Natural Language Processing Tools for Reading Level Assessment and Text Simplification for Bilingual Education

Sarah E. Petersen

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

University of Washington

2007

Program Authorized to Offer Degree: Computer Science & Engineering

University of Washington
Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Sarah E. Petersen

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Chair of the Supervisory Committee:

Mari Ostendorf

Reading Committee:

Mari Ostendorf

Oren Etzioni

Henry Kautz

Date:

In presenting this dissertation in partial fulfillment of the requirements for the doctoral degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this dissertation is allowable only for scholarly purposes, consistent with “fair use” as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to Proquest Information and Learning, 300 North Zeeb Road, Ann Arbor, MI 48106-1346, 1-800-521-0600, to whom the author has granted “the right to reproduce and sell (a) copies of the manuscript in microform and/or (b) printed copies of the manuscript made from microform.”

Signature____________________

Date____________________

University of Washington

Abstract

Natural Language Processing Tools for
Reading Level Assessment and Text Simplification
for Bilingual Education

Sarah E. Petersen

Chair of the Supervisory Committee:
Professor Mari Ostendorf
Electrical Engineering

Reading proficiency is a fundamental component of language competency. However, finding topical texts at an appropriate level for foreign and second language learners is a challenge for teachers. We address this problem using natural language processing technology to assess reading level and simplify text. In the context of foreign- and second-language learning, existing measures of reading level are not well-suited to this task. Related work has shown the benefit of using statistical language processing techniques; we extend these ideas and include other potential features to measure readability. In the first part of this dissertation we combine features from statistical language models, traditional reading level measures, and other language processing tools to produce a better method of detecting reading level. We discuss the performance of human annotators and evaluate results for our detectors with respect to human ratings. A key contribution is that our detectors are trainable; with training and test data from the same domain, our detectors outperform more general reading level tools (Flesch-Kincaid and Lexile). Trainability will allow performance to be tuned to address the needs of particular groups or students.

Next, these tools are extended to enable teachers to more effectively take advantage of the large amounts of text available on the World Wide Web. The tools are augmented to

handle web pages returned by a search engine, including filtering steps to eliminate "junk" pages with little or no text. These detectors are manually evaluated by elementary school teachers, the intended audience. We also explore adapting the detectors to the opinions of individual teachers.

In the second part of the dissertation we address the task of text simplification in the context of language learning. We begin by analyzing pairs of original and manually simplified news articles to learn what people most often do when adapting text. Based on this analysis, we investigate two steps in simplification: choosing sentences to keep and splitting sentences. We study existing summarization and syntactic simplification tools applied to these steps and discuss other data-driven methods which in the future could be tuned to particular corpora or users.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I am extremely grateful to my advisor Mari Ostendorf; I could never have done this without her help. Mari is an exceptional advisor and mentor, always knowing when to push me and when to reassure me that I was on the right track. I would also like to thank my other committee members, Oren Etzioni, Henry Kautz, Rajesh Rao, Andreas Stolcke, and Harry Bruce, for their advice and diversity of viewpoints and insights into my work.

I would like to acknowledge several institutions and individuals who provided me with data. Thank you to the Weekly Reader Corporation and Literacyworks for access to their articles to build the two primary corpora used in this work. Regina Barzilay and Noémie Elhadad provided the Britannica corpus. I also owe thanks to several people for generously allowing me to use their code. In SSLI lab, thank you to Costas Boulis for his feature selection program, Jeremy Kahn for parse tree feature extraction and general Perl help, and Xiao Li for SVM adaptation code. Elsewhere, thank you to Advaith Siddharthan at Cambridge for the use of his syntactic simplifier and Daniel Marcu at ISI for his summarizer.

Many thanks to the teachers and principal at the John Stanford School for very useful discussions that helped frame the scope and direction of this work. Thank you to the teachers and graduate students who labeled data for evaluation. Thanks also to Briana Blaser at the UW Center for Workforce Development for her cheerful and prompt labeling, aligning, and other annotation work.

Thank you to past and present students, administrators and computer support in the Signal, Speech and Language Interpretation lab. In particular, thank you to Becky Bates for welcoming me to the lab, showing me how to be a researcher, and being a friend who listened to me and provided support in the tough times as well as the good ones. Thanks to everyone else for making the lab the community that it is, and best wishes to all of you.

Thank you to students and administrators in my home department, Computer Science and Engineering. Lindsay Michimoto and Frankye Jones, present and past graduate advisors, always made sure that navigating department and university policies was as easy as possible. There are too many students to name individually, but I will start by thanking Tammy VanDeGrift for her friendship and for six years of daily lunch companionship. Thank you to the architecture group for ample opportunities for good beer and better conversation, and thanks to Andrew Putnam in particular for introducing us to hockey and starting the “Canadianization” process.

Lastly, thank you to my family for all the support and encouragement throughout the last eight years. I am very grateful to my parents for believing in me and listening to me talk about work that was sometimes comprehensible to them, and for moving to Seattle so that I could do so over dinner. Thanks in particular to Mom for being a pilot tester on some of the teacher evaluation components of my project. Thank you to my not-so-little sister, who got out of school before me, with at least as many degrees, works harder than anyone I know, and is just as proud of me as I am of her. Also, many thanks to Melissa Meyer, a friend who is as good as family, who understands the mix of emotions that come with finishing grad school and leaving Seattle. Finally, thank you to my husband Andrew for boundless love and the right mix of laughter, advice, and encouragement, especially recently, as we have both plowed through the last year of our Ph.D.s.

This material is based upon work supported by the National Science Foundation (NSF) under Grant No. IIS-0326276. Any opinions, findings or recommendations expressed here are those of the author and do not necessarily reflect the views of the NSF.

I would also like to thank the IBM Faculty Award program and the ARCS Foundation for additional support earlier in my graduate school career.

Chapter 1
INTRODUCTION

The U.S. educational system is faced with the challenging task of educating growing numbers of students for whom English is a second language [96]. In the 2001-2002 school year, Washington state had 72,215 students (7.2% of all students) in state programs for Limited English Proficient (LEP) students [10]. In 2003-2004, 3.8 million students, or 11% of all students in U.S. public schools received English language learner (ELL) services, including 26% of all public school students in California and 16% of all students in Texas [95]. Washington's LEP program has a three year "maximum stay," after which students are expected to be ready for English-only classrooms. However, in 2001-2002, 28% of LEP students had been in the program for more than three years [10].

Reading is a critical part of language and educational development, but finding appropriate reading material for LEP students is often difficult. To meet the needs of students who read below their age-appropriate grade level, bilingual education instructors seek out "high interest level" texts at low reading levels, e.g., texts at a first or second grade level that support the fifth grade science curriculum. Graded textbooks and other materials are usually available, but these do not always meet the high interest/low reading level criterion. Additionally, students often need to do supplemental reading outside of graded textbooks for class projects. Teachers also need to find material at a variety of levels, since students need different texts to read independently vs. with help from the teacher. Finding reading materials that fulfill these requirements is difficult and time-consuming, and teachers are often forced to rewrite or "adapt" texts themselves to suit the varied needs of their students.

In this dissertation we address this problem by developing automated tools to help teachers and students find reading-level appropriate texts matched to a particular topic to help provide these students with more comprehensible reading material. Natural language processing (NLP) technology is an ideal resource for automating the task of selecting appropriate reading material for bilingual students. Information retrieval systems successfully find topical materials and even answer complex queries in text databases and on the World Wide Web. However, an effective automated way to assess the reading level of the retrieved text is still needed.

In addition to students in bilingual education, these tools will also be useful for those with reading-related learning disabilities and adult literacy students. In both of these situations, as in the bilingual education case, the student's reading level does not match their intellectual level and interests.

In this chapter we present background information about bilingual education to further motivate this work and discuss related work in the area of NLP technology for language learners. Then we describe the goals and contributions of this dissertation and provide an outline of the remaining chapters.

1.1 Bilingual Education

"Bilingual education" is an umbrella term used to describe several different approaches to teaching students who speak more than one language. The goal of English as a Second Language (ESL) programs, which are the most common, is to prepare immigrant students for English-speaking classrooms as quickly as possible. Some schools also have bilingual programs known as "dual language immersion" in which students use two languages throughout the school day. In both cases, the students are learning a "second language," a language used as a medium for learning other subjects and interacting outside of the language classroom, in contrast to a "foreign language," which is taught as a subject unto itself [1]. Studies show that a language is learned better when it is used purposefully, i.e., learned in the context of other subject material [75].

For ESL students, although the focus is on learning English, instruction also continues in subjects such as history and science, which often requires adaptation of texts to present content from the student's age-appropriate grade level in an easier-to-read form. Most ESL classrooms include students at a variety of levels of English competency, and students need different levels of texts to read individually and with assistance, requiring teachers to find or create a large variety of texts.

Teachers in dual language immersion programs have a related challenge, since students learn some subjects in each language, and students are generally only native speakers of one of the two languages. For example, the John Stanford International School in Seattle uses the same curriculum (specified in English) as all Seattle public schools, but certain subjects are taught in Spanish and require texts and other materials in Spanish. In this case, most students are native English speakers, but it is not a requirement of dual-language immersion that all students share the same native language. In this dissertation, we focus on developing tools for English texts, but the approaches and techniques used are general and could be applied to other languages.

In both the dual language and ESL situations, automated tools for finding and simplifying texts from the web (or any database of reading material) would be beneficial for teachers and students. In this dissertation we focus on the following scenarios:

- Given a topic and a specific reading level, find appropriate texts on the web.

- Given an intermediate-level text, simplify it in terms of grammatical construction and length.

These tools are initially envisioned as tools for teachers, since the technology is not perfect but can still be used to speed up teachers' work. Other tools have been developed with similar intended usage, e.g., Max's interactive system [66, 67] which integrates simplification suggestions into a word processing program which could be used by teachers or other authors, and ETS's Automated Text Adaptation Tool [8, 9] which provides adaptations such as marginal notes and synonyms and is intended for use by teachers to aid lesson planning or for direct use by students.

1.2 NLP Tools for Language Learners

There are currently few NLP tools to assist language learners and teachers. The existing systems focus largely on building vocabulary and require large amounts of human intervention. Horst *et al.* observe that while reading is an effective method for vocabulary acquisition, repeated exposure to new words is required, and it is necessary for most (95%) of the text to be familiar for a reader to learn new words from context [44]. Building on these findings, Ghadirian's TextLadder system organizes teacher-selected articles for optimal presentation of vocabulary [38]. While useful, the teacher is required to find all the stories manually; the tool only sorts them.

Fletcher has developed a concordancing tool for use in foreign language teaching [36]. He often uses the web in his own classes to find examples, check questionable usage of words, and find new words that have not yet made it into dictionaries. His tool KWiCFinder finds examples of "keywords in context" (KWiC) on the web and automatically produces summary documents. It was designed primarily as a filter to speed up the task of finding examples of usage of particular words.

Researchers at Carnegie Mellon University have developed REAP, an intelligent tutoring system designed to select reading material from the web for students in second-language classes based on unigram models of grade level, curriculum, and the reading level of individual students [6, 41] with the addition of hand-crafted grammar rules in the most recent version of the system [42]. REAP is targeted at adult ESL students reading at approximately the sixth to eighth grade level. The system builds a pre-processed database of articles which contain words from the 570-word Academic Word List [30], although the particular choice of word list is not a key component of the system. For a particular student, REAP chooses articles based on finding an appropriate number of new words for that student, not based on the topic of the article. Thus, the REAP approach focuses on vocabulary acquisition for individual students, which is facilitated by the use of unigram models.

In the context of our work, we believe it is also useful to look at short phrases and/or the complexity of syntactic structure in a text. While vocabulary acquisition is an important part of language learning, students also need to learn content material. In particular, for the ESL task in our target scenario, the vocabulary (i.e., topic) and grade level are not necessarily well-matched and teachers want materials on a particular topic with some more difficult, topic-specific words but simple structure. Our work differs from previous work in that we focus on more structural features of reading level in order to allow users to specify topic separately from level.

