Research Article Open Access

Liuqing Li#, Jack Geissinger#, William A. Ingram, Edward A. Fox*

Teaching Natural Language Processing through Big Data Text Summarization with Problem-Based Learning

https://doi.org/10.2478/dim-2020-0003
received September 27, 2019; accepted January 21, 2020.

Abstract: Natural language processing (NLP) covers a large number of topics and tasks related to data and information management, leading to a complex and challenging teaching process. Meanwhile, problem-based learning is a teaching technique specifically designed to motivate students to learn efficiently, work collaboratively, and communicate effectively. With this aim, we developed a problem-based learning course for both undergraduate and graduate students to teach NLP. We provided student teams with big data sets, basic guidelines, cloud computing resources, and other aids to help different teams in summarizing two types of big collections: Web pages related to events, and electronic theses and dissertations (ETDs). Student teams then deployed different libraries, tools, methods, and algorithms to solve the task of big data text summarization. Summarization is an ideal problem to address learning NLP since it involves all levels of linguistics, as well as many of the tools and techniques used by NLP practitioners. The evaluation results showed that all teams generated coherent and readable summaries. Many summaries were of high quality and accurately described their corresponding events or ETD chapters, and the teams produced them along with NLP pipelines in a single semester. Further, both undergraduate and graduate students gave statistically significant positive feedback, relative to other courses in the Department of Computer Science. Accordingly, we encourage educators in the data and information management field to use our approach or similar methods in their teaching and hope that other researchers will also use our data sets and synergistic solutions to approach the new and challenging tasks we addressed.

Keywords: information system education, computer science education, problem-based learning, natural language processing, NLP, big data text analytics, machine learning, deep learning.

1 Introduction

Teaching natural language processing (NLP) is both exciting and challenging for faculty in universities and colleges. Traditionally, such a computing-related course is taught using hour-long, content-driven lectures. Abstract concepts and principles are often illustrated to students first, followed by idealized examples that may be far removed from their personal experiences or interests. Further, grade competition keeps students isolated; typical end-of-chapter, plug-and-chug exercises foster knowledge without conceptual understanding (Mazur, 1992). Thus, students may not be actively involved in self-learning. We made use of problem-based learning (PBL), an advantageous method that has found application in modern education, to avoid such roadblocks in teaching NLP. PBL results from the process of working toward the understanding or resolution of a problem (Barrows & Tamblyn, 1980) and can be incorporated into a variety of classroom formats, such as small-group collaborative activities, large-group case method discussion, laboratory experimentation, and interactive lecturing (Wilkerson & Feletti, 1989). All these means can motivate students' interest, improve their collaboration ability, and enhance their autonomy and practical skills.

Regarding our NLP teaching scenario, we elaborate on the following research problem: "How can we best automatically construct English language

***Corresponding author: Edward A. Fox,** Department of Computer Science, Virginia Tech, VA 24061, USA, Email: fox@vt.edu
Liuqing Li, Department of Computer Science, Virginia Tech, VA 24061, USA
Jack Geissinger, Department of Electrical and Computer Engineering, Virginia Tech, VA 24061, USA
William A. Ingram, University Libraries, Virginia Tech, VA 24061, USA
#Liuqing Li and Jack Geissinger contributed equally to this paper.

summaries of the important information in a large document collection?" This is one of many research hotspots in recent years, combining both "big data" and "text summarization", and covers a diverse set of research fields in NLP. Further, we broadly introduce students to the area of data and information analytics and management, through mining structured and unstructured data, gaining insights and knowledge from data using machine learning, and using novel methods to collect and use personal and public Web postings – via a set of case studies of big data analytics.

Two versions of the course run concurrently and use different document collections: (1) a set of Web pages about an event, topic, or trend; (2) each of the chapters in a collection of electronic theses and dissertations (ETDs). These correspond to multi-document (MDS) and single-document summarization (SDS), respectively. Later, we provide different types of resources (e.g., data, hardware, and software) to support computing, and propose a general pipeline for each version as a high-level guide for summarization. Since our course is designed for both undergraduate and graduate students, we further divide all students into different teams to foster student interaction, teamwork, and reinforcement of interpersonal skills (Vernon, 1995).

Since one of our goals is to promote student-centered learning, we do not supply student teams with all the information for problem-solving; instead, they are encouraged to explore a variety of methods, techniques, and tools to solve the problem through just-in-time self-learning. We illustrate the methodologies that the students followed to solve the problem. We also quantitatively and qualitatively measure the performance of their approaches. Afterward, we evaluate the effectiveness of our PBL course by considering the quality of the summaries generated by the student teams and the feedback from students on the course and on PBL. This leads to several interesting observations. Moreover, we also discuss multiple pedagogical solutions in our PBL course and hope they can support other PBL courses in the same or related areas. Altogether, there are multiple contributions, which can be divided into two major categories.

(1) Educational:

- We applied PBL in teaching NLP through big data text summarization, providing a good demonstration for other similar NLP teaching scenarios. Compared with some previous approaches, our problem is more motivating, realistic, and practical, and it also covers more research topics.
- We adopted multiple types of pedagogical solutions during PBL, including intra- and inter-team collaboration, peer evaluation, team presentations, and problem-driven lectures, to strengthen team dynamics, accomplish project milestones, and ensure educational value (confirmed by student assessments).
- We carried out two types of evaluations on our PBL course. One is to measure the summaries generated by student teams; the other is to measure student feedback. They both have quantitative evaluation and human evaluation results. We also treated undergraduate and graduate students as two separate groups in the latter evaluation, to gauge differences between their perceptions of the course. On all counts, results were positive.

(2) Technical:

- We devised a system architecture for summarization of big data text. This also serves as a treemap of NLP teaching and the learning points of the course. Thus, it marries a conceptualization of the problem and how to solve it, giving a layering suitable for both learning and implementation. The levels cover stages of processing, concepts, libraries/tools, and methods/algorithms (atop hardware).
- Regarding MDS, we found that topic level- and sentence level-based models perform better on our event-related collections, supporting the future work of summarization of tweets and Web pages.
- Regarding ETD chapter summarization, we identified promising approaches, which led to a successful proposal for additional research, now funded by the Institute for Museum and Library Service (IMLS).
- More broadly, we developed new data sets, including golden standard solutions to aid evaluation, to advance the field of text summarization. Little research has been done to summarize large event-oriented Web page collections or long documents, and none has been done with ETDs, so we have opened up new areas for technical contribution. Further, we have introduced, implemented, and evaluated new synergistic solutions using information retrieval, text analytics, machine learning, deep learning, and transfer learning.

The remainder of this paper is structured as follows. We present a literature review in Section 2, and the preparation for our PBL course in Section 3. In Section 4, we detail the proposed general pipelines and the methods used by student teams. We demonstrate our quantitative and human evaluation results on summarization and PBL in Sections 5 and 6, respectively. We discuss more about PBL and pedagogical solutions in Section 7. Our conclusions are drawn in the final section.

2 Related Work

2.1 Applications of PBL

PBL is an educational practice that has been in use for >4 decades. It is a popular methodology that helps students focus on the educational process. It originated in medical science education as a way to deal with the growing amount of information required to become a doctor (Barrows & Tamblyn, 1980; Barrows, 1986). This innovative learning approach is thought to improve students' understanding of the course material through active and self-directed learning. It should be noted that PBL is a form of active learning since it engages students and encourages them to perform analysis and explain their thinking (Serife, 2011). The goal of PBL and education, in general, is to encourage "deep" learning (Biggs, 1999). Deep learning in the educational context is based on comprehensive and full understanding. PBL has been found to encourage deep learning, but results are varied based on how the specific course or curriculum is structured (Dolmans, Loyens, Marcq, & Gijbels, 2016).

PBL has been applied in areas as varied as introductory science courses (Allen, Duch, & Groh, 1996), chemistry education (Cline & Powers, 1997), and courses on a wide range of engineering topics (Mills & Treagust, 2003; Costa, Honkala, & Lehtovuori, 2007; Yadav, Subedi, Lundeberg, & Bunting, 2011; Zhang, Hansen, & Andersen, 2016). In addition, computer science education is possible with PBL (Nuutila, Törmä, & Malmi, 2005). Kay et al. (2000) described some challenges in teaching foundation courses on computer science, such as "Introduction to Programming" and "Introduction to Computer Science", and how they designed PBL approaches to address them. Cavedon, Harland and Padgham (1997) leveraged two World Wide Web (WWW)-based tools – the CourseWeb and the ProblemWeb – to support their undergraduate artificial intelligence teaching. Indiramma (2014) presented a PBL approach for a course on "Theoretical Foundation of Computation", which enhanced group work and a social environment of education. Multiple projects were selected by different teams for implementation. Regarding teaching NLP through PBL, Carstensen and Hess (2003) presented an approach for combining Web-based learning with PBL and developed an interactive learning application for teaching introductory lectures on "Computational Linguistics". However, they did not carry out further experiments to evaluate their PBL approach. Currently, with the advancement of computing resources, state-of-the-art techniques, and software libraries, it is also feasible to cover data analytics in upper-level computer science courses (Núñez-del-Prado & Goméz, 2017). Kanan, Zhang, Magdy, and Fox (2015), using a newly constructed Hadoop cluster, applied PBL in a Computational Linguistics class. This and another course led to Virginia Tech's 2016 Xcaliber Award "for making extraordinary contributions to technology enriched active learning." As in our course, the problem addressed was to produce a good summary of an event, which was confirmed by an evaluation of the summaries generated by student teams. Compared with the work by Indiramma (2014), all teams in our course are focused on the same problem or research topic, which is helpful for evaluation. We conducted a comprehensive evaluation on both summarization and student views of our PBL course. Further, we also added the topic of deep learning to NLP and big data, aiming to help students keep pace with the development of new techniques and new methods.

Clearly, PBL has been shown to be an effective teaching method. Consequently, our focus when evaluating the teaching aspects of this research was on the effectiveness of what we did, as opposed to comparing teaching using PBL vs. teaching using other approaches, which, in our setting, was not possible in any case[1].

2.2 NLP Subfields

NLP is an interdisciplinary field applying both linguistics and computer science to understand, model, and process human languages (Jurafsky, 2000; Bird, Klein, & Loper, 2009; Indurkhya & Damerau, 2010). There are many subfields within NLP, e.g., those involving part-of-speech tagging, sentiment analysis, coreference resolution, neural machine translation (Sutskever, Vinyals, & Le, 2014; Bahdanau, Cho, & Bengio, 2015), natural language generation (Reiter & Dale, 1997; Stent & Bangalore, 2014), and automatic summarization.

Information extraction is an important step in text summarization, the goal of which is converting the source

[1] In another educational research study, the instructor of this course taught two sections of a same course, one as a control group, but in spite of careful design, there were too many unavoidable confounding variables to be able to show significant differences due to the main treatment. Further, our 2014 course on Computational Linguistics, which led to a university award, already confirmed the value of our use of PBL in a similar setting.

material into a shorter version while maintaining its key information and overall meaning. Thus, extracting useful information from natural language is a fundamental task within NLP. It comes in many different forms like keyword extraction, key-phrase extraction, named-entity recognition (NER), and so forth. Keyword extraction often is a purely statistical process based on finding the most frequent words in a collection of documents. The results vary depending on the quality of the data set, its size, and its uniformity, among other things. Importantly, teams in our course were tasked with finding frequent words in their documents, and many teams used these frequent words to craft templated summaries or classifiers for abstractive summarization. Key-phrase extraction builds upon keyword extraction and can be used to understand whether a document is relevant and contains important phrases. Also, key phrases can be used to determine the similarity between documents and improve summarization quality. When domain-specific information is present, as in technical reports in computer science, key-phrase extraction (e.g., using decision trees) improves in quality (Frank, Paynter, Witten, Gutwin, & Nevill-Manning, 1999).

Beyond keyword and key-phrase extraction, another type of information extraction is NER. Named entities are important linguistic features that refer to, e.g., organizations, geopolitical entities, products, or people (Bird et al., 2009). Importantly, NER systems are automated programs for detecting named entities and their types within documents. These systems are essential for information extraction tasks similar to the one given in our course. The Stanford CoreNLP toolkit contains the Stanford Named Entity Recognizer (Manning et al., 2014), which is a widely used system in NLP.

The task of clustering natural language documents also has a long history. Clustering text documents is an essential step in indexing, retrieval, management, visualization, and data mining. Vector space models are a popular method for representing text sources (Baeza-Yates & Ribeiro-Neto, 1999). The model takes documents as a $t \times d$ term-by-document matrix and compares term vectors to identify similarities between documents. Another way of clustering documents is through building ontological relationships (Gruber, 1995). Documents can be easily placed into different clusters based on prebuilt relationships. Some methods improve the algorithms to identify subspace structure (Li, Ma, & Ogihara, 2004), and some focus on finding important features during clustering (Frigui & Nasraoui, 2004).

Topic modeling is another popular choice related to clustering since each document has a probabilistic value in each topic (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990; Pritchard, Stephens, & Donnelly, 2000; Blei, Ng, & Jordan, 2003). Topic models are a type of statistical model used to determine what a document or collection contains in its content. Latent Dirichlet allocation (LDA), e.g., was a favorite choice among teams in our course. Through LDA, document collections can be intuitively represented as a mixture of various topics.

