Computational Models of Quality for Educational Digital Resource Assessment

Philipp Georg Wetzler
University of Colorado at Boulder, philipp.wetzler@colorado.edu

Follow this and additional works at: https://scholar.colorado.edu/csci_gradetds

Part of the Databases and Information Systems Commons, and the Library and Information Science Commons

Recommended Citation
https://scholar.colorado.edu/csci_gradetds/11

This Dissertation is brought to you for free and open access by Computer Science at CU Scholar. It has been accepted for inclusion in Computer Science Graduate Theses & Dissertations by an authorized administrator of CU Scholar. For more information, please contact cuscholaradmin@colorado.edu.
Computational Models of Quality for
Educational Digital Resource Assessment

by

Philipp G. Wetzler

M.S., University of Colorado, 2003

A thesis submitted to the
Faculty of the Graduate School of the
University of Colorado in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Computer Science
2010
This thesis entitled:
Computational Models of Quality for Educational Digital Resource Assessment
written by Philipp G. Wetzler
has been approved for the Department of Computer Science

James H. Martin

Tamara Sumner

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.
Educational digital libraries and peer-produced open educational resources have become integral to efforts to incorporate personalized learning into the classroom. Assuring the quality of educational content from these sources has become a major concern of the curators of such materials, and of educators who want to use them. But quality of educational materials is a multi-faceted problem, not completely understood, and often disputed. In current practice, focused manual effort by trained experts is required to assess each resource.

This work attempts to leverage the large existing corpus of work in the field of computational semantics to supplement and support human judgment in educational resource assessment. Based on an in-depth study of human expert decision processes, characterizing the quality of a resource is broken down into dimensions of quality, and further into low-level, more easily identified indicators of quality; these indicators of quality alone are strongly predictive of an expert’s overall quality assessment of a resource.

A corpus of 1000 resources from the Digital Library for Earth System Education (DLESE) was manually annotated for the presence or absence of seven important quality indicators. Human experts were able to make these assessments quite consistently. Using a supervised machine learning and document classification approach, a baseline computational system was able to train models for each of the seven indicators that achieved some agreement with the human annotation. By adjusting the computational system to make better use of the data set, these models were improved to achieve good agreement on all seven indicators.

To evaluate the generalizability of this approach, an additional corpus of 230 peer-produced open educational resources from the Instructional Architect (IA) project was manually annotated for quality indicators, using a slightly modified annotation protocol. In spite of the very different nature of the materials, the computational models trained on the DLESE corpus generalized to the new data to a small extent; models trained on the new data achieved mostly good agreement.
Acknowledgements

This thesis builds on the extensive work by my colleagues at the University of Colorado, at Digital Learning Sciences / UCAR, at the University of Utah, and at Utah State University. While I am thankful to all of them, I will not attempt to provide a complete list of all the people that contributed their work and their ideas. But I would like to thank, in particular, the professors, researchers, and students of the Computational Semantics group at CU for their continued support throughout my time here, for their advice and stimulating discussions, and for the occasional, very welcome distractions.

My advisers, Tamara Sumner and Jim Martin, were always supportive of my ideas and had new perspectives to add that I would not otherwise have thought of. I would like to thank Jim especially for the time he took reviewing the many drafts of my dissertation and providing useful feedback. Tammy always had targeted, helpful, and interesting new ideas, and she went far beyond what I could ever have expected in keeping me on track and getting me to this point.

The foundation of this research was laid by Tammy Sumner, Kirsten Butcher, and Steven Bethard, and without their advice, from the very beginning, I would not have gotten far in making sense of the rich background in digital libraries and education.

Along with our other colleagues at Digital Learning Sciences / UCAR, I would like to mention Holly Devaul, who showed a lot of interest in looking at digital libraries from my very different perspective. She was always happy to share her extensive expertise. In particular, her involvement was essential in planning and creating the DLESE corpus, and in performing the error analysis on the DLESE models.

The initial idea to create the Instructional Architect corpus came from Mimi Recker and Heather Leary, and they played the major role in planning and conducting the annotation. In addition, I would like
to thank Mimi and Heather, as well as Andrew Walker, for the helpful discussions we had on inter-rater reliability.

The work presented in this thesis was supported in part by the National Science Foundation under Grant Numbers CISE/IIS-0741326, CISE/IIS-0534581, and EHR/DRL-0554440, as well as Utah State University. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.
Contents

Chapter

1  Introduction  1
   1.1  Research Methodology ......................................................... 5
       1.1.1  Expert Study ................................................................. 6
       1.1.2  Creating a Training Corpus ............................................. 6
       1.1.3  Building Computational Models ....................................... 7
       1.1.4  Generalizing to a New Data Set ....................................... 7
   1.2  Main Contributions ............................................................. 8

2  Related Work  10
   2.1  Automatic Quality Assessment on Educational Resources ............. 11
   2.2  Definitions of Resource Quality in Education ............................ 12
   2.3  Automatic Quality Assessment on Textual Documents and Web Sites .. 13
   2.4  Information Quality Frameworks ............................................. 15

3  Characterizing Quality  16
   3.1  Analysis of Prior Studies .................................................... 17
   3.2  Expert Study ........................................................................ 19
   3.3  Indicators of Quality ............................................................ 19

4  The DLESE Corpus  22
   4.1  DLESE .............................................................................. 22
4.1.1 The DLESE Community Collection ........................................... 23
4.2 Quality Indicators ................................................................. 24
  4.2.1 Has instructions ............................................................... 27
  4.2.2 Has sponsor ................................................................. 28
  4.2.3 Has prestigious sponsor .................................................. 28
  4.2.4 Identifies age range ....................................................... 29
  4.2.5 Content is not inappropriate for age range ......................... 29
  4.2.6 Identifies learning goals ............................................... 29
  4.2.7 Organized for learning goals .......................................... 29
4.3 Annotation Protocol ............................................................. 30
4.4 Evaluation of Annotation Results .......................................... 31
4.5 Quality Indicator Predictiveness ........................................... 32

5 Stage 1: Baseline System ......................................................... 35
  5.1 Corpus Preparation .......................................................... 37
  5.2 Feature Set ................................................................. 39
    5.2.1 Feature Encoding ...................................................... 39
    5.2.2 Feature Extraction .................................................. 40
  5.3 The Machine Learning System ........................................... 42
  5.4 Results .......................................................................... 43
  5.5 Error Analysis ............................................................... 44
  5.6 Conclusion ..................................................................... 46

6 Stage 2: Model Improvements .................................................. 47
  6.1 Corpus Preparation .......................................................... 47
  6.2 Web Crawler Configurations .............................................. 51
  6.3 Relative Importance of Features ....................................... 52
  6.4 Exploiting Resource Structure .......................................... 56
6.5 Training Set Size .......................................................... 58
6.6 Machine Learning Components ....................................... 62
6.7 Summary ................................................................. 64

7 The Instructional Architect Corpus .................................................. 68
7.1 Differences Between IA and DLESE ............................................. 69
7.2 Annotation ................................................................. 73
    7.2.1 Project Selection ....................................................... 73
    7.2.2 Annotators ............................................................. 74
    7.2.3 Modified Annotation Scale ........................................... 75
    7.2.4 Quality Indicators ..................................................... 76
7.3 Evaluation ................................................................. 79
7.4 Conclusion ................................................................. 82

8 Computational Models for Instructional Architect .................. 92
8.1 Preliminary Steps .......................................................... 93
8.2 Experimental Setup ....................................................... 94
8.3 Models Trained on DLESE .................................................. 96
8.4 Models Trained on Instructional Architect ............................... 97
8.5 Vocabulary and Generalizability ......................................... 98
8.6 Conclusions ............................................................... 100

9 Discussion ........................................................................ 101
9.1 Use Cases for DLESE Models ............................................... 103
    9.1.1 Prioritizing Review Processes ...................................... 103
    9.1.2 Scaffolding Expert Reviews ........................................ 103
    9.1.3 Quality Indicators to Improve Search ............................ 104
9.2 Extensions to the Quality Indicator Models and Framework ........ 105
Tables

Table

3.1 Sample qualitative data drawn from raw data sources for three dimensions . . . . . . . . . . . . . 18
3.2 The 12 most salient dimensions of quality . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18
3.3 Comments on the presence (+) or absence (−) of quality indicators . . . . . . . . . . . . . . . . 20

4.1 List of quality indicators . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27
4.2 Details of agreement between annotators on the “has prestigious sponsor” indicator . . . . 32
4.3 Quality indicator presence in resources and inter annotator agreement . . . . . . . . . . . . . . 32
4.4 Quality indicator predictiveness and leave-one-out analysis . . . . . . . . . . . . . . . . . . . . . 33

5.1 Baseline computational system results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 44

6.1 DLESE resource types in corpus . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49
6.2 Comparison of corpus preparation configurations . . . . . . . . . . . . . . . . . . . . . . . . . . . 50
6.3 Comparison of crawler configurations . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 52
6.4 Quality indicator performance using different feature sets . . . . . . . . . . . . . . . . . . . . . 55
6.5 Effect of resource structure annotation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 58

7.1 List of quality indicators in the IA corpus . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 76
7.2 Quality indicator bias in the DLESE and IA corpora . . . . . . . . . . . . . . . . . . . . . . . . . 80
7.3 Inter-rater reliability metrics for the IA corpus . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 82
7.4 Inter-rater reliability metrics, only between annotators 1 and 3 . . . . . . . . . . . . . . . . . . 83
7.5 Inter-rater reliability metrics for the DLESE corpus ........................................... 83

8.1 Quality indicator models trained on DLESE corpus ........................................... 97

8.2 Quality indicator models trained on IA corpus ..................................................... 98

8.3 Effect of limiting vocabulary ................................................................................. 99
## Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>DLESE example resource 1: NASA SIR-C Education Program Student Lesson</td>
<td>25</td>
</tr>
<tr>
<td>4.2</td>
<td>DLESE example resource 2: UCAR Digital Image Library</td>
<td>26</td>
</tr>
<tr>
<td>6.1</td>
<td>Training set size – Page 1</td>
<td>60</td>
</tr>
<tr>
<td>6.2</td>
<td>Training set size – Page 2</td>
<td>61</td>
</tr>
<tr>
<td>6.3</td>
<td>Training Set Size – Part 1</td>
<td>66</td>
</tr>
<tr>
<td>6.4</td>
<td>Training Set Size – Part 2</td>
<td>67</td>
</tr>
<tr>
<td>7.1</td>
<td>Instructional Architect example project 1: Earthquakes in Utah</td>
<td>70</td>
</tr>
<tr>
<td>7.2</td>
<td>Instructional Architect example project 2: Addiction and the Brain</td>
<td>71</td>
</tr>
<tr>
<td>7.3</td>
<td>Distributions of ratings on the Instructional Architect annotation – “has instructions”</td>
<td>85</td>
</tr>
<tr>
<td>7.4</td>
<td>Distributions of ratings on the Instructional Architect annotation – “links to prestigious sources”</td>
<td>86</td>
</tr>
<tr>
<td>7.5</td>
<td>Distributions of ratings on the Instructional Architect annotation – “identifies learning goals”</td>
<td>87</td>
</tr>
<tr>
<td>7.6</td>
<td>Distributions of ratings on the Instructional Architect annotation – “organized for learning goals”</td>
<td>88</td>
</tr>
<tr>
<td>7.7</td>
<td>Distributions of ratings on the Instructional Architect annotation – “identifies age range”</td>
<td>89</td>
</tr>
<tr>
<td>7.8</td>
<td>Distributions of ratings on the Instructional Architect annotation – “content seems appropriate for age range”</td>
<td>90</td>
</tr>
<tr>
<td>7.9</td>
<td>Distributions of ratings on the Instructional Architect annotation – all quality indicators</td>
<td>91</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

“Advancing personalized learning” has been identified as one of the great challenges of today [40]. While teaching has traditionally followed the one-size-fits-all approach, it is being increasingly recognized that individual preferences and abilities of students must be taken into account in order to help each student achieve their potential. In 2008, the NSF Task Force on Cyberlearning declared the importance of leveraging technology to facilitate students’ learning efforts and support teachers in providing a rich educational landscape[4]. They recognized the opportunities offered by web technologies, which allow educators and institutions to easily publish, share, and adapt educational content online.

Educational digital libraries, such as the National Science Digital Library (NSDL)\(^1\) and the Digital Library for Earth System Education (DLESE)\(^2\), have been a cornerstone of these renewed efforts. Much like traditional libraries, these projects develop and curate collections of high quality web-based resources that can be useful for teaching and learning across a wide range of grade levels and educational settings. Materials include background readings and reference texts, classroom and laboratory activities, interactive and visual content (e.g. maps, animations and simulations), and scientific data. Many are published by educational and scientific institutions (such as NASA or the U.S. Geological Survey), non-profit organizations, or commercial providers of educational content; some are created by individual teachers and enthusiasts. The National Science Digital Library, which has been in operation since 2000 and is broad in scope, has cataloged over 115,000 such items.

\(^1\) [http://www.nsdl.org/](http://www.nsdl.org/)

\(^2\) [http://www.dlese.org/](http://www.dlese.org/)
Resource quality is a major concern for these libraries; each library, and each collection, has detailed guidelines on minimum expectations a resource must meet in order to be considered for inclusion. Guidelines are hand-crafted by expert educators and curators and aim to ensure a high level of quality for common use cases without excluding useful content. While most users of the library will have shared concerns in their expectations of quality, any attempt at providing one set of criteria for inclusion must be too lenient in some scenarios, and too strict in others, considering the broad scope of the libraries and the large diversity in educational settings and preferences; the definition of quality is contextual. It depends on the alignment between the user constituency being served, the educational setting where deployed, and the intended purpose of the resource. For example, **Connexions** is an online project aimed at efficiently sharing modular scholarly content that is collaboratively created. Acknowledging that there are multiple perspectives on quality, the service allows third parties to create “lenses”, which provide different views onto collections. For instance, a professional society has created a lens to view resources that have been vetted by their own peer review processes. Others have created different lenses based on algorithmic assessments of resource popularity.

As a further complication, implementing quality policies reliably and at the necessary scale is challenging to carry out in practice. Consistently judging the educational quality of resources in a generalized context is a complex cognitive task, requiring extensive training and a large time commitment from experts. For instance, the **Multimedia Educational Resource for Learning and Online Teaching** (MERLOT) has a comprehensive peer-review process for vetting resources, consisting of 15 editorial boards. Yet published data highlight the challenge MERLOT is facing in scaling up its peer-review process to keep up with the rate of contributions: Carey and Hanley report that the ratio of submitted to reviewed contributions is approximately eight to one [6].

More recently, focus has begun to shift from the traditional library model to an open content approach, attempting to leverage the collaborative power of the Web. Wikipedia, the perpetual poster child of open content, admirably demonstrates the “wisdom of crowds” effect: large numbers of non-experts, with little centralized coordination, succeed in collaboratively creating a resource of high utility. While criticism

---

of this editorial model remains, in some ways Wikipedia is able to compete with, or even surpass, traditional encyclopedias; effective use of Wikipedia demands higher information literacy of the user, but coverage on especially the more obscure topics is better and more complete. This success has proven difficult to replicate in other projects; but collaboratively created and managed content is seen as a big opportunity to directly involve the teacher community in the collection and preparation of educational content – thus bridging the gap between consumers and producers of resources. The Curriculum Customization Service (CCS)[50] is one such effort: teachers share a platform that allows them to compile their own curriculum, based on existing digital library resources, school district materials, and their own content, and as they do so, they share their experiences and recommendations with each other. Such projects can be constructed on top of existing digital library infrastructure, but they fundamentally change the way we think about curation policies and quality assessments of digital resources.

Many projects now have teachers create Open Educational Resources (OER). One such example is the Instructional Architect[28] project; essentially a simple web publishing platform for teachers, the project encourages its users to create re-usable educational resources and share them with other teachers. Teachers use the system to create guided activities that take the students to a selection of other web pages (taken from NSDL); students learn to read, analyze, and synthesize content from various sources. Users of Instructional Architect have produced many thousands of such resources, but the quality differs widely. Some are simply left over test pages or aborted attempts, others are only useful within the specific context of the class they were created for; a relatively small number are potentially very useful to other teachers and should be made accessible to them. But in order to include them in existing digital libraries, all pages must undergo the quality review process, thus making the scalability problem many times worse. Thus, Downes[18] refers to variable resource quality as a “crisis in OER” that is “hindering increased uptake and usage”.

The shifting landscape in educational content creation requires us to re-think what our expectations of quality are, how to characterize the quality of resources in a way that generalizes to the many use cases, and how to integrate such an understanding of quality (as opposed to the traditional “curation” model) into educational platforms. The dual challenges we face are generalizability on the one hand, and scalability on the other.
My approach to the fairly general problem of quality assessments on educational resources is based on previous approaches in the field of natural language processing (NLP). Computational models are being used successfully to do a wide range of things with text. The ultimate goal of NLP is, arguably, to be able to have a computer “read” a text (or understand speech) and parse and understand it the way an expert human reader would – following the conversational structure of the arguments, inferring things that aren’t explicitly stated, relating the explanations to all sorts of common knowledge about the world, and evaluating the explicit and implied statements in the context of expert domain and general knowledge. The state of the art falls far short of this goal. It is, however, possible to use basic structural and vocabulary cues to infer overall summative statements about a text; NLP is also successful at performing simple linguistic analysis tasks, such as parsing the phrase structure of a text or identifying recurring concepts.

This being the case, computational approaches are not as good at things that require a deep reading of a text, or the quasi-logical inference human readers perform, or relating what’s stated in the text to human experience. But even so, NLP has been remarkably successful at tasks one would expect to be hard, given those limitations. Examples are question answering, named entity identification, and broad assessments like essay grading or sentiment analysis[32, 43]. Simple surface features of a text have surprising predictive power on such tasks; while these computational systems don’t take the same path a human reader would, they do often arrive at the same conclusion.

I would like to mention two issues this work is not going to address: factual accuracy and intentional bias. Determining if the information stated within a resource is scientifically accurate can be hard to prove even for experts in science, and certainly for most people. In order to do so, one must have a deep and detailed understanding of the scientific matter being covered and the ability to derive a nuanced interpretation of the writer’s intent from the text. This kind of analysis is far beyond what current state-of-the-art artificial intelligence systems are capable of.

Bias, on the other hand, is more readily measurable. When the writer of a resource has an agenda they’re trying to push, one would expect to find certain “conversational cues” in the text – for example they would be more likely to use ideologically loaded terms and phrases that indicate value judgment, which a purely scientific text, arguably, should not have. But introducing intentional bias as a consideration opens
up another can of worms: One has to acknowledge that someone may attempt to “game” the system by, for example, intentionally introducing specific terminology that is designed to lead our system to rate a resource of higher quality than it is. Preventing such manipulation is beyond the scope of this project. This is not to say that intentional bias is not an important aspect of quality. Other research has considered this problem, and future work in open educational resource quality will have to incorporate these concerns.

1.1 Research Methodology

This section will give an overview of the methodology I employed during my research and the ideas that guided my processes. It will also summarize the main contributions of my work and serve as an outline of the remainder of this dissertation.

The research presented were was primarily guided by two principal ideas:

- **Human centered design.** The motivating factor of this research is found in problems that are faced by teachers and students, on materials that are produced by educators and other experts and that are used in educational environments that exist for the benefit of students. I wanted to acknowledge and make use of the vast existing expertise in these fields. The foundation of this work is to first understand how experts view these issues and what kinds of processes they employ in dealing with them.

- **Supervised machine learning using human-annotated data.** This is a time-tested approach in the natural language processing research community. In creating the computational models for quality assessment, I’m building on a large corpus of existing work in computational linguistics, document classification, and information retrieval.

The human centered design idea has previously been successfully employed in the educational domain to automatically recognize and remediate student misconceptions in a scientific domain [11] [26]. Unlike other, more common approaches in computational linguistics, instead of starting from a theoretically motivated framing of the problem, it begins by observing and understanding existing processes that human experts
already employ, and by letting the requirements of those experts guide the experimental design. In this way, it firmly anchors the work in real-world needs.

1.1.1 Expert Study

In order to systematically approach the characterization of the quality of digital library resources, specific aspects that are relevant in the context in which the resource may be evaluated must be identified. The educational domain has its own sets of criteria for what constitutes a “high quality” resource in a learning environment, and frameworks that try to define “quality” in a broader sense may not be specific enough. Approaches that have focused on quality dimensions in educational digital libraries address the specific requirements of the domain, but high-level abstract criteria such as “good pedagogical design”, while they make sense intuitively, are hard even for human experts to assess consistently; in addition to the inherent ambiguity, computational methods are hampered by their lack of common-sense knowledge and understanding, being limited to low-level features such as the presence or absence of specific keywords in a resource.

This research is based on a study conducted by Bethard et. al.[2], which examined the cognitive processes experienced educators followed in assessing resources for in-classroom use and identified the characteristics they use most commonly. Unlike previous studies, the researchers narrowed their analysis down to identify various low-level indicators that the experts routinely referred to in making their decision; these indicators are promising candidates for computational assessment. Chapter 3 summarizes this study and its findings.

1.1.2 Creating a Training Corpus

Assessing the presence or absence of these quality indicators in a resource is accomplished using supervised machine learning algorithms. These algorithms first construct a statistical model of resources that do, and do not, exhibit that indicator. This requires analyzing a large corpus of resources, some of which are known to exhibit the indicator, and some of which do not. To create such a corpus, I conducted an annotation project with my colleagues, in which we selected about 1000 educational resources from a digital
library and asked experts to assess each of them for the presence or absence of seven indicators chosen from the expert study. This corpus and the annotation protocol will be described in detail in Chapter 4.

1.1.3 Building Computational Models

In order to create and evaluate the computational models on the corpus, each resource must be compiled and processed to be brought into a form understandable by the supervised machine learning algorithms — a list of “features” describing the resource. There are many steps to this transformation, and many variations on each step. Chapter 5 and Chapter 6 describe in detail the process of going from a resource to a list of features, as well as the various experiments I conducted to determine the best way of preprocessing the corpus, and what features to use.

1.1.4 Generalizing to a New Data Set

Even if the computational models I create work well on the digital library corpus they are developed on, we cannot conclude that they would work at all on data that follows a different statistical distribution — even on digital library resources from a different, but related, scientific domain the models might perform very differently. I had the opportunity of working with a research group at the University of Utah to create a second corpus. This corpus uses not digital library resources, but peer-produced educational materials from a broader domain than the first corpus. In Chapter 7 and Chapter 8 I explain how we created this new corpus, how it differs from the first one, and how well the computational models were able to adapt to the new data.

I will finish in Chapter 9 with a discussion of what can be learned from this research, what can be improved upon, and which directions future research in the quality assessment of educational resources may take.
1.2 Main Contributions

The following are the main contributions my work has provided to research and to research infrastructure:

(1) I demonstrated that the methodology described here is a viable approach to automatically characterizing the quality of educational resources. By creating computational models for seven diverse indicators of quality, and by evaluating their performance compared to human annotators on two very different corpora, I showed that breaking up educational quality into quality indicators makes the quality assessment problem tractable with current computational methods.

(2) We created an annotated corpus of 1000 educational digital library resources from the domain of earth science, each of which are tagged for seven important quality indicators, as well as partial information about which of the linked pages are part of the resource; the annotation has been shown to be of good quality. In addition to its use here, this data set has already been used by the Cairo Microsoft Innovation Lab to further their research, and it is available for others upon request in an open XML format. It will also serve as a foundation for future research expanding on the work presented here.

(3) We further created a second annotated corpus of 230 peer-produced open educational resources from the Instructional Architect project. Each of these resources is rated for six indicators (a subset of the seven used in the first corpus), and each resource was independently annotated by three educators. This data set complements the digital library data set by expanding the topic coverage to general science content, and by shifting the focus to user-created open educational resources.