In addition to the NLP-based tools described above, there are many existing reading level classification tools which we will discuss in Chapter 3. These tools are also often focused on vocabulary, and are not well-suited to our target task since they are static, which can be problematic for different populations and the normal temporal dynamics of language.

1.3 Goals and Contributions

This work has two main objectives: to apply existing natural language processing technology to problems faced by teachers and students in the context of bilingual education, and to extend the state of the art in the relevant NLP areas. As part of this effort, we will develop tools to assist teachers in finding appropriate-level, topical texts for their students by developing new methods of reading level assessment and text simplification, making the following contributions.

Reading level detection. We develop reading level detectors for a corpus of clean text articles and then extend these detectors to apply to pages retrieved from the web using a standard search engine.

- **Development of trainable reading level detectors for clean text**
  The detectors are support vector machine (SVM) classifiers, with features that include traditional grade level features (e.g., average number of words per sentence), n-gram language model scores, and parser-based features. Experiments are conducted with texts from Weekly Reader, an educational newspaper with versions targeted at different grade levels [99]. Weekly Reader articles cover topics such as current events,

science, and history. On a held-out test set from this corpus, detectors trained for grade levels 2-5 outperform the traditional Flesch-Kincaid and Lexile reading level measures, with F-measures in the range of 0.5-0.7 (depending on grade level) for data within the 2-5 range. Additional news text at adult reading levels is rejected in 90% of the 30 cases that we tested.

- **Extension of reading level detectors for web pages**
  Moving from a static collection of good quality text to the dynamic (but mixed quality) text resources found on the web produces additional challenges. We find that the web pages returned include a large number of pages that the detectors trained on clean text simply are not designed to handle. We augmented the tools to first distinguish web pages with narrative text from those that mainly have links, advertisements, or other unwanted content, reducing the amount of unwanted pages by approximately 50% relative.

- **Investigation of algorithms for adapting detectors for individual users**
  Reading level assessment is a subjective and variable problem. Different annotators have different perceptions of the appropriateness of articles for a particular grade level, in part due to variability between individual students the may have worked with. With the goal of developing detectors that can be adapted in the style of active learning or relevance feedback, we study the applicability of existing SVM adaption techniques to this task, using annotations from each teacher to adapt the reading level detectors to better meet the needs of each individual user. We were unable to obtain improvements using existing SVM adaptation techniques, but these were developed on non-text-based tasks, and further research on alternative classifiers may lead to a gain.

Automatic text simplification. We analyze a parallel corpus of original and simplified articles, apply syntactic simplification techniques and explore other approaches to simplification based on this corpus.

- **Analysis of features of original and simplified texts**

  Using a parallel corpus of original news articles and simplified versions from a literacy website [100], we conduct a study of the characteristics of these texts to inform our simplification work. The sentences are hand-aligned with sentence "split points" indicated. The resulting corpus is an additional contribution to the field of NLP research, and our analysis yields insight into the type of changes made by people when adapting texts for students.

- **Exploration of components of a text simplification system for language learners**

  We investigate possible approaches for a text simplification system for the context of language learning and suggest a decomposition of the simplification problem into component tasks. We focus our analysis on two of these, sentence selection and sentence splitting, and evaluate the performance of existing summarization and syntactic simplification tools applied in this context.

1.4 Dissertation Outline

This dissertation is divided into two main parts related to work on reading level assessment and text simplification. As a preface, Chapter 2 provides background on NLP and machine learning technologies common to both parts, including n-gram language models, text classification, and support vector machines.

Part I of the dissertation, on reading level assessment, comprises Chapters 3 through 6. Chapter 3 provides background on related approaches to reading level assessment, and Chapter 4 describes the corpora used in this work. Our approach to reading level detection for "clean" data, such as magazine articles, is presented in Chapter 5, which includes a discussion of human annotator performance on this task and evaluation of our detectors with respect to both human ratings and test set labels. In Chapter 6 we extend these detectors to handle web pages by incorporating filtering steps to remove “junk” pages with little or no text, such as pages which consist primarily of links. We evaluate the performance of these tools on real web pages with the help of human annotators, observing that different people have different opinions, and we conduct additional experiments on adaptation of the detectors to individual users. Part I of the dissertation is based on work published in [79, 71, 70].

Text simplification for language learners is the subject of Part II of the dissertation, which consists of Chapters 7 through 9. Chapter 7 covers background and related work in the areas of text simplification, summarization and sentence compression. Chapter 8 describes the corpus used in our text simplification work, including an analysis of the different features of original news texts and simplified versions intended for language learners. Chapters 7 and 8 are based on [72]. In Chapter 9, we discuss options for an automatic simplification system, including experiments on the applicability of existing syntactic simplification and summarization tools.

Chapter 10 summarizes the dissertation and discusses future research directions based on this work.

Chapter 2
BACKGROUND

This chapter provides background information on NLP and machine learning technologies that will be used in the proposed work. Section 2.1 introduces n-gram language modeling, used in many NLP applications including text classification. We discuss selected text classification methods in section 2.2, since reading level detection can be thought of as a type of text classification problem. In Section 2.3 we describe feature selection, an important component of many text classification systems. Section 2.4 contains an overview of support vector machines (SVMs), a machine learning technique used in many types of classification problems. Statistical language models and SVMs are topics which are well-documented in the literature. Hence, in this section we provide only a brief overview of these techniques.

2.1 Language Modeling

Statistical language models are used in a variety of NLP applications including speech recognition, machine translation, information retrieval and text classification. In our work, we explore the use of LMs for reading level assessment as a classifier in and of themselves, and as features for an SVM classifier.

Statistical LMs determine the probability of a word sequence w by considering the probability of each word given its history

$$P(w) = P(w_1)\prod_{i=2}^{m} P(w_i|w_{i-1}, ..., w_1). \tag{2.1}$$

Considering the full history for each word is infeasible in practice, so truncated histories are used. This results in the most commonly used statistical language model, the n-gram model, which assumes that the word sequence is a Markov process. For example, for $n = 3$ (trigram) the probability of each word is conditioned on the two previous words and the probability of sequence w is approximated by

$$P(w) \approx P(w_1)P(w_2|w_1)\prod_{i=3}^{m} P(w_i|w_{i-1}, w_{i-2}). \tag{2.2}$$

Despite its simplicity, the trigram model is very successful in speech recognition and other applications. N-gram models of other orders (2-5) are also common. The parameters of the model are typically estimated using a simple maximum likelihood (ML) estimate based on the observed frequency in a training corpus

$$P(w_i|w_{i-1}, w_{i-2}) = \frac{n(w_{i-2}w_{i-1}w_i)}{n(w_{i-2}w_{i-1})}, \tag{2.3}$$

where $n(.)$ represents the number of occurrences of the particular n-gram in the training data. Sentences are typically treated independently, with special beginning and end of sentence tokens added.

Smoothing Due to the large number of parameters to estimate in these models, smoothing is essential. Even if a large training corpus is available, some word combinations will not be present in the training data leading to undesirable zero probabilities for the corresponding ML estimates. To compensate for this problem, the parameter estimates must be smoothed, typically using either back-off or interpolative methods. Details about common smoothing techniques can be found in [21], and standard language modeling toolkits that address this problem are available. In all language models described in this thesis, we use the SRI toolkit [90] and its implementation of modified Kneser-Ney smoothing [21], a state-of-the-art smoothing method.

N-gram order N-gram order is dependent on the task and the quantity of training data. Unigrams ($n = 1$) are used in tasks such as topic modeling, while bigrams ($n = 2$) and higher-order n-grams are used in speech recognition, The availability of larger amounts of training data, e.g., from web pages, has enabled the use of $4-$gram and $5-$gram models in recent years. Unigram models have been used in previous work on reading level assessment [25] due to the small size of the available training data set. In addition, unigrams are often used for topic-oriented text classification. In this work, we use feature selection, described in Section 2.3, and smoothing to compensate for the limited amount of data while still allowing the use of more powerful bigram and trigram models which capture syntactic as well as vocabulary information.

2.2 Text Classification

Our task of text selection based on reading level can be viewed as a type of text classification. In general, automatic text classification is the task of assigning semantic labels to segments of natural language text (e.g., sentences or documents). Many types of labels have been studied in the NLP community, including topic, genre, and author; examples will be discussed in this section.

While some text classification methods incorporate handwritten rules, machine learning is often used to automatically learn classification rules from training data. The representation of a document is also an important consideration. Bag-of-words is a common representation, consisting of a vector of counts of word occurrences in a document, which works well for many tasks but it may not be appropriate for all tasks since it does not maintain information about the sequence of words and overall structure of the document.

Several threads of previous work in text classification are relevant to this thesis. Genre classification has recently received increased interest and is similar to our task in that the goal is to classify the style or type of text independent of topic. Early work on genre detection by Kessler identified structural cues from tagged/parsed text, lexical cues such as dates and terms of address (e.g., Mrs.), character-level cues such as punctuation and capitalization, and derivative cues (e.g., ratios of the above features) [50]. Stamatatos *et al.*'s discriminant analysis classifier uses the frequency of the most common words in the text (function words) as the most important feature [88]. They note that punctuation is also helpful. Lee and Myaeng use an extension of an information retrieval statistic, term frequency-inverse document frequency (TF-IDF), to select genre-revealing terms [59]. The goal is to find words that appear often in a particular genre and not others and are evenly distributed across topics within that genre.

Farahat *et al.'s* work on the AuGEAS system for ranking authoritativeness of web pages is very similar to genre detection [34]. Most prior work on authoritativeness of web pages focuses on the hypertext link structure, which confers "social authority" (i.e., pages to which many other pages have links can be considered more authoritative). Their work focuses on textual indicators of authoritativeness and uses categories that can be viewed as genres of web pages: scientific documents, news, commercial pages, home pages, etc. The features they consider are also similar to features often used in genre and reading level assessment: occurrence of particular characters (e.g., punctuation), numerals, words, word classes, HTML features and abbreviations; part of speech tag features and grammatical patterns; scores from traditional reading level indices; and textual characteristics such as document length and average sentence length. They select specific features using stepwise regression and build authoritativeness classifiers using both linear regression and boosting. The results of their classifier are complementary to other link-based authoritativeness measures and can be used to re-rank search results to find "high-value" texts.

The word-based features used by the systems described previously are based on the bag-of-words representation. The remainder of this section highlights text classification systems that use n-gram models to consider sequences of words as features. Cavnar and Trenkle's technique for topic categorization and language identification compares character n-gram profiles of classes and documents [15]. This simple language ID technique is extremely accurate; an implementation by van Noord has been trained for almost seventy languages and is publicly available on the web [97]. Peng *et al.* use word and character n-grams to build a "chain-augmented naive Bayes classifier" for topic and authorship detection [69]. Their classifier consists of an n-gram language model for each category. Here, smoothing takes the place of other forms of feature selection commonly used in classification algorithms.

Yamron *et al.* use LMs in their topic tracking system for news text [102]. Their classifier compares the scores of a unigram LM for each topic and a background LM built from general text. Damashek and Huffman's system for document categorization represents documents as vectors of character n-grams [32, 46]. Although intended as an information retrieval system for the Text Retrieval Conference (TREC) evaluations, their approach is more effective for assessing document similarity. An interesting feature of the system described by Damashek is the idea of clustering document vectors by topic and then "subtracting" the cluster centroid to cancel out topic-related factors and allow analysis based on other features of the documents [32].

2.3 Feature Selection

Feature selection is a common part of classifier design for many classification problems; however, there are mixed results in the literature on feature selection for text classification tasks. In work by Collins-Thompson and Callan on readability assessment, LM smoothing techniques are more effective than other forms of explicit feature selection [26]. However, feature selection proves to be important in other text classification work, including genre detection work by Lee and Myaeng [59], which is similar to the reading level detection task. For the models used in this dissertation, we use both feature selection and smoothing. This section describes the feature selection method that we chose, which is based on information gain.