A pioneering work in the summarization task is that by Luhn (1958), which introduced a method to extract salient sentences from source material using features such as word and phrase frequencies. Presently, based on the type of input, summarization is of two types: SDS and MDS. One way of performing either SDS or MDS is through extractive summarization, where key sentences of the source material are extracted and concatenated together to form the summary. This methodology is popular and useful for both MDS and SDS, and it has been researched in the past through a variety of methods (Mihalcea & Tarau, 2004; Carbonell & Goldstein, 1998; Erkan & Radev, 2004). For example, extractive MDS is possible through detection of cluster centroids (Radev, Jing, Styś, & Tam, 2004). To complement extractive summarization and to enhance readability, recent attention has been given to the task of abstractive summarization, which also involves key methods of natural language generation (Reiter & Dale, 1997). Various neural networks have been proposed to accomplish this task through deep learning (Nallapati, Zhou, Gulcehre, & Xiang, 2016; See, Liu, & Manning, 2017; Chen & Bansal, 2018). The pointer-generator network (PGN) (See et al., 2017), e.g., was the state-of-the-art at the time of this course.

Considering all these points, we teach NLP through big data text summarization with PBL, covering most of the research fields in NLP.

3 PBL Course Preparation

In this section, we first introduce the background of our PBL course (Section 3.1). Next, we present the course's learning objectives (Section 3.2), showing specific units that students can complete during PBL. Then, we describe the big data sets and computational resources for summarization (Section 3.3). Finally, we list all teams and their corresponding collections to ensure a clear division of work and cooperation (Section 3.4).

3.1 Computer Science PBL Course

As a PBL course, “CS4984/CS5984: Big Data Text Summarization” was taught for both undergraduate and graduate students in the Department of Computer Science at Virginia Tech in Fall 2018. The course is based on a single question that drives and aids the student learning activities; it is organized like a “flipped classroom” with few lectures and many discussions. The task of the instructor, as well as the graduate teaching assistant (GTA) and related graduate research assistants (GRAs), is to provide a nurturing environment and to serve as facilitators/guides to help students to achieve course learning objectives.

By focusing on NLP through big data text summarization, our PBL course allows students to engage in active learning of NLP, especially on how to work with extensive event-related collections of text (e.g., tweets or Web pages) and ETD chapters. Specifically, students are encouraged to learn and use both classical and modern methods – which companies and research teams use in search engines, linguistic analysis, topic modeling, and text summarization – in analyzing big data collections, extracting vital information, and generating readable summaries of events and chapters. Just-in-time learning allows the development of an understanding of concepts, techniques, and toolkits so that students master the critical methods related to NLP through summarization.

3.2 Course Learning Objectives

The primary learning objective of our PBL course is to automatically construct summaries of the critical information in either an extensive document collection regarding an event or an ETD chapter.

The event summarization task requires students to learn many core concepts from NLP, such as information extraction, topic modeling, clustering, and extractive and abstractive summarization. Thus, the real-world motivation for the task can be recognized through the importance of the events that are summarized. Events of national and international interest can have long-term impacts on people and organizations. With a set of Web pages relevant to a national or international event, students are encouraged to extract important words and named entities, produce clusters and topics, and apply extractive and abstractive summarization techniques for summary generation. In this case, we designed a course structure with 10 units that cover diverse topics in NLP, recommended to provide scaffolding, progressing from easy to hard (see Table 1). Ultimately, all teams are charged with coming up with an NLP pipeline for getting the best results, using their best judgment, including being allowed to ignore some or all of the scaffolding in case of special knowledge or interest.

Table 1
The 10 Units for Summary Generation

Unit	Task Description
1	A set of most frequent important words
2	A set of WordNet synsets that cover the words
3	A set of words constrained by part of speech (POS), e.g., nouns and/or verbs
4	A set of words/word stems that are discriminating features
5	A set of frequent and important named entities
6	A set of important topics, e.g., identified using LDA
7	An extractive summary, as a set of important sentences
8	A set of values for each slot, matching the collection semantics
9	A readable summary explaining the slots and values
10	A readable abstractive summary, e.g., from deep learning

For the ETD summarization task, the learning objective is mainly on deep learning. Students are required to work together to devise and tailor a workflow to meet the specific challenges of ETD summarization. Specifically, this requires students to address the problems of obtaining suitable training data, determining both where deep learning will apply and how it can be combined with other NLP techniques, as well as acquire knowledge and skill with a particular deep learning methodology.

Each week during the course, teams are required to present progress and problems, share resources, and improve their presentation skills. A final presentation and report is the culmination of their work throughout the semester. As another course learning objective, collaboration is strongly encouraged so that students share their thoughts, methods, and tools within and among teams during learning.

3.3 Data and Computational Resources

3.3.1 Data Resources

Regarding event summarization, we selected 11 events from our project archives and created a small (about 500 Web pages) and a big (about 10000 Web pages)

Table 2
Characteristics of the 11 Event-related Collections and Their Corresponding Teams

Team ID	# in Team	Event-related Collection	Small Size	Big Size	Event Type
1	5	Hurricane Harvey	448	12789	Hurricane
2	4	Hurricane Matthew	495	12004	Hurricane
3	5	Attack Westminster	486	12063	Attack
4	4	Earthquake New Zealand	503	11856	Earthquake
5	5	Shooting Douglas	510	10108	Shooting
6	5	NeverAgain	499	12355	Shooting
7	5	NoDAPL	478	12323	Movement
8	5	Hurricane Irma	480	15305	Hurricane
9	4	Hurricane Florence	493	11375	Hurricane
10	5	Facebook Breach	478	10829	Breach
11	5	Shooting Maryland	505	12373	Shooting

collection for each event (see Table 2). Each collection of an event is in the Internet Archive's Web ARChive (ARC) file (WARC) format, along with a compound index (CDX) file. Small collections are used for prototype testing, while students generate summaries from big collections for evaluation. Besides the unlabeled data set above, we downloaded NEWSROOM (Grusky, Naaman, & Artzi, 2018), a large labeled data set for training and evaluating summarization systems, as an aid to event summarization. Regarding ETD summarization, in partnership with the University Libraries, we prepared a corpus of 13071 doctoral dissertations and 17890 master's theses downloaded from the VTechWorks institutional repository system in the University Libraries at Virginia Tech. This multidisciplinary document corpus reflects the large array of academic fields of study offered at Virginia Tech, and the collection has come to represent a rich and important body of graduate research and scholarship. Each ETD contains a main thesis document, almost always encoded in the portable document format (PDF), as well as bibliographic metadata. Some ETDs include a full-text version of the main thesis extracted from the PDF using optical character recognition (OCR). A moderate number of ETDs also include various supplementary files.

3.3.2 Hardware Resources

We provided four computing platforms in the PBL course, including local machines, a Hadoop cluster, and two advanced research computing (ARC) servers (i.e., Cascades and Huckleberry). With a set of powerful graphics processing units (GPUs), Cascades and Huckleberry servers are good at deep learning-based tasks. However, prototype testing can be carried out on students' laptops or desktops. Moreover, the MapReduce framework can support distributed computing tasks in our 20-node Hadoop cluster. Table 3 shows the specifications of the servers provided for our PBL course.

3.3.3 Software Resources

To help students get familiar with NLP and its related content, we shared a set of libraries and platforms that are useful for text summarization. Several Python libraries, such as NLTK (Bird et al., 2009), TextBlob (Loria et al., 2014), spaCy (Honnibal & Montani, 2017), and Gensim (Řehůřek & Sojka, 2011), can be used in linguistic analysis and clustering. MLlib (Meng et al., 2016), for efficient machine learning, and the ArchiveSpark (Holzmann, Goel, & Anand, 2016) library and extension of Apache Spark (2019), each facilitate processing on the Hadoop cluster. Regarding cloud computing and deep learning, we held a guest lecture and instructed students on how to install both TensorFlow (Abadi et al., 2016) and PyTorch (Paszke et al., 2019) and on how to run their code on the two ARC platforms.

Table 3
Specifications of the Public Servers Provided for Our PBL Course

	Hadoop Cluster	Cascades Server	Huckleberry Server
OS	CentOS 6.9	CentOS 7.4	Ubuntu 16.04
Nodes	1 Headnode	K80 GPU Engine: 4 nodes	2 Login Nodes
	19 Datanode	V100 GPU Engine: 40 nodes	14 Compute Nodes
Each Node	[Headnode]	[K80 GPU Engine]	[Compute Node]
	CPU: Intel E5-2630	CPU: 2 x Intel E5-2683	CPU: 2 x IBM Power8
	RAM: 128GB	RAM: 512GB	RAM: 256GB
	GPU: None	GPU: 2-NVIDIA K80	GPU: 4-NVIDIA P100
	[Datanode]	[V100 GPU Engine]	
	CPU: Intel i5-4570	CPU: 2 x Intel Xeon 6136	
	RAM: 32GB	RAM: 768GB	
	GPU: None	GPU: 2-NVIDIA V100	

Notes: OS: Operating System; CPU: Central processing unit; RAM: Random-access memory; GPU: Graphics processing unit

3.4 Team Learning and Collaboration

There are 67 undergraduate and graduate students interested in our PBL course, where 52 students are interested in event summarization, while 15 students, most of whom are graduate students, prefer to work on ETD summarization. According to the number of events, we divided the 52 students into 11 teams (from Team 1 to Team 11). Each team has four or five members and needs to summarize one event-related collection. Further, we also make sure that each team has at least one graduate student who is responsible for advanced processing (e.g., deep learning). Table 2 shows the teams and their corresponding event-related collections. We also make a relatively even split of students in the three ETD teams (i.e., Team 12, Team 13, and Team 14).

4 Methods Used

In this section, we describe a general pipeline that we proposed as a high-level guide for all teams in the text summarization task (Figure 1). Regarding event summarization, all teams first had to do some preprocessing work to clean their corresponding event-related collections. Next, they are required to index the data for browsing, sharing, and gold standard summary generation. Then, they can apply various methods such as topic modeling, clustering, or classification to filter out nonrelevant Web pages and enhance data quality. Finally, they can utilize different state-of-the-art text summarization (e.g., extractive or abstractive summarization) techniques or design their own approaches to produce summaries of those collections. The deep learning stage can typically be performed in the cloud (e.g., with ARC servers), giving students exposure to modern techniques. Regarding (single-document) ETD summarization, we simplify the pipeline. Thus, in Figure 1, the underlined phrases represent the key stages for ETD summarization, including preprocessing, gold standard summary generation, and abstractive summarization.

Our reason for the pipeline is to introduce the students to core principles of NLP. Moreover, we strongly encourage teams to develop or seek other approaches for better performance. We will describe each stage within the general pipeline in detail (Sections 4.1–4.4), along with selected approaches from several student teams.

4.1 Preprocessing

4.1.1 Event Collections

As mentioned above, each event has two types of files: a data file (i.e., *.warc.gz) and an index file (i.e., *.cdx). We provided a Scala script built on ArchiveSpark (Holzmann et al., 2016) so that event teams can read the two files and extract Web page payloads for further processing. By sharing the Web archive files and our code, we expect students who are interested in Web archiving and big data

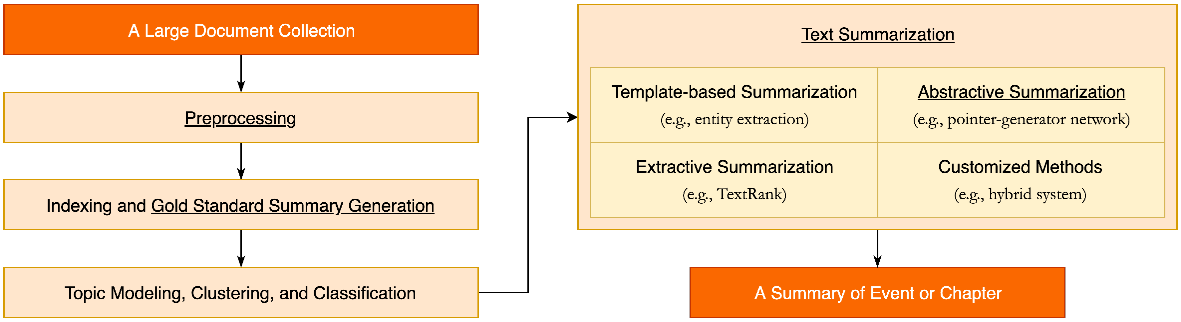

A Large Document Collection
Preprocessing
Indexing and Gold Standard Summary Generation
Topic Modeling, Clustering, and Classification
Text Summarization
Template-based Summarization
(e.g., entity extraction)
Abstractive Summarization
(e.g., pointer-generator network)
Extractive Summarization
(e.g., TextRank)
Customized Methods
(e.g., hybrid system)
A Summary of Event or Chapter

Figure 1. A general pipeline of event summarization in our PBL course

Table 4

An Example of Raw HTML Text and Clean Sentences in Attack Westminster from Team 3

(a) Before preprocessing	(b) After preprocessing
<body class=“feature-post”> <a data-buzzblock=“back-to-top” data-module=“back-to-top” href=“#top” class=“back-to-top xs-hide md-block xs-border-lighter xs-fixed xs-p1 xs-b2 xs-r6 xs-z2 circle” data-type=“to-top” data-parent=“window” data-content=“.buzz-title” data-padding=“”> <svg role=“img” aria-label=“back to top” viewbox=“0 0 22 22” class=“back-to-top__link-icon”>	Masood, born Adrian Russell Ajao, was named as the attacker. He was among those who died near parliament on March 22. He was among those who died near parliament on Wednesday. Your Image was too big. Something went wrong. Post has not been vetted or endorsed by BuzzFeed’s editorial staff. He was among those who died near parliament on March 22. Three civilians, Aysha Frade, Kurt Cochran, Leslie Rhodes, and a police officer, Keith Palmer, have been identified as his victims. 27…

to learn more about the accessible WARC format, Apache Spark, and parallel processing.