(4) We created tools to facilitate the annotation of the two corpora and the error analysis. The annotation tool, in particular, has been adopted by another project for a similar kind of annotation.

(5) I co-developed the open-source ClearTK toolkit for natural language processing, which offers a rich API for using machine learning for computational linguistics, especially within the UIMA
framework[41]. Throughout the course of this research I kept making significant contributions to the ClearTK project⁴.

(6) In addition to contributions to ClearTK, I developed an extensive research software, incorporating all the components discussed in the following chapters. This software is based on Java and ClearTK and can easily be integrated with other such software. It offers the capability of reproducing all the experiments I conducted during my research, and it is being used by my colleagues to extend upon my work.

My work represents an important foundation for future research in this field. I identified approaches that are suitable for creating computational models of educational resources, and ones that are not. The ideas discussed in this dissertation will be informing the directions taken in future research projects, bringing computational models to bear in quality assessments of educational resources in a wide range of settings.

⁴ The ClearTK project is not a focus of this dissertation. Details on the implementation, as well as the complete source code, are available from the project web site at http://code.google.com/p/cleartk/.
Chapter 2

Related Work

Assessing the educational quality of textual documents and websites with computational, automatic models has not been a focus of research so far. The application of computational methods to such abstract, difficult to grasp measures as quality is still a very challenging prospect, and this is only made more difficult by the introduction of the educational domain, which is constantly evolving and in which many diverging points of views exist. Compiling a usable annotated data set alone is a major undertaking.

Staying within the field of education, we mainly find approaches that target textual cohesion and coherence, readability, and automatic grading of essays. This research goes far back. However, while these measures are, arguably, relevant to the educational quality of a document, they fall far short of providing a full characterization of quality.

Moving away from automatic assessments, there is a body of work exploring the way experts and users assess the quality of digital resources and web sites. These studies seek to identify both the major factors influencing human judgments and the lower level features of resources that people attend to when assessing quality. They also consider the consistency (or lack of consistency) of humans when making potentially subjective assessments.

On the other hand, in recent years there have been various attempts at automatically assessing aspects of quality within more controlled domains for which data readily exists. The primary target of these efforts has been Wikipedia, which offers a large collection of peer-produced articles, all of which are freely available; Wikipedia also has support for annotation of quality problems by its contributors. General web search can
also be seen as facing the quality problem: out of the millions of pages that match a user’s query, search engines attempt to rank those highest that are most likely to address a user’s needs.

Widening our gaze further, a characterization of quality is most broadly attempted within the domain of information quality, where researchers have attempted to construct conceptual frameworks for defining what quality means, beyond the limits of one specific domain or application. These frameworks tend to be very abstract and difficult to apply to the free form data found in educational texts.

2.1 Automatic Quality Assessment on Educational Resources

In the work most closely related to my research, Custard and Sumner trained machine learning models to judge overall quality in the context of educational digital libraries using low-level features like website domain names, the number of links on a page, how recently a page was updated, and whether or not videos or sound clips were present [9]. Their models were able to identify whether a resource was from a high, medium or low quality collection with 76.67% accuracy. The corpus the models were evaluated on was much smaller, and the distinctions more coarse-grained; but the work of Custard and Sumner was the major motivation for the research I present here.

In the context of developing the QCommons project, which aims to be a platform for open educational content, Aleahmad et. al. applied machine learning methods similar to the ones used here to automatically rate the quality of resources[1]. The materials they looked at were of a specific form: worked examples, containing a short question and a series of steps with explanations leading to a solution. The study was limited to questions on one specific topic in mathematics (the Pythagorean theorem). Using features that took the specific structure of the problems into account, their models achieved high correlation with the human annotations.

Metrics based on simple text statistics such as the number of syllables in words and the average sentence length, which are easily computed automatically, have a long history of being used in an educational context to assess a text’s reading level, and more recently for automatic essay grading. Concrete formulations of these ideas are the Flesch Reading Ease Score and the Flesch-Kincaid Grade Level[22, 34].
More recent efforts make use of computational semantics. The computational framework of Latent Semantic Analysis[12] was first introduced for indexing documents by capturing their semantic content through the words they contain, but its uses have subsequently been explored, among many things, for essay grading and to match texts with the right group of students[48, 56]; these approaches rely on reference texts that can be compared against. Graesser et. al. presented Coh-Metrix[25], a software that for a given text computes over 200 measures covering aspects like cohesion and readability. This is set in contrast by Dufty et. al. with other attempts at evaluating document quality[19], but “quality” there exclusively refers to structural and stylistic quality as opposed to other aspects of a document that affect its usability in an educational setting.

2.2 Definitions of Resource Quality in Education

A core assumption motivating the approach discussed in this document is that, when asked to assess the quality of a digital resource or web site, people draw on a suite of often implicit criteria to guide their judgment. Numerous studies have documented this phenomenon. Fogg and colleagues studied how people assess “web site credibility”. Using an online survey of over 1400 participants, they identified criteria such as “ease-of-use”, “expertise” (of the site), and “trustworthiness” as being important factors influencing human judgment [23]. Rieh studied how people judge “cognitive authority” while searching for information on the web [47]. This study identified a slightly different set of factors as important for this specific judgment task, including accuracy, currency, trustworthiness, scholarliness and authoritativeness. Sumner and colleagues conducted focus groups to identify criteria to guide collections policy decisions in educational digital libraries serving both formal and informal educational audiences [51]. The criteria identified here for assessing resource quality included scientific accuracy, lack of bias, and good pedagogical design.

This body of work highlights two important issues: First, there are many potential perspectives on quality, depending on the use and audience of materials. Credibility is important when selecting an online bank, for instance, whereas cognitive authority is more important when locating scholarly resources. Second, these studies demonstrate that once the purpose of the quality assessment has been more narrowly defined,
people do draw on common (task-dependent) criteria to characterize resources. This suggests that identifying common dimensions for assessing the quality of resources for formal classroom use is feasible.

For a machine learning approach to be possible, a corpus of annotated resources is needed to train the models. Creating a training corpus requires humans to annotate each resource as exhibiting, or not, the desired characteristic. The rule of thumb is that the higher the agreement between human annotators, the more successful the machine learning algorithm is likely to be.

Prior research suggests that only approaches that break down complex judgments like “quality” into clearly defined, multiple dimensions are likely to lead to high agreement between annotators. For instance, Devaul and colleagues looked at the inter-annotator agreement on a judgment task that is very much akin to assessing resource quality: deciding if a resource supports a particular educational standard [13]. For this type of broad stroke judgment, inter-annotator agreement was very low, averaging only 32%. Reitsma and colleagues took a slightly different approach to the same standards alignment problem and used a theoretical model to break this challenging human judgment task down into nine more focused dimensions. They were able to achieve inter-annotator reliability ratings between 61% and 95% for their different alignment dimensions [46]. Bethard et. al. took a similar approach, though focused on a somewhat different judgment task (suitability for classroom use versus alignment to a standard) [2]. However, instead of basing the finer-grained dimensions of human judgments on a theoretical model, they conducted an empirical study to identify the most salient dimensions (this is discussed in Chapter 3). As our annotation data shows, this approach made it possible to develop a detailed annotation protocol which yielded very high inter-annotator reliability.

Other research efforts seek to examine the correlation between low level features amenable to machine recognition with higher-order human decisions. For example, Ivory and colleagues showed that low-level design issues, such as the amount and positioning of text, or the overall portion of a page devoted to graphics, correlated highly with expert judgments of overall site quality [30].

2.3 Automatic Quality Assessment on Textual Documents and Web Sites

Other work has explored the feasibility of training models to support quality assessments in various contexts. Most approaches have treated the quality assessment problem as a document classification
task, leveraging machine learning techniques to distinguish between high quality and low quality documents based on document content and available meta-data; this is also my approach. But because of the widely varying document types, available meta-data, and underlying definitions of quality, there is no over-arching framework that governs people’s efforts in this area; the systems are generally very task-specific.

Algorithms for assessing web site quality in its many definitions have been broadly proposed ever since web usage entered mainstream society. In the simplest scenario internet search is driven by determining the presence of keywords on web pages. However, the pages that are relevant to a query now frequently number in the millions, and ranking the results by criteria including their assumed “quality” has been an integral part of internet search engines for a long time. For example, Google’s at the time revolutionary PageRank algorithm judged a site to be more likely to provide high quality results if many other sites link to it [5]. Quality metrics such as the time since last update of a web page, presence of broken links, and the amount of textual content have also been shown to improve results on internet search [57].

More recent research has attempted to ground web page quality in existing information quality models, and found that quality of web pages needs to be considered in the context of the domain or application on one side and the targeted audience on the other side [35, 33]. These studies confirm that metrics for overall “web site quality” are not specific enough to give highly useful quality assessments for a specific task; even so, some of the very general features, such as PageRank, may be useful when combined with more specific ones.

The popularity of the online encyclopedia Wikipedia has prompted research into its overall quality of information, and more specifically into evaluating the quality of individual articles. The Wikipedia community maintains a set of featured articles, which are considered to be the best that Wikipedia has to offer; they are selected by community review according to a set of well defined criteria (e.g. well written, factually accurate, follows Wikipedia style guidelines) [55]. Stvilia et. al. used textual features like readability and metadata features such as number of editors and age of the article in combination with machine learning techniques to predict featured article status of Wikipedia articles with a relatively high accuracy of 86%[49]. Blumenstock later showed that the task can be accomplished with almost perfect accuracy (up to 97%) using only the word count of an article as a feature [3]. He also suggested that this unusual phenomenon is due to
the particular way in which Wikipedia articles evolve: long articles have received a lot of editorial attention by many people, and as a result they are generally of higher quality. This example shows that extremely superficial features of a document can have strong predictive power in practice, though it may not generalize to other tasks — it is very unlikely that document length alone would be useful in assessing quality aspects of educational resources on the web, as these are the result of a very different editorial process.

2.4 Information Quality Frameworks

The multi-faceted nature of quality has been a focus of research in many areas. The field of information quality tries to come up with frameworks for describing what quality means, from a generalized perspective. Often these are focused on structured data, such as data points in a database (e.g. how accurate or precise the data points are, or how complete a data set is). Information in that sense is used as a fairly general, abstract concept, “mathematical” one might say. Eppler and Wittig review and compare a number of such frameworks with different target domains[21].

In contrast with such efforts, in the context of this research, the concept of “quality” is not so much concerned with information being accurate and precise, but with how it is encoded in text — how well does a resource convey information, not what information, exactly, is contained. If we are willing to assume that creators of resources are generally well informed and mean well, then the quality question becomes one of mostly pedagogical utility.
Chapter 3

Characterizing Quality

This chapter reports on an extensive study of expert processes that was conducted by Bethard et. al.[2], and which forms the basis of my methodology. The study aimed to identify the characteristics most commonly used by secondary level science educators and science librarians when identifying resources for use in the classroom. Among other things, this resulted in a list of low-level indicators of quality of a resource, which are known to correlate strongly with experts’ quality judgments. These indicators are concrete enough to be consistently annotated by humans and to be recognized without deep expertise, and they cover a range of the relevant dimensions of quality, so they make a promising basis for an automatic system to characterize a resource’s quality.

The study was done in two parts: In the first part, data was gathered from three prior projects where participants were involved in assessing resource quality for use in educational settings. The compiled data was analyzed for common criteria that participants brought to bear on this complex judgment task. In addition to identifying common high-level dimensions, the lower-level indicators or features of resources that people draw on to inform their judgments were identified. The first part of the study produced an initial cut at the dimensions and indicators that might comprise a rich, multi-dimensional characterization of resource quality. The second part consisted of a mixed-method study of experts with significant experience in selecting online educational resources for classroom use. The purpose of this part was to verify and refine the initial cut of quality dimensions and indicators, and to gather more detailed information on how well (or not) different dimensions and indicators correlated with overall quality judgments.
3.1 Analysis of Prior Studies

The study began by analyzing verbal data collected from three sources:

**Educator Reviews for Digital Water Education Library (DWEL):** A total of 364 reviews, generated by 21 reviewers for 182 URLs, all of which were directed at grades 9-12.

**Educator Reviews for Climate Change Collection (CCC):** Narrative comments concerning digital resource quality from 55 individual reviews by 4 reviewers for 28 digital resources, targeted at grades 9-12.

**Educator Focus Groups:** In 2002, Sumner and colleagues hosted a series of focus groups where science educators discussed the quality of digital library resources [51] for the purpose of directing the review and curation process of the Digital Library for Earth System Education (DLESE). This resulted in transcribed verbal data of 38 educators as they reviewed 18 resources.

This verbal data was coded and filtered to leave only comments that were related to the quality of the resource. Then the most important dimensions of quality were derived in an iterative approach: Comments were grouped by similarity. The resulting categories were iteratively adjusted to best fit the data, until the researchers reached agreement. This resulted in 25 categories, or **dimensions of quality**; also, all the comments that were assigned to each dimension were available. Table 3.1 shows some examples of such educator comments.

The top 12 dimensions, based on the frequency with which they were observed in comments, were selected for further investigation. These dimensions, listed in Table 3.2, cover a wide range of quality concerns, from technical (e.g. readability and graphics) to content (e.g. presence of activities and key ideas) to contextual (e.g. source identification and real-world applications). Overall, the 12 dimensions above accounted for 78% of all the comments about resource quality in the three studies.

\footnote{1 Table as presented in [2]}
### Sample Dimensions

<table>
<thead>
<tr>
<th>Sample Dimensions</th>
<th>DWEL Reviews</th>
<th>CCC Reviews</th>
<th>Educator Focus Groups</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appropriate inclusion of graphics</td>
<td>Highly visual site...</td>
<td>The movies/graphics are well done, and seem to enhance understanding of the text</td>
<td>I would have liked to see graphics</td>
</tr>
<tr>
<td>Readability of text</td>
<td>Provides several interactive visuals that prompt students to think</td>
<td>Could be formatted differently to be easier to read. The text is dry and technical.</td>
<td>Hard to read with busy background</td>
</tr>
<tr>
<td>Focuses on key content</td>
<td>Does not take advantage of the internet, is difficult to read, and has no illustrations.</td>
<td>Looks like a really good activity for seeing... if changes in climate are part of global warming or natural variability.</td>
<td>... lessons looked like they would help convey important information in an engaging way</td>
</tr>
</tbody>
</table>

Table 3.1: Sample qualitative data drawn from raw data sources for three dimensions

These are some example comments that were identified as being relevant to one of the quality dimensions.¹

<table>
<thead>
<tr>
<th></th>
<th>Name</th>
<th>%</th>
<th>Overall</th>
<th>Accept</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Provides access to relevant data</td>
<td>6.1</td>
<td>0.67**</td>
<td>0.46*</td>
</tr>
<tr>
<td>2</td>
<td>Good general set-up</td>
<td>3.6</td>
<td>0.65**</td>
<td>0.73**</td>
</tr>
<tr>
<td>3</td>
<td>Appropriate inclusion of graphics</td>
<td>13.4</td>
<td>0.61**</td>
<td>0.53*</td>
</tr>
<tr>
<td>4</td>
<td>Robust pedagogical support</td>
<td>4.1</td>
<td>0.59**</td>
<td>0.48*</td>
</tr>
<tr>
<td>5</td>
<td>Appropriate pedagogical guidance</td>
<td>4.1</td>
<td>0.57**</td>
<td>0.52*</td>
</tr>
<tr>
<td>6</td>
<td>Reflects source authority</td>
<td>7.7</td>
<td>0.56**</td>
<td>0.54*</td>
</tr>
<tr>
<td>7</td>
<td>Readability of text</td>
<td>10.7</td>
<td>0.54**</td>
<td>0.40</td>
</tr>
<tr>
<td>8</td>
<td>Appropriateness of activities</td>
<td>2.6</td>
<td>0.49*</td>
<td>0.53*</td>
</tr>
<tr>
<td>9</td>
<td>Focuses on key content</td>
<td>13.1</td>
<td>0.42*</td>
<td>0.32</td>
</tr>
<tr>
<td>10</td>
<td>Age appropriateness</td>
<td>5.3</td>
<td>0.41*</td>
<td>0.26</td>
</tr>
<tr>
<td>11</td>
<td>Inclusion of hands-on activities</td>
<td>4.4</td>
<td>0.36</td>
<td>0.43*</td>
</tr>
<tr>
<td>12</td>
<td>Connections to real-world applications</td>
<td>3.0</td>
<td>0.36</td>
<td>0.27</td>
</tr>
</tbody>
</table>

Table 3.2: The 12 most salient dimensions of quality

These are the 12 dimensions of quality accounting for the most comments. The “%” column shows what percent of comments each dimension accounted for. The “Overall” and “Accept” columns show the dimensions’ relationship to experts’ assessment of overall quality and their decision to accept or reject a resource (*p < .05, **p < .01). The dimensions are ordered by their correlations with overall quality.¹
3.2 Expert Study

To further verify and refine the dimensions of quality, a mixed-method study of digital library curation experts was then performed. Eight digital library experts with significant experience as science educators or instructional designers were asked to perform a variety of quality judgments on a series of digital resources. The resources were taken from the Digital Library for Earth System Education (DLESE) [14], a repository of digital educational resources about Earth science. They were selected to include resources that had been peer-reviewed and identified as being of high quality, and ones that had been rejected from DLESE for being of too low quality; it was not revealed to the experts which resource belonged to which category.

The experts were asked to evaluate the quality of six resources; while they examined them, their comments about positive and negative aspects were recorded. In the end they were asked to rate each resource’s quality on a scale from -3 to +3, and to decide whether they deemed the resource good enough to be accepted for inclusion into a digital library collection. Afterwards the experts were again presented with three of the six resources (the top ranked, the bottom ranked, and one average resource) and asked to rate them on each of the 12 quality dimensions (from -3 to +3); as before, their comments were recorded.

Table 3.2 shows how the experts’ quality dimension ratings compared to both the overall quality ratings and the accept/reject decisions. The “Overall” column indicates the correlation between the expert’s dimension ratings and their overall resource quality ratings. The “Accept” column indicates the correlation between the dimension ratings and the decision to accept or reject the resource from a digital library. Ten of the twelve dimensions were significantly correlated with the experts’ overall quality rating, and 8 of the 12 dimensions were significantly correlated with their accept/reject decisions.

3.3 Indicators of Quality

The expert study confirmed that quality could be decomposed into meaningful dimensions that are more detailed and targeted than the abstract concept of “quality”. However, even the list of the 12 most important dimensions includes fairly abstract dimensions like “Good general set-up”. To make a computational approach to quality feasible, it was necessary to push the decomposition of quality further, identifying
<table>
<thead>
<tr>
<th>Indicator</th>
<th>Example expert comments of presence (+) or absence (–)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Has sponsor</td>
<td>(+) I ... look at the URL just to kind of get an idea of where I'm at, you know, is it like a NOAA site</td>
</tr>
<tr>
<td></td>
<td>(–) It looks like there's links, but I still don't know who they are.</td>
</tr>
<tr>
<td>Has prestigious sponsor</td>
<td>(+) That is a NOAA site... I generally think what NOAA offers up is good quality.</td>
</tr>
<tr>
<td></td>
<td>(–) If it said NOAA or something like that I would say, “Okay. This is USGS, this is NOAA”</td>
</tr>
<tr>
<td>Has instructions</td>
<td>(+) Well I’m looking and it’s talking about how to use the software and I like that</td>
</tr>
<tr>
<td></td>
<td>(–) Okay, what am I supposed to read here? What am I supposed to start with?</td>
</tr>
<tr>
<td>Identifies learning goals</td>
<td>(+) Again very high quality in terms of what it’s addressing and what the objectives are.</td>
</tr>
<tr>
<td></td>
<td>(–) ... it would help me if I had some information about where this might fit in the curriculum.</td>
</tr>
<tr>
<td>Identifies age range</td>
<td>(+) Well that’s kind of interesting, and it gives grade level which is very nice</td>
</tr>
<tr>
<td></td>
<td>(–) None of these are really showing me grade level which I’m kind of disappointed</td>
</tr>
<tr>
<td>Organized for learning goals</td>
<td>(+) So, you see how the objectives match up with the worksheets and the procedure.</td>
</tr>
<tr>
<td></td>
<td>(–) I’m a little unclear as to what’s going on ... and how it connects to the rest of the content.</td>
</tr>
<tr>
<td>Content is appropriate for age</td>
<td>(+) I think that’s a good middle school activity, looking at climate change.</td>
</tr>
<tr>
<td>range</td>
<td>(–) I really don’t like this already... I think middle school kids are probably beyond this.</td>
</tr>
</tbody>
</table>

Table 3.3: Comments on the presence (+) or absence (–) of quality indicators

These are examples of comments that were identified as either indicating the presence or absence of a quality indicator.¹
low-level **indicators** of quality that were concrete, easily recognizable, and yet known to be useful for making quality judgments.

The analysis of previous studies had indirectly resulted in candidates for such indicators: There, reviewers’ comments that were similar were grouped together. Similar groups were then iteratively combined until only a small number of overarching groups remained: the quality dimensions. But all the intermediate groups that were combined into the quality dimensions were known, each representing a more narrow, more specific aspect of the dimension they were a part of. These intermediate groups could serve as lower level, more specific **quality indicators**. The remaining problem was to identify which of these were most important for characterizing quality.

To answer this question, all of the recorded comments from the assessments of overall quality during the expert study were hand-coded to identify where an expert mentioned a quality indicator as being either present or absent in the resource. Table 3.3 shows some examples of such indicators and comments about their presence or absence. Using the aggregated counts, it was possible to identify the indicators where both the presence was highly correlated with acceptance and the absence was highly correlated with rejection.

This resulted in a list of low-level quality indicators that provide a concrete definition of quality which corresponds strongly to the expert processes. They provide a set of characteristics that identify the conceptual aspects of a resource that are likely to be considered when judging the quality of a resource. In addition, they provide a means of characterizing quality in terms of low-level features that should be more amenable to computational approaches.
Chapter 4

The DLESE Corpus

The first step in creating a computational system to characterize the quality of educational resources using the quality indicators the expert study identified was to build a corpus that would be suitable for training and evaluating machine learning models to recognize the presence of each of a set of indicators in a resource. In order to build such a corpus, we needed to:

- select a test bed in the domain of educational digital libraries that would offer a varied selection of resources
- choose a set of quality indicators to annotate that computational methods would be likely to be able to learn, and that would likely be useful in a real-world quality task
- create annotation guidelines that would help our experts annotate resources consistently and objectively.

4.1 DLESE

The Digital Library for Earth System Education (DLESE) is a digital repository of educational resources in the domain of Earth science. DLESE’s development started in 1999, and it was envisioned as a “single comprehensive online source for geoscience education, aggregating a wide variety of pedagogically sound, technologically robust, and scientifically accurate resources, collections of resources, datasets, services, and communications to support inquiry-based, active, student-centered learning about the Earth system”[16]. Its catalog contains over 13000 educational resources organized in 38 collections and provides
extensive meta-data to its resources, including information such as the intended grade level and alignment with educational standards. Beyond simply providing a repository of information, DLESE has served as the basis for many experimental and ground-breaking efforts to involve the wider community of educators, students and scientists more directly in resource curation and dissemination, through initiatives such as the Curriculum Customization Service, which provides a collaborative platform for bringing open educational resources into the classroom[50]. Initially funded by the NSF, DLESE is now operated by the National Center for Atmospheric Research (NCAR); it serves as the geoscience node of the National Science Digital Library (NSDL).