Words are sorted based on their information gain (IG) [104] using software developed by Boulis [5]. Information gain measures the difference in entropy when w is and is not included as a feature:

$$\begin{aligned} IG(w) = & - \sum_{c \in C} P(c) \log P(c) \\ & + P(w) \sum_{c \in C} P(c|w) \log P(c|w) + P(\bar{w}) \sum_{c \in C} P(c|\bar{w}) \log P(c|\bar{w}), \end{aligned} \tag{2.4}$$

and it corresponds to the mutual information between the class and the binary indicator random variable for word w. The most discriminative words are selected as features by plotting the sorted IG values and keeping only those words above the "knee" in the curve, as determined by manual inspection of the graph. In early development experiments we replaced all remaining words with a single "unknown" tag. This did not result in an effective classifier, so in later experiments the remaining words were replaced by their POS tag, as labeled by a maximum entropy tagger [74]. The use of POS tags was motivated by our goal of representing syntax; the tags allow the model to represent patterns in the text at a higher level than that of individual words, using sequences of POS tags to capture rough syntactic information.

2.4 Support Vector Machines

Support vector machines were introduced by Vapnik [98] and were promoted in the area of text classification by Joachims [48]. We chose SVMs for this work because of their popularity and success for a variety of other classification problems.

SVMs are a machine learning technique based on the principle of structural risk minimization. Starting with a set of labeled training examples, a kernel function is used to, in effect, map the points into a higher-dimensional feature space. Then the goal is to fit a hyperplane between the positive and negative examples in the high dimensional feature space so as to maximize the distance between the data points and the plane; this distance is called the margin. The examples closest to the separating hyperplane are called support vectors and they form two maximum-margin hyperplanes on either side of the separating hyperplane. However, in many situations, no hyperplane exists that correctly separates all the examples, so typically the “soft margin” method is used to chose a hyperplane that best separates as much of the data as possible.

To design an SVM classifier, it is necessary to choose a kernel, tune parameters, and possibly tune a threshold for the output. Hsu *et al.* recommends the radial basis function (RBF) kernel for classification problems with small numbers of features [45]. Our task falls into this category, and in initial experiments using development data, we confirmed that this kernel was a good choice for our task. We tuned parameters using cross validation and grid search, also as described in [45]. The RBF kernel has one parameter to tune, and there is a cost parameter for all soft-margin SVMs, regardless of kernel choice, which trades off error on the training set and the size of the margin.

In addition to these two parameters, in some cases it is necessary to tune a threshold for the classifier output. The training data is labeled with +1 and -1 for the two classes, so ideally, outputs greater than 0 indicate the positive class, and numbers less than 0 indicate the negative class. In practice, it is sometimes necessary to tune this threshold to be something other than 0. This is particularly useful when the classes are not well-separated, e.g., in a topic classification task with multiple possible labels per document [63]. Yang [103] describes several approaches to threshold tuning, including sorting values and choosing the top n, and tuning the threshold on a development set (or by cross-validation). Liu *et al.* [63] applied these methods to SVM classifiers for the task of classification in the Yahoo web hierarchy, which has thousands of categories. They found that tuning on a development set worked well for categories with relatively large priors. (For rare categories, it was better to compare outputs of all the classifiers, although this situation is not analogous to our task, in which all of the categories have fairly large, although not identical, prior probability.) Sun *et al.* [91] also tuned thresholds on development data for SVM classifiers for the WebKB data set, which consists of university web pages labeled with seven classes, e.g., department, student, or faculty. Our reading level detection task is somewhat similar to these other text classification tasks, and we also chose to tune thresholds on development data.

For our initial SVM experiments, we used the SVM^{light} toolkit developed by Joachims [49]. We also used the SVMTorch toolkit [28] in adaptation experiments to take advantage of an augmented version of the toolkit developed by Li [61].

PART I: READING LEVEL ASSESSMENT

Part I of this dissertation addresses reading level assessment, starting with a discussion of related approaches in Chapter 3 followed by a description of the corpora used in this work in Chapter 4. Chapter 5 presents our approach to reading level assessment for clean corpora, developing reading level detectors based on n-gram language models and SVM classifiers. Chapter 6 extends these detectors for use on web pages and in an adaptive setting.

Chapter 3
RELATED APPROACHES TO READING LEVEL ASSESSMENT

This chapter highlights examples and features of some commonly used measures of reading level and discusses current research on the topic of reading level assessment.

The process used by teachers to select appropriate reading material for their students is complicated and subjective, taking into account subject matter as well as characteristics of the text itself. For example, Fountas and Pinnell's well-known system of matching books to readers takes into account more than a dozen high-level characteristics, including vocabulary, grammatical structure of phrases and sentences, use of literary devices, illustrations, and layout on the page [37]. Automatic tools currently do not capture this entire range of characteristics, but a variety of methods and formulae have been developed to estimate reading level based on characteristics which are easily measured.

Many traditional formulae for reading level assessment focus on simple approximations of syntactic complexity such as sentence length. The widely used Flesch-Kincaid Grade Level index is based on the average number of syllables per word and the average sentence length in a passage of text [53] (as cited in [25]). Similarly, the Gunning-Fog index is based on the average number of words per sentence and the percentage of words with three or more syllables [39]. These methods are quick and easy to calculate but have drawbacks: sentence length is not an accurate measure of syntactic complexity, and syllable count does not necessarily indicate the difficulty of a word. Additionally, a student may be familiar with a few complex words (such as dinosaur names) but unable to understand complex syntactic constructions.

Other measures of readability focus on semantics, which is usually approximated by word frequency with respect to a reference list or corpus. The Dale-Chall formula uses a combination of average sentence length and percentage of words not on a list of 3000 "easy" words [16]. The Lexile framework combines measures of semantics, represented by word frequency counts, and syntax, represented by sentence length [89]. These measures are inadequate for our task; in many cases, teachers want materials with more difficult, topic-specific words but simple structure. Measures of reading level based on word lists do not capture this important information about structure. An additional drawback of these traditional approaches is that they use word lists that are updated manually. Our system is automatically trained, with the advantage that it can be customized for different levels or domains given only a small set of training documents.

In addition to the traditional reading level metrics, researchers at Carnegie Mellon University have applied probabilistic language modeling techniques to this task. Si and Callan conducted preliminary work to classify science web pages using unigram models [82]. More recently, Collins-Thompson and Callan manually collected a corpus of web pages ranked by grade level and observed that vocabulary words are not distributed evenly across grade levels. They developed a "smoothed unigram" classifier to better capture the variance in word usage across grade levels [26]. On web text, their classifier outperformed several other measures of semantic difficulty: the fraction of unknown words in the text, the number of distinct types per 100 token passage, the mean log frequency of the text relative to a large corpus, and the Flesch-Kincaid measure. The traditional measures performed better on some commercial corpora, but these corpora were calibrated using similar measures, so this is not a fair comparison. More importantly, the smoothed unigram measure worked better on the web corpus, especially on short passages. The smoothed unigram classifier is also more generalizable, since it can be trained on any collection of data. The smoothed unigram classifier is used in the REAP system, described in Chapter 1. Syntactic features based on a set of 22 "relevant grammatical constructions" which were hand-selected from ESL grammar textbooks are a recent addition to REAP [42].

Although the smoothed unigram classifier outperforms other vocabulary-based semantic measures, it does not capture syntactic information. In some contexts, it is also useful to look at short phrases and/or the complexity of syntactic structure in a text. In particular, this may be important for the ESL task, where the vocabulary (i.e., topic) and grade level are not necessarily well-matched and teachers want materials with more difficult, topic-specific words but simple structure. We believe that higher order n-gram models can achieve better performance by capturing both semantic and syntactic information. An automatic parser can provide additional syntactic features.

Chapter 4
READING LEVEL CORPORA

Our detectors are trained and tested on a corpus obtained from Weekly Reader, an educational newspaper with versions targeted at different grade levels [99]. These data consist of short articles on a variety of non-fiction topics, including science, history, and current events. Our corpus consists of articles from the second, third, fourth, and fifth grade editions of the newspaper because these grade levels were available in electronic form. This corpus contains just under 2400 articles, distributed as shown in Table 4.1. This table also includes the mean and standard deviation of the article lengths (in words), although article length was not used as a feature for our detectors. We want our detectors to perform well on a variety of texts, not just Weekly Reader articles, and article length is closely tied to this particular domain. In general, it is intuitive that lower grade levels often have shorter texts, but we would like to be able to classify short and long texts of all levels without assuming that short length is always an indicator of low reading level.

We divide the Weekly Reader corpus into separate training, development, and test sets,

Table 4.1: Distribution of articles and words in the Weekly Reader corpus.

Grade Level	**Number of Articles**	**Number of Words**	**Article Length (Words)**	
			Mean	**Std Dev**
2	351	71.5k	161.1	146.5
3	589	444k	151.4	174.6
4	766	927k	254.3	197.8
5	691	1M	314.4	264.4

Table 4.2: Number of articles in the Weekly Reader corpus as divided into training, development and evaluation test sets.

Grade	**Training**	**Dev**	**Eval**
2	315	18	18
3	529	30	30
4	690	38	38
5	623	34	34

as shown in Table 4.2. The development data is used as a test set for tuning parameters, and the results presented in Chapter 5 are based on the evaluation test set. The development and evaluation test sets are the same size, and each consist of approximately 5% of the data for each grade level.

Additionally, we have two smaller corpora consisting of articles for adults and corresponding simplified versions for children or other language learners. Barzilay and Elhadad have allowed us to use their corpus from Encyclopedia Britannica, which contains articles from the full version of the encyclopedia and corresponding articles from Britannica Elementary, a new version targeted at children [2]. The Western Pacific Literacy Network/Literacyworks web site has an archive of CNN news stories and abridged versions that we have also received permission to use [100]. Although these corpora do not provide an explicit grade-level ranking for each article, the adult and child/language-learner versions allow us to learn models that distinguish broad reading level categories. We can use these models to score articles from the Weekly Reader corpus or other sources to provide additional features for detection.

We use one other corpus in training, consisting of Associated Press newswire data from the TIPSTER corpus [40]. These are newspaper articles on a variety of topics; we selected this corpus as an example of text at an adult reading level in the same non-fiction/news domain as the Weekly Reader corpus. We use this corpus as “negative training data” to

Table 4.3: Distribution of articles and words in supplemental training corpora.

Corpus	**Num Articles**	**Num Words**
Britannica	115	277k
Britannica Elementary	115	74k
Literacyworks	111	51k
Literacyworks Abridged	111	37k
Washington Post	30	5.9k
KidsPost	30	3.5k
TIPSTER Newswire	979	420k

improve the accuracy of our detectors on text outside the Weekly Reader corpus. We will elaborate on this strategy in Chapter 5.

Finally, for tests related to the generalizability of the approach, i.e., using data outside the Weekly Reader corpus, we downloaded 30 randomly selected newspaper articles from the "KidsPost" edition of The Washington Post [92]. We do not know the specific grade level of each article, only that KidsPost is intended for grades 3-8. We also downloaded 30 randomly chosen articles from the standard edition of The Washington Post. Table 4.3 shows the sizes of the supplemental corpora.

Chapter 5
READING LEVEL DETECTION FOR CLEAN TEXT

In this chapter, we develop a method of reading level assessment that uses support vector machines (SVMs) to combine features from statistical language models (LMs) and parse trees, with several traditional features used in reading level assessment. Section 5.1 describes our basic approach and target scenarios for this application. Preliminary work described in sections 5.2 and 5.3 shows that SVM-based detectors incorporating features from LMs and other sources outperform LM-based detectors. Section 5.4 explores the generalization performance of the SVM detectors and describes improved detectors which include the use of negative training data from newswire text. Section 5.5 presents an analysis of the contribution of different types of features, and Section 5.7 compares the SVM detectors with regression-based SVMs and other methods of reading level assessment. Section 5.8 discusses experiments with human annotators to provide insights into the difficulty of the task and to present a different means of evaluating our detectors in comparison to traditional methods of reading level assessment.