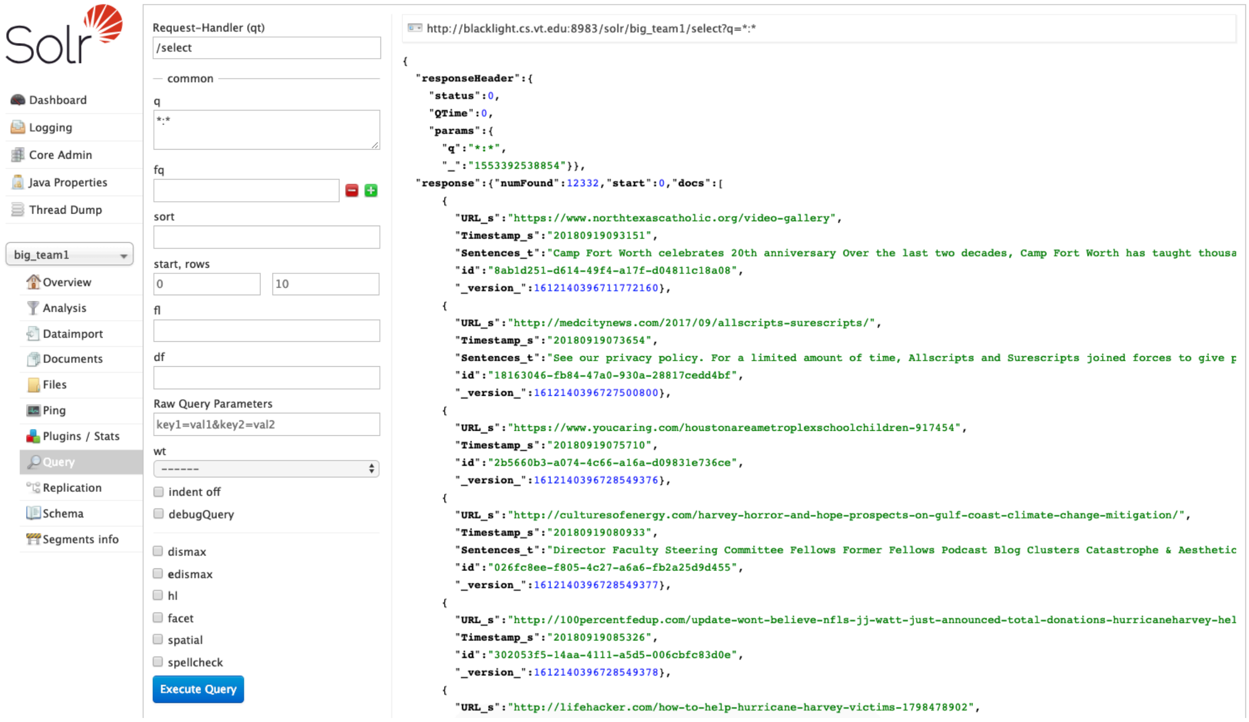
The payloads extracted from the WARC files are raw hypertext markup language (HTML) pages that need further processing (e.g., removal of unusable content). The expectation for this part of the PBL course was that the students would try and share some methods either of their own creation or from open source projects. In general, the teams discovered and made use of jusText (Pomikálek, 2011), a Python package, to filter the Web pages in their data sets. The groups wrote Python scripts incorporating the jusText package to parse HTML and determine whether the text in <p> tags was relevant text or boilerplate. Some teams modified the original Scala script and filtered out error pages using response code (e.g., 403, 404, and 408) or text (e.g., Access Denied). Table 4 shows an example of raw HTML text and clean sentences in the *Attack Westminster* collection from Team 3.

Further, the students removed pages generated by jusText that contained no English text, e.g., pages in a foreign language or redirected pages with links and no useful text. In some cases, as with Team 10, it was necessary to transfer their data sets from American Standard Code for Information Interchange (ASCII) to Unicode, removing hex characters. Some teams had data sets that jusText did not clean sufficiently. Team 4, e.g., made use of Apache OpenNLP (Baldridge, 2005), which solved most of the issues with punctuation and boilerplate, and then built a word list to ensure that documents contained the most frequent words in their data set.

As a result, most teams discarded approximately 33%–67% of the Web pages in their data sets by following the above steps. Even after cleaning their data sets, Team 9 had issues with image captions appearing in their summaries. It was necessary to manually remove image captions due to these issues.

4.1.2 ETD Collections

Before the ETD teams could begin the main task of creating summaries from chapters, they first needed to devise a process for segmenting the documents and extracting text from individual chapters. Initial attempts using the OCR text files were not successful. The trouble with OCR-extracted text is that running headers, tabular data, image/figure captions, and page numbers are jumbled together with paragraph text. So, the students focused their efforts



Request-Handler (qt)
/select

common

q
:

fq

sort

start, rows
0 10

fl

df

Raw Query Parameters
key1=val1&key2=val2

wt

□ indent off
□ debugQuery

□ dismax
□ edismax
□ hl
□ facet
□ spatial
□ spellcheck

http://blacklight.cs.vt.edu:8983/solr/big_team1/select?q=*:*

```
{
  "responseHeader":{
    "status":0,
    "QTime":0,
    "params":{
      "q":"*:*",
      "_":"1553392538854"}},
  "response":{"numFound":12332,"start":0,"docs":[
      {
        "URL_s":"https://www.northtexascatholic.org/video-gallery",
        "Timestamp_s":"20180919093151",
        "Sentences_t":"Camp Fort Worth celebrates 20th anniversary Over the last two decades, Camp Fort Worth has taught thousa
        "id":"8ab1d251-d614-49f4-a17f-d04811c18a08",
        "_version_":1612140396711772160},
      {
        "URL_s":"http://medcitynews.com/2017/09/allscripts-surescripts/",
        "Timestamp_s":"20180919073654",
        "Sentences_t":"See our privacy policy. For a limited amount of time, Allscripts and Surescripts joined forces to give p
        "id":"18163046-fb84-47a0-930a-28817cedd4bf",
        "_version_":1612140396727500800},
      {
        "URL_s":"https://www.youcaring.com/houstonareametroplexschoolchildren-917454",
        "Timestamp_s":"20180919075710",
        "id":"2b5660b3-a074-4c66-a16a-d09831e736ce",
        "_version_":1612140396728549376},
      {
        "URL_s":"http://culturesofenergy.com/harvey-horror-and-hope-prospects-on-gulf-coast-climate-change-mitigation/",
        "Timestamp_s":"20180919080933",
        "Sentences_t":"Director Faculty Steering Committee Fellows Former Fellows Podcast Blog Clusters Catastrophe & Aesthetic
        "id":"026fc8ee-f805-4c27-a6a6-fb2a25d9d455",
        "_version_":1612140396728549377},
      {
        "URL_s":"http://100percentfedup.com/update-wont-believe-nfls-jj-watt-just-announced-total-donations-hurricaneharvey-hel
        "Timestamp_s":"20180919085326",
        "id":"302053f5-14aa-4111-a5d5-006cbfc83d0e",
        "_version_":1612140396728549378},
      {
        "URL_s":"http://lifehacker.com/how-to-help-hurricane-harvey-victims-1798478902",
```

Figure 2. The event-related collection (i.e., Hurricane Harvey) created by Team 1 on Solr

on segmenting and extracting chapter text directly from PDF files, eschewing the OCR files altogether.

Students experimented with using two state-of-the-art scholarly PDF data extraction tools, Grobid (Lopez, 2009) and Science Parse (AI2, 2019). Both operate by applying conditional random fields to automatic text extraction of bibliographic data and document segmentation, following the approach of Peng and McCallum (2006). They both attempt to convert PDFs into eXtensible Markup Language (XML) or JavaScript Object Notation (JSON) files. Grobid marks up its output using the Text Encoding Initiative (TEI) (Sperberg-McQueen & Bernard, 1990) extensible XML schema, and Science Parse structures its output as JSON. As a result, the teams were able to extract individual chapters and strip away citations, notes, tables, figures, captions, and extraneous text. Finally, the teams used various Python modules from NLTK (Bird et al., 2009) to clean up the text and fix punctuation errors.

4.2 Indexing and Gold Standard Summary Generation

After preprocessing, we held a guest lecture about Apache Solr (2019), and all teams were required to import their data sets into a prebuilt Solr server. The goal is to help students get familiar with search engines and provide searchable data for the gold standard summary generation across teams. Similar to the WARC conversion task, we shared a tutorial on how to index data with Solr, along with an executable script for immediate use and further study. Figure 2 illustrates the event-related collection (i.e., *Hurricane Harvey*) created by Team 1.

With the help of Solr, all large document collections can be managed, read, and shared easily among teams. To aid further evaluation, we set up a cross-labeling task for event teams, so each team prepared a summary used to evaluate the work of another team. Specifically, each team generated a gold standard summary by the following steps: (1) constructing a template to cover the main details of an event; (2) browsing Wikipedia pages and external links of the event entry; (3) searching and browsing relevant Web pages through Solr; (4) generating a summary based on the above resources; and (5) receiving feedback from the instructor and refining the gold standard summary.

Also, we invited a guest lecturer from the Department of Communication at Virginia Tech to speak to the students on the critical factors of journalism, how to report events, and how journalists construct news articles, as well as to answer questions from teams. Through the above instructions and the guest lecture, each team completed the cross-labeling task and generated one summary for another team. These gold standard summaries were compared with the summaries generated by the teams

Table 5
Partial Gold Standard Summaries Generated by the Event Teams through Manual Labeling

Team ID	Gold Standard Summary
1	Hurricane Harvey initially developed in the Atlantic Ocean, to the east of the Wind ward Islands, making landfall in Barbados and Saint Vincent, and then moved into the Caribbean Sea. It achieved tropical storm status between August 17th and 19th 2017. It crossed the Yucatan Peninsula and then intensified over the Gulf of Mexico...
2	On September 22, 2016 a mass of thunderstorms formed off Africa. On September 28, moving away from the Antilles (where it caused extensive damage to landmasses), with sustained winds of 60 mph, the storm gained tropical cyclone status when southeast of St. Lucia. It became Hurricane Matthew on September 29, ...
3	On Wednesday March 22nd 2017, an attack took place at the Houses of Parliament in London. A car was driven at high speed across Westminster Bridge - hitting victims as it went. The horror began at the south London end of Westminster Bridge when a driver in a Hyundai Tucson jumped onto the pavement at around 2.30pm, ...
4	On Tuesday February 22, 2011 at 12:51 pm, an earthquake of magnitude 6.3 hit Christchurch, New Zealand, a South Island city of nearly 400,000 people. The country‘s deadliest natural disaster in 80 years killed 185 people from more than 20 countries and injured several thousand, 164 seriously. The earthquake epicenter was near Lyttelton, ...
5	On February 14, 2018, gunman Nikolas Cruz killed seventeen people at Marjory Stoneman Douglas High School in Parkland, Florida. Fourteen students and three administrators were killed in the shooting. Seventeen people were also injured non-fatally in the shooting. Cruz began the attack at 2:21 p.m. and left the premises at 2:28 p.m ...
6	This collection is about the NeverAgain student group. NeverAgain (also known by the Twitter hashtags #NeverAgain and #EnoughIsEnough; see also NeverAgainMSD and neveragain.com) is a U.S. political action committee that promotes tighter regulation of guns to prevent gun violence ...
7	The Dakota Access Pipeline, also known as DAPL, is a $3.8 Billion construction program that began in July 2014. It‘s a 1,172-mile-long (1,886 km) underground oil pipeline project that starts in the Bakken shale oil fields in northwest North Dakota, passes through South Dakota and Iowa, and ends in Illinois at the oil tank farm near Patoka ...
8	Hurricane Irma was a Category 5 Atlantic hurricane, the most powerful in history, with winds of 185 mph for 37 hours, longer than any storm recorded. Tropical force winds extended 185 miles from the center. Irma held 7 trillion watts of energy. Storm surges brought waves 20 feet higher than normal ...
9	The National Hurricane Center identified a potential tropical storm in the eastern Atlantic Ocean with a wind speed of around 30 mph on August 30, 2018. It originated near Cape Verde, off the coast of West Africa. This became a tropical storm named Florence on September 1. It developed into a Category 2 Hurricane on September 4 ...
10	In 2014 Facebook authorized a Russian/American researcher named Aleksandr Kogan to view information about people who used his personality quiz app “thisisyourdigitallife”. This data was to be used for research and should have only given him information on the 270,000 people who agreed to download and use his app ...
11	*Event 1*: On Tuesday, June 28, 2018 at 2:33pm, 38 year-old Jarrod Warren Ramos opened fire on the glass front door of the Capital Gazette newsroom in Annapolis, MD. Of the 11 employees there that day, 5 were killed: ... *Event 2*: On Tuesday, March 20, 2018 at 7:57 a.m, 16 year-old Jaelynn Willey was shot in the halls of Great Mills High School in Great Mills, Maryland. The bullet that hit Willey also struck 14-year-old Desmond Barnes in the leg ...

during evaluation; see details in Section 5. Table 5 lists the first few sentences from the gold standard summary of each event team.