4.1.1 The DLESE Community Collection

One of the collections within DLESE is the DLESE Community Collection (DCC). It contains interdisciplinary resources with a general focus on “bringing the Earth system into the classroom”, demonstrating “the application of science to solving real world problems”. In order to be considered for inclusion, resources should be “well-documented, easy to use, bug-free, motivational for learners, pedagogically effective, scientifically accurate”, and “foster mastery of significant understandings or skills”. Examples for resources that are found in the DCC can be seen in Figures 4.1 and 4.2. A defining characteristic of the DCC collection within DLESE is that it includes resources that were submitted by individual DLESE users [15].

All submitted resources are currently manually reviewed by committee for compliance with the standards of DCC content. After the decision has been made to include a resource in the collection, it is then annotated with various metadata describing the new catalog entry. Because of the widely varying nature of resources, the wide range of topics, and the distributed nature of the submission process, we chose the DCC to be our test-bed.

Initially, our aim was that the corpus should represent the typical distribution of resources that a reviewer for the DCC might encounter. Thus it needed to include not only resources that had already passed the review process, but also ones that were considered for inclusion, but ultimately rejected. Submissions to the DCC are guided by a very explicit collection scope statement, clearly indicating to submitters what
the expectations are for a resource to be accepted; probably because of that, relatively few resources are submitted that are not good candidates for inclusion. At the time when we selected resources for our corpus, the rate of rejection was determined by DLESE reviewers to average around five percent.

Reflecting this skewed distribution, we selected 1000 educational resources directed at high school students: 950 were selected randomly from DCC, and 50 were selected from resources that had been submitted and rejected. When a DCC reviewer decides to reject a resource, they write a short free-form note explaining their reasons. Common reasons for rejection, besides quality-related problems, are: the resource is outside the scope of DLESE; the type of the resource is one not cataloged by DLESE; or the resource suggested is already in the collection. Based on the reviewer’s notes we only selected resources that were rejected for what appeared to be quality-related reasons. In addition to these 1000 resources, we selected a small set of 48 resources to be used only to test and refine the annotation protocol.

4.2 Quality Indicators

The expert study this work was motivated by identified experts’ quality considerations on multiple levels: higher-level quality dimensions such as “provides robust pedagogical support” and “reflects source authority”, and lower-level quality indicators such as “has instructions” or “identifies age range”. The quality dimensions represent a level at which an expert might break down the problem of quality assessment, and they necessitate complex analyses of resources, using deep understanding, and expert knowledge and experience. The quality indicators, on the other hand, are simple, low-level, concrete measures which, as the study found, strongly feed into higher level quality assessments. For example, whether or not a resource has instructions might be identified by only skimming a resource, mostly relying on certain words to show up; but if a resource does have instructions, that strongly supports the hypothesis that a resource offers pedagogical support (although many other things feed into that as well).

It was the goal of our annotation project to produce a corpus that would be suitable for training and evaluating models to recognize some of the most important quality indicators that the expert study identified. We wanted to focus on the indicators that the study had found to be most influential on the educators’ ultimate judgment, but other criteria also influenced our selection: For one thing, there would
Module 1 Lesson Guide

These Lesson Guides include written and computer activities on concepts of remote sensing in general and NASA’s Mission to Planet Earth. More information about Mission to Planet Earth can be obtained on the World Wide Web through NASA’s Spacelink site (URL: http://spacelink.msfc.nasa.gov/). Spacelink lists educational materials which are available through NASA's network of Teacher Resource Centers and through NASA’s Central Operation of Resources for Educators (CORE), at the address below:

NASA CORE  
Lorain County JVS  
15181 Route 58 South  
Oberlin, OHIO 44074  
Tel: (216) 774-1051 (ext. 293/294)  
Fax: (216) 774-2144

Section A - What is Mission to Planet Earth?

This is a written activity asking the students to consider what about the earth they would want to study.

Section B - Viewing the Earth from Space

Combines written activity on the Galileo spacecraft with a computer activity. Students will view images of the earth taken as the spacecraft flew by us on December 8, 1990. Students receive their first introduction to the image processing programs as they view the two earth images. One view was taken using optical filters and one was taken using an infrared filter. Students are asked to discover what differences in information about the earth the two images reveal. The images used are GalOptic.gif (115K GIF) and GallInfra.gif (164K GIF).

Section C - Remote Sensing

Students are introduced to the idea that images are often displayed in 256 shades of gray, with each shade (DN or Digital Number) representing a light level. They work with several software tools and become comfortable opening files and applying various image processing techniques. This lesson will almost certainly take more than one day to complete. The images used are GalOptic.gif (115K GIF), GallInfra.gif (164K GIF), and GalAndes.gif (151K GIF).

Section D - Zoom-In on Los Angeles

In this section, students work with whole earth optical images and then open up their first radar image, seeing first a SEASAT image of Los Angeles taken in 1978 and then a closeup view of Elysium Park and Dodger Stadium.

Figure 4.1: DLESE example resource 1: NASA SIR-C Education Program Student Lesson

This resource presents a number of guided activities for students focusing on remote sensing. Each section is introduced with a summary of what is involved in the activity and what students are expected to get out of it. Such activities can be used by teachers in the classroom as they are.
The UCAR Digital Image Library offers a large variety of images on topics such as atmospheric phenomena, climate, and pollution, as well as a range of other topics outside the domain of earth science. These images can be a valuable resource to teachers and students by providing material for presentations and discussions.\[53\]
These seven quality indicators were selected to be annotated in the DLESE corpus. have been little benefit in spending time annotating indicators that are computationally trivial (e.g. “does the resource have broken links”). While this information is important, it would be a waste of effort to have a human annotator determine if links are broken, if the same can be done automatically with almost the same accuracy. Conversely, indicators of a deeply semantic nature (e.g. “does the resource use examples that intuitively make sense”) would have been very hard to annotate with a high degree of consistency, as the assessment is highly subjective. It would also have been unlikely that we’d be able to achieve good performance using computational methods. Our final set of seven indicators contained some that were easier to annotate consistently, and some that were more subjective. Table 4.1 lists the selected indicators, and I will describe them in detail in the following sections.

Two people with previous experience cataloging DLESE resources were asked to judge the presence or absence of the seven quality indicators on each resource. In order to achieve reliable results we carefully formulated instructions for annotation, outlining our definitions for each indicator using concrete terms and examples taken from our expert study. After a short test-run and in cooperation with DLESE experts we slightly revised these annotation guidelines to avoid ambiguities. These are the seven indicators we selected, and how we directed the annotators to evaluate each of them:

### 4.2.1 Has instructions

The annotators were instructed to label a resource as having instructions if the resource explains how the content should be approached by the user. For example, a resource may indicate the sequence in which

<table>
<thead>
<tr>
<th>Quality Indicator</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Has Instructions</td>
</tr>
<tr>
<td>2 Has Sponsor</td>
</tr>
<tr>
<td>3 Has Prestigious Sponsor</td>
</tr>
<tr>
<td>4 Identifies Age Range</td>
</tr>
<tr>
<td>5 Content is not Inappropriate for Age</td>
</tr>
<tr>
<td>6 Identifies Learning Goals</td>
</tr>
<tr>
<td>7 Organized for Learning Goals</td>
</tr>
</tbody>
</table>

Table 4.1: List of quality indicators
the user should visit its pages, describe the steps in a classroom activity, or explain how to install and use a piece of software.

4.2.2 Has sponsor

A resource that has a “sponsor” in this sense explicitly identifies a publisher, who has chosen to make this content publicly accessible; they may not be the creator of the content and may or may not have any cognitive authority in the respective field. For example, a page may state that it is organized or maintained by a professor, a public school, a research lab (e.g. NOAA or NASA), or a politician or celebrity (e.g. Al Gore). Wikipedia-like sites where the content is managed by the community are not considered to have sponsors. Being part of a particular domain is also not sufficient for having a sponsor – the content must be explicitly attributed to someone.

When we conducted the annotation, we chose to call this quality indicator “has sponsor”, and for that reason I will continue to use this term here. However, we have subsequently found that a more appropriate way to name this indicator (as we intended and annotated it) would be “has publisher”.

4.2.3 Has prestigious sponsor

A resource has a “prestigious sponsor” if the manager or organizer of the page is respected in the field of study relevant to the resource. For earth systems resources, prestigious sponsors could include entities like NASA, USGS or NOAA, but also respected universities (when the resource is maintained by university itself – not just a student’s webpage).

This indicator is orthogonal to the “has sponsor” indicator above, as that one refers to a sponsor more in the sense of a publisher, while this indicator sees a sponsor as an entity that takes responsibility for the content of the resource. As confirmed by the inter-annotator agreement numbers below, this is the most ambiguous of the selected indicators; it’s assessment strongly depends on which organizations an annotator is familiar with.
4.2.4 Identifies age range

A resource identifies its age range if the text states the expected age or grade level of its intended users, or if the resource is divided into sections targeted to users with different levels of knowledge.

4.2.5 Content is not inappropriate for age range

Rather than asking them to positively judge if a resource is appropriate for the given age range, which is a complex question that even education experts often will not agree on, annotators were instructed to label this as “yes” if the resource is not obviously inappropriate for its age range, that is, if someone with little expertise in education would judge that the reading or activities were neither much too hard nor much too easy for the given grade level. For example, playing with clay is generally inappropriate for high school audiences, and very technical terminology is inappropriate for elementary school students. In both these cases, the resource should be annotated as “No”. A resource that has no obvious problems would be annotated as “Yes”.

This indicator is dependent on the “identifies age range” indicator: if a resource does not identify a target age range, this is to be labeled as “n/a”.

4.2.6 Identifies learning goals

A resource that identifies learning goals articulates the knowledge and skills a student is expected to acquire over the course of using the resource. Specific state or national standards may be given, or the resource may more informally identify its best use in an educational setting.

4.2.7 Organized for learning goals

A resource is organized appropriately for its learning goals if the site is clearly organized so that each goal has a corresponding description or activity. This often means that each learning goal is addressed under a separate heading, tab or page. A page with a single learning goal may be appropriately organized if the headings, etc. give useful sub-structure to the learning activities.
This indicator is dependent on the “identifies learning goals” indicator: if a resource does not identify learning goals, this is to be labeled as “n/a”.

4.3 Annotation Protocol

First we conducted an initial test run, where each annotator was given the same 48 resources. By comparing their performance on this small set, and through interviews with them, we refined the annotation protocol and instructions to make it easier for them to annotate consistently, and to make sure that their annotations captured what we were aiming for. The 48 resources from the test run were not used in the subsequent annotation steps and are not counted as part of the 1000 resources in the corpus.

After the initial test run, each annotator was asked to independently look at a subset of the 1000 selected resources; they were presented with the home page of the resource and allowed to navigate freely. Every resource was annotated by at least one annotator, and 200 were double-annotated to allow us to measure agreement between the two annotators; thus each annotator saw a total of 600 resources. If agreement on an indicator is low, it either indicates that the annotation guidelines for that indicator were too inexact, thus letting each annotator come up with their own interpretation, or that annotating that indicator is inherently difficult for people, and their judgment is subjective. It is commonly assumed by the natural language processing research community that the higher the agreement between annotators, the more likely a machine learning system will be able to also perform well on it.

In addition to the annotators’ quality indicator judgments, we also recorded the URLs of all web pages that they visited during their review and that they marked to be part of the resource. DLESE only stores the URL of first entry into a resource; but many resources consist of multiple linked pages. Automatically identifying the extent of a resource is a complex problem in itself [20, 17]; having this information about the annotators’ navigation of resources available has been useful in evaluating the impact of different ways of crawling resources.

After determining the inter-annotator agreement (reported in the next section) we asked both annotators to co-operatively cross-check the resources and indicators where they disagreed, and discuss and resolve
their disagreement. The resulting set of 200 double-annotated and cross-checked resources can be assumed to be more consistently annotated. These 200 resources were chosen to serve as the test set.

4.4 Evaluation of Annotation Results

Table 4.3 shows inter-annotator agreement for each indicator, as well as the percentage of resources where the indicator was marked as present. The agreement is calculated as the number of resources on which both annotators gave the same response, divided by the total number of resources. The table also includes Cohen’s $\kappa$ values[8]. Cohen’s kappa is a statistical measure of inter-rater reliability, which differs from simple agreement numbers because it takes agreement by chance into account; a high $\kappa$ value indicates that not only did the annotators agree much of the time, it is also unlikely that they did so by chance.

During the annotation, a small number of resources (less than 1% of the complete set) were offline for one or both annotators. Those resources were not annotated. This reduces the number of double annotated resources to 198.

Furthermore, on two of the indicators, “not inappropriate for age range” and “organized for learning goals”, a possible answer was “n/a” (if the indicator they depend on was labeled as “no”). When computing the agreement number for those two indicators, resources with a label of “n/a” by either annotator were excluded from the calculation. This left 33 of the 198 double-annotated resources for “not inappropriate for age” and 32 for “organized for learning goals”.

Agreement was above 80% for 6 of the 7 indicators, suggesting that our guidelines were clear and our characterization of quality was not too subjective. The indicator “has prestigious sponsor” was not as consistently annotated, which seems to confirm the intuition that this measure is more subjective, as it depends on the familiarity of the annotator with the domain the resources are sampled from. This is supported by the analysis in Table 4.2: Both annotators had experience with metadata annotation on digital library resources, but annotator 2 was less familiar with the field of earth science than annotator 1. As a result, annotator 2 answered “no” on “has prestigious sponsor” 159 out of 198 times, while annotator 1 was more likely to recognize a sponsor as “prestigious”. During the adjudication process, 52 of the 69 cases of
disagreement on this indicator were decided in favor of annotator 1. The $\kappa$ values indicate fair to substantial agreement on most indicators.\(^1\)

### 4.5 Quality Indicator Predictiveness

In order to confirm that the indicators capture aspects of a resource relevant to quality-related judgments, as the expert study suggested, I performed an analysis of their predictiveness of a resource being considered of high or low quality on a given task: As explained in Section 4.1.1, of the annotated corpus, 950 resources were randomly selected from the DCC, and 50 resources were not allowed into DLESE for quality reasons; from the perspective of a DCC curator, the 50 rejected resources are of lower quality than the rest. Using a simple linear kernel Support Vector Machine, with only the annotated quality indicators as features to characterize a resource, I attempted to predict whether a resource was accepted into the DCC.

In addition to looking at the predictiveness of all the indicators in combination, I performed a leave-one-out analysis, by in turn excluding each of the seven indicators from the features, predicting acceptance

\(^1\) These characterizations of $\kappa$ values are based on Landis & Koch[36]. They define values of 0 – 0.2 as slight agreement, 0.2 – 0.4 as fair, 0.4 – 0.6 as moderate, 0.6 – 0.8 as substantial, and 0.8 – 1.0 as “almost perfect” agreement. Note that these characterizations are fairly arbitrary and by no means universally accepted.

<table>
<thead>
<tr>
<th>Quality indicator</th>
<th>present in agreement</th>
<th>Cohen’s $\kappa$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1  Has instructions</td>
<td>39%</td>
<td>85.2%</td>
</tr>
<tr>
<td>2  Has sponsor</td>
<td>97%</td>
<td>99.5%</td>
</tr>
<tr>
<td>3  Has prestigious sponsor</td>
<td>34%</td>
<td>63.6%</td>
</tr>
<tr>
<td>4  Identifies age range</td>
<td>20%</td>
<td>87.3%</td>
</tr>
<tr>
<td>5  Not inappropriate for age</td>
<td>99%</td>
<td>100.0%</td>
</tr>
<tr>
<td>6  Identifies learning goals</td>
<td>28%</td>
<td>83.1%</td>
</tr>
<tr>
<td>7  Organized for goals</td>
<td>76%</td>
<td>80.6%</td>
</tr>
</tbody>
</table>

Table 4.3: Quality indicator presence in resources and inter annotator agreement
of a resource using only the remaining six indicators. The results of this analysis can be seen in Table 4.4. These experiments were run on a reduced training set of 100 resources: 50 randomly selected from the set of accepted resources, and 50 rejected ones.

The seven quality indicators together were able to accurately predict whether a resource was ultimately accepted into the DCC with an accuracy of 71%. Phrased in another way: given a resource to be reviewed, if we never look at the resource itself, but we know if it has instructions, if it has a sponsor, if it has a prestigious sponsor, if it identifies a target age range and isn’t clearly inappropriate for it, and if it identifies learning goals and is structured for them, we can predict if it is good enough for inclusion in the collection in about 71% of cases. This is encouraging, as it shows that the quality indicators truly capture relevant aspects of quality. We also see that, while all of the indicators contribute to the outcome, for this specific task of identifying the five percent of rejected resources the indicator “has prestigious sponsor” was by far the most useful – if we leave it out, the predictiveness of the remaining six indicators drops by 16%. This is particularly remarkable when considering that the annotation on “has prestigious sponsor” was the noisiest during annotation, with the largest amount of disagreement.

I also want to point out that, clearly, the selected set of seven indicators is not exhaustive. As discussed in Chapter 3, they were among the most useful in predicting experts’ quality judgments, but they only partially cover some of the 12 dimensions of quality that were identified in the study. Future work will explore an extended set of quality indicators to cover all of the 12 dimensions; some of these additional indicators may be recognizable by a machine learning based approach, others may leverage other methods.

<table>
<thead>
<tr>
<th></th>
<th>accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>all indicators</td>
<td>71%</td>
</tr>
<tr>
<td>w/o Has instructions</td>
<td>-5%</td>
</tr>
<tr>
<td>w/o Has sponsor</td>
<td>-2%</td>
</tr>
<tr>
<td>w/o Has prestigious sponsor</td>
<td>-16%</td>
</tr>
<tr>
<td>w/o Identifies age range</td>
<td>-7%</td>
</tr>
<tr>
<td>w/o Not inappropriate for age</td>
<td>-2%</td>
</tr>
<tr>
<td>w/o Identifies goals</td>
<td>-4%</td>
</tr>
<tr>
<td>w/o Organized for goals</td>
<td>-4%</td>
</tr>
</tbody>
</table>

Table 4.4: Quality indicator predictiveness and leave-one-out analysis
(e.g. cohesion metrics). But at this stage my focus is only on the seven selected indicators, to achieve a **partial** characterization of a resource's quality.
Chapter 5

Stage 1: Baseline System

The methodology of this work is based on supervised learning, and is similar to approaches people have followed for tasks of document classification. In the supervised learning paradigm, we begin with a corpus of instances (i.e. documents, or resources, in this case) for which the proper classification or label (i.e. whether or not a specific indicator is present in this resource) is known. Using numerical or categorical features that adequately describe what we consider relevant about each resource, machine learning algorithms create statistical models, which can be used to predict the proper classification, or label, based on the respective values of these features. In general, being able to make good predictions on unseen instances requires a training corpus that is sufficiently diverse to more or less cover the space of possible inputs. A large variety of these machine learning algorithms exist, which vary in a number of ways: what the computational requirements are for training a model, and for applying it to make predictions; what kind of features they can handle; what kinds of predictions they can make (“yes” / “no”, categorical, numerical, i.e. regression); and under what conditions they are likely to produce accurate results, to name a few. Some of the more common ones are Decision Trees, Artificial Neural Networks, Support Vector Machines (which most of this work uses), or Maximum Entropy[38, 54]. Most algorithms internally operate on feature vectors, that is, the features describing an instance are represented as a vector in (possibly very high-dimensional) Euclidian space.

One non-trivial problem, then, is to find a way to encode the relevant information about a resource (i.e. everything that might help us determine if an indicator is present or absent) into a numerical vector. If we consider all the information that can be taken or inferred from a resource by a skilled reader, then
ideally, we would want all of that information to also be inferable from the corresponding feature vector by the machine learning algorithm. Unfortunately this is infeasible, for at least two reasons: First, a machine learning algorithm doesn’t have the deep understanding a human expert does, both domain and more general “common-sense” knowledge that’s rooted in the human experience of the world. And second, machine learning algorithms are severely limited in their ability to do “deep inference”, i.e. multi-step reasoning. These limitations are not as serious in problems that are inherently numeric to begin with and that our expertise as humans doesn’t translate to easily (example), but when dealing with natural language, as is the case with almost all educational materials, they pose a big problem. Nonetheless the natural language processing research community has identified a number of ways to partially encode text such that, while much of the semantics are lost, a few salient aspects are preserved and become accessible to a machine learning algorithm. Such surface level features have been used with sometimes surprising success in a number of domains (examples). But the only way to determine which of these surface level features work well on a new problem is to experiment.

My methodology, then, is to begin with the corpus we annotated on DLESE (so it has the labels we require for the supervised learning approach). We split the corpus into a training and a testing portion – they need to be separate to assure we report fair numbers. If we were to use any of the testing portion during creation of models, the reported numbers would be unfairly good, because if we were to use our models in a realistic scenario, we’d have no up-front knowledge about the resources they will be applied to in the future. So we extract features from each of the resources in the training set into one feature vector each, label them according to the quality indicator annotation, and feed this into a machine learning algorithm. This produces models – one for each quality indicator. Then, for each resource in the test set, we again extract features into a feature vector, feed this vector into each of the models and get a prediction for each indicator: is it present or absent? By comparing the models’ answers to the ones the annotators gave, we can determine in what percentage of cases they agreed. While the annotators’ answer isn’t always the best one, we can generally assume that if the models agree more with the annotators the model output is more accurate.
In the following section I will describe my first, baseline system: how the corpus was split for training and testing, how each resource was collected and pre-processed, the feature set, and the machine learning setup. The goal of this system was to apply a document classification approach to the DLESE data set in a straight-forward fashion.

5.1 Corpus Preparation

The DLESE corpus consists of 1000 annotated resources. 200 of these 1000 were annotated by both annotators, then disagreements were resolved through discussion. The remaining 800 resources were annotated by one of the annotators only. For the purpose of this first round of experiments, I decided to only use the 800 single-annotated resources in order to reserve the 200 double-annotated ones for later evaluations.

The compiled corpus created during the DLESE annotation in fact does not contain the resources themselves. Similar to the digital library it was based on, it contains references by URL to resources, and associated meta-data (in this case, the answers of each of the annotators). Thus the first step is to retrieve the content of each resource for further processing. This brings up the first question: which of the pages of a resource should be included? Rarely does a resource only consist of a single page. Most of the time, other pages on the same site are linked, and at least some of them are integral parts of the resource without which a fair evaluation is not possible. During the annotation project we acknowledged this fact by allowing the annotators to navigate the resource freely while making their decision on the presence or absence of a quality indicator. This gave us a simple, quick way to start: while annotators navigated the resource, we recorded which pages they visited, and on each page we asked them if they considered this to still be part of the same resource (as opposed to when they’d accidentally navigate to a page that turns out to be an external link). For this round of experiments I included all the pages of a resource that the annotators visited and considered part of the resource. This will usually only be a subset of the full extent of the resource, since we didn’t ask the annotators to navigate the complete resource; but one would expect that this subset will contain all the relevant information – after all, the annotators didn’t find it necessary to examine any pages outside of this set when making their decisions. One issue with this approach for deciding resource extent,
of course, is that it relies on having that kind of annotation available; on unseen data that will rarely be the case.

Now we have a set of HTML pages for each resource (there could also be images, media files, and in some cases PDF files – but for now, I am ignoring those). While HTML is a text-based format, there is a lot of mark-up in there which doesn’t add directly to the content, and different encodings are also possible. Theoretically one can parse the HTML, which will take care of these problems; the textual content can then be simply extracted. Unfortunately many different versions of HTML are being used, and on top of that the HTML found in the wild is rarely according to standard. It’s very common to see even respected web sites declare encodings incorrectly, use proprietary extension to HTML (which unfortunately most browsers support), or simply have broken code (which browsers will attempt to interpret as well as they can). I used a custom parser based on the TagSoup package[52], with manually added heuristics to improve stability. The output of this step is the plain text of the resource, as well as a structured representation of the HTML markup on it; some elements that would be invisible in a browser, such as script sections, are omitted. The content of all the pages within the resource is concatenated into one long document.