5.1 Approach

There are two main ways in which reading level assessment tools could be used. In the first case, we imagine a teacher who is looking for texts at a particular level for an individual student or class. In the second case, we want to classify a group of articles into a variety of categories, perhaps for inclusion in a database. The first case corresponds to a binary detection problem, and the second involves either n-way classification or regression. In this work, we focus primarily on the first case, in which a typical scenario is a teacher or student searching the Web (or other large collection of documents) for articles on a certain topic at a particular grade level. We would like to be able to filter articles by level just as search engines currently filter by topic. However, we also include some experiments using regression to demonstrate the generality of the method and for more direct comparison to other techniques.

To address the detection scenario, we construct one detector per category which decides whether a document belongs in that category or not. We compare LM-based detectors with SVM-based detectors which incorporate a variety of additional features. To address the second scenario, we train a regression-based SVM model using SVM^{light} and round the predicted continuous value to the nearest integer grade level.

Existing reading level measures are inadequate for our intended task due to their reliance on vocabulary lists and/or a superficial representation of syntax. A trainable classifier is better than a non-trainable classifier when the training data is reasonably well matched to the test data. This is an obvious point that has not yet been put forward in this application. Our approach uses n-gram language models as a low-cost automatic approximation of both syntactic and semantic analysis, since syntactic dependencies are local in a large percentage of cases. Statistical LMs are used successfully in this way in other areas of language processing such as speech recognition and machine translation. We also use a standard statistical parser [19] to provide richer syntactic analysis. These and other features, described further in the next sections, are combined in a SVM framework to build grade level detectors.

5.2 N-gram LM and SVM Detectors

In this section, we compare LM-based detectors with SVM-based detectors using n-gram LMs and additional features.

5.2.1 Lanuage Model Detectors

Our first set of detectors consists of one n-gram language model per class c in the set of possible classes C. For each text document t, we can calculate the likelihood ratio between the probability given by the model for class c and the probabilities given by the other models for the other classes:

$$LR(t) = \frac{P(t|c)P(c)}{\sum_{c' \neq c} P(t|c')P(c')} \tag{5.1}$$

where we assume uniform prior probabilities $P(c)$. The resulting value can be compared to an empirically chosen threshold to determine if the document is in class c or not. For each class c, a language model is estimated from a corpus of training texts.

5.2.2 SVM Detectors

In addition to using the likelihood ratio for classification as described above, we can use scores from language models as features in another classifier (in this case, an SVM). Our SVM detectors for reading level use the following features:

- Traditional features:
  - Average sentence length
  - Average number of syllables per word[1]
  - Flesch-Kincaid score
- 6 out-of-vocabulary (OOV) rate scores
- Parse features (per sentence):
  - Average parse tree height
  - Average number of noun phrases
  - Average number of verb phrases
  - Average number of "SBAR"s.[2]

[1] The number of syllables per word came from a 85k word dictionary used in automatic speech recognition systems. Syllable counts for words not found in the dictionary were generated by the publicly-available Perl module Lingua::En::Hyphenate.

[2] SBAR is defined in the Penn Treebank tag set as a "clause introduced by a (possibly empty) subordinating conjunction." It is an indicator of sentence complexity.

- 12 language model perplexity scores

The OOV scores are relative to the most common 100, 200 and 500 words in the lowest grade level (grade 2) in the training data. For each article, we calculated the percentage of a) all word instances (tokens) and b) all unique words (types) not on these lists, resulting in three token OOV rate features and three type OOV rate features per article.

The parse features are generated using the Charniak parser [19] trained on the standard Wall Street Journal (WSJ) Treebank corpus. We chose to use this standard training set due to our interest in a scenario where text is selected from the web. In this scenario we anticipate that most texts will be more news-like than conversational and many of the retrieved texts will be at a reading level similar to that in a newspaper. While the lower level (children's) text may have different distributions of syntactic constructions than the newspaper text, we assume that the WSJ corpus at least covers the simple types of constructions typically observed in text written for primary grade levels. Inspection of some of the resulting parses in the Weekly Reader corpus showed satisfactory results. These particular parse tree features were selected because we expect the height of the tree and the distribution of phrases to indicate the overall complexity of the sentences in an article and help indicate its grade level.

Language model perplexity scores are used as indicators of semantic and syntactic similarity to each of several reference corpora. If we had unlimited training data, these reference corpora would consist of additional Weekly Reader articles for each grade level. However, our corpus is limited and preliminary experiments in which the training data was split for LM and SVM training were unsuccessful due to the small size of the resulting data sets. Thus we made use of the Britannica and Literacyworks corpora described in Chapter 4 to train trigram, bigram, and unigram models on each corpus of "child" text and "adult" text. This resulted in 12 LM perplexity features per article. These features were based on unigram, bigram and trigram models trained from each of the following individual corpora: Britannica (adult), Britannica Elementary, Literacyworks (adult) and Literacyworks abridged. Although these corpora do not map directly to Weekly Reader grade levels, they do represent broad differences in reading level and provide informative features for our detectors.

We used the Flesch-Kincaid score as a feature since this traditional measure of reading level can be calculated automatically. To our knowledge, there are no automatic tools available for the Lexile and Dale-Chall measures.

5.2.3 Feature Selection for N-gram Language Modeling

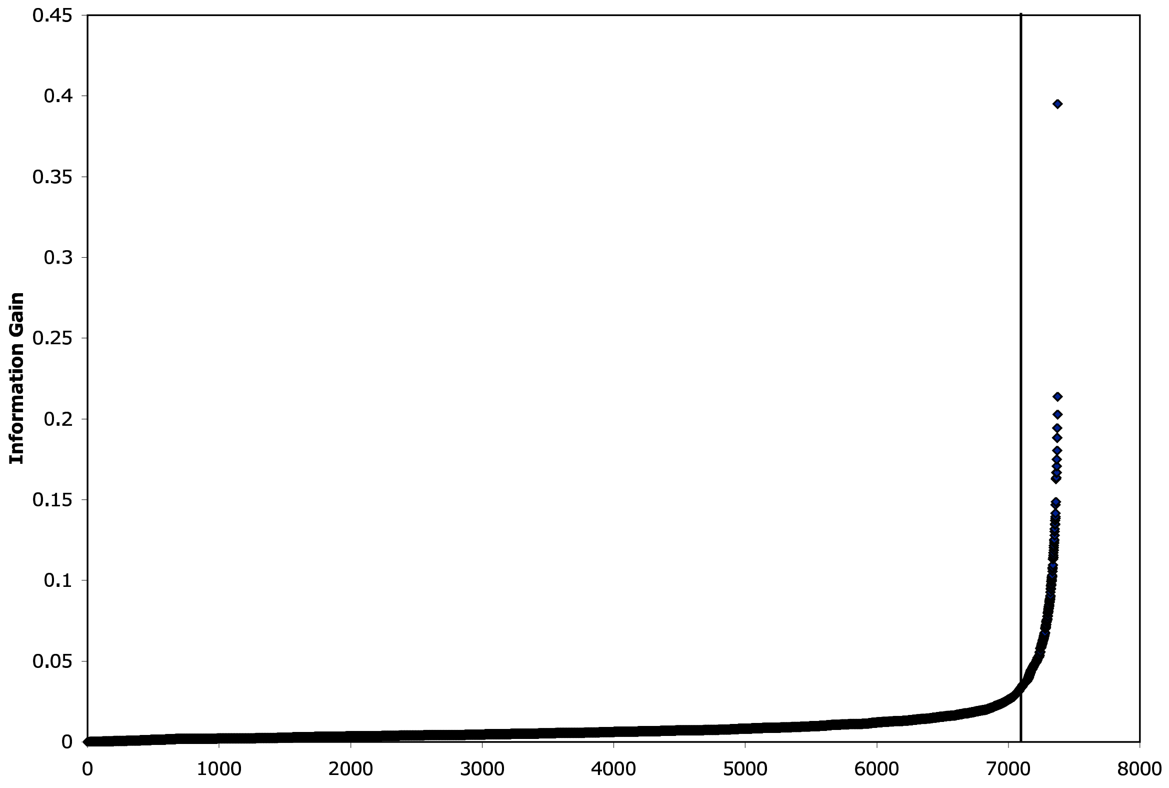

Figure 5.1: Information gain for words in the Weekly Reader corpus. Words above the knee in the curve (indicated by the vertical line) are used as is for language modeling, while others are replaced with POS tags.

For the language models in this chapter, we use feature selection based on information gain, as described in Section 2.3. Given $P(c|w)$, the probability of class c given word w, estimated empirically from the Weekly Reader training set, we sorted words based on their information gain to select which words will be used as is vs. replaced by generic part-of-speech (POS) tokens. These values are plotted in Figure 5.1; words with information gain above the knee in the curve (indicated by the vertical line on the plot) are retained. The resulting vocabulary consisted of 276 words and 56 POS tags. An n-gram language model is used to characterize the resulting mixed word-POS sequence, estimated using standard smoothing techniques. We use LMs trained in this fashion for both the LM-based detectors and the LM features of the SVM detectors.

5.3 Experiments with the Weekly Reader Corpus

5.3.1 Evaluation Criteria

Results are assessed using multiple criteria. For analyzing our binary detectors, we use Detection Error Tradeoff (DET) curves and precision/recall measures. Detection Error Tradeoff curves show the tradeoff between false negatives and false positives for different threshold values for the detectors. DET curves have been used in other detection tasks in language processing, e.g., [65]. We use these curves to visualize the tradeoff between the two types of errors, and select the minimum cost operating point in order to get a threshold for precision and recall calculations.

The minimum cost operating point on the DET curve depends on the relative costs of false negatives and false positives. It is conceivable that one type of error might be more serious than the other. However, teachers that we consulted with did not have a clear consensus as to which should be weighted higher, and in practice it may be useful to work at different operating points depending on the scenario. Hence, for the purpose of this analysis we weighted the two types of errors equally. In this work, the minimum cost operating point is selected by averaging the percentages of false negatives and false positives at each point and choosing the point with the lowest average.

Different operating points on the DET curve correspond to different tradeoffs of precision and recall, where precision indicates the percentage of detected documents that match the target grade level, and recall indicates the percentage of the total number of target documents in the data set that are retrieved. Precision and recall are intuitively meaningful measures for this application, which is similar to information retrieval. On this task, precision indicates the percentage of articles labeled by the detector as positive for a particular grade level that are also associated with that grade level in the Weekly Reader categorization, including those that are positively detected at multiple levels. Recall indicates what percentage of articles at a particular Weekly Reader grade level which are detected as positive for that grade level, regardless of other grade levels which might also have been chosen. Thus, positive detector results for multiple grades per article penalize precision but benefit recall. Due to the possibility of trading off one measure for gains in the other, the F-measure ($F = 2PR/(P + R)$) is often used to give a single system performance figure. Precision, recall and F-measures reported are associated with the minimum cost operating point.

5.3.2 Language Model Classifier

Figure 5.2 shows DET curves for the trigram LM-based classifiers. The minimum cost error rates for these classifiers, indicated by large dots in the plot, are in the range of 33-43%, with only one over 40%. The curves for bigram and unigram models have similar shapes, but the trigram models outperform the lower-order models. Error rates for the bigram models range from 37-45% and the unigram models have error rates in the 39-49% range, with all but one over 40%. Although our training corpus is small the feature selection described in Section 5.2.3 allows us to use these higher-order trigram models.