Regarding the ETD teams, they manually created gold standard summaries at the chapter level. Each member chose one thesis and one dissertation from which to create his or her summaries. For each document chosen, the student would (1) manually extract the text for each chapter; and (2) carefully read each chapter and write a coherent overview, including interesting details from the chapter. Students were instructed to (1) make sure that each summary stands on its own and (2) provide enough context so that researchers would understand the overall topic of the thesis if they found it independently. As a result, the ETD teams generated about 150 gold standard chapter summaries from about 30 ETDs; these were all edited by the instructor to ensure quality and consistency. Table 6 lists the first few sentences from one gold standard summary from each ETD team. A key difference in the gold standard summaries created for the ETD task is that each summary is created from a single source chapter, whereas the event summaries are created from multiple documents. The other key difference is the large amount of domain-specific jargon used in the ETD chapter summaries.

Table 6
Partial Gold Standard Summaries Generated by ETD Teams through Manual Labeling

Team ID	Gold Standard Summary
1	The two-point correlation structure and turbulence statistics of a cylinder wake are studied in order to develop accurate prediction methods for an open rotor ingesting turbulence. Understanding wake flow is necessary for understanding the noise produced by a wake generator. Proper Orthogonal Decomposition is used to determine ...
2	Experimental setups for optimization and parametric studies include the AGMD module and superhydrophobic surface. An air-cooled air gap membrane distillation system works well, with lower energy requirements, due to its modular design. The conductivity of the support mesh is an important factor in flux values, with copper mesh giving good ...
3	The chapter proposes different steps to improve Spectral Clustering algorithm but the method of using the sub-sampling method is proved better. The new algorithm splits the original graph into subgraphs by maintaining a certain number of nodes as common in all the subgraphs. After performing spectral clustering over each of these subgraphs, community ...

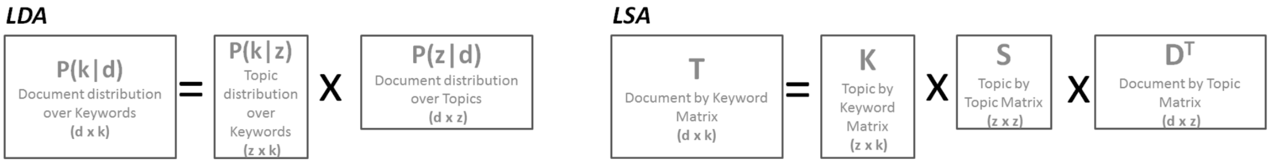


Figure 3. Matrix decomposition for LDA and LSA

4.3 Topic Modeling, Clustering, and Classification

As an MDS task, event summarization requires further processing. A good summary should cover various aspects (e.g., background, human or social reaction, and aftermath) of an event. Unfortunately, each data set has thousands of Web pages after cleaning; automatic summarization of such collections remains a challenging problem. Accordingly, we encouraged the students to learn and implement topic modeling, clustering, and classification techniques.

We noticed that event teams were divided into two major groups: one preferred using the (binary) bag-of-words model, while the other decided to use the (weighted) vector space model. LDA (Blei et al., 2003) using Gensim (Řehůřek & Sojka, 2011) was the most popular approach for topic modeling. Team 2, e.g., noted that choosing LDA required them to learn and understand the bag-of-words model, the *id2word* function, and the details of the LDA algorithm. Further, teams such as Team 10 utilized alternative methods such as latent semantic analysis (LSA) (Deerwester et al., 1990) when they found that LDA did not work well on their data sets. Team 10 reported that LSA yielded much better topics for their *Facebook Breach* data set than LDA. Figure 3 shows the matrix decompositions for LDA and LSA.

In addition to LDA and LSA, some teams worked on vector space models, carrying out similarity computations and applying scikit-learn (Pedregosa et al., 2011), a simple and efficient machine learning library, for clustering. Further evaluation of clustering was carried out by a few teams. Specifically, Team 10 used multiple clustering algorithms to determine which worked best for their data set. They compared K-means (Lloyd, 1982) and various hierarchical clustering approaches (e.g., single linkage, average linkage, and complete linkage) to evaluate how the distance function, the number of topics, and the number of clusters affected each clustering algorithm. The team settled on K-means after tuning the parameters of the model during a large part of the course.

Teams did not use classification as prominently as clustering. However, with classification, Team 9 was able to create sizable collections that were directly relevant to their event. Through feature engineering, they determined which articles were most relevant to their event. They made use of important words, bigrams, and synsets to determine whether articles contained the most important information in their event-related collection. We discuss this approach further in Section 5.4.

Meanwhile, many teams noted the difficulties of topic modeling, clustering, or classification in their reports. Team 5 made use of the Doc2Vec model (Le & Mikolov, 2014) followed by the K-means method, and described how they found segmenting the document space to be the most challenging part of the project. Team 8 detailed how they spent a significant amount of their time trying complicated methods like K-means and LDA, while their

Table 7
A Summary Template Related to Westminster Attack Created by Team 3

On ***DATE*** a ***TYPE_OF_ATTACK*** occurred in ***LOCATION*** near ***LOCATION***. The police investigated that the attacker was ***ATTACKER***. During the ***TYPE_OF_ATTACK*** the attacker killed ***NUM_OF_PEOPLE*** people and injured ***NUM_OF_PEOPLE***.

final approach, a simple word filter, outperformed each complicated process that they tried. This situation was mainly due to their event, *Hurricane Irma*, overlapping in time with *Hurricane Harvey*; that complicated their data set analysis.

4.4 Text Summarization

To begin, we hosted a guest lecture about text summarization that described the major types of the task and a set of state-of-the-art approaches. Afterward, different teams applied a multitude of methods to summarize their corresponding event collections or ETD chapters.

4.4.1 Event Summarization

First, as a basic course-learning unit, all event teams applied template-based summarization. They created templates without knowledge of the gold standard summaries; extracted key factors such as dates, locations, and victims from Web pages by applying NER with libraries like spaCy; and filled template slots with the extracted terms. As an example, Table 7 shows the summary template created by Team 3 about the *Westminster Attack*.

Second, some teams began with extractive summarization techniques such as TextRank (Mihalcea & Tarau, 2004) and sentence extraction. Regarding the MDS task, they designed and implemented two types of approaches. Some teams merged all Web pages about a topic into a single, lengthy document, while other teams proposed some hierarchical strategies to solve the problem. Specifically, they first applied existing tools to create a summary for each Web page. Then, they concatenated these summaries as a new document and reused the above-mentioned tools to generate the final output.

Third, regarding abstractive summarization, most teams applied deep learning based techniques to solve the problem. Their widely used approach is the PGN (See et al., 2017); see Figure 4. Due to the time limits involved, many teams made use of a pretrained model of PGN, which is accessible from the GitHub page of PGN. A few teams, such as Team 8, did not utilize the pretrained model but trained the network on their data set to generate better results.

Some teams, such as Team 5 and Team 6, focused exclusively on deep learning techniques. Subteams were formed, so that the team as a whole could deploy different deep learning models such as the seq2seq model (Sutskever et al., 2014), the PGN, and the reinforced extractor-abstractor network (REAN) (Chen & Bansal, 2018) (shown in Figure 5).

4.4.2 ETD Summarization

The three ETD teams compared the performance of the different deep learning techniques listed above, namely, seq2seq, PGN, and REAN, for generating abstractive summaries. These models require a large set of labeled training data to produce satisfactory results, but no such training set exists for ETD chapters. To overcome this significant limitation, each of the ETD teams implemented various transfer-learning techniques, in which a base language model trained on a different large data set is used to generate summaries for ETD chapters. Two models were trained using news article data sets – one originating from British Broadcasting Corporation (BBC) News (Greene & Cunningham, 2006) and the other from Cable News Network (CNN)/Daily Mail (Hermann et al., 2015) – each containing thousands of article–summary pairs. Another model was trained on a data set of scientific papers downloaded from the arXiv.org e-Print archive. Article–summary pairs were made by using a paper's abstract section as the training summary and the remaining content as training input. A fourth model was trained with a data set of Wikipedia articles, using article abstracts as training summaries and article content as training input.

As previously mentioned, a key difficulty with ETD summarization is the large amount of domain-specific jargon used in ETD chapters. To ensure more overlap in the vocabularies of the training and test data, one team attempted to create a smaller subset of the Wikipedia data, which only included articles that were related to documents in the ETD corpus. To create this data set, students constructed a neural network language model of the corpus in order to compute a similarity score for relating the text of Wikipedia articles with the ETD corpus. Only Wikipedia articles with a high similarity score relative to the ETD corpus were included in this smaller training data set.

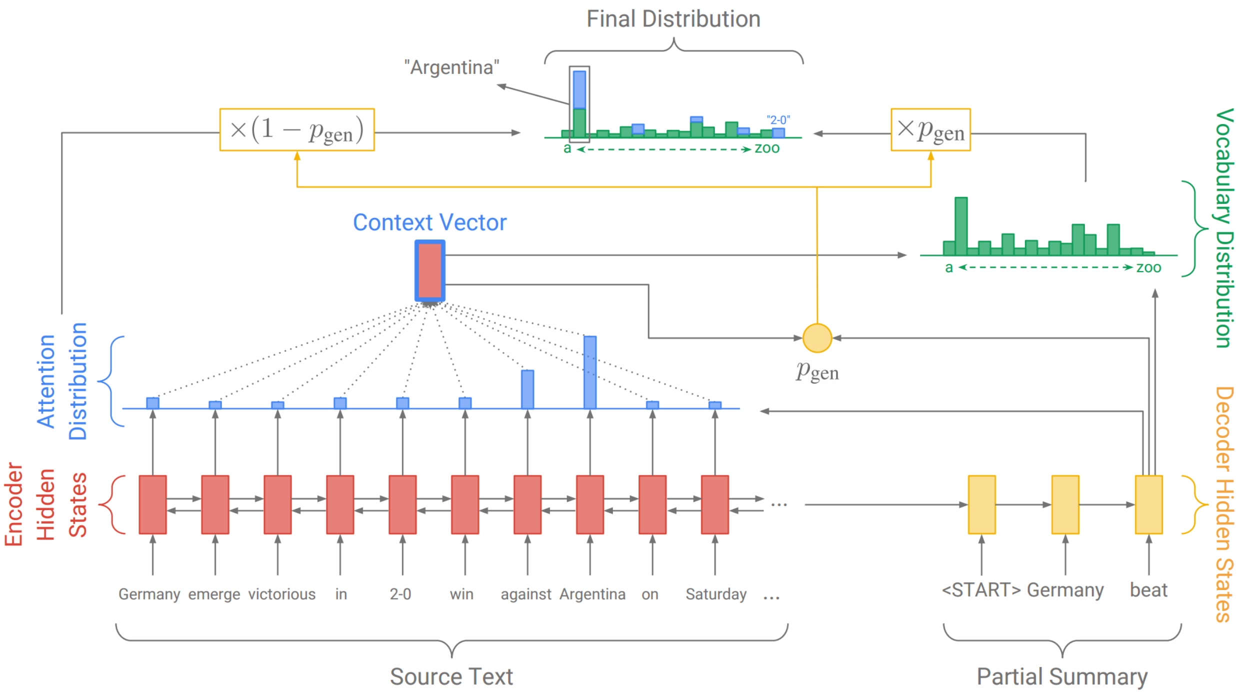


Figure 4. Architecture of the pointer-generator network (PGN) (See et al., 2017)

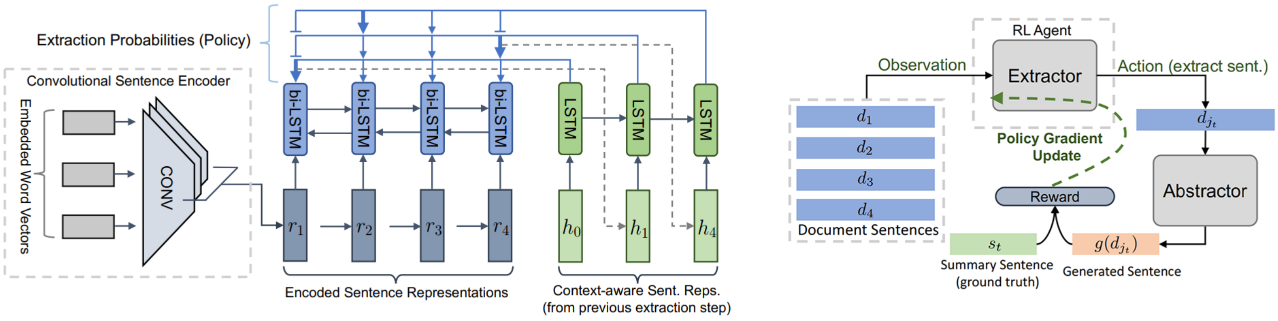


Figure 5. Architecture of the reinforced extractor–abstractor network (REAN) (Chen & Bansal, 2018)

Additionally, the ETD teams experimented with training deep learning models on a combination of data sets (e.g., CNN/Daily Mail + Wikipedia) to see whether these models would perform better than those trained on a single source.