At this point, the text exists only as a sequence of characters without any further structure. In the next step, I break the text in sentences, and then into tokens (i.e. individual words and punctuation). Sentence segmentation is handled by the OpenNLP[42] package’s built-in sentence segmenter. Treebank style tokenization1 is handled by custom code, which was translated from a script by Robert MacIntyre at the University of Pennsylvania2. The script was translated into Java with the help of Steven Bethard and Philip Ogren at the University of Colorado and is now part of the ClearTK distribution.

In a final step, I determine the stem of each word. In linguistics, conceptually, the stem of a word is the part of it that’s common to all of its inflected forms. This is a common thing to do in natural language processing. The suffix of a word (the part that’s affixed to the stem) primarily serves to indicate the grammatical function of the word in the specific context of the sentence. As grammatical structure is not normally present in the feature vector representation of a text, not much is lost by dropping the suffix. On

---

1 The Treebank project, which manually annotated the syntactic structure of sentences in a corpus of Wall Street Journal articles, also defined a way in which sentences should be split into tokens (i.e. tokenized). This style has become the standard among natural language processing researchers.

2 The original script can be found at [http://www.cis.upenn.edu/~treebank/tokenization.html](http://www.cis.upenn.edu/~treebank/tokenization.html)
the other hand, two instances of a word with different suffixes can be expected to have the same semantics, when viewed independently of their grammatical context.

To summarize, the steps involved in preparing the corpus for the next stage are:

1. Retrieve the resource’s front page and associated pages
2. Clean and parse the HTML code of the pages
3. Extract the plain text from the HTML
4. Break the plain text into sentences and tokens
5. Determine the stem of each token

5.2 Feature Set

The step of transforming a resource into the feature vector required by the machine learning system is broken into two parts by the ClearTK framework: feature extraction, and feature encoding. While feature extraction happens first during an experiment, I will begin by explaining the feature encoding, which defines the way conceptual features are transformed into format required by the machine learning algorithm.

5.2.1 Feature Encoding

Features extracted from resources are of different types: some are binary, or “yes” / “no” features (e.g. “does this resource consist of more than one page”); some are numeric (e.g. “how many other websites link to this page”); and some are categorical (e.g. “what is the top level domain of this website”, where typical values would be “com”, “org”, or “edu”). Converting these into the right format for a machine learning algorithm can be done in many ways, and not all of them will work equally well for all algorithms, because each algorithm analyzes and manipulates data in different ways.

For my experiments I used the SVMlight package[31], which uses the support vector machine (SVM) algorithm for creating models. This algorithm has in the past been found to be very effective in natural language processing applications, and it can handle very large numbers of features. It expects the features to be presented as a numeric vector. In my experiments, features were encoded as follows:
**Binary** “Yes” / “no” features are encoded into the feature vector as a single one or zero. Each such feature that shows up in the training set is mapped to a specific index in the feature vector, and the value at that index is set to one or to zero based on the value of the feature for the current resource.

**Categorical** Features that can have a number of values, but which are not inherently numerical (i.e. there’s no “order” to the values, they’re all equally related or unrelated to each other) are encoded as multiple binary features: for each possible value $x$ that the feature takes in the training set, we create a new binary feature “feature has value $x$”, which is then encoded in the fashion described above. An example of such a feature is the top level domain of the resource URL; some possible values are “edu”, “org”, or “com”.

**Numerical** Features that are numerical in nature can be used as they are, since the Support Vector Machine algorithm is able to handle them well. Each such feature, like the binary features, is mapped to an index in the feature vector; the value at that index is then set to the numeric value of the feature for the current resource.

After all the features extracted from a digital library resource are combined into one numeric feature vector, the vector is normalized to Euclidian length 1. If a feature and value occurred less than 5 times in the training set it was discarded, as it was presumed that its presence in a resource did not carry useful information.

### 5.2.2 Feature Extraction

Most features I extract are taken straight from the text (e.g. individual words that show up somewhere in the document); some features make use of non-textual elements in the HTML markup (e.g. HTML anchor elements); other features include the domain the resource is hosted in, or information offered by an external source (e.g. the amount of traffic the site receives). Specifically, the first stage of experiments used these features:

**Bag-of-words** This feature is a common starting point for many natural language processing applications, and frequently the most powerful. It simply indicates the presence or absence of every known word
in this resource. Words that occur multiple times are only counted once. This is encoded as a categorical feature. For example: “resource contains the word ‘a’”, “resource contains the word ‘seismic’”, “resource contains the word ‘record’”, and so on, for every distinct word that a resource contains.\(^3\) The DLESE corpus used in these experiments contains almost 28000 distinct words. At this stage, I did not remove stop words — all words were included.

**Bag-of-bigrams** Similar to the **bag-of-words** feature, this is a categorical feature that looks not at single words, but at pairs of words. For example: “resource contains the word ‘long’ followed by ‘period’”, “resource contains the word ‘period’ followed by ‘Rayleigh’”, “resource contains the word ‘Rayleigh’ followed by ‘waves’”.

**TF-IDF term frequency – inverse document frequency**, a refinement of the **bag-of-words** feature. This was encoded as a set of numeric features, where each word’s feature’s value is based on how often the word shows up in the current resource versus in all resources. For example, the word “and” will show up many times in all resources, so the feature “resource contains the word ‘and’” will have a low value. On the other hand, the feature “resource contains the word ‘Rayleigh’”, assuming the word “Rayleigh” shows up a number of times in the current resource, but almost never anywhere else, will have a particularly high value.\(^3\)

**Resource URL** This feature presents the resource URL to the machine learning system. In addition to the full URL I included the domain and super-domains, encoded as categorical features. For example: “http://web.ics.purdue.edu/~braile/edumod/surfwav/surfwav.htm”, “web.ics.purdue.edu”, “ics.purdue.edu”, “purdue.edu”, “edu”. This allows the machine learning system to make potentially useful generalizations about the domain a resource is hosted in.

**URLs linked to** All the URLs that a resource links to. Each URL was presented in the same way as explained for the resource URL feature.

**Google PageRank** For all URLs I included a feature that indicates the Google PageRank of the respective site. This indicates the relative importance of that site on the internet, measured by how many other

\(^3\) These examples are from http://web.ics.purdue.edu/~braile/edumod/surfwav/surfwav.htm
sites link to it. For example, a site like \url{http://www.nasa.gov/} has a high PageRank value, while e.g. a largely unknown and small university web site will have a low value.

**Alexa TrafficRank** Alexa is a company offering traffic statistics on web sites based on analyzing user behavior\(^4\). For all URLs I included their reported **TrafficRank** in the feature set, which indicates the amount of user traffic a web site receives relative to other sites.

### 5.3 The Machine Learning System

As resources go through feature extraction and feature encoding, one numeric feature vector is generated for each resource. During model training the machine learning software analyses these vectors, generated from the training corpus, to learn a statistical model for each of the seven indicators. During evaluation the machine learning system decides if the indicators are present or absent in a resource by applying those models to the vector generated from that resource.

For the machine learning algorithm I chose support vector machines, using the SVMlight software[31]. SVMlight is one of the most usable, up-to-date, and actively developed implementations of the SVM algorithm and easily integrates into the ClearTK workflow.

One computational model was trained for each quality indicator. The same feature vector was used for all models (i.e. feature extraction and encoding were identical for the seven indicators).

The parameters to SVMlight were chosen using five-fold cross validation: The training data was split randomly into five roughly equal parts. Starting with a number of possible sets of training parameters, for each parameter set five models were trained, using in turn each part of the training data as the test set, the other four as the training set, then averaging the result over the five models. I tested linear, quadratic, and cubic kernels and a range of \(C\) values. The parameter set that consistently performed best during cross-validation was a simple linear kernel configuration; this was used on the final model, which was trained on the **complete** training set.

\(^4\) \url{http://www.alexa.com/}
Access to Alexa is offered through Amazon Web Services.
5.4 Results

In order to evaluate the effectiveness of my system I trained and evaluated models for each of the quality indicators on the training data. I then compared the results to a simple majority-class baseline: always assume the most common case, ignoring any information about the resource. For example, the has instructions indicator is present in 39% of resources. If I always assumed that a resource has no instructions, I would be correct in 61% of cases.

It is worth pointing out that there are multiple ways to compute a “majority class” baseline. The simplest and strictest way (when using as a point of comparison) is to determine the most frequent class in the test data and to always choose that in the evaluation. The comparison here is, strictly speaking, not fair: it’s biased in favor of the baseline, because the baseline system then uses information about the labels in the test set in its decision process. A more proper way would be to determine the majority class from the training data only, and then apply it to the test data. If the training and test set are from the same distribution, and if the data sets are large enough, and if the majority class is frequent enough (i.e. much more frequent than the next most frequent one), the two approaches are likely to give the exact same result. But if these conditions are not met, it can happen that the majority class in the training set is not the same as the majority class in the test set, and the performance, when using only the data that would be available to other machine learning algorithms, is much worse.

For simplicity the majority-class results I report use the test data in determining the most frequent answer. Still, an effective machine learning model will show significant improvement over this trivial baseline. Table 5.1 shows the results of this evaluation.

Improvements over the majority class baseline were achieved on the has instructions and has prestigious sponsor indicators, and smaller improvements on the indicates age range, identifies learning goals, and organized for goals indicators. The baseline machine learning system was unable to improve performance over the already high baseline on has sponsor and not inappropriate for age. The limiting factor, here, is the corpus: 950 of the 1000 resources in the corpus were taken directly from DLESE; the remaining 50 were candidates for inclusion. Because of this, almost all of the resources will satisfy certain
Table 5.1: Baseline computational system results

<table>
<thead>
<tr>
<th>Quality indicator</th>
<th>majority class baseline</th>
<th>SVM model performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Has instructions</td>
<td>61%</td>
<td>78%</td>
</tr>
<tr>
<td>2 Has sponsor</td>
<td>96%</td>
<td>96%</td>
</tr>
<tr>
<td>3 Has prestigious sponsor</td>
<td>70%</td>
<td>81%</td>
</tr>
<tr>
<td>4 Indicates age range</td>
<td>79%</td>
<td>87%</td>
</tr>
<tr>
<td>5 Not inappropriate for age</td>
<td>99%</td>
<td>99%</td>
</tr>
<tr>
<td>6 Identifies learning goals</td>
<td>72%</td>
<td>81%</td>
</tr>
<tr>
<td>7 Organized for goals</td>
<td>75%</td>
<td>83%</td>
</tr>
</tbody>
</table>

minimum standards of quality. Only one percent of resources were marked as inappropriate for age, and only four percent of resources did not identify their “sponsor”. The very small number of negative examples on these resources make statistical training and evaluation extremely hard. Other corpora might not exhibit this severely skewed distribution; training of these indicators might be possible then.

5.5 Error Analysis

In order to better understand the weaknesses and strengths of these models I conducted a study in cooperation with my colleagues at Digital Learning Sciences / UCAR to analyze the errors the system makes. For the purpose of this study we ignored the indicators has sponsor and not inappropriate for age, because the annotated data provides insufficient variation to conclusively train and evaluate models. Our goal was to manually compare the models’ output to the human annotator assessment and try to identify what factors may have led to the model making an incorrect decision (or, conversely, to determine if the human annotator had made a mistake on this resource).

I randomly split the annotated training data into a training set (650 instances) and a test set (150 instances), trained quality indicator models on the training set and ran them on the test set. By comparing the models’ results with the manual annotation on the test set I identified resources where two or more of the remaining five quality indicator models produced an apparently incorrect result, giving me a total of 39 resources with between 2 and 4 errors each. Since in each step the human expert was to evaluate one indicator on one resource, choosing resources where more than one mistake was made meant that the expert
would not have to familiarize themselves with a new resource on every step, thus allowing a more efficient use of their time.

A DLESE curation expert was asked to perform the analysis. The expert had helped develop the original annotation protocol, but had not taken part in the actual annotation. For each error on the selected set of 39 resources the expert completed an online questionnaire consisting of both enumerated choice and open-ended questions. This questionnaire asked the expert:

- which is correct: the human annotation or the automatic model’s result?
- is the indicator clearly present or absent, or is it ambiguous?
- are there cues in the text which clearly signal the indicator and should have been found; or are the cues implied, but not explicitly stated; or are the cues present in graphic elements or other parts not examined by the system (e.g. flash, images, etc.)?

Analysis of the data suggests that in 55% of the cases where the models did not detect the presence of an indicator (i.e. Type II error), cues were clearly present in the text and should have been found. As an example, the Heat Transfer and El Niño resource clearly lists “curriculum standards”, but in spite of this the model rated the resource negative on “has learning goals”.

In approximately 29% of the cases where the models disagreed with the annotators and rated an indicator as present (i.e. Type I error), potentially misleading text could be found in the resource. For example, for the resource Fossils in the Field the machine learning model rated the resource as “has instructions”. The expert noted that parts of the text may have been mistaken as instructions, stating that “the resource is a discussion of pedagogy” and “the words used provide ideas for instructing students […] but there are no instructions about using this resource as a professional reference.”

In less than 5% of cases when the models failed to detect an indicator, the cues were buried in graphic elements that are not examined by the system. For example, the EarthStorm – Relative Humidity &

---

5 http://projects.edtech.sandi.net/roosevelt/elnino/heatandelnino.html
6 http://www.ucmp.berkeley.edu/fosrec/GrifAll.html
Dew Point\textsuperscript{7} resource lists the grade ranges it supports in clickable buttons and not in the text of the resource.

On the whole, I found that it is difficult for a human expert to attempt to analyze where the models may or may not have gone wrong. This is part of the nature of machine learning algorithms; their power lies in their ability to leverage very small, hard to detect statistical dependencies. Often what pushes a model’s judgment one way or the other does not intuitively make sense to a human observer, even if the effect is statistically correct and works on real world data. For this reason, I would conclude that this type of analysis is not the most useful for informing targeted improvements to these models.

However, as in the examples above, there are some useful pieces of information to be gained. One concern with this approach was that important information for assessing a resource might be contained in media elements that are not accessible to these models; the error analysis seems to refute that. In fact, many of the errors seem to lie in the models missing textual cues that are present, or in not being discriminating enough in interpreting the relevant portions of the text. While my approach follows a whole-document classification model, this suggests that in future experiments it will be worth exploring models that specifically target the paragraphs that are relevant to an indicator; but with this data set that approach is difficult.

5.6 Conclusion

The baseline machine learning system served to validate this particular approach to constructing computational models for the quality indicators. The system successfully detected the presence or absence of most of the quality indicators at accuracies well above the majority class baseline. The next step was to identify the weaknesses in the existing system and find ways to improve performance on all indicators — and in the process creating a robust software platform that can serve as a solid foundation for future research.

\textsuperscript{7}http://earthstorm.mesonet.org/materials/les_rel_humid.php
Chapter 6

Stage 2: Model Improvements

The results of the baseline system showed the feasibility of modeling quality indicators with natural language processing and machine learning techniques. Performance was promising on some of the indicators, but less so on others. The error analysis showed that the feature set still missed many cues that, to human readers, are readily available in the text. Guided by these insights, and in order to improve performance across all indicator models, I identified areas for improvement. Here I will outline the steps I took to improve the various components of the system and discuss the effects they had.

6.1 Corpus Preparation

Similar to the digital library it was based on, the annotated corpus did not contain the actual resources, only references to their front page by URL. As our experiments proceeded, attrition of resources became a problem: over time, some of the resources would be shut down by their publisher, or would change substantially. In order to preserve the utility of the annotated corpus, I implemented a local cache of all web content that was retrieved during my experiments. While doing so, I attempted to find ways to determine if a resource was already offline or broken; making this determination is complicated by the fact that, frequently, servers will not report an error upon requesting a page that doesn’t exist anymore, instead forwarding the browser to another page (e.g. the front page of the site, or a directory page). In the end, I implemented a retrieval algorithm using manually tweaked heuristics based on redirects and error messages encountered on any of the known pages within a resource. As of today, I was able to maintain a cache of 890 of the 1000
The indicators content is not inappropriate for age range and organized for learning goals are dependent on identifies age range and identifies learning goals, respectively. Because they can only be meaningfully evaluated where the indicator they depend on is present, only a subset of the whole corpus is available for their training and testing: 147 + 36 resources for content is not inappropriate for age range, and 199 + 44 resources for organized for learning goals.

In another attempt to improve robustness of the corpus, I decided to start using the metadata records kept by DLESE for each of the resources. Resources for which no metadata record is available at DLESE can be assumed to have been offline for some time and can be considered dead. Of the 1000 resources used in the original annotation project, metadata records were available for 736 (714 of which are included in the cached set). Using the DLESE metadata records allowed another way of improving utility of the corpus: The full corpus contained a wide range of resource types, including things like portal sites and references. The DLESE metadata record of a resource contains the resource type, as determined by curators. While all types of resources are important in their own way, our assessments lend themselves particularly well to textual content that is to be read by students or teachers. For that reason, I implemented a mechanism to filter resources by DLESE resource type; in many of my experiments, I limited the resource type to those under the categories of DLESE : Learning Materials and DLESE : Text. Table 6.1 shows a list of the DLESE resource types that showed up in the corpus, as well as the number of times they each occurred. For reasons of readability, only the types DLESE : Learning Materials and DLESE : Text have been expanded into their sub-types. Note that a resource may be categorized as more than one type, so the numbers add up to more than the full 736.

Table 6.2 shows the effect on system performance when preparing the corpus in these ways. These results, and all the results I will present from here on, are computed on the held-out test set (i.e. the 200 double-annotated resources). The first column shows the majority class baseline for the test set. “Not inappropriate for age” has 100% performance across all columns, because in this test all resources were marked “yes”, so a perfect score can be trivially achieved. For completeness, I will include the row anyway.
<table>
<thead>
<tr>
<th>DLESE Resource Type</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLESE:Audio</td>
<td>39</td>
</tr>
<tr>
<td>DLESE:Data</td>
<td>49</td>
</tr>
<tr>
<td>DLESE:Learning materials</td>
<td>369</td>
</tr>
<tr>
<td>DLESE:Learning materials:Assessment</td>
<td>9</td>
</tr>
<tr>
<td>DLESE:Learning materials:Case study</td>
<td>14</td>
</tr>
<tr>
<td>DLESE:Learning materials:Classroom activity</td>
<td>110</td>
</tr>
<tr>
<td>DLESE:Learning materials:Computer activity</td>
<td>85</td>
</tr>
<tr>
<td>DLESE:Learning materials:Course</td>
<td>7</td>
</tr>
<tr>
<td>DLESE:Learning materials:Curriculum</td>
<td>7</td>
</tr>
<tr>
<td>DLESE:Learning materials:Field activity</td>
<td>12</td>
</tr>
<tr>
<td>DLESE:Learning materials:Field trip guide</td>
<td>4</td>
</tr>
<tr>
<td>DLESE:Learning materials:Instructor guide</td>
<td>52</td>
</tr>
<tr>
<td>DLESE:Learning materials:Lab activity</td>
<td>45</td>
</tr>
<tr>
<td>DLESE:Learning materials:Lesson plan</td>
<td>79</td>
</tr>
<tr>
<td>DLESE:Learning materials:Module or unit</td>
<td>33</td>
</tr>
<tr>
<td>DLESE:Learning materials:Presentation or demonstration</td>
<td>15</td>
</tr>
<tr>
<td>DLESE:Learning materials:Problem set</td>
<td>13</td>
</tr>
<tr>
<td>DLESE:Learning materials:Project</td>
<td>9</td>
</tr>
<tr>
<td>DLESE:Learning materials:Syllabus</td>
<td>2</td>
</tr>
<tr>
<td>DLESE:Learning materials:Tutorial</td>
<td>20</td>
</tr>
<tr>
<td>DLESE:Learning materials:Virtual field trip</td>
<td>15</td>
</tr>
<tr>
<td>DLESE:Offline</td>
<td>1</td>
</tr>
<tr>
<td>DLESE:Portal</td>
<td>72</td>
</tr>
<tr>
<td>DLESE:Service</td>
<td>32</td>
</tr>
<tr>
<td>DLESE:Text</td>
<td>310</td>
</tr>
<tr>
<td>DLESE:Text:Abstract or summary</td>
<td>5</td>
</tr>
<tr>
<td>DLESE:Text:Book</td>
<td>3</td>
</tr>
<tr>
<td>DLESE:Text:Glossary</td>
<td>26</td>
</tr>
<tr>
<td>DLESE:Text:Index or bibliography</td>
<td>7</td>
</tr>
<tr>
<td>DLESE:Text:Journal article</td>
<td>6</td>
</tr>
<tr>
<td>DLESE:Text:Periodical</td>
<td>8</td>
</tr>
<tr>
<td>DLESE:Text:Proceedings</td>
<td>3</td>
</tr>
<tr>
<td>DLESE:Text:Reference</td>
<td>248</td>
</tr>
<tr>
<td>DLESE:Text:Report</td>
<td>34</td>
</tr>
<tr>
<td>DLESE:Tool</td>
<td>31</td>
</tr>
<tr>
<td>DLESE:Visual</td>
<td>269</td>
</tr>
</tbody>
</table>

Table 6.1: DLESE resource types in corpus
Table 6.2: Comparison of corpus preparation configurations

<table>
<thead>
<tr>
<th>Quality indicator</th>
<th>majority class</th>
<th>impr. retrieval</th>
<th>req. metadata</th>
<th>Text</th>
<th>Learning materials</th>
<th>Text + L.M.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Has instructions</td>
<td>65.2%</td>
<td>88.8%</td>
<td>88.2%</td>
<td>69.6%</td>
<td>66.5%</td>
<td>90.0%</td>
</tr>
<tr>
<td>Has sponsor</td>
<td>98.8%</td>
<td>98.8%</td>
<td>98.8%</td>
<td>98.8%</td>
<td>98.8%</td>
<td>98.8%</td>
</tr>
<tr>
<td>Has prestigious sponsor</td>
<td>59.6%</td>
<td>80.1%</td>
<td>80.1%</td>
<td>74.5%</td>
<td>75.1%</td>
<td>77.0%</td>
</tr>
<tr>
<td>Indicates age range</td>
<td>83.2%</td>
<td>91.3%</td>
<td>91.9%</td>
<td>83.2%</td>
<td>88.8%</td>
<td>90.7%</td>
</tr>
<tr>
<td>Not inappropriate for age</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
</tr>
<tr>
<td>Identifies learning goals</td>
<td>76.4%</td>
<td>90.1%</td>
<td>88.8%</td>
<td>77.0%</td>
<td>85.7%</td>
<td>89.4%</td>
</tr>
<tr>
<td>Organized for goals</td>
<td>92.1%</td>
<td>92.1%</td>
<td>92.1%</td>
<td>92.1%</td>
<td>92.1%</td>
<td>92.1%</td>
</tr>
</tbody>
</table>

In the second column, “improved retrieval”, we see that, by improving the reliability of the HTTP retrieval code and detecting if resources are offline (and filtering resources in which some of the pages the annotators originally saw during annotation are now inaccessible), the new system’s performance is noticeably higher than the baseline system I presented in the previous chapter. The assumption is that, if the structure of the resource web site has changed, it is likely that the content of the resource is also different from what it was at the time of annotation. The results represent a statistically significant\(^1\) improvement over the majority class baseline on four indicators (“has instructions”, “has prestigious sponsor”, “indicates age range”, and “identifies learning goals”). Also restricting the training set to resources for which metadata records are available from DLESE did not have any further impact.

The last three columns show the effect of restricting the training set to resources of a given type, according to the DLESE metadata. Performance dropped significantly on some indicators when using only resources of type “DLESE:Text”, or of type “DLESE:Learning materials”. When allowing both types, however, it was essentially the same as when not filtering by resource type at all.