5.3.3 SVM Classifiers

Figure 5.3 shows DET curves for detector performance on the test set. The grade 2 and 5 detectors have the best performance, probably because in these cases the decision effectively involves only one boundary, whereas there are two boundaries (higher vs. lower) for the cases of grades 3 and 4. The minimum cost error rates for these classifiers range from 14-

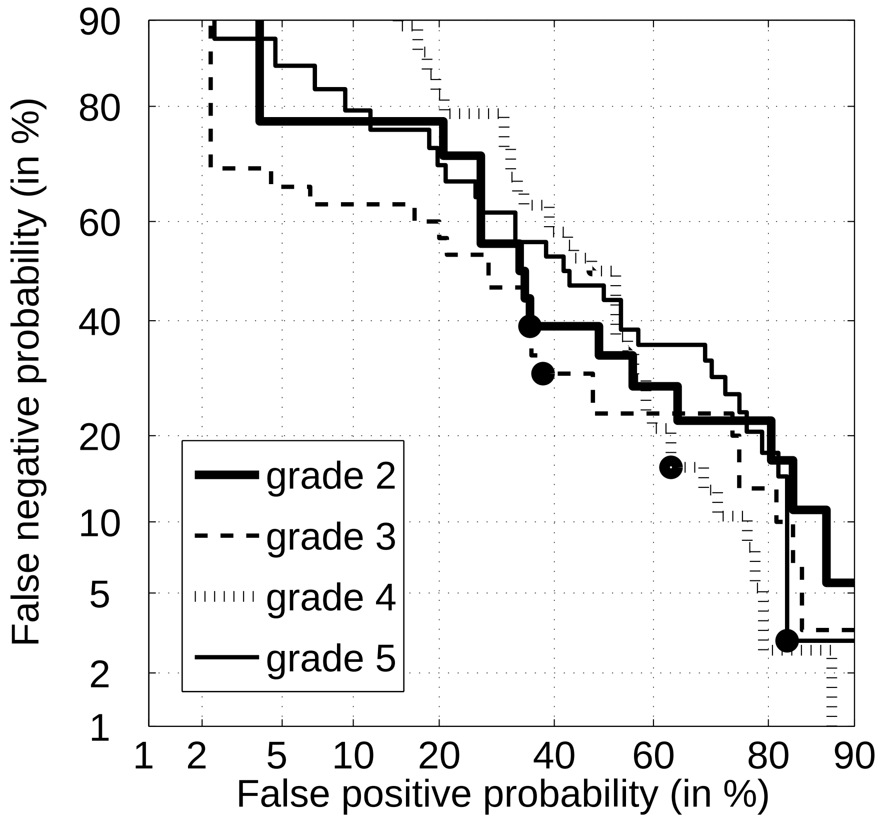

Figure 5.2: DET curves (test set) for classifiers based on trigram language models. The large dot on each curve indicates the minimum cost error point.

25%. Since these error rates are much lower than those for the LM-only detectors, we focus the remainder of our work on the SVM detectors. Using threshold values selected based on minimum cost on the development set, we calculated precision and recall on the test set, shown in Table 5.1. The grade 3 detector has high recall but relatively low precision; the grade 4 detector does better on precision and reasonably well on recall. Note that the minimum cost operating points do not correspond to the equal error rate (i.e., equal percentage of false negatives and false positives), so there is variation in the precision-recall tradeoff for the different grade level detectors.

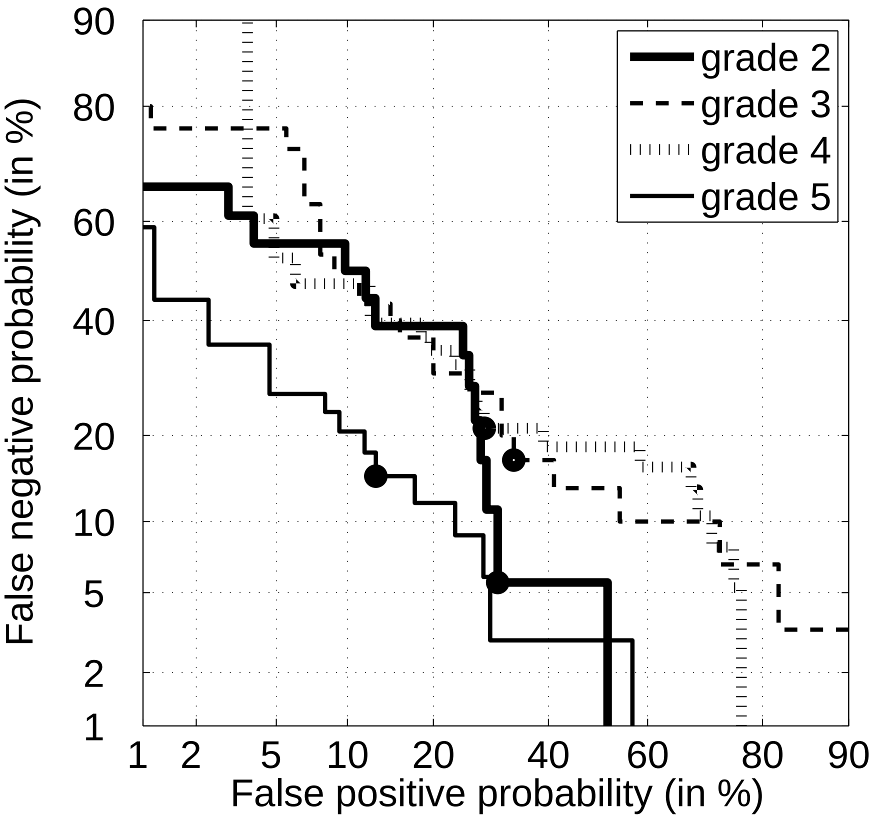

Figure 5.3: DET curves (test set) for SVM detectors. The large dot on each curve indicates the minimum cost error point.

5.4 Generalization Experiments and Improved SVM Detectors

To assess the performance of the system on new data, the detectors were applied to data downloaded from the KidsPost and standard editions of The Washington Post newspaper, as described in Chapter 4. In addition, since the detectors had only been trained on data reflecting reading levels 2-5, we trained new versions of the SVMs with TIPSTER newswire data included as additional negative training examples for each grade level. These negative training data were used to reduce the number of false positives for higher-level articles, particularly in the case of the grade 5 detector. It also leads to more realistic performance

Table 5.1: Precision, recall and F-measure on the test set for SVM-based detectors.

Grade	**Precision**	**Recall**	**F-measure**
2	38%	61%	47%
3	38%	87%	53%
4	70%	60%	65%
5	75%	79%	77%

Table 5.2: Number of KidsPost articles (out of 30) detected by each grade level detector for SVMs trained on the Weekly Reader (WR) data only vs. the WR data plus negative examples from TIPSTER newswire data. Articles that are not detected by any of the classifiers for grades 2-5 are counted under "Undetected".

Classifier Grade	**Classifier Training**	
	WR only	**WR + Newswire**
2	0	0
3	4	2
4	11	10
5	21	12
Undetected	0	12

for the grade 5 detector on the lower-level articles, since the grade 5 detector now has the potential to reject articles as being at a higher as well as a lower level.

Table 5.2 includes detection results for the KidsPost articles for both the original SVM detectors and the new version with augmented training data. No information about the target grade range of the articles was provided to the SVM detectors. Both grade 2 detectors correctly rejected all 30 articles. As we expected, the detector trained only with Weekly Reader data detects a much larger number of articles at grade 5. This model also fails to leave any article unclassified, which is unrealistic. Since the KidsPost data is targeted

Table 5.3: Precision, recall and F-measure on original Weekly Reader test set for SVM detectors trained on the Weekly Reader (WR) data (covering only grades 2-5) plus negative examples from TIPSTER newswire data.

Grade	**WR + Newswire**		
	Precision	**Recall**	**F-measure**
2	38%	78%	51%
3	56%	83%	67%
4	66%	55%	60%
5	74%	68%	71%

for the 3-8 grade range, one would expect that some of these articles would be above the grade 5 level. The detectors trained on Weekly Reader plus newswire data detect a more reasonable percentage of articles at grade 5 and leave 12 articles unclassified.

The benefit of the augmented training is particularly notable with the 30 articles from the standard edition of The Washington Post. All 30 of these articles were classified positively by the original grade 5 detector. The detector trained with newswire data as additional negative training data only positively classified 3 of these higher-level articles, leaving the remaining 27 articles undetected.

These results show that our detectors' performance generalizes to test data from a source outside the Weekly Reader corpus. We also observe that the addition of negative training examples from above the target grade level range is very important in reducing the number of false positives on test articles from above the target range.

Adding newswire data as additional training data does change the performance of the new detectors on the original Weekly Reader corpus, as shown in Table 5.3 and Figure 5.4. Recall improves for the grade 2 detector, as does precision for grade 3, resulting in higher F-measures in both cases. These differences are statistically significant at a 0.05 confidence level. There are small but insignificant losses in both precision and recall in other cases, leading to more balanced F-measures for the detectors for grades 3-5. While the higher

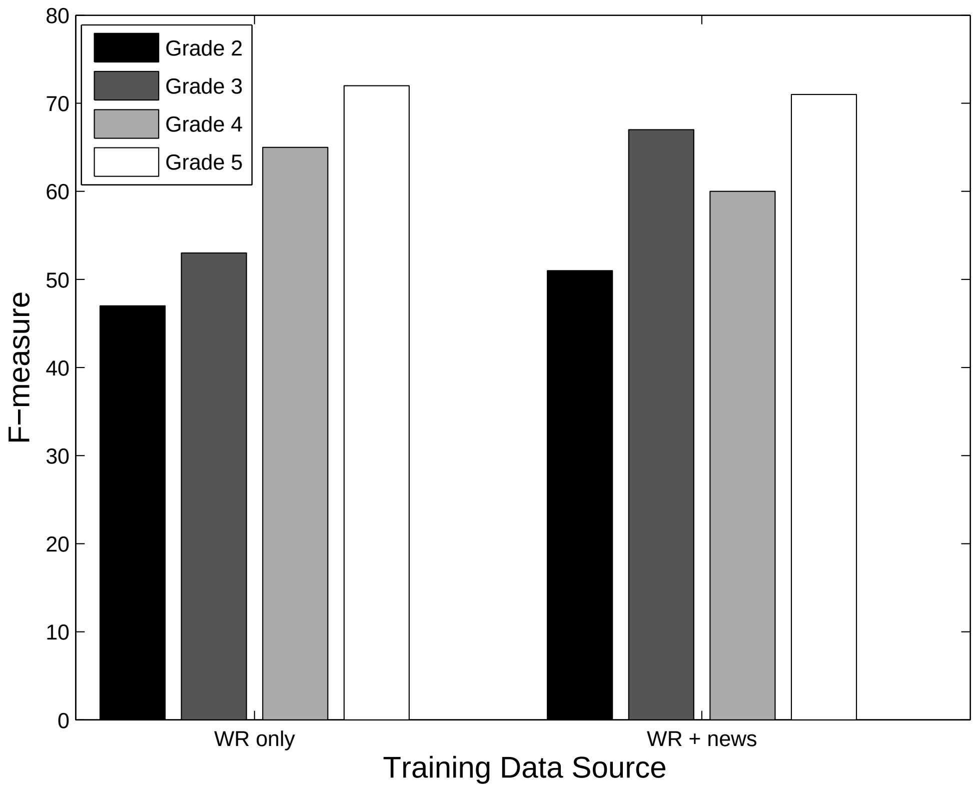

Figure 5.4: Comparison of F-measures on the original Weekly Reader test set for SVM detectors trained on the Weekly Reader (WR) data only, covering only grades 2-5, vs. the WR data plus negative examples from TIPSTER newswire data.

grades have slightly worse performance, these differences are not statistically significant, and we believe that the numbers are more representative of the real task, and the advantage in generalization performance on other data is of substantial real-world importance.

5.5 SVM Feature Analysis

We investigated the relative contributions of different feature types to the overall performance of the SVM classifiers, dividing the features into the following groups:

- lexical-only: OOV and average number of syllables per word
- syntactic: parse and n-gram scores (unigram, bigram, trigram)
- non-syntactic: all lexical-only features plus average sentence length in words and Flesch-Kincaid score

The syntactic features represent the new information used in our approach, and the non-syntactic features correspond to the traditional approach. The lexical-only features omit sentence length (and thus Flesch-Kincaid which incorporates sentence length), and they are included to assess the relative importance of vocabulary vs. structural cues. All n-gram features including unigram are considered syntactic, since the unigram LM includes POS tags, which can be considered a basic representation of syntax.