5 Evaluation of Summarization

In this section, we describe a comprehensive evaluation of summarization, especially of the abstractive summaries generated by the teams. We first propose our metrics for text summarization (Section 5.1). Then we present both quantitative and human evaluation results from all teams (Sections 5.2 and 5.3). Finally, we describe two of the best solutions from the event teams: the first is built on self-customization, while the other is built on existing techniques and achieved the best quantitative scores (Section 5.4).

5.1 Metrics

We used Recall-Oriented Understudy for Gisting Evaluation (ROUGE) scores (Lin & Hovy, 2002) to compare the generated summaries with the gold standard

Table 8

*Quantitative Evaluation of the Best Summaries Generated by the Event Teams**

Team ID	Main Solution	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-SU4
1	PGN + MMR	0.034	0.000	0.034	0.006
2	Template summary	0.161	0.033	0.064	0.035
3	LSA + PGN	0.074	0.000	0.074	0.014
4	PGN + K-means	0.250	0.029	0.167	0.070
5	Doc2Vec + K-means + REAN	0.091	0.031	0.091	0.022
6	LDA + PGN	0.115	0.040	0.115	0.021
7	Customized hybrid system	0.189	0.055	0.135	0.053
8	RBC + K-means + PGN	0.222	0.029	0.139	0.006
9	MLP + LDA + PGN	**0.267**	0.069	**0.267**	**0.085**
10	LSA + K-means + PGN	0.115	0.000	0.115	0.021
11	K-means + PGN	0.218	**0.129**	0.187	**0.085**

*Because each event team applied its proposed model on a specific event, it is not suitable to conduct significance tests. In the future, we plan to select one event-related data set and evaluate all their models, including a significance test.

summaries. The recall score of ROUGE metrics is used in most cases in text summarization; Equation 1 shows how to calculate the value.

$$ROUGE_{recall} = \frac{\#\ of\ overlapping\ words}{\#\ of\ words\ in\ gold\ standard\ summary} \quad (1)$$

Regarding the overlapping words, there are four popular types: (1) ROUGE-1 refers to the overlap of unigrams; (2) ROUGE-2 refers to the overlap of bigrams; (3) ROUGE-L refers to the longest matching sequence of words using the longest common subsequence (LCS); and (4) ROUGE-SU4 is a bigram measure that enables, at the most, four unigrams inside bigram components to be skipped.

5.2 Quantitative Evaluation

5.2.1 Event Summarization Results

Table 8 lists the various approaches developed by the different event teams and their corresponding ROUGE scores. In addition to the techniques mentioned in Section 4, we noticed that some teams applied other methods during implementation, including a rule-based classifier (RBF) or a multilayer perceptron (MLP) classifier for relevance judgment, and the pointer-generator with maximal marginal relevance (PG-MMR) developed by Lebanoff, Song, & Liu (2018) for summarization.

Regarding the evaluation results, Team 9 achieved the best ROUGE-1 (0.267), ROUGE-L (0.267), and ROUGE-SU4 (0.085) scores among all teams, while Team 11 performed the best in ROUGE-2 (0.129) and ROUGE-SU4 (0.085). Focusing on ROUGE-1, we discovered that four out of 11 teams had scores >0.2, which represented good performance on text summarization. Besides these teams, the ROUGE-1 score of Team 7, which developed their hybrid system for abstractive summarization, is 0.189, which is promising for further exploration and improvement. Team 2 also has a relatively high ROUGE-1 score. Unfortunately, they did not complete the abstractive summarization task but used a template-based summary during evaluation.

After looking into the students' approaches and comparing the ROUGE scores among different teams, we made several interesting observations. First, teams having better performance used event topics as good indicators in summarization. Meanwhile, Team 1, Team 3, and Team 5 achieved low ROUGE-1 scores. Team 1 did not apply any topic modeling or clustering techniques. Though Team 3 applied LSA for topic modeling, they only chose three relevant topics that did not cover enough to describe an event comprehensively. Furthermore, Team 5 used Doc2Vec to represent a news article and K-means for clustering. However, detailed information might be lost

Table 9
Quantitative Evaluation of the Best Summaries Generated by the ETD Teams

Team ID	Model + Data set	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-SU4
12	REAN + CNN/Daily Mail + arXiv	0.139	**0.122**	–	0.048
13	PGN + CNN/Daily Mail	**0.238**	0.097	**0.213**	–
14	PGN + arXiv	0.222	0.038	0.198	–

during conversion to Doc2Vec. Second, summarization at the sentence level seems to be a proper approach in our specific scenario. Some teams like Team 9 and Team 11 implemented clustering at the sentence level and had better results. Team 7 focused on ranking abstractive sentences generated by the PGN, and extractive sentences extracted from each topic, for summarization. Teams working at the article level sometimes had difficulty with topic modeling and clustering since each document reports an event from various aspects. Such a problem may have a negative effect on summarization. Third, a simple concatenation of all documents or a set of documents into one topic before abstractive summarization leads to poor performance as has been illustrated from the final results of Team 1 and Team 3 and the interim results of Team 10. Further, by comparing the two best approaches (i.e., Team 9 and Team 11), both of which worked with sentences, we found that Team 9 built a feature-based classifier and focused on event-related sentences, while Team 11 applied K-means clustering on all sentences. The results show that Team 9's approach performs better on ROUGE-1 and ROUGE-L, while Team 11's approach has a higher ROUGE-2 score.

5.2.2 ETD Summarization Results

Table 9 lists the best-performing approach developed by each of the three ETD teams and their corresponding ROUGE scores. The best-performing deep learning technique was PGN. The results also show that models trained on a single data set perform better than those trained on a combination of two or more training sets. However, despite the superior vocabulary coverage of the Wikipedia training data, models trained on Wikipedia did not perform as well as those trained on CNN/Daily Mail. The best Wikipedia-trained model only achieved a ROUGE-1 score of 0.172 and a ROUGE-L score of 0.154. The best combination of model and training data set, PGN + CNN/Daily Mail, achieved a ROUGE-1 score of 0.238 and a ROUGE-L score of 0.213.

5.3 Human Evaluation

We evaluated selected summaries from the event teams and the ETD teams based on criteria found in a previous paper (Di Fabbrizio, Stent, & Gaizauskas, 2014). In short, four criteria were used to determine the quality of a summary: (1) readability; (2) correctness; (3) completeness; and (4) compactness.

5.3.1 Event Summarization Results

Table 10 lists parts of the summaries generated by the three selected event teams. In general, these summaries, as well as the sentences in each summary, are well structured and easy to read.

5.3.1.1 Readability

Each of the summaries is readable, with reasonably correct grammar and flow. Furthermore, the summaries make use of complicated sentence structure with multiple clauses in some sentences. Team 7's summary refers to a Web site correctly. The summaries are high-quality readable summaries that illustrate that the students achieved a good grasp of NLP through big data text summarization during our PBL course.

However, we also discovered that sentence sequence is a general problem among all teams, which is also a vital issue of MDS. Simple sentence concatenation generally does not preserve the contextual information and ordering expected in summaries. It also decreases the readability if the thoughts conveyed in the summary are not organized.

5.3.1.2 Correctness

Regarding the correctness of the summaries in Table 10, we found that the summaries took content from the news articles and were aligned factually with what was contained in those articles. For example, Team 11's summary accurately stated that Austin Rollins was the

Table 10
Representative Summaries Generated by the Three Selected Event Teams

Team 7	President Obama says the Dakota Access Pipeline will be delayed. Thank you for temporarily halting the Dakota Access. Thank you for temporarily halting the Dakota Access Pipeline. We know that President Elect Trump has a serious conflict of interest by owning large investments in DAPL and other fossil fuel assets; and his energy team includes Harold Hamm, billionaire founder of the oil company Continental Resources, and someone Mr. Trump might name as his secretary of energy. After the failed keystone pipeline, a new and virtually regulation-free Dakota Access Pipeline was approved. The Dakota Access Pipeline will be built on top of several burial grounds and sacred sites. Veterans of the United States Armed Forces, including the U.S. Army, U.S. Air Force, and U.S. Coast Guard, and we are calling for our fellow veterans to assemble as a peaceful, unarmed militia at the Standing Rock Indian Reservation on Dec 4–7....
Team 9	The National Hurricane Center predicts the storm will make landfall: some latest updates. The storm has increased in strength and speed from earlier updates on Monday. Florence will likely hit land as a Category 4 hurricane late Thursday evening before weakening again. More than 1.5 million people have already been ordered to evacuate coastal areas. Models have come into agreement that a northward turn before reaching the United States is unlikely and that a building high-pressure zone north of the storm will cause it to slow or stall once it reaches the coast or shortly thereafter. Where exactly the zone of heaviest rain will be is a big uncertainty....
Team 11	Austin Rollins, a student at Great Mills High School in St. Mary's County, MD, started firing in a hallway at the start of the school day. The shooting broke out just before classes were scheduled to start at Great Mills High, a 1500-student school 65 miles south of Washington, D.C. All of this occurred before calls were placed to 911 starting at 7:58 a.m., just before classes started. Tyriq Wheeler, 17, was headed to English class when he heard a bang. Isaiah Quarles, a 10th-grader, was walking to his first-period class on Tuesday. A student named Jonathan Freese called into CNN and said the shooting began early in the morning and seven people could possibly be hurt. The entire incident played out in less than a minute at 7:55 a.m. in a hallway at Great Mills, a school 90 miles south of Baltimore....

shooter in the *Great Mills High School Shooting*. One or multiple sentences also cover different aspects of a specific event. The summary from Team 11 presents the key factors in the *Maryland shooting* event, including date–time, location, shooter, and a shooting report from a student.

Though most of the *NoDAPL* (No Dakota Access Pipeline) summary generated by Team 7 is accurate, it also expresses opinions in the summary itself. This is due to the nature of news articles in the data set used for summarizing the event. Systems developed by the teams do not detect and therefore do not filter out opinions included in the data; so, opinions are inappropriately added to the summary. This shows a limitation of the current systems for MDS.

Some summaries contain information that is outdated or missing contextual information. For example, Team 9's summary mentions that *Hurricane Florence* will make landfall Thursday but does not describe what date this is or where it fits into the timeline. In a human-generated summary of hurricane events, one would expect a timeline of events to be accurate contextually concerning time.

5.3.1.3 Completeness

The summaries are not complete and only contain a few topics of the corresponding event. This is particularly true for the hurricane event summaries. *Hurricane Florence*, e.g., is not described using a timeline; so, there are correctness issues, as stated above. This problem also causes failures in completeness.

Further, the *NoDAPL* summary lacks a definition of the DAPL itself. Such contextual information would help readers understand the event, but it is not present in the summary.

5.3.1.4 Compactness

Additionally, compactness evaluates how concise a summary is. This metric indicates the capability to avoid long summaries. The three summaries presented in Table 10 contained 860, 296, and 463 words, respectively. Impressively, Team 9 had the highest ROUGE-1, -L, and -SU4 scores, with a quite compact summary.

Team 11 had a relatively compact summary, while Team 7 had a long drawn-out summary. Again, there were opinions present in Team 7's summary, which, if removed, would have made the summary more concise (and more correct).

5.3.2 ETD Summarization Results

Table 11 lists two summaries generated by Team 13 using two different training data sets. All of the three teams working with ETDs were able to generate abstractive summaries of chapter text using deep learning techniques. Accordingly, we chose to focus on a single team to give a brief overview of the results.

5.3.2.1 Readability

In general, we found that the ETD chapter summaries were harder to read and understand than the event summaries. We believe that the hardest part of the problem of creating coherent summaries from ETD chapters comes from the widespread use of domain-specific jargon in ETDs, coupled with the wide-ranging breadth of subject matter contained in the ETD corpus. However, the ETD summaries do contain complete, complex sentences, which aids in readability.

A major flaw in the Wikipedia model, shown in Table 11, is the repetition of the words “of the wake”. The summary becomes unreadable and incomprehensible due to these repetitions.

Table 11
Representative Summaries Generated by Team 13 through Transfer Learning (CNN/Daily Mail Model vs. Wikipedia Model)

Team 13
CNN/Daily Mail Model
This thesis describes an experimental study of the two-point correlation structure of a cylinder center. These four-dimensional space–time correlations of the wake at the rotor disk location, measured without the rotor installed, provide the complete linear inflow boundary condition to this problem. The upwash velocity is the velocity perpendicular to the blades of a rotor disk, and its correlation function directly depends. None...
Wikipedia Model
A circular cylinder is a type of wake flow in which the wake of a cylinder is used to measure the velocity fluctuations of an aircraft or aircraft. It is used to measure the velocity fluctuations of a wake flow in a wake of the wake of the wake of the wake of the wake of the wake of the wake of the cylinder wake. The wake of a wake flow is the wake of the wake of the wake of the wake of the wake of the cylinder wake. The cylinder wake can be caused by a wake of the wake of the wake of the wake of the cylinder wake....

5.3.2.2 Correctness

The CNN/Daily Mail model is fairly readable, but the correctness is questionable. The first sentence starts the summary reasonably, but the second sentence is a quick change to an unrelated concept without any definition or transition. The Wikipedia model results are incorrect and mostly incomprehensible.

5.3.2.3 Completeness

The vocabulary size for the model trained on CNN/Daily Mail data seems to be too small to create summaries with much substance. On the other hand, the Wikipedia models have better vocabulary coverage, but nonetheless, they fail to create good summaries. The ideas presented in both summaries are also not complete and are lacking in flow and structure, which one would expect from a human-generated summary.