From these results it can be concluded that the resource type was not a useful tool for filtering the training set for this task. Near optimal performance was achieved just by avoiding resources that may have become corrupted or changed since annotation. While no improvements over majority class were achieved on “has sponsor” and “organized for goals”, this can be explained by the extreme one-sidedness of the annotation on those indicators (and, in fact, the achieved score is not statistically significantly different from a perfect score).

\(^1\) Statistical significance was computed using McNemar’s test with $p = 0.05$. 
6.2 Web Crawler Configurations

The next step was to address the issue of resource extent. During the annotation of the DLESE corpus, we not only recorded each annotator’s decision on the presence or absence of each quality indicator; we also recorded which web pages they visited in making their decisions, and for each visited web page we asked them if the page was still part of the resource (as opposed to when the annotators accidentally clicked a link that took them to a different site or resource). For each resource the annotators viewed on average 2.2 pages (including the front page) that they considered as belonging to the resource.

In the baseline system I used this information as a kind of oracle to determine which web pages were part of the resource, and thus which pages to include in the processing. This approach has some drawbacks: most importantly it doesn’t extend to data from other sources, as we’ll generally not have such information available. But on top of that, this approach only looks at the pages the annotators saw – but the annotators weren’t required to navigate through the whole resource, only the parts they found relevant in making their decision on the quality indicators. While, presumably, most of the relevant information is thus contained in the selected set of pages, additional cues may be hidden in other sub-pages of the resource. To explore this effect, I implemented a more flexible web crawler framework, which supports a number of different configurations:

**PartOfResource** This is the original crawler configuration, which simply includes all the pages seen by an annotator, and which they deemed part of the resource.

**FrontPage** This simple configuration doesn’t perform any crawling — it simply includes the front page of the resource, and nothing else.

**SameSiteDepthOne** In this configuration the front page is included, as well as any pages linked directly from it that reside on the same server (i.e. the url has the same server string). The total number of pages is limited to seven to avoid the system being bogged down by pages with large amounts of parallel content.
Table 6.3: Comparison of crawler configurations

<table>
<thead>
<tr>
<th>Quality indicator</th>
<th>Part of Resource</th>
<th>Same Site Depth One</th>
<th>Front Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Has instructions</td>
<td>88.8%</td>
<td>86.1%</td>
<td>83.7%</td>
</tr>
<tr>
<td>2 Has sponsor</td>
<td>98.8%</td>
<td>97.6%</td>
<td>97.6%</td>
</tr>
<tr>
<td>3 Has prestigious sponsor</td>
<td>80.1%</td>
<td>76.5%</td>
<td>77.7%</td>
</tr>
<tr>
<td>4 Indicates age range</td>
<td>91.3%</td>
<td>88.6%</td>
<td>89.8%</td>
</tr>
<tr>
<td>5 Not inappropriate for age</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
</tr>
<tr>
<td>6 Identifies learning goals</td>
<td>90.1%</td>
<td>89.2%</td>
<td>86.8%</td>
</tr>
<tr>
<td>7 Organized for goals</td>
<td>92.1%</td>
<td>87.5%</td>
<td>92.5%</td>
</tr>
</tbody>
</table>

Table 6.3 shows the effect of using these different crawler configurations, using the “improved retrieval” corpus preparation introduced in the previous section. All results up to this point were using the “part of resource” crawler configuration. We see a slightly lower performance when switching to the “same site, depth one” crawler or when only using the front page of a resource. These drops are small (and not statistically significant), and we can conclude that both approaches are viable alternatives to using the information gathered during annotation for determining resource extent. This is an important finding, as in most realistic applications of this system, manually determining the resource extent would be infeasible.

### 6.3 Relative Importance of Features

The feature set I started out with was a combination of a range of commonly used features in natural language processing applications (e.g. bag of words, bigrams, TF-IDF). The machine learning algorithms are able to determine which of the available features are actually useful in predicting the outcome, that is, which features can be used to predict if a resource exhibits a given indicator. Because of this, including extraneous features that have no predictive power in the training set, or including redundant features that add no information beyond what the other features already encode, may not be a problem at all. In fact, the Support Vector Machine learning algorithm used here has generally been found to be quite robust to “unhelpful” features. Throwing all conceivable features at a problem is a common approach and it tends to work fairly well.
On the other hand, in order to target future efforts in improving system performance, it can be very helpful to know which features contribute significantly to the solution, which features are redundant (i.e. they duplicate the effect of other features in the set), and which simply do not contribute at all. Having an insight into this means we can focus on improving the features that we know potentially contribute a lot to the overall performance. Leaving out the features that don’t contribute also simplifies the overall system, and this means that we can start from a simpler, more solidly founded base when approaching a new problem with a similar structure.

The original feature set has been described in detail in Section 5.2. It can be roughly separated into these feature groups:

**Resource URL** This is the URL of the resource’s front page, and information associated with the URL, such as the Google PageRank.

**Bag of Words** The collection of all words (or, more specifically, the word stems) that occur in the resource.

**TFIDF** All words, weighted according to how frequent they are in the whole corpus.

**Bigrams** All pairs of words that occur in the resource.

**Links** All URLs of outgoing links in the resource, including information associated with those URLs.

Table 6.4 shows the results obtained when using only a subset of this feature set. The top row shows the performance when including the whole feature set (this corresponds to the results reported in Section 6.2). At the bottom, I have included the majority class baseline performance, which presents a lower limit.

Excluding only one of the five feature groups while keeping the remaining four only has a small effect on the outcome, suggesting that the full feature set has quite a bit of redundancy. Only when excluding the “bigrams” feature does a statistically significant drop occur in the “has instructions” indicator.

Approaching the problem from the other side, the results are a bit more varied: when using only one out of the five feature groups, only the “bag of words” and “bigrams” features offer relatively good performance across all indicators. In either case, the weakness lies in the “has prestigious sponsor” indicator,
which is handled better by the “resource URL” and “links” feature groups. When we combine one of each of them, the performance is about as good as on the full feature set.

This greatly simplifies the overall system. Whether one would choose to employ bag of words or bigram features on a task depends on the specifics of the situation; bigrams are slightly more complex to implement and may result in sparsity problems — there are many more possible pairs of words than single words, and a small training set may not adequately cover the relevant ones. But either type of feature, combined with a URL based one, is sufficient – at least on this data set.
<table>
<thead>
<tr>
<th>Feature Set</th>
<th>HI</th>
<th>HS</th>
<th>HPS</th>
<th>IAR</th>
<th>NIA</th>
<th>ILG</th>
<th>OLG</th>
</tr>
</thead>
<tbody>
<tr>
<td>all features</td>
<td>88.8%</td>
<td>98.8%</td>
<td>80.1%</td>
<td>91.3%</td>
<td>100.0%</td>
<td>90.1%</td>
<td>92.1%</td>
</tr>
<tr>
<td>no resource URL</td>
<td>88.8%</td>
<td>98.8%</td>
<td>79.5%</td>
<td>91.3%</td>
<td>100.0%</td>
<td>88.8%</td>
<td>92.1%</td>
</tr>
<tr>
<td>no bag of words</td>
<td>88.8%</td>
<td>98.8%</td>
<td>83.2%</td>
<td>92.5%</td>
<td>100.0%</td>
<td>89.4%</td>
<td>92.1%</td>
</tr>
<tr>
<td>no TFIDF</td>
<td>87.0%</td>
<td>98.8%</td>
<td>80.1%</td>
<td>90.7%</td>
<td>100.0%</td>
<td>90.7%</td>
<td>92.1%</td>
</tr>
<tr>
<td>no bigrams</td>
<td><strong>82.6%</strong></td>
<td>98.8%</td>
<td>80.8%</td>
<td>90.7%</td>
<td>100.0%</td>
<td>90.7%</td>
<td>92.1%</td>
</tr>
<tr>
<td>no links</td>
<td>90.1%</td>
<td>98.8%</td>
<td>79.5%</td>
<td>92.5%</td>
<td>100.0%</td>
<td>90.1%</td>
<td>92.1%</td>
</tr>
<tr>
<td>only resource URL</td>
<td><strong>73.3%</strong></td>
<td>98.8%</td>
<td>78.9%</td>
<td><strong>84.5%</strong></td>
<td>100.0%</td>
<td>83.2%</td>
<td>92.1%</td>
</tr>
<tr>
<td>only bag of words</td>
<td>88.2%</td>
<td>98.8%</td>
<td>73.9%</td>
<td>87.6%</td>
<td>100.0%</td>
<td>88.8%</td>
<td>92.1%</td>
</tr>
<tr>
<td>only TFIDF</td>
<td>87.0%</td>
<td>98.8%</td>
<td><strong>71.4%</strong></td>
<td>91.3%</td>
<td>100.0%</td>
<td>88.8%</td>
<td>92.1%</td>
</tr>
<tr>
<td>only bigrams</td>
<td>88.8%</td>
<td>98.8%</td>
<td>75.2%</td>
<td>91.9%</td>
<td>100.0%</td>
<td>90.1%</td>
<td>92.1%</td>
</tr>
<tr>
<td>only links</td>
<td><strong>66.5%</strong></td>
<td>98.8%</td>
<td>82.6%</td>
<td><strong>83.2%</strong></td>
<td>100.0%</td>
<td><strong>77.0%</strong></td>
<td>92.1%</td>
</tr>
<tr>
<td>bag of words + links</td>
<td><strong>83.3%</strong></td>
<td>98.8%</td>
<td>77.0%</td>
<td>90.7%</td>
<td>100.0%</td>
<td>89.4%</td>
<td>92.1%</td>
</tr>
<tr>
<td>bigrams + links</td>
<td>87.0%</td>
<td>98.8%</td>
<td>80.8%</td>
<td>91.9%</td>
<td>100.0%</td>
<td>89.4%</td>
<td>92.1%</td>
</tr>
<tr>
<td>bag of words + resource URL</td>
<td>87.0%</td>
<td>98.8%</td>
<td>78.3%</td>
<td>88.2%</td>
<td>100.0%</td>
<td>86.3%</td>
<td>92.1%</td>
</tr>
<tr>
<td>bigrams + resource URL</td>
<td>88.2%</td>
<td>98.8%</td>
<td>81.4%</td>
<td>90.7%</td>
<td>100.0%</td>
<td>90.1%</td>
<td>92.1%</td>
</tr>
<tr>
<td>majority class</td>
<td>65.2%</td>
<td>98.8%</td>
<td>59.6%</td>
<td>83.2%</td>
<td>100.0%</td>
<td>76.4%</td>
<td>92.1%</td>
</tr>
</tbody>
</table>

Table 6.4: Quality indicator performance using different feature sets

This shows performance of the models using different feature sets. Numbers in bold represent a statistically significant drop in performance from the full feature set.

6.4 Exploiting Resource Structure

Resources that are cataloged by DLESE and other libraries are for the most part in HTML format, potentially linking to PDF files or containing rich media. I followed the common approach: running a simple parser that strips out HTML tags and discards presumed irrelevant portions of a page. Various implementations are available that perform this task (e.g., TagSoup\textsuperscript{2}, Beautiful Soup\textsuperscript{3}). My system combines the TagSoup parser with a simple, manually compiled set of heuristics to arrive at a plain-text representation of the resource. This representation is then fed into the feature extraction process.

Such a plain-text representation still contains many things that are not part of the interesting content of the web page. Most web pages consist of a frame that is shared across the site; this may consist of a header identifying the site, a footer giving copyright notices or disclaimers, navigation bars at the top or at the side, advertisements (in text or as banners), and similar elements. A web page offers many visual cues (different fonts, font sizes, visual separation, headings / paragraphs, . . .) to help the user identify the parts of the page containing the “payload”, i.e. the educational content; in the plain-text format, those cues are lost.

CLEANEVAL\textsuperscript{[7]}, organized by ACL SIGWAC in 2007, was a contest of automatic systems for preparing web sites for use for linguistic and language technology research. The contest focused on removing all HTML and Java code, boilerplate, advertisements, and other content that was deemed not part of the “proper, coherent content” of the page. Paragraphs, headings and list elements were marked. Arguably, the boilerplate parts of a page (title, copyright statements), navigation elements (e.g. a navigation bar on the side) and other parts that are not proper content still contain information that may be useful to identify quality indicators; for example, the boilerplate sections are exactly where you would expect a resource to identify its publisher. Thus CLEANEVAL’s approach can be expanded upon by not simply throwing away those sections, but by categorizing them into content classes, thus allowing features to focus on the specific information they contain.

\textsuperscript{2} http://home.ccil.org/~cowan/XML/tagsoup/
\textsuperscript{3} http://www.crummy.com/software/BeautifulSoup/
I built a system that splits a web page along HTML block element tags (for example div, p and table tags) using a set of hand-tuned heuristics. The resulting HTML fragments are then classified along two dimensions: content section, and content type. The content section of a fragment describes the place it holds in the site layout: is it a header element, is it part of the main content, is it an extraneous section that’s interleaved with the main content (e.g. an ad that lies between two content paragraphs), or is it a footer element? The content type could be textual (e.g. flowing text, lists and tables), media (images, embedded flash elements), or junk (e.g. parts of HTML code, scripts – while they are mostly filtered out during parsing, small segments can sometimes make it into the plain text form of the web page, depending on how they are encoded in the HTML).

I manually annotated a small number of resources by marking each paragraph as belonging to a content section and a content type; from this data a classifier was built using a simple linear kernel Support Vector Machine. Features for classification are a bag of words, bag of part-of-speech tags, fraction of capitalized words vs. non-capitalized, overall length of the fragment in tokens and in characters, fraction of the fragment that consists of links, bag of the names of all contained HTML tags, bag of all HTML tags that contain the fragment, and the attributes of the image tag, if the fragment is made up of one image. As the training set is very small, I do not have a quantitative analysis of the performance of this system.

In order to attempt to determine the usefulness of this added information to the task of detecting quality indicators, I ran different combinations of bag features, making use of the resource structure annotation (Table 6.5). Models trained on only the resource URL and a bag of words feature serve as the baseline. The other models make use of multiple separate bags of words, one for each content section / type. For example, the second column, “by content section”, uses one bag of words for each of the “content section” classes (header, main content, interleaved, footer). One might expect that such a system would be able to find that certain words are particularly important when they show up in a specific section of a page, for example a reference to “instructions” might be expected to show up within the main content of a page instead of in a link at the very bottom of it.

There are two ways to separate features by both content section and content type. In the first case, I include a bag of words for each content section, and one for each content type; every word is then counted
Table 6.5: Effect of resource structure annotation

<table>
<thead>
<tr>
<th>Quality indicator</th>
<th>no structure</th>
<th>by content section</th>
<th>by content type</th>
<th>section &amp; type</th>
<th>type (joint)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Has instructions</td>
<td>86.9%</td>
<td>81.3%</td>
<td>81.3%</td>
<td>83.1%</td>
<td>82.5%</td>
</tr>
<tr>
<td>Has sponsor</td>
<td>98.8%</td>
<td>98.8%</td>
<td>98.8%</td>
<td>98.8%</td>
<td>98.8%</td>
</tr>
<tr>
<td>Has prestigious sponsor</td>
<td>78.1%</td>
<td>81.9%</td>
<td>77.5%</td>
<td>78.8%</td>
<td>81.3%</td>
</tr>
<tr>
<td>Indicates age range</td>
<td>88.1%</td>
<td>88.1%</td>
<td>86.9%</td>
<td>88.8%</td>
<td>88.8%</td>
</tr>
<tr>
<td>Not inappropriate for age</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
</tr>
<tr>
<td>Identifies learning goals</td>
<td>86.3%</td>
<td>90.7%</td>
<td>86.9%</td>
<td>90.0%</td>
<td>90.0%</td>
</tr>
<tr>
<td>Organized for goals</td>
<td>92.1%</td>
<td>92.1%</td>
<td>92.1%</td>
<td>92.1%</td>
<td>92.1%</td>
</tr>
</tbody>
</table>

Twice, once in the bag of its paragraph’s content section, and once in its paragraph’s content type. In the second, joint, case, I include a bag of words for each **combination** of section and type. For example, one bag of words counts all words that occur in paragraphs that have media content and are part of the page header. Since four content section classes and three content type classes were annotated, this model uses twelve separate bags of words.

As Table 6.5 shows, separating the bag of words into separate bags has very little effect. None of the models using the structure annotation are significantly different from the baseline. This does not mean that structural information of this type is of no use for this task; but it does suggest that this is not a good way to make use of that information. Alternatively, it is possible that the lack of improvement here is due to weaknesses in my structural annotation scheme, or that the automatic annotation is simply not very accurate.

6.5 Training Set Size

Another factor in model performance that I wanted to explore lies in the size of the training set. In all the experiments reported on in this chapter, the training set consisted of up to 800 resources (the actual set is smaller than 800 where some resources were excluded because of errors during retrieval, wrong metadata, etc.). On this data set, using the full training set has resulted in good performance, but in order to consider the role this approach might play in future digital library interfaces or peer production systems, it is important to understand if a smaller set of annotated resources may be sufficient to reliably detect
indicator presence; after all, reliable manual annotation of resources for training is an expensive and time consuming process.

To explore this issue I ran a further experiment: For different sizes of training set $n$, I randomly selected $n$ resources from the full set, trained the system on those, and evaluated on the full test set. For each $n$, I repeated the experiment five times and averaged the results. Figures 6.1 and 6.2 show the outcome of this experiment (the majority class baseline is shown in light grey). The models require about 50 to 100 resources to beat the majority class baseline. Beyond that, we see a gradual increase in performance, which reaches near its peak between 200 and 300 resources (only “indicates age range” doesn’t get there until about 400 resources). This result is encouraging, as it shows that a smaller training set is sufficient for achieving full performance; it also suggests that training set size is not the limiting factor for performance on this data set.
Figure 6.1: Training set size – Page 1
Figure 6.2: Training set size – Page 2
6.6 Machine Learning Components

The last extension to the quality indicator models that I would like to discuss here concerns the form of predictions being made. Up to this point, the problem of quality indicator presence has been framed as a binary decision problem: an indicator is either present, or it is not. This is the way the DLESE annotation study was conducted (annotators were able only to choose between “yes, the indicator is present” and “no, the indicator is not present”), and it is the way the computational models were designed. The Support Vector Machine algorithm, at least in the form I have used it here, learns to distinguish between two distinct classes.

Considering that quality indicator assessment, even when conducted by human experts, can be very ambiguous, and allowing for the fact that no computational model can be perfect at detecting an indicator, it makes sense to want a more graduated output from the models. If a computational model is able to indicate when it is “less certain” about its output, we might be able to rely on it more in the remaining cases; a model that is able to judge some resources reliably can be more useful than a model that is able to judge all resources with only moderate reliability.

The linear-kernel Support Vector Machine algorithm used in this work essentially performs a linear mapping from the space of features (i.e. highly complex vectors of numerically encoded information about a resource) to a simple, scalar number. This mapping is constructed by selecting some of the training resources as “support vectors”; these support vectors are then linearly combined to create (in the case of a linear kernel) a single vector. Applying the model to a resource simply means computing the dot product of the resource’s feature vector with the model’s vector. The result of this computation is a scalar value that encodes how the evidence for an indicator’s presence and for its absence balances out, also called the decision value.

In its normal configuration, the Support Vector Machine algorithm will select and combine the support vectors in such a way that, if the decision value for a resource is positive the evidence for the indicator’s presence outweighs the evidence for its absence; and if the decision value is negative, the evidence for absence outweighs the evidence for presence. Beyond this simple application of a threshold, the decision value isn’t
calibrated in any way. For example, if the decision value is +1.5, this suggests that the indicator is probably present, but it doesn’t say how much more likely the indicator’s presence is than its absence. Also, decision values are not comparable between different models: a +1.5 on “has instructions” may correspond to a completely different degree of certainty than a 1.5 on “has prestigious sponsor”.

Nevertheless, the decision value, within a single model’s results, gives an ordering of resources. If resource A has a decision value of +0.7 on “has instructions”, and resource B has a decision value of +1.3 on “has instructions”, we can conclude that the evidence for resource B having instructions is stronger than for resource A. This being the case, we should be able to use the decision value to measure a model’s confidence in its assessment. Ideally, we would like to end up with a probability. For example, we would like to be able to say that resource A has instructions with 60% probability, while resource B has instructions with 85% probability.

It is important to note that true probabilities can only be arrived at by a probabilistic reasoning process. The difference is more ideal than real, but it is an important one in principle: While a probabilistic reasoning process may begin with faulty assumptions and incorrect models, thus arriving at probabilities that don’t align well with the physical reality, it is internally consistent according to the laws of probability. But in this case, a rough approximation of the “true” probability, is sufficient. This is a pseudo-probability.

Various approaches to mapping SVM decision values to pseudo-probabilities have been explored. I implemented an algorithm published by Hsuan-Tien Lin, Chih-Jen Lin, and Ruby Weng, which in turn extended an algorithm by John C. Platt. After training a standard SVM model, this algorithm applies a logistic sigmoid function to the decision value to form a pseudo-probability. The logistic sigmoid function follows the formula:

\[
\frac{1}{1 + e^{-Ax+B}}
\]  

Using a training set, Lin, Lin, and Weng’s algorithm chooses the parameters A and B in order to fit the output to the observed probability of the model making a correct assessment.

Figures 6.3 and 6.4 show the result of applying this algorithm to the quality indicator models. For any threshold value \(x\), the curve “negative examples” shows the number of resources that do not have the
indicator for which the computed probability is higher than the threshold; on other words, those would be misclassified as having the indicator. Conversely, the curve “positive examples” shows the number of resources that do have the indicator but that would be classified as not having it at a given threshold value.

The challenge, then, is to pick the threshold value such that the number of misclassifications in either direction is minimized. The default threshold is 0.5, corresponding to the point where, according to Lin, Lin, and Weng’s algorithm, there’s an equal chance of the indicator being present or absent in the resource. However, depending on the application of the models, one might want to choose the threshold in a different way. For example, in Figure 6.3 we see that, if we were to choose a threshold of about 0.97, we would almost never say a resource had instructions when in fact it doesn’t, but we would still accurately detect instructions in a third of resources that have them. If the priority is high precision while low recall is acceptable, that ability can be very useful. Consider, for example, a search engine: If a given query results in hundreds of results, excluding all the bad ones at the cost of excluding some of the good ones as well is quite acceptable. We might in fact have two thresholds, one for positive examples and one for negative ones. If the probability is below the “negative threshold”, the resource is classified as “indicator not present”; if it’s above the “positive threshold”, the resource is classified as “indicator present”; and if the probability lies between the two values, we do not rely on the output of the automatic assessment and flag the resource for manual evaluation instead.

6.7 Summary

In this chapter I discussed a wide range of factors that influence the performance of computational models for quality indicator assessment, from the way the data is selected and collected to the feature set and machine learning system being used. It is very difficult to make an argument that results such as the one presented here represent “good” performance or “bad” performance; how good a system is can, ultimately, only be said once it is put into use in a real world application.

However, it is possible to look at system performance in comparing it to the human annotators’ ability to make the same assessments. On all quality indicators, the system presented here was able to perform at least at or near the same level, compared with the adjudicated data, as the human annotators’ agreement
with each other. On some indicators (“has prestigious sponsor” and “organized for goals”), in fact, the system agreed with the adjudicated data significantly more than the annotators agreed with each other.

With accuracy on all indicators being above 80%, this data set has become a limiting factor. The methodology used here and the quality indicator models were developed on the DLESE data set, and it is maybe not surprising that, even though this task is highly complex, they perform relatively well on this data. This demonstrated the necessity to begin exploring the effectiveness of this approach on a new, more diverse data set. In the next chapters I will explain how the opportunity to create such a data set arose in the Instructional Architect project, and how I adapted the existing quality models to this new domain.
Figure 6.3: Training Set Size – Part 1
Figure 6.4: Training Set Size – Part 2
Chapter 7

The Instructional Architect Corpus

The Instructional Architect (IA) is a simple, web-based authoring tool for open educational resources. The web site provides users with tools to gather and sort educational resources from the web, including existing collections of resources such as the National Science Digital Library (NSDL) or DLESE; to use some of these selected resources, sequence, and annotate them with one’s own text in order to create learning activities (or “projects”); and once they are finished, to control how to publish the created materials to students, or to the public.