We trained new versions of the SVM grade level detectors with each of the above categories of features. Figure 5.5 shows F-measures for these classifiers compared to the classifiers using all features, trained on the Weekly Reader training set augmented with newswire data. The SVMs trained with lexical features perform comparably to the SVMs trained with all features for grades 2 and 3, while for the higher grades, the classifiers that use all types of features give better results. The SVMs trained with syntactic features alone do not perform as well as the other classifiers, but these features still appear to contribute to the overall performance of the SVMs trained with all features. Heilman *et al.* found similar results when grammar-based features were added to the REAP system's vocabulary-based (unigram) measure of reading level. In their experiments, the vocabulary features alone gave better results than the grammar features alone, while the combination of both feature sets gave better results than either alone [42].

To study the relative importance of the four parser features, we trained a decision tree classifier with the Weekly Reader training set using only the parser features to classify articles in grades 2 through 5.[3] Our goal was not to use this classifier explicitly to do grade

[3]We used the C4.5 decision tree package [73].

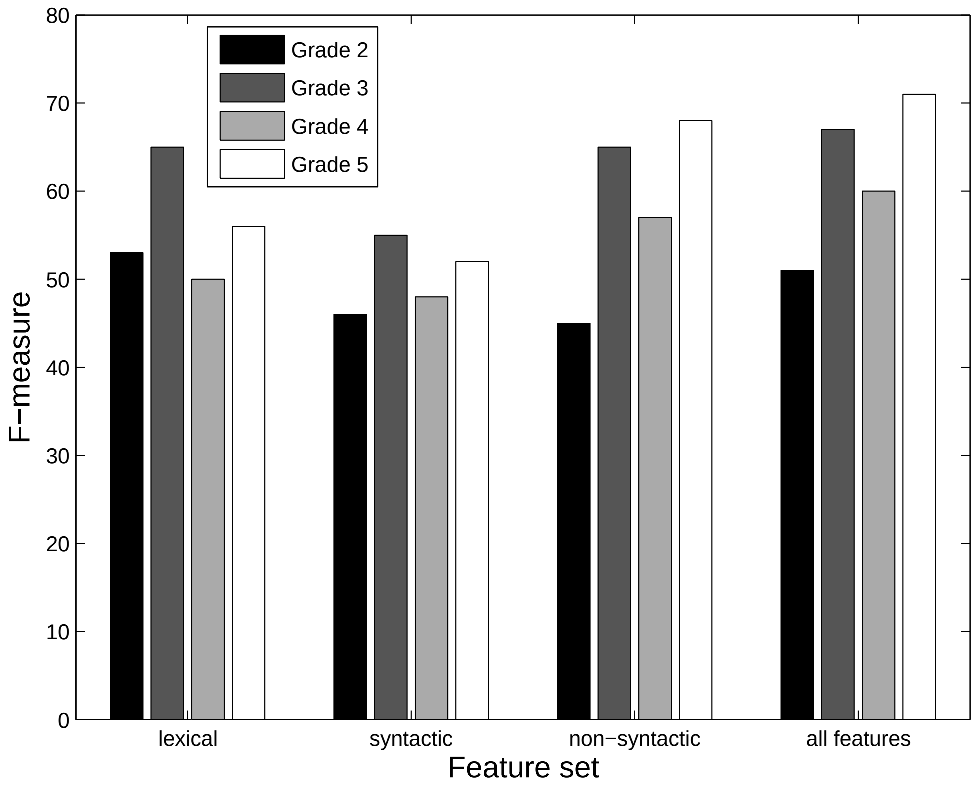

Figure 5.5: Comparison of F-measures for SVM detectors trained with lexical, syntactic and non-syntactic features on the Weekly Reader data, covering grades 2 through 5, plus negative examples from TIPSTER newswire data.

level classification, but to see how it made use of the four features. All four parse features were used by the decision tree. The features for the average number of noun phrases and verb phrases were used at higher nodes in the tree, while the features for the average number of SBARs and the average parse tree height were used for more fine-grained decisions at lower nodes of the tree. Additional trees trained in one vs. all fashion for each of the four grade levels showed that the height and SBAR features appeared higher in the trees that were identifying the higher grade levels, 4 and 5.

We also conducted an SVM detection experiment with a larger number of LM features using LMs based on three new vocabularies: POS tags only, the top 100 words according to information gain (plus POS tags for the remaining words), and the top 500 words plus POS tags for others. As with the previous LM features, we trained unigram, bigram, and trigram LMs for each of the four supplemental corpora for each of these vocabularies, resulting in 36 additional features for the SVMs. The resulting SVMs performed worse than the SVM detectors with the original 25 features, most likely due to overtraining since the Weekly Reader data set is small. It is possible that with a larger amount of training data, these additional LM features could be useful.

5.6 Alternative Classifier Configuration

One alternative to training a single detector for each grade level is to train a classifier for each boundary between grade levels and use the output of this set of classifiers as input to a metaclassifier for each grade level. This approach has the potential advantage that the basic classifiers have a simpler task, with a single boundary vs. multiple boundaries for the middle grades, but the disadvantage that some of the training data must be held out for metaclassifier design. We kept 10% of the original training set for metaclassifier training and trained the following SVM classifiers using the same features and parameter-tuning techniques as our previous SVM detection experiments:

- Grade 2 vs. grades 3-5 + newswire
- Grades 2-3 vs. grades 4-5 + newswire
- Grades 2-4 vs. grade 5 + newswire
- Grades 2-5 vs. newswire

The SVM scores are used directly as input to the metaclassifier. This has the advantage of not requiring additional held out data to tune a decision threshold for the SVMs, and potentially provides more information to the metaclassifier than a yes or no decision. We tested two types of metaclassifiers, a multilayer perceptron (MLP) trained using the Torch3 toolkit [27], and an SVM. Parameters for the MLP (number of hidden units, number of iterations, weight decay, and learning rate decay) were tuned using the development set. For the SVM metaclassifier, parameters were tuned using cross validation and the development set was used to find thresholds, as in previous experiments. The MLP metaclassifiers did not perform well, with F-measures between 42-49% depending on the grade level. This may be due to the very small amount of training data available, a problem which has been observed for MLPs in other contexts, e.g., [61]. The SVM metaclassifiers performed about 10% (absolute) worse than the original SVM detectors for grades 2 and 3, and about the same for grades 4 and 5.

Since the SVMs in this experiment have less training data than in the previous experiments, due to the need to hold out data for metaclassifier training, it is not too surprising that this approach did not provide a benefit. However, it is possible that the metaclassifier approach may be useful for adaptation, since only the metaclassifier would need to be adapted, and it could be simpler since the input has lower dimensionality.

5.7 Comparison with Other Methods

We compared the performance of both versions of our SVM detectors (with and without newswire data) with a regression-based SVM classifier, trained using SVM^{light}'s regression mode, using the same features and training data. For the regression classifier, we use "9" as the target grade level for the newswire data. There is no grade level explicitly assigned to this corpus, but most news articles are targeted for a high school reading level. In classification with the regression model, we round the predicted value to the nearest level.

Figure 5.6 shows F-measures for the SVM detectors trained on Weekly Reader data only and Weekly Reader plus newswire data, and SVM regression classifiers trained on both datasets. When trained on the Weekly Reader data, the SVM regression classifier performs comparably to the SVM detectors for grades 2 and 3. Grade 4 performance is

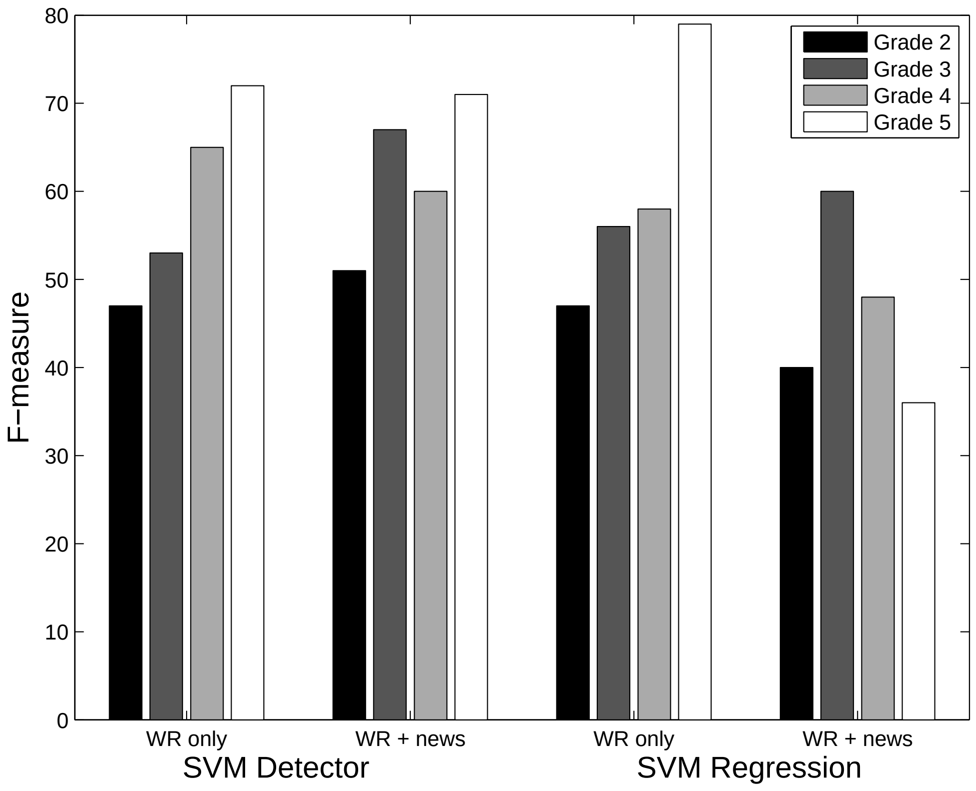

Figure 5.6: Comparison of SVM detectors and SVM regression classifier.

better for the detectors, while the regression classifier performs better at grade 5. However, the regression classifier trained on Weekly Reader plus newswire data has worse performance than the detectors at all grade levels. Regression may be better suited to a matched training and test condition (as in the Weekly Reader-only data), while the SVM detectors allow the use of higher (and potentially lower) level training data without accurate categorization of that data.

We also compared the regression-based SVM with two traditional reading level measures, Flesch-Kincaid and Lexile. The Flesch-Kincaid Grade Level index is a commonly used measure of reading level based on the average number of syllables per word and average sentence length. The Flesch-Kincaid score for a document is intended to directly correspond with its grade level. This score is a continuous number, which we rounded to get the predicted level for this comparison. We chose the Lexile measure as an example of a reading level classifier based on word lists.[4] Lexile scores do not correlate directly to numeric grade levels. However, a mapping of ranges of Lexile scores to their corresponding grade levels is available on the Lexile web site [60]. We used the regression classifier for this comparison since, like Flesch-Kincaid and Lexile, it provides a single classifier for all levels. However, results for the SVM detectors are also included in the figures for comparison purposes. Detection is a better fit to the target problem since there are gray areas and not strict divisions between grade levels.

Since these numbers correspond to classifiers (vs. detectors), performance can be evaluated in terms of accuracy or F-measure. The accuracy of the Flesch-Kincaid index is only 5%, while Lexile's accuracy is 36% and the SVM detectors achieve 43%. Figure 5.7 shows F-measures for the Flesch-Kincaid and Lexile measures compared to the regression classifier and SVM detectors trained on Weekly Reader plus newswire data. Flesch-Kincaid performs poorly, as expected since its only features are sentence length and average syllable count. Although this index is commonly used, perhaps due to its simplicity, it is not accurate enough for our intended application. The SVM regression classifier also outperforms the Lexile metric. Lexile is a more general measure while our regression classifier is trained on this particular domain, so the better performance of our model is not entirely surprising. Importantly, however, our classifier is easily tuned to any corpus of interest.

