.

functionality creates opportunities for new functionality in ClassTranscribe video playback system; students will be able to directly search and seek to the exact moments in a video where the the query term is first presented or re-used.

7 Conclusion and Future Works

We presented the value and a method of extracting domain-specific terms and phrases within the context of engineering education. Student understanding of domain-specific terms is an important step for students to comprehend course materials. Under the principles of Universal Design for Learning, this paper proposed PhraseHinter, a system that was designed for domain-term extraction from engineering lecture videos. We presented the design of the system pipeline, which included text pre-processing, TF-IDF Filtering, and Sequential Pattern Mining. We analyzed the performance of the proposed system with a dataset of (N = 10) lecture videos from multiple engineering disciplines. Our evaluation indicated that the PhraseHinter system achieved a sufficient performance with reasonable accuracy. Finally, we presented three learning-focused applications of the PhraseHinter service; improving Speech-to-Text transcriptions, a Glossary Tool, and search of visual text.

This work introduced additional opportunities to the engineering education community. For example, with the domain-specific terms identified, test questions can be automatically included based on the terms. It is an open question if domain term information could be used to understand dependencies and relationships across lecture videos. We encourage the engineering education practitioners who are interested in adopting new

approaches into their course, to contact us for more information. Finally, we hope our system and its initial results will inspire additional innovation in inclusive and accessible education.

8 Acknowledgment

We wish to thank the Illinois students who contributed to the ClassTranscribe project, members of the Illinois Computer Science Education group. We also acknowledge the invaluable technical support from University of Illinois students, staff, and faculty, including Rob Kooper, and technical support from National Center for Supercomputing Applications (NCSA). Portions of this research were supported by a Microsoft Corporation gift to the University of Illinois as part of the 2019 Lighthouse Accessibility Microsoft-Illinois partnership and a 2022 Microsoft Corporation research gift, “Accessible Multimedia for Learning.”

References

[1] R. Agrawal and R. Srikant. “Mining sequential patterns”. In: *Proceedings of the Eleventh International Conference on Data Engineering*. 1995, pp. 3–14. DOI: 10.1109/ICDE.1995.380415.

[2] Lawrence Angrave, Jiaxi Li, and Ninghan Zhong. “Creating TikToks, Memes, Accessible Content, and Books from Engineering Videos? First Solve the Scene Detection Problem”. In: *2022 ASEE Annual Conference & Exposition*. 2022. URL: https://peer.asee.org/41185.

[3] Lawrence Angrave et al. “Improving Student Accessibility, Equity, Course Performance, and Lab Skills: How Introduction of ClassTranscribe is Changing Engineering Education at the University of Illinois”. In: *2020 ASEE Annual Conference*. June 2020. DOI: 10.18260/1-2--34796.

[4] Lawrence Angrave et al. “Who Benefits? Positive Learner Outcomes from Behavioral Analytics of Online Lecture Video Viewing Using ClassTranscribe”. In: *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. SIGCSE ’20. Portland, OR, USA: Association for Computing Machinery, 2020, pp. 1193–1199. ISBN: 9781450367936. DOI: 10.1145/3328778.3366953.

[5] Steven Bird and Edward Loper. “NLTK: The Natural Language Toolkit”. In: *Proceedings of the ACL Interactive Poster and Demonstration Sessions*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 214–217.

[6] Didier Bourigault. “Surface grammatical analysis for the extraction of terminological noun phrases”. In: *COLING 1992 Volume 3: The 14th International Conference on Computational Linguistics*. 1992.

[7] W. Nelson Francis. “A Standard Corpus of Edited Present-Day American English”. In: *College English* 26.4 (1965), pp. 267–273.

[8] John S Justeson and Slava M Katz. “Technical terminology: some linguistic properties and an algorithm for identification in text”. In: *Natural language engineering* 1.1 (1995), pp. 9–27.

[9] Su Nam Kim, Timothy Baldwin, and Min-Yen Kan. “Extracting domain-specific words-a statistical approach”. In: *Proceedings of the Australasian Language Technology Association Workshop 2009*. 2009, pp. 94–98.

[10] Hongye Liu et al. “A Digital Book Based Pedagogy to Improve Course Content Accessibility for Students with and without Disabilities in Engineering or other STEM courses (WIP)”. In: *2022 ASEE Annual Conference & Exposition*. Minneapolis, MN: ASEE Conferences, Aug. 2022. URL: https://peer.asee.org/41438.

[11] Chirantan Mahipal et al. ““What did I just miss?!” Presenting ClassTranscribe, an Automated Live-captioning and Text-searchable Lecture Video System, and Related Pedagogical Best Practices”. In: *2019 ASEE Annual Conference*. June 2019. DOI: 10.18260/1-2--31926.

[12] Stephanie Moore. “David H. Rose, Anne Meyer, Teaching Every Student in the Digital Age: Universal Design for Learning”. In: *Educational Technology Research and Development* 55 (Oct. 2007), pp. 521–525. DOI: 10.1007/s11423-007-9056-3.

[13] Despoina Mouratidis et al. “Domain-Specific Term Extraction: A Case Study on Greek Maritime Legal Texts”. In: *Proceedings of the 12th Hellenic Conference on Artificial Intelligence*. 2022, pp. 1–6.

[14] Rogelio Nazar. “Distributional analysis applied to terminology extraction: First results in the domain of psychiatry in Spanish”. In: *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication* 22.2 (2016), pp. 141–170.

[15] Jian Pei et al. “PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth”. In: *Proceedings 17th International Conference on Data Engineering*. 2001, pp. 215–224. DOI: 10.1109/ICDE.2001.914830.

[16] Anand Rajaraman and Jeffrey David Ullman. “Data Mining”. In: *Mining of Massive Datasets*. Cambridge University Press, 2011, pp. 1–17.

[17] David H. Rose et al. In: *Teaching Every Student in the Digital Age : Universal Design for Learning*. Association for Supervision and Curriculum Development, 2002.

[18] Chong Wang et al. “A learning-based approach for automatic construction of domain glossary from source code and documentation”. In: *Proceedings of the 2019 27th ACM joint meeting on european software engineering conference and symposium on the foundations of software engineering*. 2019, pp. 97–108.

[19] Zhilin Zhang et al. “How Students Search Video Captions to Learn: An Analysis of Search Terms and Behavioral Timing Data”. In: *2021 ASEE Virtual Annual Conference Content Access*. Virtual Conference: ASEE Conferences, July 2021. URL: https://strategy.asee.org/37257.