Often the materials created by IA users take the form of a guided activity: in a sequence of steps, the teacher guides the students to read and interact with specific parts of web based resources, thus giving the students an opportunity to directly take information from a variety of sources without requiring the students to go on a self-directed quest for information that may or may not be relevant to the task at hand. This sequence of activities helps the students to integrate what they’ve read into a coherent mental model. By providing clear instructions for how to approach each resource, and by integrating the materials into a coherent activity (possibly adding questions to focus students’ attention and to evaluate their progress), the activity provides educational value beyond the content of the resources it uses. Instructional Architect encourages teachers to share the materials they produce; others can then use them for their own students, or they might adapt them or even build their own materials on top of them. In this way, the Instructional Architect project has aimed to offer a fertile peer production environment.

Instructional Architect has over 5200 users. Its users have created over 11000 separate projects using over 50000 different online resources. In many cases, these projects are short, single-use documents that a
teacher created for a specific class context and that have little meaning outside of that context. Others are activities that were designed to fit within a class setting (for example, they might refer to activities that took place during the class period), but their main content could stand on its own and could conceivably be of use to other teachers, if only as a starting point for their own materials. And in some cases, the teacher who produced the activities designed them in such a way that they could be used as they are by other teachers to teach their class. Project size varies a lot, from just a handful of words to lengthy, fully self-contained documents. Figures 7.1 and 7.2 show two such IA projects.

A wide range of quality of content can be found among Instructional Architect projects. Not all users produce content that not only has meaningful informational value beyond what the linked resources offer, but that can also stand on its own in an educational setting or be used productively by other teachers. Expert and peer review can play a role in helping seekers of materials sift through the large pile of projects, but with increasing adoption of peer production practices among educators, being able to assess aspects of quality automatically becomes more and more important. In particular, this problem came up when it was decided to try to provide a channel for including high quality IA projects in the NSDL. Simple proxy metrics such as number of words in a project or number of linked resources can exclude some obviously uninteresting projects, but beyond that a more nuanced view of project quality becomes necessary. This situation offered an opportunity to test the resource quality framework we developed for DLESE on a different, but related, data set. In order to do so, we decided to create a second corpus using Instructional Architect projects.

7.1 Differences Between IA and DLESE

In some ways, Instructional Architect projects are fairly similar to the kinds of educational resources we find in DLESE, or in other digital libraries: they are web sites that are designed to be used in an educational setting, providing materials that teachers can use in the context of their class. In fact, some of the resources found in DLESE (including some in the DLESE corpus described here) are learning activities written and published by individual teachers, similar to how Instructional Architect users write and publish their content.
Mr. Ribera

Earthquakes in Utah

Students will gain a better understanding of how earthquakes and volcanoes are formed and how they have contributed to the geology of Utah.

This web-based lesson will help guide you through a number of websites that will help you gain a better understanding of earthquakes and volcanoes especially happening in Utah.

Follow the instructions for each and enjoy. You will need your headphones on for the videos.

Site #1. Watch video on earthquake destruction.
Earthquakes 2

Site #2. What has happened recently in Utah?
Latest Quakes

Site #3. Read about volcanoes and tsunamis and look at different animations. First volcano animation and then look at tsunamis

Site #4. Go through the basics and words to know to test yourself. Click on the following website for #4
Scholastic: Weather Watch

Figure 7.1: Instructional Architect example project 1: Earthquakes in Utah

This short project takes students through a series of videos, texts, and evaluations to help them understand earthquakes in the context of their own state.[29]
Learn about the structure of the human brain and how it is affected by drugs of abuse.

Use the resources below to

1) List at least 10 structures in the brain, and explain their function. Be sure to include the reward pathway.

2) Make your own sketch of the brain and show the location of the 10 structures above. You may need more than one sketch.

3) Describe how drugs of abuse affect different parts of the brain.

   Neurobiology of Drug Addiction Sec. I

   Reward Pathway and Addiction

   Understanding Drug Abuse NIDA

   Pleasure Centers in the Brain

   Brain Activities

4) Describe the long term damage that drugs can do to the brain and other body systems. Be very specific.

   Drugs and the Brain NIDA

   Brain Health

   Utah Genetics

5) Describe the brain and neuronal issues involved in abuse, tolerance, craving, and addiction.

   Teen Brain

6) Describe 4 techniques we can use to visualize the living brain.

Figure 7.2: Instructional Architect example project 2: Addiction and the Brain

This project gives students a series of activities and questions, providing various supporting materials in the form of external resources, to help the students learn about the neurobiological aspects of drug addiction. (Only the first page of the project is shown.)[27]
However, on the whole the distribution of resource types and formats is quite different between DLESE and Instructional Architect. Most DLESE resources are offered by web sites that specialize in educational or scientific content, and that offer many related resources on the topics they cover; frequently, these resources are interconnected, sharing common materials across resources (for example through headers, footers, and sidebars), and linking from one resource to the next. In contrast, IA projects stand alone, each project being developed independently from the others. From a purely structural perspective, it’s just the opposite: DLESE embodies the full range of formats that web materials can take, as each publisher uses their own HTML publishing and content managing platform, and the resulting HTML code differs significantly in how content is displayed. On the other hand, Instructional Architect uses a simple rich-text format for all projects; they all share the same style and the same underlying HTML code.

Most DLESE resources, being offered by organizations that specialize in educational content, have undergone extended editorial processes. Each resource tends to cover a substantive amount of material, often exploring a subject from multiple sides. Usually, the main value of resources is the teaching materials they themselves contain, that is, they do not depend much on other linked resources to be effective. Instructional Architect projects, on the other hand, are by definition developed by individual authors. The peer production environment encourages ad-hoc content creation, so materials, once they are produced, usually aren’t edited further; one can assume that the standards of the writing are lower on average because of that. Projects tend to be focused on one activity, not aiming to cover the subject comprehensively; many projects are quite short. And further, most IA projects are by their nature derivative works, that is, the value they offer is in how they arrange and introduce materials from other sources, not in the material they cover directly.

Lastly, DLESE is the Digital Library for Earth System Education – that is, all resources within DLESE cover topics within the domain of Earth Science. On top of that, in selecting resources for the DLESE corpus we chose to only include materials that are targeted at the high school level. Instructional Architect has no predefined domain, everything that falls under educational content qualifies. Not only does that include other fields of science, but also humanities, languages, and so on.
7.2 Annotation

We had two main goals in mind in creating an Instructional Architect corpus: First, to assess how well our approach of characterizing the educational quality of a resource using low-level indicators works on IA projects, compared to its effectiveness on DLESE resources; that is, to test if annotators would be able to annotate a set of IA projects with reasonably high agreement, and if the annotated results would be diverse enough to differentiate between different qualities of projects. And second, to evaluate how well the computational models I developed using the DLESE corpus are able to recognize the same indicators on this new, different data set, and what adaptations might become necessary in order to allow for the differences between DLESE and Instructional Architect.

The original motivating scenario for this work was the idea that high quality Instructional Architect projects would make suitable candidates for inclusion in the National Science Digital Library. If successful, our framework for characterizing quality could provide the underpinnings of a system for doing that; and in particular, if computational models can be effectively applied to this data, it would make it much more likely that such a system could be put in place. Beyond that, there are many other initiatives for collecting peer produced educational content (for example, the QCommons project offers an open bank of questions for practice and assessment[44]) that such a system may generalize to.

In the following I will describe how we selected the set of IA projects to be included in the corpus, and how the annotators were chosen. Then I will explain the modifications we decided to make to the annotation protocol from the DLESE corpus.

7.2.1 Project Selection

While the Instructional Architect web site hosts over 11,000 distinct “projects,” not all of them are likely to be amenable to our model of quality, which has been developed within the context of science education and, in particular, Earth science. While it is interesting to evaluate the approach in an extended domain, I see little chance of gaining valuable insights from applying the model, as it is, to content like literary texts and poems; education in those fields necessarily follows somewhat different structures. On
top of that, one goal was to be able to find projects that could be included in NSDL, which focuses on science. For these reasons, we decided to focus on projects that could be categorized as science, technology, engineering, or mathematics (STEM); projects outside these domains were excluded.

We further made the decision to exclude projects below a minimum length of 70 words. Projects that are shorter than that are likely to be of less original value, possibly only linking to some external resources; by excluding an easily filtered set of projects which are likely to be of less interest, we were able to ensure that the limited time our annotators had would be focused on evaluating more promising ones. Another reason was that the computational models’ decisions are based on the distribution of words within a project. If the number of words is very low, this becomes harder, and the results are more likely to become erratic and random. We also required the projects to refer to at least three external resources, since including external materials is considered central to the purpose of IA projects. Finally, projects had to be set to be publicly available and have been viewed at least twenty times. This filtering of the available resources was done in stages, first automatically excluding resources that didn’t fit the strict criteria, then manually filtering the remaining resources by content. In this way 300 projects were identified; more would have been available, but this number was deemed sufficient for the purposes of this study.

After determining the amount of time it would take the annotators to evaluate a project, and how much annotator time was available to us, we decided on a preliminary split of 20 projects for an initial practice round, and another 200 projects for the main annotation round, with an option of extending the main annotation if resources permitted.

7.2.2 Annotators

Teachers who had previously used Instructional Architect and who had taken part in a professional development workshop within the last two years were targeted as annotators for the study. Using workshop survey data and IA usage logs, teachers were identified who had taught STEM subjects in grades 6–12 for at least three years, and had used their IA content in their classroom. Teachers were offered a small stipend for their participation. Out of the candidates, three were chosen to participate in the study.
The annotation was performed using custom software (a modified version of the tool used during the DLESE study). After being introduced to the annotation protocol, annotators were given 20 projects as a practice round. After completing those, the annotators had the chance to provide feedback on problems they encountered, but no serious issues were noted, and so the remaining annotation followed the same protocol. The main annotation round consisted of 200 projects, which was supplemented with an additional 10 after completion, making a total of 230 annotated IA projects. Unlike the DLESE corpus, each annotator evaluated each of the 230 projects, so each project was triple-annotated. Mostly due to resource and time constraints we decided against an additional adjudication round.

### 7.2.3 Modified Annotation Scale

For the DLESE corpus annotation the annotators were asked to determine the presence or absence of each of seven quality indicators, indicating their answer as “yes” or “no”; where the presence of an indicator could only be meaningfully stated if another indicator was present, the answer was recorded as “n/a” if that other indicator was marked as not present (i.e., if “identifies age range” was marked as “no”, then “appropriate for age” was marked as “n/a”).

As Section 5.5 detailed, one finding during our initial error analysis was that indicator presence can sometimes be less clear cut. On the one hand, the **degree of indicator presence** can vary. In some cases an indicator is only weakly present, for example a resource might have the barest minimum of instructions; when put to the question of “does the resource have instructions, yes or no?”, the answer would still be “yes”, but it’s a much weaker “yes” than in other cases, where extensive and complete instructions are present. On the other hand, the **certainty of indicator presence** that an annotator has can vary as well. An annotator might look at a resource and find themselves uncertain if it is “organized for learning goals”, maybe feeling that it is, but they’re not quite convinced. With the DLESE corpus annotation protocol, in either case the annotator was required to make a clear decision between “yes” and “no”.

For the Instructional Architect corpus annotation protocol, we decided to allow the annotators to indicate an indicator’s presence on a five point Likert scale[37]. Each indicator was formulated as a declarative statement, for example “the IA project includes instructions for the user indicating how to navigate and
<table>
<thead>
<tr>
<th>Quality Indicator</th>
</tr>
</thead>
<tbody>
<tr>
<td>1  Has Instructions</td>
</tr>
<tr>
<td>2  Links to Prestigious Sources</td>
</tr>
<tr>
<td>3  Identifies Learning Goals</td>
</tr>
<tr>
<td>4  Organized Appropriately for Learning Goals</td>
</tr>
<tr>
<td>5  Identifies Age Range</td>
</tr>
<tr>
<td>6  Content Seems Appropriate for Age Range</td>
</tr>
</tbody>
</table>

Table 7.1: List of quality indicators in the IA corpus

These six quality indicators were selected to be annotated in the IA corpus.

use the project and linked resources," with some clarifying examples. The annotator could then state their level of agreement with that statement by choosing from “strongly disagree,” “disagree,” “neither agree nor disagree,” “agree,” and “strongly agree”. Similar to the DLESE corpus, dependent quality indicators were set to “n/a” if the indicators they depended on were marked as “strongly disagree”.

This more nuanced annotation scale gives a more fine grained look at agreement between annotators. It also helps during computational model training and evaluation: If annotators were not too certain if a project does or does not present a given indicator, that project might not be a very suitable example for either side, and it might be better not to include it in training at all. And further, it will be interesting to see if the projects that the computational models have a harder time evaluating correspond to the ones that the annotators were less certain about.

7.2.4 Quality Indicators

Because of some systematic differences between DLESE resources and Instructional Architect projects, the set of quality indicators and their corresponding descriptions couldn’t be used one-to-one for the new corpus. Even so, our goal was to keep the new set of indicators closely aligned with the original set in order to be able to compare results across the two studies more easily.

Table 7.1 lists the six quality indicators we chose to annotate on the Instructional Architect corpus. In the following I will describe how each of these indicators was defined to the annotators and how it relates to the corresponding indicator from the DLESE corpus annotation. Because every IA project lists the handle of the user who created it, the indicator “has sponsor” would, by definition, always be marked as “yes”; for
this reason we chose not to include it in the annotated set. A copy of the exact instructions given to the annotators can be found in Appendix B.

7.2.4.1 Has Instructions

“The IA project includes instructions for the user indicating how to navigate and use the project and linked resources. Example: project indicates a sequence in which the user should visit pages, describes steps in a classroom activity, or explains how to install and use a piece of software.”

This quality indicator corresponds very closely to the indicator of the same name in the DLESE corpus. Only the wording of the explanation was changed and examples were replaced to fit more closely with the structure of IA projects.

7.2.4.2 Links to Prestigious Sources

“The IA project provides a linked resource(s) to a ‘prestigious’ source. A ‘prestigious’ source includes a site where the manager or organizer is highly respected in the relevant subject area, government organizations, respected digital libraries, university maintained sites, and non-profit organizations.

“This indicator will require you to click on the resource links. In visiting resources, follow any instructions from the IA project itself (e.g. try to go to the location that students are supposed to go). Once you arrive, do not click on any additional links to make your determination.”

Instructional Architect projects are authored and published by IA users, generally individual teachers. By most definitions, they would not be considered prestigious authorities in their field – even though they may be excellent teachers, prestigious authorities are more likely to be major research institutions or well known researchers with many peer-reviewed publications. The DLESE corpus indicator of “has prestigious sponsor”, then, doesn’t really apply to the IA corpus.

Considering how relatively important that indicator was in the case of DLESE, however, we wanted to find a way to include it here as well. Since IA projects generally make extensive use of other resources (as explained at the beginning of this chapter), an essential part of their value lies in the web resources that they link to, and so we asked the annotators to evaluate the ‘prestigiousness’ of the linked sources, rather than of the author of the IA project itself.
7.2.4.3 Identifies Learning Goals

“The IA project identifies learning goals and articulates the knowledge and skills a student is expected to acquire over the course of using the project. Specific state or national standards may be given, or the resource may more informally identify its best use in an educational setting.

“Remember to use the ‘more information’ link at the bottom of the project as another way to identify learning goals and/or age range.”

This indicator mostly corresponds to the same indicator in the DLESE corpus. We specifically instructed the annotators to follow the ‘more information’ link at the bottom of each IA project. The ‘more information’ page is part of the template all IA projects follow, and it includes some descriptive information that the author entered, including the target age range and a summary of the content.

7.2.4.4 Organized Appropriately for Learning Goals

“The IA project is organized appropriately for its learning goals. The goals are clearly organized so that each goal has a corresponding description or activity.”

This closely corresponds to the same indicator in the DLESE corpus. The wording was only modified slightly to account for the fact that all IA projects are contained within one page (not counting the external links).

7.2.4.5 Identifies Age Range

“The IA Project identifies its target student age range by stating the expected age or grade level of its intended users in the project itself or in the ‘more information’ link at the bottom of the project; or the project structure, using sections targeted to users with different levels of knowledge, implies a certain age range.”

This closely corresponds to the same indicator in the DLESE corpus. Again we instructed the annotators to take into account the ‘more information’ page.

7.2.4.6 Content Seems Appropriate for Age Range

“The IA project’s content seems appropriate for its age range; someone with little expertise in education would judge that the reading or activities were neither much too hard nor much too easy for the given grade level. For example, playing with crayons is generally inappropriate for high school audiences, and very technical terminology is inappropriate for elementary school students.”
This closely corresponds to the indicator “not inappropriate for age” in the DLESE corpus. We decided to phrase the indicator in the positive instead of the double negative, which had been confusing to people. As explained in Section 4.2, our aim was not to determine that a resource was positively appropriate for the target age group, as age appropriateness is a highly complex and frequently subjective matter. Rather, we wanted to capture instances where the project’s content was very clearly not appropriate.

7.3 Evaluation

The evaluation of the Instructional Architect corpus is a bit more complex than for the DLESE corpus; because the IA projects were annotated on an interval, simple agreement numbers are insufficient to measure inter-rater reliability. In the following I will look at a few different metrics in order to characterize the results of the IA annotation, and I will try to compare the quality of the annotation to the original DLESE corpus, as far as that’s possible.

Figures 7.3 through 7.9 show the distribution of ratings used during annotation, both broken up by annotator and by quality indicator, and totaled up. Ratings of “N/A” were omitted for simplicity, and because their number is quite low (less than 100 out of 3600 possible ratings), and they don’t contribute much to the overall analysis.

Looking at the totals (Figure 7.9), the first observation is a strong bias towards high ratings of “5”, indicating a likely ceiling effect – that is, the distribution of ratings doesn’t accurately capture differences in highly rated resources, because the scale limits ratings to no higher than 5. This may not pose a major problem, because the lower tail of the distribution is represented well. Another interesting fact is that there are distinct differences between how the three annotators used the scale available to them. While the ratings of annotators 1 and 3 could be assumed to follow a normal distribution (truncated at the higher end, thus becoming a one-sided Gaussian), annotator 2 follows a distinctly bimodal distribution, strongly preferring assertive judgments (“strongly agree” vs. “strongly disagree”) over the more ambiguous middle values. In fact, annotator 2 never made use of rating 3, “neither agree nor disagree”.

---

1 In part, this is because the criteria for rating an indicator “N/A” were fairly strict: In the DLESE corpus, if an indicator was labeled “No”, its dependent indicator would be labeled “N/A”, but in the IA corpus, the dependent indicator was only rated “N/A” if the depended-on indicator was labeled “1 – strongly disagree” out of 5.
Table 7.2: Quality indicator bias in the DLESE and IA corpora

<table>
<thead>
<tr>
<th>Quality indicator</th>
<th>DLESE % “Yes”</th>
<th>IA % “5”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Has instructions</td>
<td>39%</td>
<td>69.3%</td>
</tr>
<tr>
<td>Has sponsor</td>
<td>97%</td>
<td>(not rated)</td>
</tr>
<tr>
<td>Has prestigious sponsor</td>
<td>34%</td>
<td>30.5%</td>
</tr>
<tr>
<td>Identifies age range</td>
<td>20%</td>
<td>84.3%</td>
</tr>
<tr>
<td>“Appropriate for age”</td>
<td>99%</td>
<td>69.6%</td>
</tr>
<tr>
<td>Identifies learning goals</td>
<td>28%</td>
<td>45.1%</td>
</tr>
<tr>
<td>Organized for goals</td>
<td>76%</td>
<td>43.9%</td>
</tr>
</tbody>
</table>

The percentage of “Yes” and “5” ratings on the DLESE and IA corpora, respectively. Both very high and very low numbers are undesirable, as they indicate a strongly one-sided annotation. Such a data set does not provide good statistical separation between quality levels. (DLESE numbers are from Table 4.3.)

In comparing the distribution across individual quality indicators, we find that for some of them the bias towards high ratings is much more pronounced than for others. For example 84.3% of ratings on “identifies age range” have value 5 (see Figure 7.7), and 69.3% for indicator “has instructions” (Figure 7.3). Ideally the ratings would be more distributed, but we encountered this same effect during the DLESE annotation, as shown in Table 7.2. The most biased indicators in the DLESE corpus were “has sponsor” and “not inappropriate for age”, which were labeled as present in 97% and 99% of resources, respectively. By that measure, the Instructional Architect corpus succeeds in providing on average somewhat more balanced data than the DLESE corpus.

After looking at the distribution of annotators independently, I will look at the agreement of their joint ratings on IA projects. Unlike the DLESE corpus, measuring only exact agreement is less interesting. In one case, annotator 1 might rate a project as “strongly agree” on “has instructions”, while annotator 2 rates the same project as “strongly disagree”; on another project, annotator 1 might say “agree”, while annotator 2 says “neither agree nor disagree”. Clearly these two cases are not equivalent, and our evaluation metric needs to take into account the numeric nature of the ratings.

Table 7.3 shows a few different metrics related to the inter-rater reliability on the IA corpus. The first measure I present is mainly an attempt to reproduce the reliability measure used to evaluate the DLESE corpus. In the DLESE corpus, agreement was measured using Cohen’s $\kappa$, a metric that aims to measure the agreement between annotators while controlling for chance agreement (compare Section 4.4). Cohen’s $\kappa$,
however, is only defined for the special case of categorical annotations (e.g. “Yes” vs. “No”), not the interval annotations used in the IA corpus; it also is not defined for more than two annotators. In order to compute $\kappa$ values that are at least somewhat comparable to the DLESE values, I mapped all ratings of projects above “3” (“agree” and “strongly agree”) to “Yes”, all ratings below “3” (“disagree” and “strongly disagree”) to “No”, ignoring the middle point in the scale (“neither agree nor disagree”). I then computed the $\kappa$ values for each pairing of the annotators (three pairings total) and found the average. A possible interpretation of $\kappa$ values, according to Landis & Koch\[36\], is: less than 0.2 is “slight agreement”, 0.2 – 0.4 is “fair agreement”, 0.4 – 0.6 is “moderate agreement”, 0.6 – 0.8 is “substantial agreement”, and 0.8 – 1.0 is “almost perfect agreement”.

When comparing these values with DLESE, we see a notable drop in performance. The evaluation, however, is not yet conclusive, because this use of $\kappa$ discards a lot of the recorded information in the IA data set and thus might put it at an undue disadvantage.

Unfortunately, further measures confirm this drop in performance. The next I report is Krippendorff’s $\alpha$, which can be seen as a generalization of $\kappa$ to multiple annotators and arbitrary rating scales; while the values are not strictly comparable, the value one would expect for a given “goodness” of agreement is similar. In this case, $\alpha$ confirms the $\kappa$ measure, indicating even lower agreement. I also provide averaged Pearson product-moment correlation coefficient; this does not account for chance agreement between annotators, and so it tends to overestimate performance in some cases\(^2\).