[4]Other classifiers such as Dale-Chall do not have automatic software available, while Lexile has a semi-automated web-based tool.

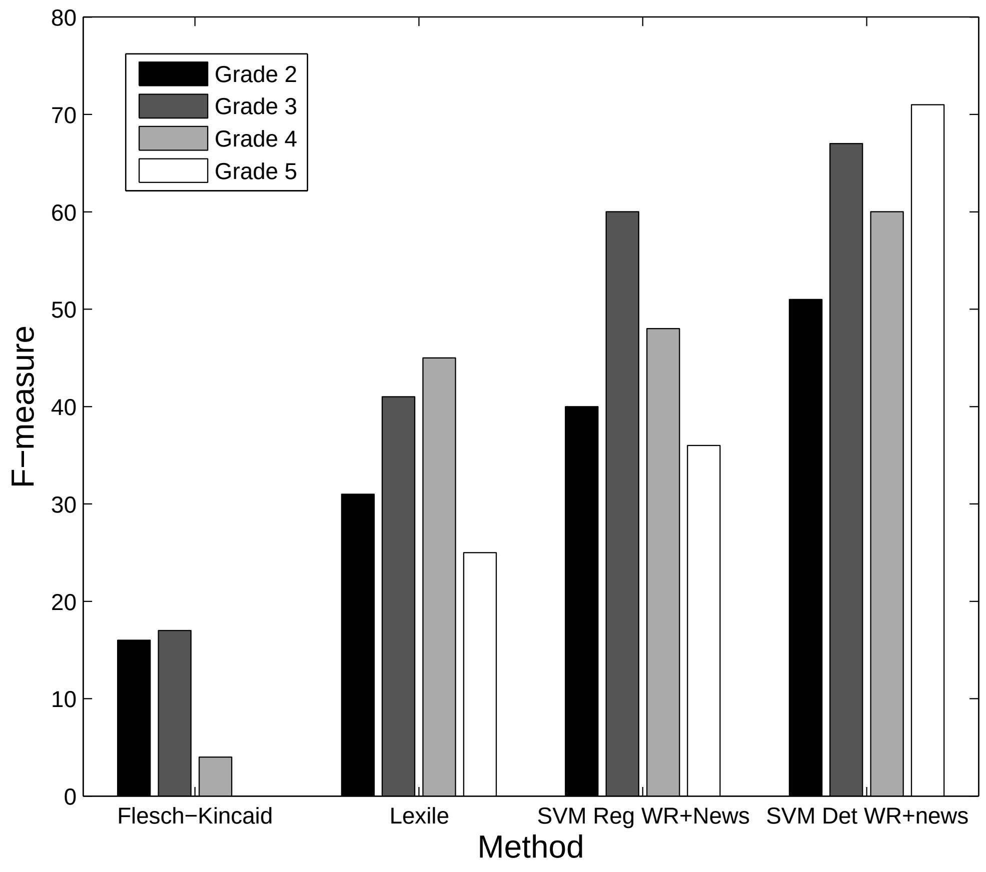

Figure 5.7: Comparison of Flesch-Kincaid, Lexile, SVM regression classifier and SVM detectors trained on Weekly Reader plus news data.

Figure 5.8 shows results for the same set of classifiers as Figure 5.7, evaluated only on the 73 articles in the test set on which three human annotators did not disagree by more than one grade level. (See Section 5.8 for details of these annotations.) On this cleaner test set, the results are a bit higher for all the classifiers, but the trends remain the same.

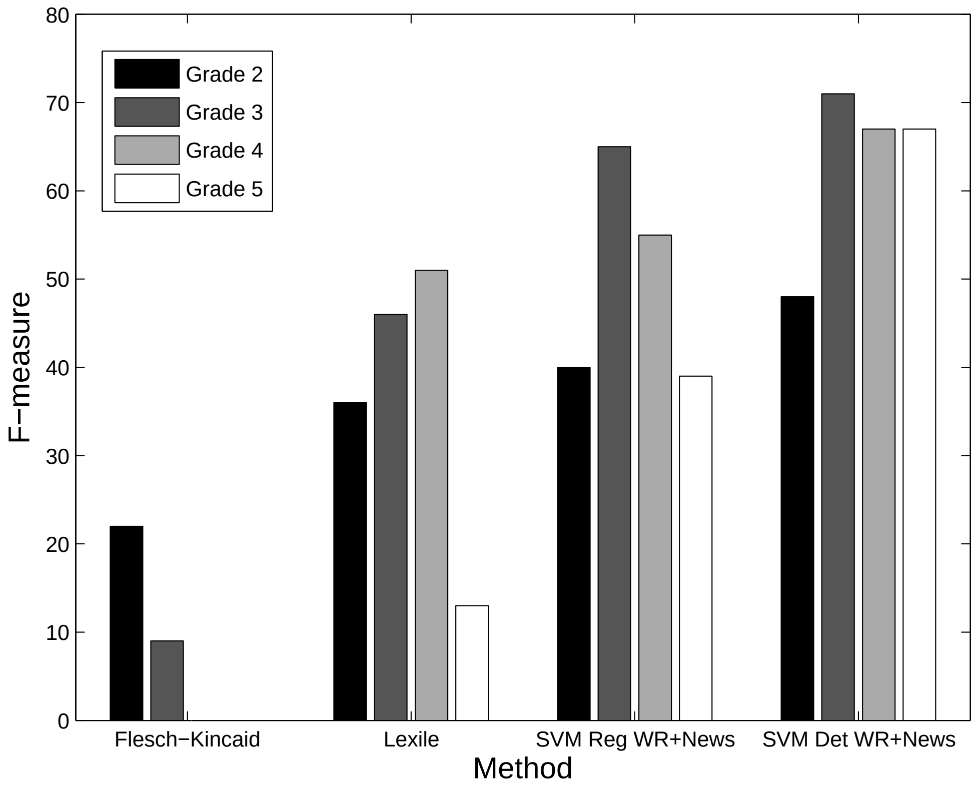

Figure 5.8: Comparison of Flesch-Kincaid, Lexile, SVM regression classifier and SVM detectors trained on Weekly Reader plus news data on subset of 73 test set articles on which 3 annotators do not disagree by more than one grade level.

Since there is overlap between grade levels, misclassifications of a single adjacent grade level are more minor than other misclassifications. Table 5.4 shows the percentage of articles which are misclassified by more than one grade level by Flesch-Kincaid, Lexile, and the SVM

Table 5.4: Percentage of articles which are misclassified by more than one grade level by traditional and SVM classifiers.

Grade	**Percentage of errors greater than one level**			
	Traditional		**SVM**	
	Flesch-Kincaid	**Lexile**	**WR**	**WR + Newswire**
2	78%	33%	6%	0%
3	67%	27%	3%	3%
4	74%	26%	13%	13%
5	59%	24%	21%	9%

detectors. Each SVM detector is independent of the detectors for the other grades, so it is possible for more than one detector to return "true" for a given article. In this case, we use a pessimistic measure and count errors even if another detector gave the correct answer. For example, if the grade 2 and grade 4 detectors both returned "true" for a grade 4 article, it counts as one error in this analysis. Consistent with the results in Figure 5.8, Flesch-Kincaid performs poorly, and the SVM detectors also outperform Lexile. The SVM detector trained on Weekly Reader data plus negative examples from TIPSTER newswire data performs the same as the original SVM detector for the middle grades, but does better on grades 2 and 5. This actually indicates that the negative training data improves the detector at all levels, since fewer misclassifications of grade 2 articles result from fewer mistakes by the grade 4 and 5 detectors, and similarly fewer grade 5 misclassifications result from better performance for the grade 2 and 3 detectors.

For further comparison, in the following section, we present results for the SVM detectors and Lexile evaluated on human annotations of the test set.

5.8 Assessment with Multiple Annotations

One of the challenges in the area of reading level assessment is knowing the right answer. In the experiments described above, we take the grade level assigned to each article in the corpus by the writers and editors of Weekly Reader as the "gold standard." However, we were interested to see how difficult this kind of annotation task is for human experts, how well human annotators agreed with each other and with the labels given in the corpus, and how well our detectors perform when compared to human evaluations of this corpus. In our informal discussions with teachers, we have learned that experienced teachers feel that they are able to easily identify whether or not a text is the appropriate grade level for their students.

To investigate this issue, we conducted a study of the performance of human annotators on the Weekly Reader data. We hired three experts to annotate our test corpus. The first annotator (referred to as A in the following discussion) was an elementary school bilingual education teacher. The other two annotators were graduate students in fields relevant to reading and pedagogy: one student was earning a Master's in Teaching in elementary education (referred to as B), and the other was an English student with a particular interest in pedagogy (referred to as C). We provided the annotators with a few example articles of each grade level chosen randomly from the training data. Then we asked them to read each article in the test set (unlabeled and in random order) and mark which grade level(s) they thought were appropriate. In a small number of cases, the annotators did mark more than one grade level for a single article. We included all of these annotations in our analysis, since this is comparable to the way our SVM detectors work. Since our detectors are independent, a single article can have hits from more than one detector; likewise, an article can be classified as more than one grade level by a human annotator.

One way to evaluate human annotation performance is to use Cohen's kappa statistic to see whether or not the annotators are consistent with each other. Cohen's kappa is a measure of inter-rater reliability, used to characterize the consistency of subjective annotations by human labelers in a variety of domains, including natural language processing tasks [12].

Table 5.5: Matrix of kappa values for pairwise inter-rater agreement for annotators A, B, and C, Weekly Reader test set truth values (WR), and the SVM detectors (SVM). Kappa values indicating moderate agreement are shown in bold.

	B	**C**	**WR**	**SVM**
A	**0.41**	**0.40**	0.20	0.25
B	-	**0.54**	0.36	0.27
C	-	-	0.28	0.36
WR	-	-	-	**0.52**

Cohen's kappa is calculated by comparing pairs of annotations from two labelers. For this task, the annotators could choose one or more labels per article so in our kappa calculations we consider, for each article, whether or not it is annotated positively for each grade level. This allows us to treat each annotator as a set of four "virtual detectors," which also allows for a more fair comparison with the SVM detectors. We are also interested in how well the annotators agree with the test set labels and the SVM detectors. Cohen's kappa is a pairwise measure, so we computed kappa values for all pairs of the following: the three human annotators, the Weekly Reader test set labels and the SVM detector results (using the SVMs trained on Weekly Reader plus newswire data.) Table 5.5 shows these kappa values. Kappa values observed for all pairs of human annotators and for the test set labels compared to the SVM detector results are between 0.4 and 0.6, indicating moderate agreement. Good agreement results in a kappa above 0.6, which did not occur in this experiment. Kappas for annotators compared to the SVMs and the test set labels show poor agreement. Thus we note that human experts agree reasonably well with each other, but not necessarily with the test set labels, while the SVM detectors agree better with the test set than with the annotators. It is not surprising that the SVM detectors agree best with the test set labels since the detectors were trained for the Weekly Reader corpus, but it is interesting that our annotators seem to use somewhat different criteria than Weekly Reader. These results indicate the usefulness of a classifier that can be easily trained for specific tasks.