5.3.2.4 Compactness

Each summary is compact and around the size of an abstract one would see in a journal paper. This is in line with the goals stated in the course.

5.4 Best Solutions

We identified the two event teams that developed the best summarization system, as indicated by (1) the uniqueness and quality of the system and (2) the ROUGE scores. Many teams used the PGN, which was state-of-the-art at the time of the course and produced the best results. Among those teams, we found that Team 7 and Team 9 provided the best systems.

5.4.1 Solution 1

The first solution we chose based on these metrics was of Team 7, which used LDA to perform topic modeling and then followed it with LDA-based sentence extraction. The team members then used the PGN model to generate a set of summaries of the articles in their data set. Following this, they incorporated an interesting hybrid methodology presented in Figure 6 to combine the abstractive and extractive sentences through LDA-based similarities and named-entity frequency. This proposed methodology overcomes two challenges. First, it allows for the extractive sentences to be ranked based on the named entities, which topic modeling does not incorporate. Second, since the PGN is used only on the original articles and not on the topics, it allows for the re-ranking of these sentences with respect to the LDA topics.

Team 7 also went beyond the other teams in their evaluation methodology. Its members devised a system for extrinsic subjective evaluation, which involved using a set of questions that should be answered by their summary. By evaluating each sentence in their summary,

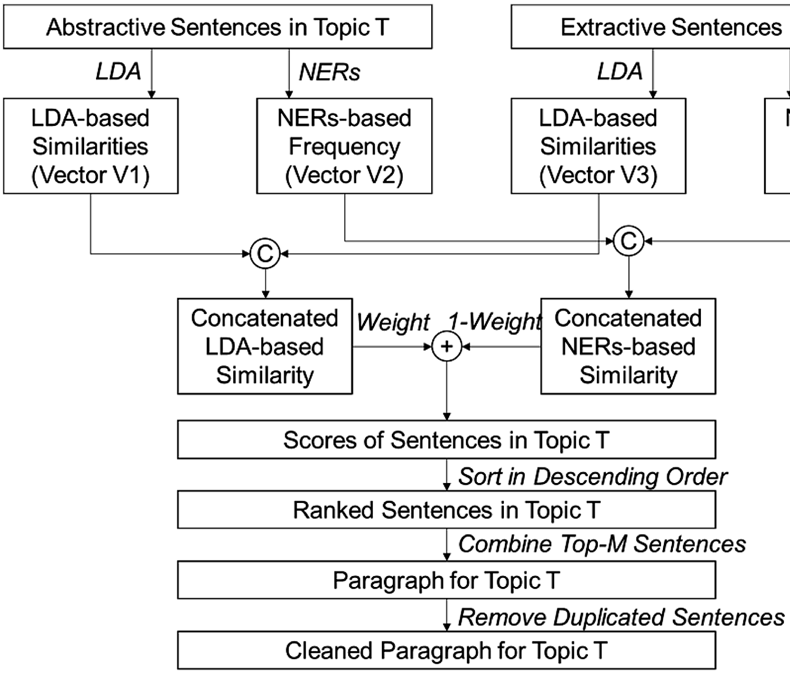

Abstractive Sentences in Topic T
Extractive Sentences in Topic T
LDA
NERs
LDA
NERs
LDA-based Similarities (Vector V1)
NERs-based Frequency (Vector V2)
LDA-based Similarities (Vector V3)
NERs-based Frequency (Vector V4)
C
C
Concatenated LDA-based Similarity
Weight
1-Weight
+
Concatenated NERs-based Similarity
Scores of Sentences in Topic T
Sort in Descending Order
Ranked Sentences in Topic T
Combine Top-M Sentences
Paragraph for Topic T
Remove Duplicated Sentences
Cleaned Paragraph for Topic T

Figure 6. A customized hybrid method developed by team 7 for the NoDAPL data set.

they found that 91.3% of the sentences had to do with the *NoDAPL* event, and that the summary answered 69.6% of the questions they wrote for the event.

5.4.2 Solution 2

The other solution we chose was from Team 9, which achieved the highest ROUGE-1, ROUGE-L, and ROUGE-SU4 scores, as shown in Table 8, but also had a unique methodology for summarizing *Hurricane Florence*. The team members first performed feature engineering to determine the relevance of the articles to the topic. They extracted the important words, frequently appearing bigrams, and synsets in the collection. If a bigram contained words that were in the important word list, they added the bigram to their set of features. They then added to the feature list the remaining words that did not appear in the bigrams. If any of these words that now were in the feature list were also synsets, they replaced the word with the synset. By following this process, they constructed a feature list that could be used to determine both the presence of a feature and its frequency in an article.

Using these features, they tested labeling methods for different thresholds on the number of features present in the news articles and extracted the most relevant sentences for their topic. They then performed LDA-based clustering on the records, chose the article closest to the center of the cluster, and summarized those articles with PGN (see Figure 7). Through testing, they found that 10 clusters worked best among the tests using 5, 10, and 15 clusters.

6 Evaluation on PBL

In this section, we used Student Perceptions of Teaching (SPOT), the university's centrally supported method for gathering student feedback on courses and instruction, to conduct both quantitative and qualitative evaluations on our PBL course (Sections 6.1 and 6.2).

6.1 Quantitative Evaluation

We selected five course evaluation items from the entire SPOT report, which are relatively close to PBL (see Table 12). Each item has a six-point scale, where one is the lowest point (i.e., strongly disagree) and six is the highest point (i.e., strongly agree) on the scale. Because this course is for both undergraduate and graduate students, we treated the two groups separately. We hypothesized that each group in our PBL course should have a higher feedback score, compared with the average score of all courses

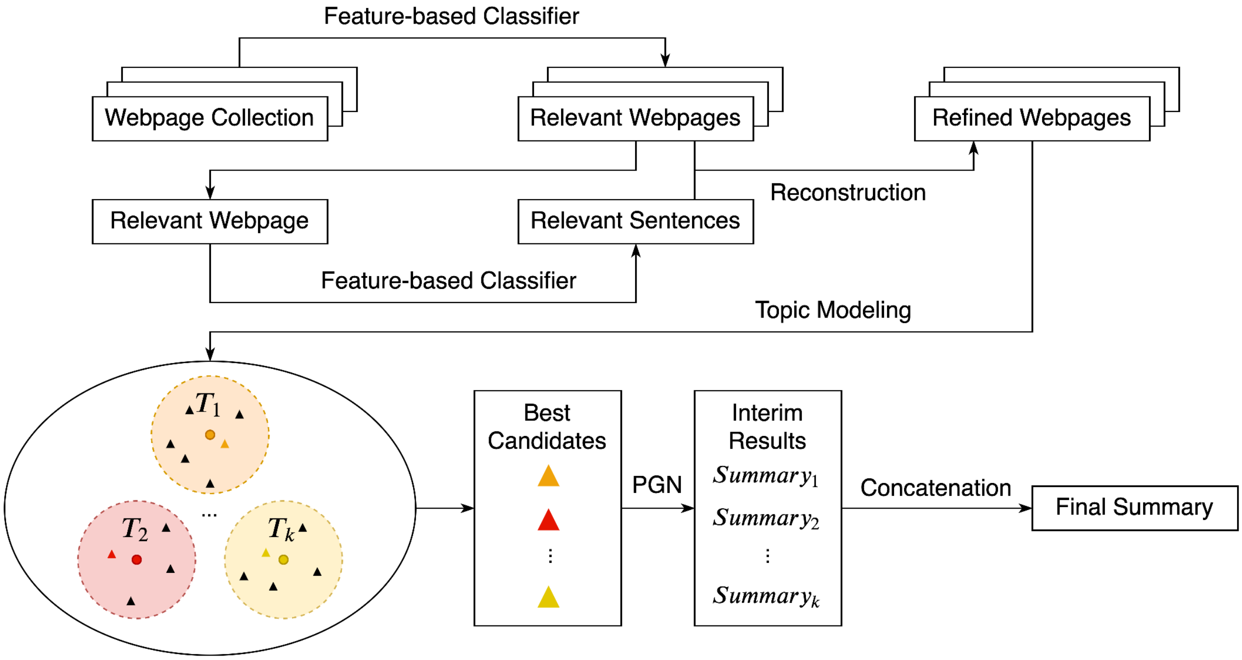


Figure 7. An integrated method developed by team 9 for the Hurricane Florence data set

in our department. We also checked whether there are some differences between undergraduate and graduate students regarding some evaluation items.

We gathered 42 student reports from SPOT. Of these, 11 reports came from undergraduate students, and 31 reports came from graduate students. Regarding our department, the number of participants from Item 1 to Item 6 varied: 2459, 2471, 2434, 2453, and 3749. Accordingly, Figure 8 shows the average score of each item among the undergraduate and graduate students in our PBL course, as well as among all students in the CS department. The lower bound of the Y-axis is set to 3.5, which is the mean of the six-point scale. We analyzed the results with a one-sided *t*-test (α = 0.05) and calculated the effect size to measure the differences among groups. The three sub-tables in Table 13, namely, Tables 13a, 13b, and 13c, list the pairwise comparison results, separately.

Table 13a shows the results of the difference test between the undergraduate students in our course (UGRD) and all students in our department (ALL). All *p*-values are <0.05, which demonstrates that the average score of UGRD is statistically significantly higher than the average score of ALL regarding each evaluation item in Figure 8, indicating that our course gave students a better experience through PBL. Especially, regarding Item 1, Item 4, and Item 5, the differences are taken to be large enough (i.e., effect size >0.8) to be significant.

Table 12
Five Course Evaluation Items Related to PBL

Item No.	Description
1	I improved my ability to problem solve.
2	My interest in the subject matter was stimulated by this course.
3	I learned to apply principles from this course to new situations.
4	My experiences encourage me to continue studying computer science.
5	The instructor related theories and concepts to practical issues.

Table 13b shows the results of the difference test between the graduate students in our course (GRAD) and all students in our department (ALL). The *p*-values in the first four items are <0.05, showing that the graduate students also received benefits from our PBL course. However, because of the small effect sizes, the improvements are not so obvious. Moreover, regarding Item 5, the average score of GRAD is not significantly greater than that of ALL. One possible reason is that combining theories and concepts with practical issues is fairly common in graduate research study.

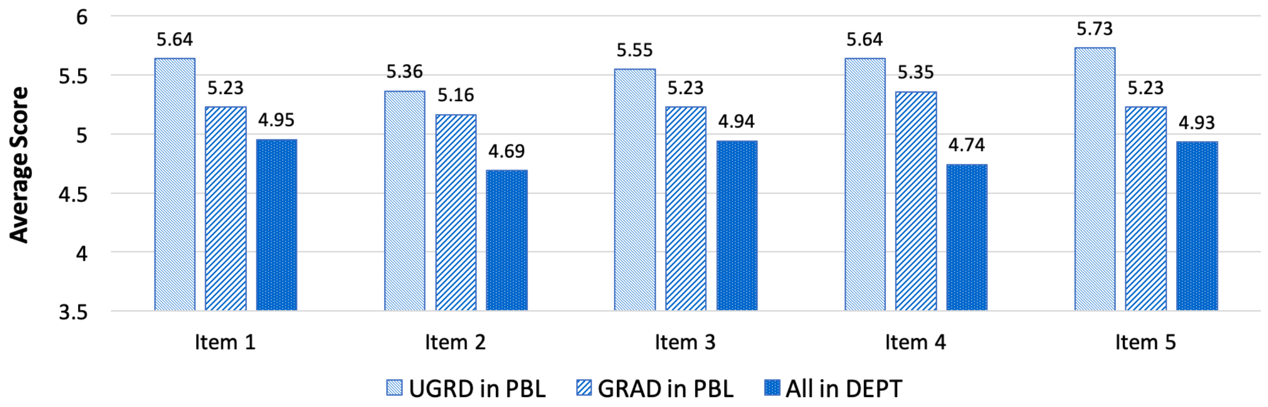


Figure 8. Average evaluation scores across students from our PBL course and all courses in the CS department

Table 13

P-Value and Effect Size Across Students from our PBL Course and all Courses in the CS Department

(a) UGRD vs. ALL			(b) GRAD vs. ALL			(c) UGRD vs. GRAD		
	***p*-Value**	**Effect size**		***p*-Value**	**Effect size**		***p*-Value**	**Effect size**
Item 1	3.2e-6	0.86	Item 1	0.033	0.27	Item 1	0.031	0.86
Item 2	6.6e-6	0.58	Item 2	0.004	0.31	Item 2	0.195	0.34
Item 3	0.002	0.66	Item 3	0.047	0.25	Item 3	0.122	0.46
Item 4	2.5e-9	0.81	Item 4	1.0e-4	0.44	Item 4	0.100	0.54
Item 5	1.2e-8	0.96	Item 5	0.068	0.22	Item 5	0.024	0.68

As shown in Table 13c, undergraduate students had even higher evaluation scores than graduate students in Item 1 and Item 5, with large effect sizes. We also observed that although Item 2, Item 3, and Item 4 had large *p*-values, they also had medium effect sizes. One cause might be that the *t*-test focuses on the difference between means, and the sample sizes of UGRD and GRAD are relatively small.