Both correlation coefficients are invariant to affine transformations of the input data, that is, multiplying all ratings of one annotator by a constant number (except zero) and / or adding or subtracting a constant does not affect the correlation coefficient. The same is not true for Krippendorff’s $\alpha$ (nor for my modification of $\kappa$, which uses a constant threshold to map ratings to “Yes” and “No”). To explore how much the differences between the annotators’ use of the 1 – 5 scale affects the outcome, I normalized each annotator’s ratings by subtracting the mean and dividing by the standard deviation (thus each annotator’s distribution of ratings is standardized to a mean of 0 and a standard deviation of 1). As seen in the table,

\(^2\) One might argue that Spearman’s rank correlation coefficient is more appropriate here than Pearson, considering that the relative rankings of IA projects are more important than the magnitude of the rating. I am not providing those numbers here, but they are mostly in line with Pearson.
### Table 7.3: Inter-rater reliability metrics for the IA corpus

<table>
<thead>
<tr>
<th>Quality Indicator</th>
<th>&quot;Cohen's $\kappa$&quot;</th>
<th>$\alpha$ (normalized)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Has instructions</td>
<td>0.390</td>
<td>0.382</td>
</tr>
<tr>
<td>Has prestigious sponsor</td>
<td>0.264</td>
<td>0.363</td>
</tr>
<tr>
<td>Identifies age range</td>
<td>0.481</td>
<td>0.551</td>
</tr>
<tr>
<td>&quot;Appropriate for age&quot;</td>
<td>0.259</td>
<td>0.312</td>
</tr>
<tr>
<td>Identifies learning goals</td>
<td>0.260</td>
<td>0.316</td>
</tr>
<tr>
<td>Organized for goals</td>
<td>0.185</td>
<td>0.125</td>
</tr>
</tbody>
</table>

Normalization raised $\alpha$ values considerably on most indicators, although agreement numbers still only rank “fair” to “moderate”.

Earlier I remarked on the fact that it appears as though annotators 1 and 3 are using the available rating scale in a similar fashion, while annotator 2’s distribution of ratings is noticeably different. It stands to reason that annotators 1 and 3 might, then, also tend to agree more on how they rate each resource, and that annotator 2 is, essentially, an outlier. Table 7.4 shows what happens to the agreement numbers if we discard annotator 2’s ratings and only compare between annotators 1 and 3. Indeed we find significantly higher agreement on indicators “has prestigious sponsor” and “organized for goals”. On the other indicators the effects are divided, with slightly higher agreement on some, and slightly lower agreement on others.

In order to provide a fair comparison, Table 7.5 shows these same metrics computed on the DLESE corpus annotation. Normalized $\alpha$ is omitted as it has no meaningful effect on nominal data; so is Spearman’s rank correlation coefficient, as it mathematically works out to be identical to Pearson for this format of data.

### 7.4 Conclusion

The aim of creating the Instructional Architect corpus was to evaluate the applicability of this approach to characterizing quality to peer produced educational resources, as well as to provide a test bed for evaluating the quality indicator models on a new data set and adapting them to the new set of requirements. Unfortunately, the inter-rater reliability achieved in this corpus is notably lower than the good agreement we achieved on the DLESE corpus. There are a number of factors that may have contributed to this. The scale of the study was significantly smaller, both in amount of time available for devising and optimizing the
<table>
<thead>
<tr>
<th>Quality Indicator</th>
<th>“Cohen’s $\kappa$”</th>
<th>$\alpha$</th>
<th>Pearson</th>
<th>$\alpha$ (normalized)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Has instructions</td>
<td>0.373</td>
<td>0.272</td>
<td>0.359</td>
<td>0.360</td>
</tr>
<tr>
<td>Has prestigious sponsor</td>
<td>0.524</td>
<td>0.366</td>
<td>0.434</td>
<td>0.435</td>
</tr>
<tr>
<td>Identifies age range</td>
<td>0.365</td>
<td>0.357</td>
<td>0.498</td>
<td>0.499</td>
</tr>
<tr>
<td>“Appropriate for age”</td>
<td>0.397</td>
<td>0.351</td>
<td>0.453</td>
<td>0.521</td>
</tr>
<tr>
<td>Identifies learning goals</td>
<td>0.334</td>
<td>0.372</td>
<td>0.423</td>
<td>0.424</td>
</tr>
<tr>
<td>Organized for goals</td>
<td>0.526</td>
<td>0.322</td>
<td>0.273</td>
<td>0.324</td>
</tr>
</tbody>
</table>

Table 7.4: Inter-rater reliability metrics, only between annotators 1 and 3

<table>
<thead>
<tr>
<th>Quality Indicator</th>
<th>Cohen’s $\kappa$</th>
<th>$\alpha$</th>
<th>Pearson</th>
</tr>
</thead>
<tbody>
<tr>
<td>Has instructions</td>
<td>0.675</td>
<td>0.765</td>
<td>0.769</td>
</tr>
<tr>
<td>Has sponsor</td>
<td>0.828</td>
<td>0.670</td>
<td>0.721</td>
</tr>
<tr>
<td>Has prestigious sponsor</td>
<td>0.316</td>
<td>0.485</td>
<td>0.555</td>
</tr>
<tr>
<td>Identifies age range</td>
<td>0.643</td>
<td>0.738</td>
<td>0.739</td>
</tr>
<tr>
<td>“Appropriate for age”</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Identifies learning goals</td>
<td>0.552</td>
<td>0.692</td>
<td>0.696</td>
</tr>
<tr>
<td>Organized for goals</td>
<td>0.333</td>
<td>0.691</td>
<td>0.574</td>
</tr>
</tbody>
</table>

Table 7.5: Inter-rater reliability metrics for the DLESE corpus

For comparison with Table 7.3, the DLESE inter-rater reliability measures. The data set used for this computation showed perfect agreement on the “appropriate for age” indicator, so correlation and agreement numbers can’t be meaningfully computed.
annotation protocol, and in resources for training annotators. The annotators were teachers who may not have had any prior experience with studies of this kind, and more rounds of training would probably have improved their consistency. And finally, the IA projects that were the focus of this annotation are overall of much smaller size and simpler structure.

Even with the noted deficiencies in the data, this corpus will still prove very useful. While the reliability numbers are not very good, agreement is by no means random; this clearly indicates that the indicators are applicable, and that they capture aspects of the projects that are perceived similarly by all annotators. This does not necessarily mean that these specific quality indicators offer a meaningful and complete characterization of the educational quality of IA projects; the data we have does not allow that kind of analysis. But, when taking into account the necessary caveats, the corpus is a suitable target for evaluating the quality indicator models. I will describe my experiments on that front in the following chapter.
Figure 7.3: Distributions of ratings on the Instructional Architect annotation – “has instructions”

This shows the distribution of ratings of the Instructional Architect annotation for the “has instructions” quality indicator split by annotator, and the total. The indicated ratings of 1–5 correspond to: 1 – strongly disagree, 2 – disagree, 3 – neither agree nor disagree, 4 – agree, 5 – strongly agree.
Figure 7.4: Distributions of ratings on the Instructional Architect annotation – “links to prestigious sources”

This shows the distribution of ratings of the Instructional Architect annotation for the “links to prestigious sources” quality indicator split by annotator, and the total. The indicated ratings of 1–5 correspond to: 1 – strongly disagree, 2 – disagree, 3 – neither agree nor disagree, 4 – agree, 5 – strongly agree.
Figure 7.5: Distributions of ratings on the Instructional Architect annotation – "identifies learning goals"

This shows the distribution of ratings of the Instructional Architect annotation for the “identifies learning goals” quality indicator split by annotator, and the total. The indicated ratings of 1–5 correspond to: 1 – strongly disagree, 2 – disagree, 3 – neither agree nor disagree, 4 – agree, 5 – strongly agree.
Figure 7.6: Distributions of ratings on the Instructional Architect annotation – “organized for learning goals”

This shows the distribution of ratings of the Instructional Architect annotation for the “resource is organized appropriately for learning goals” quality indicator split by annotator, and the total. The indicated ratings of 1–5 correspond to: 1 – strongly disagree, 2 – disagree, 3 – neither agree nor disagree, 4 – agree, 5 – strongly agree.
Figure 7.7: Distributions of ratings on the Instructional Architect annotation – “identifies age range”

This shows the distribution of ratings of the Instructional Architect annotation for the “identifies age range” quality indicator split by annotator, and the total. The indicated ratings of 1–5 correspond to: 1 – strongly disagree, 2 – disagree, 3 – neither agree nor disagree, 4 – agree, 5 – strongly agree.
Figure 7.8: Distributions of ratings on the Instructional Architect annotation – “content seems appropriate for age range”

This shows the distribution of ratings of the Instructional Architect annotation for the “content seems appropriate for age range” quality indicator split by annotator, and the total. The indicated ratings of 1–5 correspond to: 1 – strongly disagree, 2 – disagree, 3 – neither agree nor disagree, 4 – agree, 5 – strongly agree.
Figure 7.9: Distributions of ratings on the Instructional Architect annotation – all quality indicators

This shows the distribution of ratings of the Instructional Architect annotation for all quality indicator combined split by annotator, and the total. The indicated ratings of 1–5 correspond to: 1 – strongly disagree, 2 – disagree, 3 – neither agree nor disagree, 4 – agree, 5 – strongly agree.
In the following I will discuss my experiments on how to apply the existing quality indicator models to the new data that the Instructional Architect corpus provides. The most obvious way to evaluate the existing models is to run them, trained on the DLESE data as they are, on the new corpus and compare the results to the human annotation, as done on the DLESE corpus. Considering the differences in structure and content between DLESE and IA, however, it will also be interesting to see how the models can be adapted to generalize better. There are two main directions in which this can be taken:

1. By incorporating part of the IA corpus into the model training (along with the DLESE training data), the models should learn to cue in on features that are present in both corpora instead of focusing on ones that only show up in DLESE. The proportion of DLESE data to IA data can be varied.

2. By modifying the feature set used by the models, it may be possible to artificially exclude features that would tend to be too specific to one or the other corpus, thus making the models more generalizable.

I will explore these approaches below.
8.1 Preliminary Steps

Before it becomes possible to run the quality indicator models on the IA corpus at all, it is necessary
to define how to translate between the different forms of annotation. As detailed in Chapter 4 and Chapter 7,
the annotation on both corpora differed in a few important ways:

- Annotators in the DLESE corpus rated each indicator on each resource as “yes” or “no”. There
  were two annotators, and they did not rate all the same resources. The subset of resources that they
  both rated was manually adjudicated in a separate step. Thus, every resource is ultimately labeled
  authoritatively as either “yes” or “no” – in the case of single-annotated resources, this represents
  one annotator’s assessment, in the case of double-annotated resources, it represents the adjudication,
  based on both annotators’ assessments.

- Annotators in the IA corpus rated each indicator one each resource on a scale from 1 – “strongly
  disagree” to 5 – “strongly agree”. There were three annotators, and all three annotators rated every
  resource in the corpus. No adjudication took place, so there are three independent assessments.

The quality indicator models were developed within the framework of a classification task, that is,
to decide if a quality indicator is present within a resource or not. This aligns well with the way the DLESE
corpus was set up. During training, this means that a resource that has been labeled as “has instructions”
can serve as a positive example to learn from, while a resource that has not been labeled as “has instructions”
can serve as a negative example. Conversely, when evaluating model performance, if the corpus annotation
rates a resource as “has instructions”, and the computational model does not rate it so, it can be concluded
that the model made a mistake, whereas if the model and the corpus annotation match up, the model can
be assumed to have judged correctly.

Even if we use only one annotator, these procedures become more complicated in the case of the
IA corpus annotation. It might be possible to map the output of the quality models to a 1 – 5 scale by
applying thresholds to the pseudo-probability output (compare Section 6.6). But even if both annotator
and computational model use the same scale, does it make sense to measure for exact agreement? If the
annotator rated a resource as “strongly agree” on “has instructions” while the model only said “agree”,
clearly this is not the same as if the model and the annotator completely disagree in their assessment. This problem mirrors the complications we ran into when evaluating agreement between annotators in Section 7.3, and the same methods might be applicable here.

This might allow us to evaluate the model performance on the IA corpus, but training is different: The Support Vector Machine algorithm requires the training examples to be labeled as either positive or negative examples\(^1\). In order to include part of the IA corpus as training examples, the graduated labels have to be mapped to either “yes” or “no”. I see two basic approaches to this:

1. Ratings 1 and 2 are mapped to “no”, ratings 4 and 5 are mapped to “yes”, and rating 3 is left unlabeled, as it presumably indicates a “neutral” assessment. One might also include rating 2 and 4 in the “neutral” category, since they might not serve as particular good examples, since they are presumed to be more ambiguous than their 1 and 5 counterparts.

2. If we assume that the annotators’ internal representation of the scale is not necessarily the one that is spelled out in the instructions (as supported by the fact that the annotators’ use of the scale differs), we might decide to select a different threshold value that is closer to the annotators’ mean rating. In the case of the IA data, this would most likely result in mapping a rating of 5 to “yes”, and all other rating to “no”, considering the very strong bias towards ratings of 5.

This approach of mapping the ratings to “yes” / “no” labels could, of course, also be applied to the evaluation of the computational models’ performance, thus making it more comparable with the evaluations done on the DLESE corpus.

### 8.2 Experimental Setup

Some properties of the IA corpus have to be taken into account when evaluating the existing DLESE models. The “part of resource” crawler configuration from DLESE is not applicable to this data, because the extent of each Instructional Architect project is strictly defined by the site structure – it consists almost

\(^1\) Depending on the SVM implementation, unlabeled examples are also possible. In fact, SVM has also been used for regression problems, in which case examples can have a numerical label. In my experiments I have made use of unlabeled examples, but I did not try to re-cast the problem as regression.
exclusively of the main project page. For this reason we also did not record which pages annotators thought were “relevant” to the resource during IA annotation. For these reasons I decided to run these experiments using the simple “front page” crawler configuration, which only takes the front page of the resource or IA project into account.

One of the more useful features during DLESE evaluation was the “resource URL” feature. This feature tends to key in on the host and domain that the resource is found under; but all IA projects are hosted on the Instructional Architect server, so the URL can have no distinguishing power. Instead I used the “links” feature, which showed similar discriminative power to “resource URL” (compare Table 6.4 for feature importance in DLESE). The following experiments thus use the “links” and the “bag of words” features.

To facilitate later experiments that incorporate part of the IA corpus into the training set, I decided to separate the available data into two parts: a training set and a test set. The data from the training round (20 IA projects) was ignored; of the remaining ca. 200 projects, 100 were assigned to the training set, the rest to the test set. All the numbers reported below are on this test set.

The available data can be combined in different ways to train models. The most direct approach uses computational models trained on only the DLESE corpus and applies them, as they are, to the Instructional Architect test set; this will be discussed in Section 8.3. It is also the most challenging approach: although the IA corpus was designed to follow similar guidelines as the DLESE corpus, and to use similar quality indicators, IA projects are still quite different from DLESE resources, and attempts to apply machine learning models to data that is as different are often not successful.

Instead of using the DLESE data to train these models, they can be trained on only the IA training set (shown in Section 8.4). This is analogous to the experiments on the DLESE corpus: training and test set are sampled from the same set and are (presumably) similar in format and style. This allows conclusions about the generalizability of the methodology, but not about the generalizability of existing models to new domains – it requires re-training models for the new data.

A third alternative is a combination of both approaches: using some of the training data from DLESE and some of the training data from the IA corpus, to train models that generalize across both corpora. For
data sets that are as diverse as DLESE and IA, the differences in distribution between corpora may pose a problem here. My experiments in this direction did not yield interesting or promising results, and they are not discussed here.

It is not immediately clear how the performance of the quality indicator models should be evaluated on this data set; the previous section outlined the difficulties in comparing the output of the quality indicator models with the annotators’ ratings on IA projects. I will begin with the obvious: mapping the annotators’ ratings to a binary scale ("agree" / "disagree"), and comparing them directly to the models’ classification. This mirrors closely the way I evaluated the performance on DLESE (where the annotators’ ratings were binary to begin with). In order to do so, I mapped ratings 1 and 2 to "disagree", and ratings 4 and 5 to "agree"; IA projects that were rated as 3, “neither agree nor disagree”, were ignored. Similar to the previous evaluations, I include the majority class baseline, that is, what accuracy can be achieved by always labeling a resource with the most common rating.

8.3 Models Trained on DLESE

As Table 8.1 shows, the majority class baseline is very high. This is result of the heavily skewed annotation, as explained in Section 7.3. Because the annotators had a strong tendency to choose a rating of 5, most ratings, in the binary mapping, will be “agree”. Thus always choosing “agree” as a rating shows high agreement with the annotators. The models trained on the DLESE data set generally underperform the majority class baseline (except on “seems appropriate for age” and “organized for goals”), but this is not surprising: the DLESE data set is very different from the IA corpus. In particular it is worth pointing out that the majority class baseline is determined from the rating proportions in the test set. This information, of course, is not available to the trained models.

If we restrict the evaluation only to those projects on which the quality indicator models indicated strong evidence for their assessment (i.e. a decision value of less than -1 or more than 1), the agreement with the human annotators improves on two indicators (but declines further on some). However, in this evaluation on average about 58% of projects are left unrated, because the models’ assessment is too close to neutral.
Quality indicator baseline DLESE trained models trained models trained models

<table>
<thead>
<tr>
<th>Quality indicator</th>
<th>baseline</th>
<th>DLESE trained model</th>
<th>trained models (high confidence)</th>
<th>trained models (crawler)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Has instructions</td>
<td>92.4%</td>
<td>89.8%</td>
<td>100.0%</td>
<td>72.4%</td>
</tr>
<tr>
<td>Has prestigious sponsor</td>
<td>53.0%</td>
<td>44.0%</td>
<td>63.6%</td>
<td>57.6%</td>
</tr>
<tr>
<td>Indicates age range</td>
<td>91.0%</td>
<td>12.6%</td>
<td>8.3%</td>
<td>17.5%</td>
</tr>
<tr>
<td>Seems appropriate for age</td>
<td>93.0%</td>
<td>98.9%</td>
<td>98.9%</td>
<td>98.9%</td>
</tr>
<tr>
<td>Identifies learning goals</td>
<td>91.3%</td>
<td>50.0%</td>
<td>50.0%</td>
<td>46.5%</td>
</tr>
<tr>
<td>Organized for goals</td>
<td>91.2%</td>
<td>96.8%</td>
<td>96.0%</td>
<td>92.6%</td>
</tr>
</tbody>
</table>

Table 8.1: Quality indicator models trained on DLESE corpus

These results used only the front page of the IA project during feature extraction. I implemented a specialized crawler that includes the descriptive “more information” page for each project, as well as each external page directly linked from the project. When evaluating the DLESE models in this configuration, we see increased performance in “has prestigious sponsor” and “indicates age range” — this is expected, because the information about the target age group is often found on the “more information” page, and “prestigious sponsor” during IA annotation specifically referred to the linked pages, not the IA project itself. But at the same time, performance drops on some other indicators with the added information.

8.4 Models Trained on Instructional Architect

When training the quality indicator models on the IA corpus itself (that is, the held-out training set of 100 projects) instead of on the DLESE corpus, the models are able to take the statistical distribution of the IA data into account, and performance increases. Table 8.2 shows that, in this case, the quality indicator models are able to improve upon the majority class baseline on all indicators; restricting the evaluation to the high confidence ratings improves the numbers by a small amount (most notably on “has prestigious sponsor”), at the cost of leaving about 16% of projects unrated.

Considering the relatively small number of annotated projects in the IA corpus, these results do not carry too much statistical weight. According to McNemar’s test for statistical significance, close to perfect accuracy would be required on most indicators to show a statistically significant improvement over the baseline on the full test set, and the difference between test configurations is often only one or two correctly classified projects more or less. But the experiments do suggest that the methodology presented here is valid.
### Table 8.2: Quality indicator models trained on IA corpus

<table>
<thead>
<tr>
<th>Quality indicator</th>
<th>baseline</th>
<th>IA trained model</th>
<th>trained models (high confidence)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Has instructions</td>
<td>92.4%</td>
<td>98.0%</td>
<td>98.0%</td>
</tr>
<tr>
<td>Has prestigious sponsor</td>
<td>53.0%</td>
<td>61.0%</td>
<td>76.5%</td>
</tr>
<tr>
<td>Indicates age range</td>
<td>91.0%</td>
<td>96.1%</td>
<td>96.8%</td>
</tr>
<tr>
<td>Seems appropriate for age</td>
<td>93.0%</td>
<td>98.9%</td>
<td>98.8%</td>
</tr>
<tr>
<td>Identifies learning goals</td>
<td>91.3%</td>
<td>97.7%</td>
<td>98.8%</td>
</tr>
<tr>
<td>Organized for goals</td>
<td>91.2%</td>
<td>96.8%</td>
<td>96.8%</td>
</tr>
</tbody>
</table>

Beyond the domain explored in the DLESE corpus. Ideally, the knowledge gained from the DLESE corpus combined with the training data in the IA corpus would produce machine learning models that improve upon either data set alone. The fact that the DLESE-trained models could be applied to some effect on the IA data (see Table 8.1) suggests that this may be viable, that the same model could perform well on such diverse documents as a DLESE resource and an IA project. Training on a combination of both corpora, however, does not have this desired effect and underperforms the more specialized models shown here. I will omit these results here, as they do not give any further insight into the matter.

### 8.5 Vocabulary and Generalizability

There are many directions one could take to attempt to improve generalizability of the models to new data sets. In general, these approaches will try to induce the machine learning algorithm to focus on the aspects of resources that tend to be similar across data sets rather than the details in which they differ. I would like to briefly explain one direction I explored within the limits of this data set: controlling the vocabulary.

Since a bag-of-words serves as the main feature by which a resource is presented to the algorithms, which words are used by a resource strongly affects the quality indicator models. Training on a fairly narrow domain corpus, such as DLESE, one can expect that resources tend to use a similar vocabulary, one that is noticeably different from a data set with a broader focus such as Instructional Architect. For example, DLESE contains many specialized scientific terms from the field of Earth science. While some scientific terms also show up in the IA corpus, the distribution is quite different. The machine learning algorithms effectively
Table 8.3: Effect of limiting vocabulary

<table>
<thead>
<tr>
<th>Quality indicator</th>
<th>all words</th>
<th>filtered by IA corpus</th>
<th>filtered by Brown corpus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Has instructions</td>
<td>89.8%</td>
<td>82.7%</td>
<td>90.8%</td>
</tr>
<tr>
<td>Has prestigious sponsor</td>
<td>44.0%</td>
<td>45.8%</td>
<td>47.5%</td>
</tr>
<tr>
<td>Indicates age range</td>
<td>12.6%</td>
<td>10.7%</td>
<td>17.5%</td>
</tr>
<tr>
<td>Seems appropriate for age</td>
<td>98.9%</td>
<td>98.9%</td>
<td>98.9%</td>
</tr>
<tr>
<td>Identifies learning goals</td>
<td>50.0%</td>
<td>36.0%</td>
<td>51.1%</td>
</tr>
<tr>
<td>Organized for goals</td>
<td>96.8%</td>
<td>95.8%</td>
<td>95.8%</td>
</tr>
</tbody>
</table>

discard all words in the training corpus that don’t occur at least five times, but even of the remaining set 57% of words that occur in DLESE do not show up in the IA corpus at all. Any information that the models gain from these words during training on the DLESE corpus will be useless when applied to IA.

Table 8.3 shows the effect of limiting the available vocabulary during training to words that also occur in another corpus; that is, the bag-of-words features during training on the DLESE corpus only contain words that not only show up in the respective resource, but also occur somewhere in the other corpus. Interestingly, only allowing the DLESE models to use words that also occur in the IA corpus reduces the models’ performance on the IA test set (in addition to reducing performance on the DLESE test set, which I’m not showing here). This is probably an effect of the small size of the IA corpus: it only contains 5057 distinct tokens, as opposed to 27993 in the DLESE corpus, and only 3678 of them also occur in DLESE. Limiting the vocabulary this far makes it difficult to find a viable statistical model on the DLESE training corpus to begin with.

On the other hand, when using a larger, broader corpus to filter the DLESE bag-of-words features, results improve slightly. Here I am using the Brown corpus, which contains a broad range of normal written English, including news coverage, political and industry reports, and fiction. Intuitively this would make sense — the quality indicators cover not specialized scientific concepts but educational ideas that one would expect to find mentioned in such materials, not requiring any specialized terminology but relying on standard English. However, it is important to keep in mind that the improvements shown here are based on a very small number of documents, and no definite conclusions can be drawn from them.
8.6 Conclusions

The experiments on the Instructional Architect corpus have shown that the methodology reported here is applicable not only to the kind of data found in the DLESE corpus that it was developed on, but also to the shorter, less professional, peer-produced materials found in Instructional Architect. When the models were trained on the new data, they adapted well and achieved high agreement with the human annotators. When trained on DLESE, the models only generalized well on some of the indicators. In future applications, it will be important to explore ways in which existing models can be induced to generalize to new data with only small amounts of additional annotation.

Ultimately, the limiting factor in these experiments was, again, the data set. To draw strong conclusions about the effectiveness of various approaches, more data is needed overall. In future annotation projects, care should be taken to ensure that annotators rate consistently, and that the format of annotation is adequate to the task and captures the full variability of the data. For these computational methods, in particular, a strongly biased or inconsistent data set is problematic in two ways: Training a good model becomes harder, and determining if observed improvements reflect real advances in the methodology becomes impossible.
Chapter 9

Discussion

The motivation of this work was to explore the role computational methods can play in assessing and characterizing the quality of online educational digital resources. It began by analyzing the processes human experts go through when making similar assessments; this clearly showed that experts think about educational quality in a multi-dimensional way, weighing strengths and weaknesses against each other. It also showed that, often, complex assessments (e.g. “does the resource provide adequate pedagogical guidance”) can be approximated by looking for less complex, more easily recognized indicators (e.g. “does the resource text contain instructions”). This opened up the possibility of applying computational methods to the problem — while complex assessments require deep reasoning, the quality indicators can be recognized using available methods of computational semantics.

We created the annotated DLESE corpus with the explicit goal of using it as a test bed for developing computational models for some of these quality indicators. In creating this corpus, we demonstrated that human annotators, given adequate training and guidance, can consistently annotate such indicators; there is a balance between decisions that are “too subjective” and are hard to find agreement on (as in the case of the “has prestigious sponsor” indicator in the DLESE corpus), and decisions that are “too easy”, where human annotation is not of value (e.g. the “not inappropriate for age” indicator in the DLESE corpus, which was true for just about every resource in the corpus).

The DLESE corpus provided a good basis for developing, training, and testing computational models for quality indicators. In spite of its limitations — most importantly, insufficient diversity of ratings on some of the quality indicators — we were able to create models that approximated human assessment on the
indicators covered by the corpus. In doing so, we explored a range of different computational methods to deal with the large diversity and noise in the data. The quality indicator models demonstrated here, when run on data similar to the resources in the DLESE corpus, could already be used in a range of settings, some of which I will go into in Section 9.1 below. I will also discuss some logical extensions to the computational approach in Section 9.2.

The Instructional Architect corpus offered the opportunity to test this same methodology on quite different data. IA projects are different from DLESE resources in many ways: they are shorter, more focused, the editorial process is less professional, they rely more on outside materials, and the data we looked at covered a much wider range of content matter (all of STEM, as opposed to the Earth Science focus in DLESE). The set of quality indicators used during all these experiments was originally derived from DLESE; of course, the objective of the original study that yielded the indicators was to avoid aspects that are too specific to one kind of educational content, or too specific to one area of science, but even so the indicators had to be adapted to some degree to fit the kinds of resources found in the IA corpus. We also decided to rate resources along a Likert scale instead of the binary ratings done on DLESE. In Section 9.3 I will discuss some lessons to be taken away from our annotation efforts, both on the DLESE and the IA corpus.

Running the computational models on this new IA corpus gave mixed results; it was surprising to find that the DLESE trained models generalized to some degree to the new data. This was, of course, hoped for: the computational models should identify those features that help recognize a quality indicator across a range of different types of resources. But the big differences between the data in the DLESE and IA corpora violate fundamental assumptions about similarity of data distributions in the underlying machine learning algorithms; adapting a model to a new domain is challenging even for closely related domains (as discussed, for example, in [10]), and much more so for data sets as different as here. Unfortunately, the experiments did not show a benefit in combining the DLESE and IA training corpora. Finding a way to train accurate models from a diverse collection of resources, and that generalize to multiple types of data, should be a priority for future research in this field. I will discuss this in some more detail in Section 9.4 below.
9.1 **Use Cases for DLESE Models**

The current state of the quality indicator models, as demonstrated on the DLESE corpus, may not be applicable to all educational resources that are of interest; we saw some of the limitations of the models when applied to the IA corpus, but even on the DLESE data that the models were developed on, the models' performance is not so good that one would consider using them to replace human assessments. However, there are a number of ways that a quality profile of a resource, automatically created, could be applied to supplement and enhance existing processes in digital libraries, and to support tasks that experts and users of educational digital libraries are faced with now.

9.1.1 **Prioritizing Review Processes**

Current DLESE review processes require expert reviewers to manually evaluate a large number of resources. As a result, individual highly useful resources may be stuck in the queue for a long time, while the reviewer processes a large number of resources of lower quality that were submitted first. Speeding up this process and reducing the time from submission of a resource to making it accessible to users would be a valuable contribution.

Even though the quality indicator models, in a generalized digital library context, cannot approach human reviewer reliability, they can identify resources that are particularly promising to be of exceptionally high quality (or that are especially unlikely to fulfill the required criteria). High quality resources can then be moved to the start of the review queue, while low quality ones can be pushed to the end; the human reviewer can thus focus on the more interesting resources first, and high quality resources are likely to be made accessible more quickly.

9.1.2 **Scaffolding Expert Reviews**

During review, the expert is confronted with the resource and little information to get them started. To help them identify problem areas more quickly, the digital library review system could flag potential problems for the reviewer to check, e.g. “this resource does not appear to indicate a target age range”. The reviewer would then check if that assessment is accurate and decide if the identified flaw is relevant in that
particular instance. Such a system could shorten the average time required to review a resource. Care must be taken to avoid the reviewer being overly biased towards agreeing with the automatic assessment, since, considering the models’ limitations, the ultimate decision should always be based on the content of the resource, not just on the output of the quality indicator models.

9.1.3 Quality Indicators to Improve Search

Users of educational digital libraries such as DLESE frequently interact with the library using a search interface, that is, the user has a specific goal in mind and hopes that the library will provide materials that will help achieve this goal. Keyword search alone goes a long way towards helping users find relevant materials, but quality aspects of resources could become a part of such a system. As is the case with most searches on the web, a simple keyword search query is likely to return vast amounts of information — far too much to be useful. The user will maybe sample a few of the first results, which hopefully contain one that matches their requirements — but if the good resources are found further down the list of results, the user is unlikely to discover them at all.

An overall quality score, computed by combining the individual quality indicator assessments, could be a factor in a ranking scheme, which attempts to boost the high quality resources. The good resources would be more likely to show up within the first results, and the user would be more likely to discover them. Considering the large number of resources in many digital libraries, such a scheme can be tweaked to favor precision (i.e. the percentage of resources that were correctly detected as high quality) over recall (i.e. the percentage of high quality resources that were missed).

The exact way in which all of the quality indicator models’ outputs are combined in such a ranking scheme can be manually adjusted to reflect the expectations of the average digital library user. In that case, modifying existing digital library search interfaces to make use of the quality indicator models would be completely transparent to the user and would only necessitate an extension to the library’s information retrieval architecture.

On the other hand, an existing search interface could be extended to allow users to specifically request resources that score higher on certain indicators. For example, a teacher looking for a homework activity to
assign to their students might specify that they are looking for resources that “have instructions” and are “age appropriate”, whereas someone looking for reference materials might instead prefer resources that are published by a prestigious entity. The search interface would allow the user to specify their needs, and each user’s result list would be ranked according to their individual requirements.

Digital library search interfaces could also be enhanced by providing the user with “snippets” of each result. Instead of a simple list of links to resources to match their query, each resource would be shown with a representative short section, taken from the resource’s text, which highlights parts of the resource that seem particularly relevant to the user’s query. Web search engines such as Google already do this by displaying parts of a web site that contain the query terms. Using the quality indicator models (with a slight extension, which I’ll discuss in Section 9.2.2 below), these snippets could be chosen to reflect high-quality aspects of a resource; for example, they could show the parts of a resource that appear to carry instructions for the students. The user can use this information to quickly assess if a specific resource adequately addresses their needs.

9.2 Extensions to the Quality Indicator Models and Framework

The work presented in this dissertation represents the first serious effort at characterizing the educational quality of online resources with computational methods. As such, this system can only cover a small part of the vast range of methods from computational semantics, natural language processing, information retrieval, education, etc., which may be applicable to this problem.

9.2.1 Indicator-Specific Modifications

A natural way to improve model performance on the indicators would have been to begin by extracting the relevant pieces of information for each indicator. For example, to determine if a resource has a “prestigious sponsor”, the first step would seem to be to identify who the “sponsor” is, that is, extract the phrase in the text that refers to the entity that published or sponsored the content. Similarly, if a resource identifies its target age range, most likely there will be a phrase to that effect (e.g. “for high school students” or “students age 13 – 16”). Using simple heuristics and possibly some manual annotation, such target phrases
could, conceivably, be extracted. However, I decided not to pursue that direction in my approach, because such heuristics would almost necessarily be hand-tuned to each indicator — they would not generalize to the others.

The idea that the methods I explored should apply equally to all the quality indicators identified here, as well as to many other indicators that will be chosen in the future, was a major tenet in my approach. In all the experiments presented here, the same feature set and machine learning configuration was used on all seven indicators. I considered this an important aspect of my approach, since the set of indicators used here is by no means exhaustive; in particular, when implemented in a live system, new indicators might emerge as particularly relevant to that set of users. Extending a framework for characterizing educational quality by including such new indicators should not require new custom-tuned models (which would necessarily incur great extra effort to implement). As long as each quality indicator model is based on a common, generalizable approach, a small set of newly annotated data is enough to create an initial model for a new indicator.

### 9.2.2 Identifying Quality-Relevant Passages

As discussed in the previous section, I don’t consider hand-tuned heuristics for identifying relevant bits of text within a resource a particularly good approach for making quality assessments, as it is too dependent on renewed manual effort for each indicator. On the other hand, given a model that detects a quality indicator within a resource with some accuracy, an interesting side benefit might be if we could leverage that same model to identify the sections (e.g. sentences or paragraphs) within the resource that most strongly support the model’s judgment.

For example, not only would a model assert that a resource “has instructions”, it would also point out the paragraph of the text that most strongly represents the instructions. The computational quality indicator models are not perfect and will always make some mistakes; providing a user with this additional information gives them the opportunity to quickly validate or reject the model’s output. If the model is correct, the user will be more confident in the assessment because they are provided with supporting evidence; if the model is incorrect, this will be apparent from the fact that the provided snippet is not relevant or does not substantiate the model’s claim.
A simple extension to the current models may be sufficient to provide this additional functionality. By far the most important feature (as discussed in Chapter 6) is the bag-of-words feature, that is, the presence or absence of specific words within the resource. When assessing a resource along a given indicator, the bag-of-words feature is computed from the complete content of the resource. With only a slight modification, the same feature could be computed from subsets of the resource, and the model applied to those as well. Using a simple heuristic for breaking a resource into paragraphs or groups of related sentences (and one such heuristic is already implemented in the current system), each such segment could be individually tested by the model. The segment that scores highest on the given indicator will, presumably, provide the strongest evidence for the indicator’s presence. This fairly direct approach is currently being implemented and evaluated by my colleagues.

9.3 Corpus Annotation

In creating and working with the DLESE and IA corpora, some inherent limitations in the data became apparent which caused problems during model training and evaluation. In both corpora, annotations on most indicators were strongly biased toward one side. It may be the case that these skewed distributions of ratings accurately reflect the human experts’ assessments of resources’ quality, maybe most of them truly seemed exceptionally good — or maybe this is an issue of calibrating annotators’ use of the available scale. But whether the skew reflects the true distribution of the data or is an artifact of the annotation process, more evenly distributed ratings are preferable for building powerful and generalizable computational models.

A more thorough training process for the annotators to go through might lead them to better use the full breadth of the available rating scale. In the IA corpus, in particular (but also to some extent in the DLESE corpus) we saw that annotators are by no means interchangeable — each annotator may have their own strengths and weaknesses in assessing certain indicators, and each annotator may have their own way of mapping their assessments to the available rating scale (e.g. in the IA corpus, annotator 2 preferred a bi-modal rating scheme — strongly “yes” or strongly “no” — whereas the other annotators’ ratings were more normally distributed). More carefully planned annotator training may reduce such effects as well as avoid heavily skewed ratings. Alternative modes of annotation may be worth exploring as well; for example,
instead of asking annotators to give a rating (binary or on a Likert scale) to each resource, one might ask them to rank two resources, picking out the one that better represents an indicator.

The large differences in agreement numbers between annotators on the DLESE corpus and on the IA corpus brings up another question to explore: To what extent is the disagreement between annotators due to lack of proper training and preparation (i.e. the annotators simply have a different understanding of what each indicator should measure), and to what extent does it reflect genuine ambiguities in the data in respect to the indicators we used. It is quite apparent that complete agreement between annotators, on assessments as subjective as the quality indicators, will not be possible; after all, how explicit and extensive do instructions to students have to be in order to “count” as instructions? Which organizations can truly be considered as “prestigious” in their field of study? Human annotators will always differ in where exactly they draw the line on questions like these. This is not a problem.

But one of the goals in identifying the low level quality indicators for annotation was to minimize the “avoidable” causes for disagreement. By telling the annotators explicitly what to look for in each indicator, and by rooting these descriptions in the educational purpose of the resource, we wanted to leverage their “common sense” understanding of the indicators while encouraging them to stick to an objective measure in rating each resource. The DLESE corpus gave promising results: agreement was good on all indicators but “has prestigious sponsor”. This success did not translate as well to the IA corpus, however. There, agreement on all indicators was noticeably lower. It would be worth exploring to what extent the disagreement could be avoided through better annotator training (time and resources for training were much more limited on that corpus), and to what extent the increased disagreement is simply due to more ambiguity in the data.

Finally, it is worth pointing out again that the DLESE and IA data sets represent very different kinds of data. In order to evaluate the generalizability of the computational models better, a data set that is more closely related to either of them would be a useful asset. For example, this could consist of a collection of educational digital resources annotated for quality indicators, analogous to the DLESE corpus, but from a different field of science; such a data set would be much closer to the DLESE corpus while still providing variation for assessing model generalizability.
9.4 Future Directions

Work in personalized learning and with educational digital resources is moving more and more towards open, collaborative environments. Projects such as Instructional Architect, in which the teachers play a central, active role, are becoming more the norm than purely centrally managed and curated digital libraries. Increasingly, users are not passive consumers of content anymore; they interact with existing content in rich ways, enhancing it through their own experiences (for example by providing reviews and sharing content among their colleagues), and creating their own content along the way.

More than finding a place in existing review structures within digital libraries, computational models of quality can become an active part of these new systems. As users first approach such interactive portals, automatic quality assessments of the materials can help them focus their search and identify the materials that are right for them; as they compose their own content, automatic quality assessments can help guide them towards producing materials that are of high quality and can be useful to others as well; and as they become proficient with the portal and interact with the materials, the users can provide feedback (either direct feedback through a questionnaire, or indirect feedback through their actions, such as marking resources as favorites, sharing them with others, etc.), which in turn serves to improve the computational models by providing further data for training.

For this vision to become reality, on the computational side the models have to become more robust to different types of materials; ideally, one model should work equally well on digital library resources and on the type of data found in IA. With more diverse data sets future research should lead to such models, leveraging richer computational representations of resources than the models presented here. In addition to that, the way in which typical un-trained users of educational portals reason about quality needs to be explored in more detail — and in particular, how inexperienced users can be guided towards making use of the richer, more nuanced quality characterizations used by experts, and which the quality indicator models aim to capture.


Appendix A

Annotation Guidelines for Digital Resource Quality — DLESE Corpus

(The following is an exact reproduction of the written annotation guide given to the annotators of the DLESE corpus.)

The goal of this annotation project is to provide a corpus from which machines can learn to judge the quality of digital resources. In particular, we are interested in the kinds of quality judgments that managers of digital collections make when deciding whether or not to allow a resource into their collection. Two components are needed for machines to learn such quality judgments: a set of resources, and annotations on those resources that show how humans have judged their quality. The resources have already been selected. The purpose of this project is to collect the human judgments.

A.1 Annotating Quality Indicators

Annotators will tag each resource (not each page) for seven different indicators of quality. These indicators were selected by interviewing experts in digital resource selection, and identifying which indicators were most predictive of a final decision to include a resource in their collection. Annotators will examine the resource pages for these indicators, and tag the resource for each quality indicator that is present.

The seven indicators of quality to be tagged are:

- Has Instructions
- Has Publisher or Sponsor
- Has Prestigious Publisher or Sponsor
• Identifies Learning Goals

• Resource is Organized Appropriately for Learning Goals

• Identifies Age Range

• Content Not Obviously Inappropriate for Age Range

For each of these indicators, annotators should examine the resource, and tag it with either Yes, if the indicator is present in the resource, No if the indicator is not present in the resource or Not Applicable if the indicator is not applicable to the resource (e.g. Resource is Organized Appropriately for Learning Goals when no learning goals were identified.)

Annotators should visit as many pages within the resource as they feel necessary to evaluate the resource for the presence or absence of all the quality indicators. These pages may be of any type, e.g. html or PDF. The selection of a quality indicator value (Yes, No or N/A) should be made as soon as a page is viewed (or enough pages have been viewed) that makes the presence or absence of that quality indicator clear. Each page viewed by the annotators will be recorded, as will each page at which the value of a quality indicator was selected. To help distinguish between pages viewed that were part of the resource and pages viewed that were from other resources (e.g. a link that left the resource), annotators will indicate for each page whether or not they considered it part of the resource.

More thorough descriptions the quality indicators are given in the following sections.

A.1.1 Has Instructions

A resource that has instructions explains how the content of the resource should be approached by the user. For example, a resource may indicate the sequence in which the user should visit its pages, describe the steps in a classroom activity, or explain how to install and use a piece of software. Simply identifying the contents or purpose of a resource is not considered equivalent to having instructions – there must be some guide that helps understand how to navigate and use the resource.
A.1.2 **Has Sponsor**

A resource that has a sponsor explicitly attributes its content to a person or institution. For example, a page may state that it is organized or maintained by a professor, a public school, a research lab (e.g. NOAA or NASA), or a politician/celebrity (e.g. Al Gore). Wikipedia-like sites where the content is managed by the community are not considered to have sponsors. Being part of a particular domain is also not sufficient for having a sponsor – the content must be explicitly attributed to someone.

A.1.3 **Has Prestigious Sponsor**

A resource has a prestigious sponsor if the manager or organizer of the page is respected in the field of study relevant to the resource. For earth systems resources, prestigious sponsors could include entities like NASA, USGS or NOAA, but also respected universities (when the resource is maintained by university itself – not just a student’s webpage).

A.1.4 **Identifies Learning Goals**

A resource that identifies learning goals articulates the knowledge and skills a student is expected to acquire over the course of using the resource. Specific state or national standards may be given, or the resource may more informally identify its best use in an educational setting.

A.1.5 **Resource is Organized Appropriately for Learning Goals**

A resource is organized appropriately for its learning goals if the site is clearly organized so that each goal has a corresponding description or activity. This often means that each learning goal is addressed under a separate heading, tab or page. A page with a single learning goal may be appropriately organized if the headings, etc. give useful sub-structure to the learning activities. If no learning goals were identified by the resource, this indicator should be given the value Not Applicable (N/A).
A.1.6 Identifyes Age Range

A resource identifies its age range if the text states the expected age or grade level of its intended users, or if the resource is divided into sections targeted to users with different levels of knowledge.

A.1.7 Content Not Obviously Inappropriate for Age Range

A resource’s content is not obviously inappropriate for its age range if someone with little expertise in education would judge that the reading or activities were neither much too hard nor much too easy for the given grade level. For example, playing with clay is generally inappropriate for high school audiences, and very technical terminology is inappropriate for elementary school students. In both these cases, the resource should be annotated as No. If no age range was specified by the resource, this indicator should be given the value Not Applicable (N/A).
Appendix B

Annotation Guidelines for Project Quality — IA Corpus

(The following is an exact reproduction of the written annotation guide given to the annotators of the Instructional Architect corpus.)

B.1 Instructions

Please review the IA project listed and provide your judgment as to how it meets each of the indicators below. Remember, these are your judgments, and there is not necessarily a “right” or “wrong” answer. Two of the indicators (“Resource is Organized Appropriately for Learning Goals”, and “Content is Appropriate for Age Range”) have an “N/A” option in the event that learning goals or age range are absent (marked strongly disagree).

B.2 Indicator Definitions

B.2.1 Has Instructions

The IA project includes instructions for the user indicating how to navigate and use the project and linked resources. Example: project indicates a sequence in which the user should visit pages or describes steps in a classroom activity.

Scale:

(1) Strongly disagree

(2) Disagree
(3) Neither agree nor disagree

(4) Agree

(5) Strongly agree

**B.2.2 Links to Prestigious Sources**

The IA project provides a linked resource(s) to a ‘prestigious’ source. A ‘prestigious’ source includes a site where the manager or organizer is highly respected in the relevant subject area, government organizations, respected digital libraries, university maintained sites, and non-profit organizations.

This indicator will require you to click on the resource links. In visiting resources, follow any instructions from the IA project itself (e.g. try to go to the location that students are supposed to go). Once you arrive, do not click on any additional links to make your judgement.

Scale:

(1) Strongly disagree

(2) Disagree

(3) Neither agree nor disagree

(4) Agree

(5) Strongly agree

**B.2.3 Identifies Learning Goals**

The IA project identifies learning goals and articulates the knowledge and skills a student is expected to acquire over the course of using the project. Specific state or national standards may be given, or the resource may more informally identify its best use in an educational setting.

Remember to use the ”more information” link at the bottom of the project as another way to identify learning goals and/or age range.
B.2.4 Resource is Organized Appropriately for Learning Goals

The IA project is organized appropriately for its learning goals. The goals are clearly organized so that each goal has a corresponding description or activity.

If you strongly disagreed that no learning goals were identified by the resource (previous indicator), this indicator will automatically fill with the value Not Applicable (N/A).

Scale:

(N/A) no learning goals articulated.

(1) Strongly disagree

(2) Disagree

(3) Neither agree nor disagree

(4) Agree

(5) Strongly agree

B.2.5 Identifies Age Range

The IA Project identifies its target student age range by stating the expected age or grade level of its intended users in the project itself or in the “more information” link at the bottom of the project.
Alternatively, the project structure could, using sections with different levels of knowledge, imply a certain age range.

Scale:

(1) Strongly disagree

(2) Disagree

(3) Neither agree nor disagree

(4) Agree

(5) Strongly agree

B.2.6 Content Seems Appropriate for Age Range

The IA project’s content seems appropriate for its age range; someone with little expertise in education would judge that the reading or activities were neither too difficult or too easy for the given grade level. For example, playing with crayons is generally inappropriate for high school audiences, and very technical terminology is inappropriate for elementary school students.

If you strongly disagreed that the age range was specified in the previous indicator, this indicator will automatically fill with the value Not Applicable (N/A).

Scale:

(N/A) no age range articulated.

(1) Strongly disagree

(2) Disagree

(3) Neither agree nor disagree

(4) Agree

(5) Strongly agree