6.2 Qualitative Evaluation

Student comments are also helpful for course evaluation. Both undergraduate and graduate students expressed their views on our PBL course. We categorized students' comments into five classes: self-learning/just-in-time learning, problem-solving, group collaboration/presentation, guest lecture, and environment. Table 14 lists different categories and selective comments on the question "What did the instructor do that most helped in your learning?"

According to their feedback, the students enjoyed great benefit from the course. The goal of our PBL course and the course learning targets have already been achieved through the key factors they mentioned in comments, such as self-learning, periodic presentation, thinking for ourselves, group participation, flipped classroom, invited lectures, and "just-in-time" learning.

In addition, the students produced a number of materials from their time in the course. This material is available online and has been downloaded by others, in some cases, hundreds of times. Hence, the students, through our course, have gained exposure to the sharing of materials that they can confidently say they produced. In Table 15, we present the number of downloads for the final reports, final presentations, and software for each team.

The students have produced work that is interesting to a number of people and that has been downloaded, in some cases, >300 times, e.g., Team 10. Meaningful output is another positive result from our course, and we think having work downloaded many times is quite rewarding for students who worked hard in our course.

Table 14

Student Comments across Different Categories on the SPOT Question "What Did the Instructor Do that Most Helped in Your Learning?"

Category	Comment
Self-learning / Just-in-time Learning	1) He designed the course material such that we were to encouraged to learn ourselves. Self-learning is the best kind of learning. 2) He challenged us to think for ourselves 3) Dr. Fox was all about "just in time" learning. He gave us access to all resources that we could potentially want or need in order to complete our projects.
Problem-Solving	1) Problem solving skills. 2) For all of my question he had suggestions how I can start working on a new problem or improve my work.
Group Collaboration / Group Presentation	1) He had an environment very conducive to learning as a group and encouraged group participation and tasks. 2) Periodic presentations around our project were very helpful in keeping us motivated to keep working for our project. 3) One of the best thing is group work and group presentation. 4) There was mutual benefit in these presentations. 5) It would help us learn more about how to work with each other in a group.
Guest Lecture	1) Some invited lectures really helps on the project.
Environment	1) The flipped classroom was great. I definitely learned more in this class than I probably have in any other. 2) Helped provide a nurturing environment where were given freedom to explore and learn. 3) Really flexible with how student approaches the problem. There is a lot of freedom for students to do things they want to do. 4) Dr. Fox provided a very encouraging environment where we were able to ask questions easily.

Table 15

*Number of Downloads for the Specific Items Produced by Students Throughout the Semester as of August 21, 2019**

# of Downloads	Team ID													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Final Report	347	66	284	211	510	298	515	154	311	**929**	169	389	530	310
Final Presentation	89	66	123	85	99	86	152	82	93	144	84	42	**179**	103
Software	8	1	25	0	27	16	10	5	17	3	7	3	**34**	12

*Notably, Team 10's Final Report Has Been Downloaded 929 Times. Furthermore, Team 13 and Team 5 Have Had Their Final Reports Downloaded 530 and 510 Times, Respectively.

7 Discussion

7.1 PBL in Teaching

Dr. Fox has been designing PBL courses for both undergraduate (e.g., CS4624) and graduate students (e.g., CS5604, CS6604) for years. The purpose is to raise students' interest in studying computer science. This course pays close attention to teaching NLP through a practical task (i.e., big data text summarization), which is a classical, popular, and complicated problem. We expect students to not only apply traditional techniques but also understand and master state-of-the-art approaches. NLP is different from a few research topics in other PBL courses because it covers numerous tasks such as traditional computational linguistics, information extraction, natural language generation, and machine learning (e.g., clustering, deep learning).

Students have varying degrees of interest in these topics. Therefore, we have introduced our students to NLP through PBL, to cover these topics and provide students a chance to explore what interests them. Our PBL course offers an environment for students to take the initiative in self-directed learning and have the freedom to find suitable approaches to achieve course targets and satisfy their curiosity.

As shown in Table 13, PBL is beneficial for both undergraduate and graduate students in our class.

By taking responsibility for their learning, students strengthen understanding of domain knowledge, develop learning strategies, and consolidate learning abilities, which will serve them well for further problem-solving, particularly when they encounter unfamiliar problems. In attempting to solve problems, students are self-motivated to learn more about disciplinary concepts. For example, regarding topic modeling and clustering, they first need to understand document representation and how models work. These principles can be effectively transferred from our PBL course to new situations. PBL courses can also encourage students to discover methods on their own during learning and experimentation. Thus, Event Team 7 designed and implemented their hybrid summarization system and proposed a question-based approach to evaluate the generated summaries. Event Team 5 had each team member rank the six summary candidates that they produced and average the score to choose their best summary. Based on our initial investigation, only 20% of undergraduate students and 40% of graduate students had some experience in deep learning. After a semester of PBL, 13 of 14 teams applied deep learning techniques through ARC servers for abstractive summarization. Meanwhile, in addition to the above positive aspects, Table 13c shows some difference between undergraduate and graduate students, which prompts us to improve our course and aim to better meet the needs of graduate students.

7.2 Pedagogical Solutions in PBL

The original PBL in medical education focused on small-group discussion with a faculty tutor (Barrows & Tamblyn, 1980). We applied a set of different educational settings to support our pedagogical solutions.

7.2.1 Intra- and Inter-team Collaboration

We strongly encourage both intra- and inter-team collaboration in our PBL course. The course is structured with one of the two sessions per week focused on team collaboration. Students are free to discuss anything with their team, and different teams are able to discuss anything (e.g., problems, methods, and tools) related to NLP and get support from our faculty, GTA, and related GRAs. Undergraduate students can learn a lot from graduate students, while the latter could also consolidate domain knowledge through intra-team collaboration. The inter-team cooperation was beneficial too, allowing each team to share techniques or methods relevant to their collections. For example, regarding abstractive summarization, Event Team 6 struggled for several weeks on using the pretrained PGN with their data set. After trial and error, they wrote a script to help other teams generate the correct input format, which expedited the process for the other teams.

7.2.2 Peer Evaluation of Team Members

As part of the course grading, peer evaluation of team members was used to measure the work during PBL. Each student is required to provide two numbers for each other member in the same team. One reflects the quality of that student's work; the other reflects the amount of work. Each number is on a scale of 0–10, where 0 represents no contribution and 10 represents superlative contribution. As an aid to team collaboration, peer evaluation is very helpful to motivate student efforts to make more contributions to their team.

7.2.3 Team Presentation

The other session each week included teams each presenting for 7–10 minutes, describing their weekly progress, the problems they faced, and their plan for the coming week. This session acted as a weekly deadline for students to work toward helping to set the pace for their semester-long project. Before each presentation session, each team shared slides with the professor and the GTA, who were able to address potential misconceptions and provide guidance for the formal presentation. Furthermore, each team had a question-and-answer period to answer questions from other teams. All these components helped students improve their abilities regarding slide preparation, oral presentation, and time management; they also helped us assess their understanding of critical concepts in NLP.

7.2.4 Problem-driven Lectures

We had four problem-driven lectures during the semester that covered essential topics in text summarization, including indexing, deep learning, manual summarization, and automatic summarization. These topics are related to NLP; most students lacked a keen awareness of them. Problem-driven lectures filled the gaps in their knowledge about such issues. As an outcome of the two lectures about indexing and manual summarization, teams produced gold

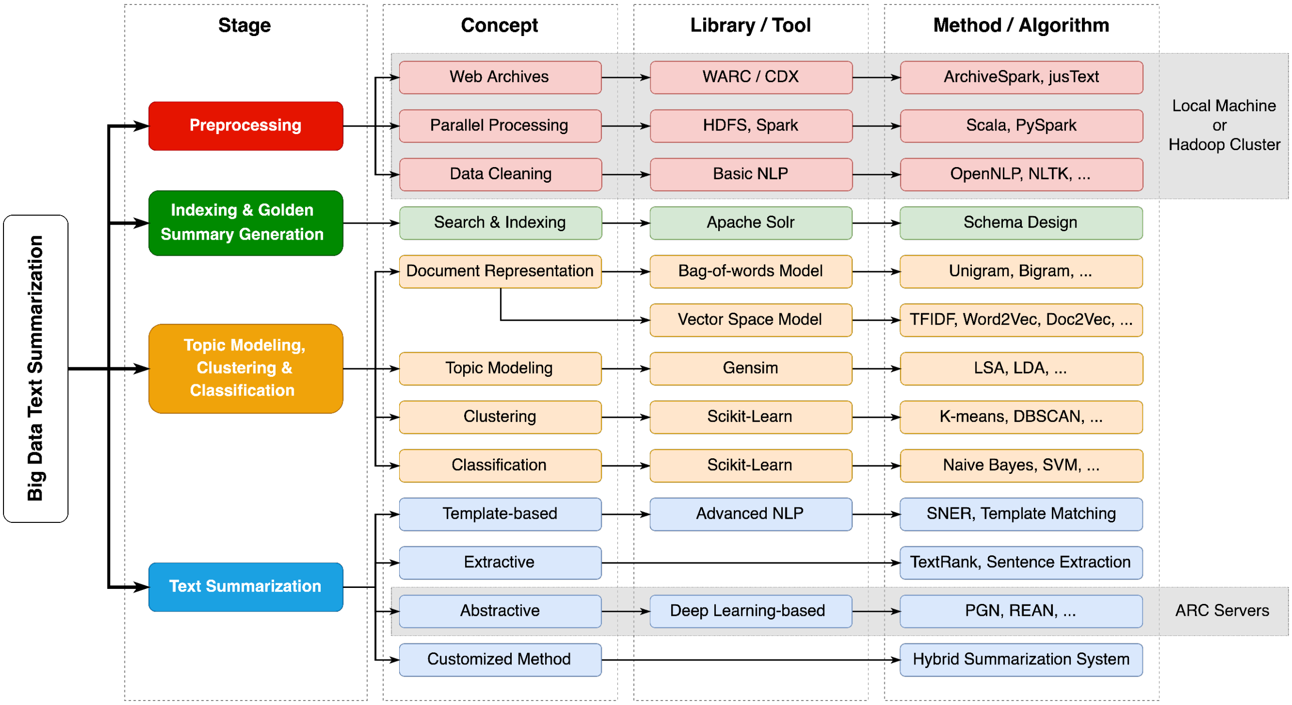


Figure 9. Treemap of the teaching and learning points in our PBL course

standard summaries through cross-labeling. Similarly, the other two lectures helped students create their summaries and evaluate the summarization quality.

7.3 Integration of Learning and Research

Unlike the traditional curriculum, our PBL course provides a more flexible way to integrate both learning and research. Regarding NLP, students can learn various methods, techniques, and algorithms through traditional teaching. However, the disadvantage is that these units are relatively independent of each other, so it is difficult for students to establish a systematic solution when facing a practical problem or research topic. Instead, we selected a “hot topic” in NLP and guided students to broaden their knowledge in that area. Students can take an active role in determining methods through the different educational settings mentioned above. Notably, multiple students have since applied their approaches through follow-on independent study or graduate research investigations.

8 Conclusions

We developed a PBL course to teach NLP through big data text summarization for both undergraduate and graduate students. The goal given to students is to automatically construct English language summaries of the important information in a large document collection (i.e., Web articles or ETD chapters). To guide students to solve the problem and learn NLP (at levels ranging from elementary to intermediate), we first proposed a general pipeline with four stages: preprocessing, indexing and gold summary generation, topic modeling and classification/clustering, and text summarization. Regarding each step, we introduced several fundamental concepts in NLP for PBL. Then, we provided multiple resources (e.g., Hadoop cluster, ARC servers) to reduce the computing difficulties of big data. Some key concepts, such as indexing and text summarization, were explained through guest lectures. Collaborative learning is also an essential part of our course. We designed our PBL course at the team level to strengthen resource sharing and team collaboration.

The student teams studied the relevant libraries by themselves to broaden their understanding of NLP. They also produced software repositories, utilized open source tools, developed or reapplied algorithms, and implemented their approaches during problem-solving. Additionally, their professional skills were further improved through weekly presentations and a final official report as a conclusion of their teamwork.

Figure 9 depicts a treemap of the teaching and learning points in our PBL course, expanding the general pipeline discussed in Section 4. We primarily worked on the top levels (i.e., stage and concept). Regarding some new concepts such as WARC, Hadoop distributed file system (HDFS), and Solr, we provided scripts, tutorials, and guest lectures for better understanding. Meanwhile, the student teams mainly worked on the detailed levels (i.e., library/tool and method/algorithm). They utilized suitable techniques for NLP through summarization and also gained experience with big data, parallel computing, cloud computing, and deep learning – to extend their 21st-century education.

As a result, most teams completed all the course learning targets by applying various types of NLP and relevant techniques, such as tokenization, part-of-speech (POS) tagging, NER, classification, clustering, and extractive and abstractive summarization. Regarding the specific summarization task, some teams designed their approaches and achieved relatively high ROUGE scores through quantitative evaluation. Meanwhile, the summaries generated by the teams are readable and correctly describe the corresponding events or ETD chapters.

The SPOT scores indicate that our PBL course significantly stimulates students' interest in problem-solving and encourages them to continue studying computer science. Additionally, our PBL course impressed students – with self-learning, "just-in-time" learning, and flipped classroom – based on their comments.

In the future, we plan to conduct further quantitative analysis for this course and other PBL courses in our department by collecting additional information on student preferences and perspectives as a supplement to the SPOT report and as another important aspect of comprehensive evaluation. Regarding the same PBL course, we will also aim to alter course settings (e.g., team size, learning targets, and pedagogical solutions), leading to a longitudinal analysis across different semesters.

We hope that the data sets we have developed will be of interest to other faculty and researchers, and also encourage them to contact us regarding their use. We believe that the student team results represent advances in synergistic approaches to the two types of summarization we are studying; we are continuing our studies of each of these tasks, building upon their findings.

Acknowledgments: Thanks go to the US National Science Foundation for its support of the Coordinated, Behaviorally Aware Recovery for Transportation and Power Disruptions (CBAR-tpd), Global Event and Trend Archive Research (GETAR), and Integrated Digital Event Archiving and Library (IDEAL) projects, through grants CMMI-1638207, IIS-1319578, and IIS-1619028, as well as IIS-1619371 to partner Internet Archive. Thanks go to the Institute for Museum and Library Services (IMLS), for the support of Opening Books and the National Corpus of Graduate Research, through LG-37-19-0078-19. We also thank the student teams in CS4984/CS5984 for their work.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Kudlur, M. (2016, November). TensorFlow: a system for large-scale machine learning. In K. Keeton & Timothy Roscoe (Eds.), *Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation* (pp. 265–283). Berkeley, CA: USENIX Association.

AI2. (2019, October) *Science parse*. Retrieved from https://github.com/allenai/science-parse

Allen, D. E., Duch, B. J., & Groh, S. E. (1996). The power of problem-based learning in teaching introductory science courses. *New Directions for Teaching and Learning, 1996*(68), 43–52.

Apache Solr (8.4.1)[Computer software]. Retrieved from http://lucene.apache.org/solr/

Apache Spark. [Computer software] Retrieved from https://spark.apache.org/

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval* (Vol. 463). New York: ACM Press.

Bahdanau, D., Cho, K., & Bengio, Y. (2015, January). Neural machine translation by jointly learning to align and translate. *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015*.1-15.

Baldridge, J. (2005). The opennlp project. Retrieved from http://opennlp.apache.org/index.html

Barrows, H. S. (1986). A taxonomy of problem-based learning methods. *Medical Education, 20*(6), 481–486.

Barrows, H. S., & Tamblyn, R. M. (1980). *Problem-based learning: An approach to medical education.* New York, NY: Springer Publishing Company.

Biggs, J. (1999). What the student does: Teaching for enhanced learning. *Higher Education Research & Development, 18*(1), 57–75.

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit.* Sebastopol, CA: O'Reilly Media, Inc.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research, 3*(Jan), 993–1022.

Carbonell, J., & Goldstein, J. (1998, August). The use of MMR, diversity-based reranking for reordering documents and producing summaries. *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 335-336). New York, NY: Association for Computing Machinery. doi:10.1145/290941.291025

Carstensen, K. U., & Hess, M. (2003). Problem-based web-based teaching in a computational linguistics curriculum. *Linguistik*

Online: Learning and teaching (in) Computational Linguistics, 17(5), 7–22.

Cavedon, L., Harland, J., & Padgham, L. (1997, July). Problem based learning with technological support in an AI subject: description and evaluation. In H. Søndergaard & J. Hurst (Eds.), *Proceedings of the 2nd Australasian Conference on Computer Science Education* (pp. 191-200). New York, NY: Association for Computing Machinery. doi:10.1145/299359.299387

Chen, Y. C., & Bansal, M. (2018). Fast abstractive summarization with reinforce-selected sentence rewriting. In C. Cardie, I. Gurevych, & Y. Miyao (Eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* (Vol. 1, pp. 675-686). Stroudsburg, PA: Association for Computational Linguistics. doi:10.18653/v1/P18-1063

Cline, M., & Powers, G. J. (1997, November). Problem based learning via open ended projects in Carnegie Mellon University's Chemical Engineering undergraduate laboratory. *Proceedings Frontiers in Education 1997 27th Annual Conference. Teaching and Learning in an Era of Change* (Vol. 1, pp. 350-354). Washington, DC: IEEE Computer Society.

Costa, L. R., Honkala, M., & Lehtovuori, A. (2007). Applying the problem-based learning approach to teach elementary circuit analysis. *IEEE Transactions on Education*, *50*(1), 41–48.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, *41*(6), 391–407.

Di Fabbrizio, G., Stent, A., & Gaizauskas, R. (2014, June). A hybrid approach to multi-document summarization of opinions in reviews. *Proceedings of the 8th International Natural Language Generation Conference (INLG)*, 54–63. Red Hook, NY: Curran Associates, Inc.

Dolmans, D. H. J. M., Loyens, S. M. M., Marcq, H., & Gijbels, D. (2016). Deep and surface learning in problem-based learning: A review of the literature. *Advances in Health Sciences Education: Theory and Practice, 21*(5), 1087–1112.

Erkan, G., & Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22, 457–479.

Frank, E., Paynter, G. W., Witten, I. H., Gutwin, C., & Nevill-Manning, C. G. (1999). Domain-Specific Keyphrase Extraction, *Proceeding of 16th International Joint Conference on Artificial Intelligence* (pp. 668–673). San Francisco, USA: Morgan Kaufmann Publishers.

Frigui, H., & Nasraoui, O. (2004). Simultaneous clustering and dynamic keyword weighting for text documents. In M. W. Berry (Ed.), *Survey of Text Mining* (pp. 45–72). New York, NY: Springer. doi:10.1007/978-1-4757-4305-0_3

Greene, D., & Cunningham, P. (2006, June). Practical solutions to the problem of diagonal dominance in kernel document clustering. In W. Cohen & A. Moore (Eds.), *Proceedings of the 23rd International Conference on Machine Learning* (pp. 377–384). New York, NY: Association for Computing Machinery. doi:10.1145/1143844.1143892

Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing? *International Journal of Human-Computer Studies*, *43*(5–6), 907–928.

Grusky, M., Naaman, M., & Artzi, Y. (2018). Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 708–719)*. doi:10.18653/v1/N18-1065

Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., & Blunsom, P. (2015). Teaching machines to read and comprehend. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, & R. Garnett(Eds.), *Advances in Neural Information Processing Systems* (pp. 1693–1701). Red Hook, NY: Curran Associates, Inc.

Holzmann, H., Goel, V., & Anand, A. (2016, June). Archivespark: Efficient web archive access, extraction and derivation. In N. R. Adam, B. Cassel, Y. Yesha, R. Furuta, & M. C. Weigle, *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries* (pp. 83–92). New York, NY: Association for Computing Machinery doi:10.1145/2910896.2910902

Honnibal, M., & Montani, I. (2017). spacy2: Natural language understanding with bloom embeddings, *Convolutional Neural Networks and Incremental Parsing. 7*(1).

Indiramma, M. (2014, December). Project based learning—Theoretical foundation of computation course. *2014 International Conference on Interactive Collaborative Learning (ICL)* (pp. 841–844). IEEE.

Indurkhya, N., & Damerau, F. J. (2010). *Handbook of natural language processing.* New York, NY: Chapman and Hall/CRC.

Jurafsky, D. (2000). *Speech & language processing.* Upper Saddle River, New Jersey: Pearson Education India.

Kanan, T., Zhang, X., Magdy, M., & Fox, E. (2015, June). Big data text summarization for events: A problem based learning course. In P. L. Bogen, S. Allard, H Mercer, M. Beck, S. J. Cunningham, D. Goh, & G Henry (Eds.), *Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries* (pp. 87–90). New York, NY: Association for Computing Machinery. doi:10.1145/2756406.2756943

Kay, J., Barg, M., Fekete, A., Greening, T., Hollands, O., Kingston, J. H., & Crawford, K. (2000). Problem-based learning for foundation computer science courses. *Computer Science Education, 10*(2), 109–128.

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, 1188–1196.

Lebanoff, L., Song, K., & Liu, F. (2018). Adapting the Neural Encoder-Decoder Framework from Single to Multi-document Summarization. In P. Merlo, R. Barzilay, & M. Johnson (Eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 4131–4141). Stroudsburg, PA: Association for Computational Linguistics. 10.18653/v1/D18-1446

Li, T., Ma, S., & Ogihara, M. (2004). Document clustering via adaptive subspace iteration. In K. Järvelin, J. Allan, P. Bruza, & M. Sanderson (Eds.), *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 218–225). New York, NY: Association for Computing Machinery. doi: 10.1145/1008992.1009031

Lin, C. Y., & Hovy, E. (2002, July). Manual and automatic evaluation of summaries. *Proceedings of the ACL-02 Workshop on Automatic Summarization-Volume 4* (pp. 45–51). Stroudsburg, PA: Association for Computational Linguistics. doi:10.3115/1118162.1118168

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, *28*(2), 129–137.

Lopez, P. (2009, September). GROBID: Combining automatic bibliographic data recognition and term extraction for scholarship publications. In M. Agosti, J. Borbinha, S. Kapidakis, C. Papatheodorou, & G. Tsakonas (Eds.), *demos in Computer Science: Vol 5714. International Conference on Theory and Practice of Digital Libraries* (pp. 473–474). Berlin, Heidelberg: Springer-Verlag. doi:10.1007/978-3-642-04346-8_62

Loria, S., Keen, P., Honnibal, M., Yankovsky, R., Karesh, D., & Dempsey, E. (2014). Textblob: simplified text processing. *Secondary TextBlob: Simplified Text Processing, 3*.

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research and Development, 2*(2), 159–165.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., & McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 55–60. doi:10.3115/v1/P14-5010

Mazur, E. (1992). Qualitative versus quantitative thinking: Are we teaching the right thing. *Optics and Photonics News, 3*(2), 38–40.

Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., . . . Xin, D. (2016). MLlib: Machine learning in Apache Spark. *Journal of Machine Learning Research, 17*(1), 1235–1241.

Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into text. *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing* (pp. 404–411). Stroudsburg, Pennsylvania: Association for Computational Linguistics.

Mills, J. E., & Treagust, D. F. (2003). Engineering education—Is problem-based or project-based learning the answer. *Australasian Journal of Engineering Education, 3*(2), 2–16.

Nallapati, R., Zhou, B., Gulcehre, C., & Xiang, B. (2016). Abstractive Text Summarization Using Sequence-to-sequence RNNs and Beyond. *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. 280–290. doi:10.18653/v1/K16-1028

Núñez-del-Prado, M., & Goméz, R. (2017, March). Learning data analytics through a problem based learning course. In *2017 IEEE World Engineering Education Conference (EDUNINE)* , 52–56.

Nuutila, E., Törmä, S., & Malmi, L. (2005). PBL and computer programming—The seven steps method with adaptations. *Computer Science Education*, *15*(2), 123–142.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Desmaison, A. (2019). PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.): *Advances in Neural Information Processing Systems* (pp. 8024–8035). Cambridge, MA: Curran Associates, Inc.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Peng, F., & McCallum, A. (2006). Information extraction from research papers using conditional random fields. *Information Processing & Management, 42*(4), 963–979.

Pomikálek, J. (2011). *Removing boilerplate and duplicate content from web corpora* (Doctoral dissertation, Masarykova univerzita, Fakulta informatiky).

Pritchard, J. K., Stephens, M., & Donnelly, P. (2000). Inference of population structure using multilocus genotype data. *Genetics, 155*(2), 945–959.

Radev, D. R., Jing, H., Styś, M., & Tam, D. (2004). Centroid-based summarization of multiple documents. *Information Processing & Management, 40*(6), 919–938.

Řehůřek, R., & Sojka, P. (2011). *Gensim—statistical semantics in Python*. Retrieved from gensim.org.

Reiter, E., & Dale, R. (1997). Building applied natural language generation systems. *Natural Language Engineering, 3*(1), 57–87. doi: 10.1017/S1351324997001502

See, A., Liu, P. J., & Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. In R. Barzilay, & Min-Yen Kan (Eds.): *Long Paper in Computer Science: Vol. 1. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (pp. 1073–1083), Vancouver, Canada: Association for Computational Linguistics. doi:10.18653/v1/P17-1099

Serife, A. K. (2011). The effect of computer supported problem based learning on students' approaches to learning. *Current Issues in Education, 14*(1), 3–18.

Sperberg-McQueen, C. M., & Burnard, L. (Eds.). (1990). *Guidelines for the encoding and interchange of machine-readable texts.* Chicago, Oxford: Text Encoding Initiative.

Stent, A., & Bangalore, S. (Eds.). (2014). *Natural language generation in interactive systems*. Cambridge, England: Cambridge University Press.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems* (pp. 3104–3112). Cambridge, MA: Curran Associates, Inc.

Vernon, D. T. (1995). Attitudes and opinions of faculty tutors about problem-based learning. *Academic Medicine, 70*(3)*,* 216–223.

Wilkerson, L., & Feletti, G. (1989). Problem-based learning: One approach to increasing student participation. *New Directions for Teaching and Learning*, *1989*(37), 51–60.

Yadav, A., Subedi, D., Lundeberg, M. A., & Bunting, C. F. (2011). Problem-based learning: Influence on students' learning in an electrical engineering course. *Journal of Engineering Education, 100*(2), 253–280.

Zhang, Z., Hansen, C. T., & Andersen, M. A. (2016). Teaching power electronics with a design-oriented, project-based learning method at the Technical University of Denmark. *IEEE Transactions on Education, 59*(1), 32–38.