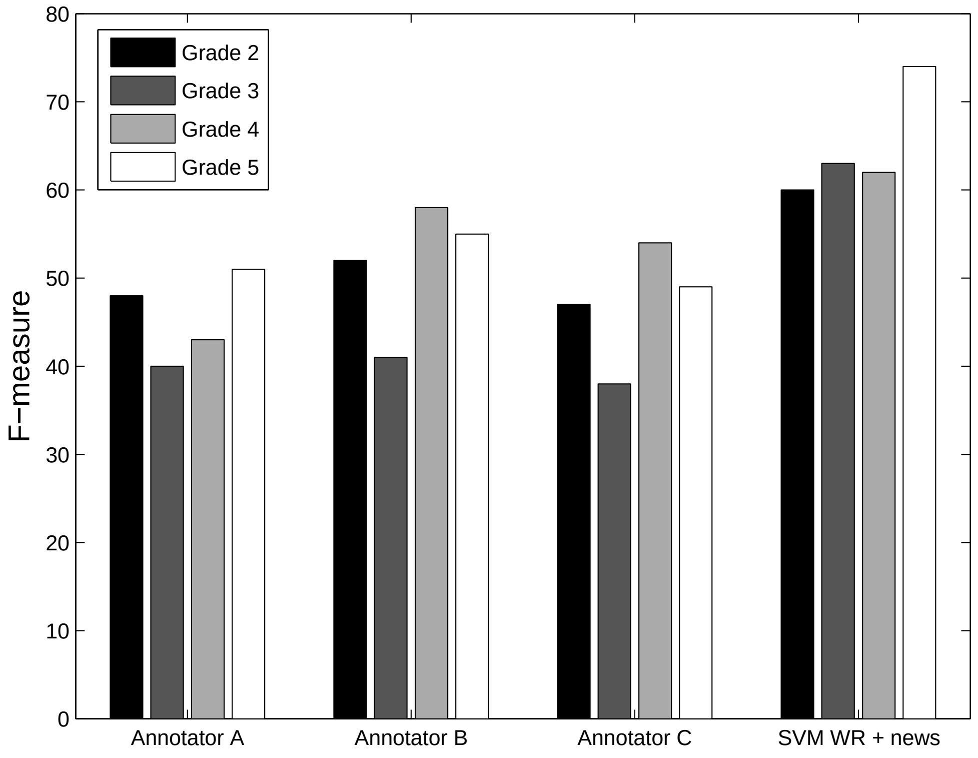

Figure 5.9: F-measures for annotators A, B, and C compared with SVM detectors trained on Weekly Reader plus newswire data.

Next, we look at F-measure for the annotators' labels evaluated on the Weekly Reader test set labels. As in the evaluation of the SVM detectors described in Section 5.3.1, for the human annotation results precision and recall indicate the percentage of articles annotated at each grade level, including articles with multiple annotations. Since annotating multiple grades per article penalizes precision but benefits recall, we use F-measure as a more balanced way to compare precision and recall results for this task. Figure 5.9 shows F-measure results for the labels provided by annotators A, B and C compared with results for the SVM detectors.[5] We observe three interesting trends in this chart. First, the SVM detectors outperform the annotators. Again, the SVM detectors are trained on this specific corpus, while the human annotators are likely to draw on personal experience as well as the sample articles, which may explain this difference. Second, all of the numbers for the human annotators are less than 60%, and some are significantly lower. This is a difficult task, even for people with appropriate education and preparation for the task. Third, some grade levels seem consistently harder than others. For example, F-measure is low for grade 3 for all three annotators. This could in part be an artifact of this particular task; the human labelers knew that there were no articles higher than grade 5 or lower than grade 2, so precision was higher for grades 2 and 5. However, it is still clear that this is a challenging task with some inherent ambiguity.

Next, we evaluate the performance of the SVM detectors trained on Weekly Reader and newswire data, and the Lexile measure, using the labels provided by the human annotators as a gold standard. We calculate F-measure for the SVMs and Lexile compared to each human annotator's labels individually and present the results in Figure 5.10. In the figure, lines connect the classifier scores for each grade relative to all three annotators. Asterisks indicate the F-measure for each classifier evaluated on the Weekly Reader test set labels. The SVM results are generally higher than the Lexile results, with some overlap, especially at grade 4. The SVM results compared to the Weekly Reader test set labels are the same or higher than the results compared to the human annotator labels. Except for grade 4, the Lexile results compared to human annotators tend to be better than compared to the Weekly Reader test set labels, which may also be due to the fact that Lexile is a general classifier that is not tuned to the Weekly Reader corpus.

The results of this annotation study indicate that reading level assessment is a challenging task even for people with moderate experience. Performance of the automatic system is better than that of human annotators on this focused task, although there is moderate

[5]The annotation experiments used a subset of about 80% of the original test set. The SVM and Lexile results in this figure are for this subset only and do not exactly correspond to the results in Section 5.7.

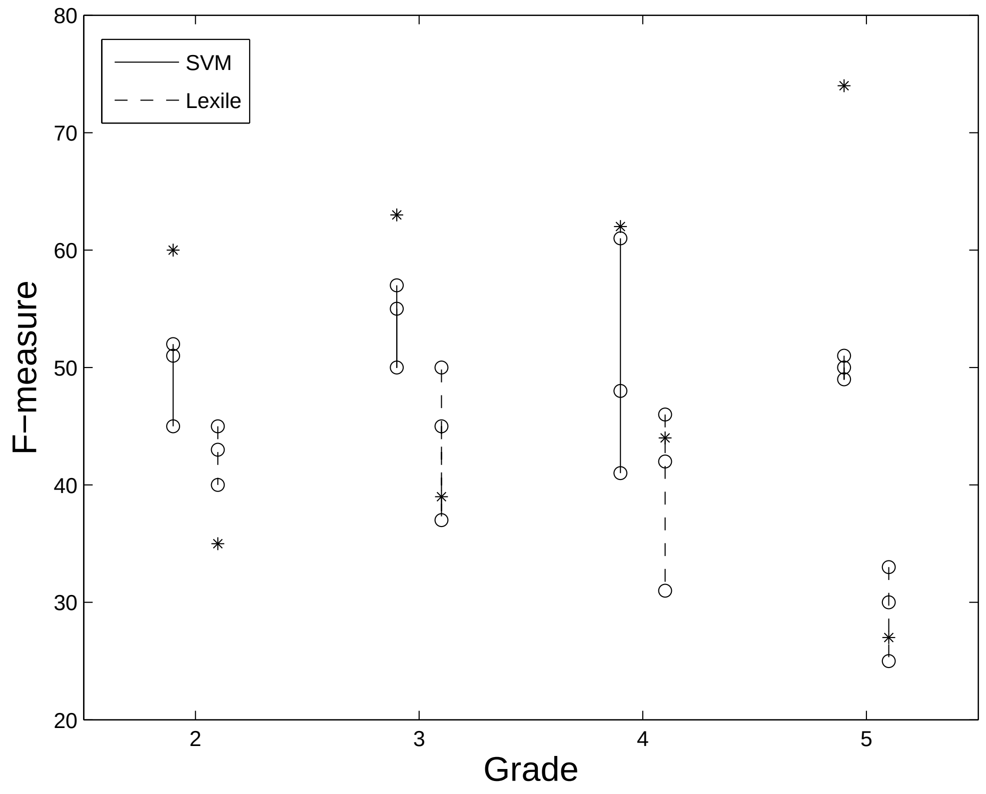

Figure 5.10: F-measures for Lexile and SVM detectors trained on Weekly Reader and newswire data compared to labels provided by human annotators. Asterisks indicate the F-measure for each classifier with respect to the Weekly Reader test set labels.

agreement among the annotators, indicating that the SVMs are more easily tuned to a particular task but not necessarily better in general. We also see that when evaluated on either the original test set labels or the human annotators' labels, there is an advantage to using trainable SVM detectors when compared to a traditional reading level classifier, Lexile.

5.9 Summary

We combine features from n-gram LMs, an automatic parser, and traditional methods of readability assessment in an SVM framework to classify texts based on reading level. On the Weekly Reader corpus, the full feature set outperforms classifiers based on the language models alone; the SVM detectors compare favorably to other existing methods; and their performance is better than that of human annotators on this corpus. These SVM detectors can be generalized to apply to newspaper text outside the initial domain with reasonable success. Adding higher-level negative training data improves generalization performance by reducing false positives without seriously degrading performance on the original test set. Experiments with human annotators show that the task of reading level assessment is challenging for people, too. The SVM detectors perform relatively well when evaluated on the labels provided by the human annotators, in addition to when evaluated on the original test set labels. The SVM detectors are trainable, which makes it not surprising that they outperform general classifiers, but this is an important characteristic for tuning performance for the needs of particular groups (e.g., native language learners vs. second language learners) or specific needs of particular students.

Individual variations in reading difficulties and text ratings indicate a need to improve classification tools using automatic learning techniques. In the next chapter, we work with teachers to evaluate the accuracy of our detectors on data from the World Wide Web and to further refine the detectors to help account for individual differences.

Chapter 6
ASSESSING THE READING LEVEL OF WEB PAGES

In this chapter, we address the problem of moving from a static collection of good quality text to the dynamic (but mixed quality) text resources found on the web, with the goal of online access for teachers and students. Graded textbooks and magazines such as Weekly Reader are one source of texts for educational purposes. We would also like to leverage the World Wide Web, with vast amounts of text on many topics, as a source of supplemental texts for teachers and students; the difficulty is efficiently sifting through these texts. We assume that the search engine (Google) does a good job on finding topic-relevant pages, constraining the focus of this effort to filtering the results to find text at the appropriate reading level. Teachers do not always have time to sift through many pages of search engine results to find web pages at the right reading level for their students. The goal of this work is to automate this task to enable teachers and students to make better use of texts available on the Web. This chapter describes our approach to augmenting the detectors described in the previous chapter to handle texts from the web. We also investigate adapting the detectors to individual teachers' needs and opinions.

The motivating scenario for this work is a teacher or student looking for web pages on a particular topic, e.g., for a research project, at a specified grade level. We use topics such as animal names, countries, and natural phenomena (e.g., tornadoes). We developed this scenario and these topic lists in consultation with teachers at a local elementary school with bilingual education and ESL programs. Our approach combines the work on reading level detection described in the previous chapter with ideas drawn from web classification results by Sethy et al. [81], work on incremental learning with SVMs by Rüping [76] and SVM adaptation methods developed by Li [61].

In the pilot study described in Section 6.1, we find that the web pages returned include a large number of pages that the detectors trained on clean text simply are not designed to handle. Hence, the tools need to be augmented to first distinguish web pages with narrative text from those that mainly have links, advertisements, or other unwanted content.

After discussing results of applying the detectors to raw web text in Section 6.1, we develop additional tools to filter web pages to remove "junk" automatically in Section 6.2 and also to remove pages that only contain links in Section 6.3, presenting results using both sets of filters in Section 6.4. These results confirm our observations about individual differences from teachers from Section 5.8, and in Section 6.5 we address this issue by investigating adaption of the detectors to individual users.

6.1 Pilot Study

We conducted a pilot study in which we applied our grade level detectors to web pages returned by Google for several topics and presented the top 15 positive results in each category to two elementary school bilingual education teachers.[1] The topics were elephants, giraffes, Japan and Mexico. The queries we submitted to Google consisted of the query term and the word "kids" with the intention that this would help find pages that were geared towards children. In some cases this was successful, but in other cases it returned an excessive number of sales pages with no real content, e.g., sites selling kids t-shirts with pictures of elephants on them. In subsequent experiments, we used the topic term only.

After retrieving topical web pages from Google, we used simple heuristic filters to clean up the text. We removed HTML tags and filtered the articles to keep only contiguous blocks of text consisting of sentences with an out of vocabulary word (OOV) rate of less than 50% relative to a general-purpose word list of 36k English words. Then we ignored articles without at least 50 words remaining and applied the reading level detectors described in the previous section to the remaining texts. Finally, we sorted the results by the score given by each grade level detector and presented the top 15 pages for each topic and grade to the annotators. In some cases, the total number of articles was less than 15. Despite

[1]These teachers served as annotators for the reading level experiments described in Sections 6.1 and 6.2.

downloading a large number of articles for each topic, we discovered that some topic/grade combinations did not result in many hits for the grade level classifiers.

Annotators were asked to view each page and choose from the following labels:

- Good = Acceptable for this grade level.
- Too low = Too easy for this grade level.
- Too high = Too high for this grade level.
- N/A = Off topic or not appropriate for students.

The annotators were not trained on sample texts for each grade level, with the premise that eventual users of the system (teachers) will not want to have special purpose training (other than their education backgrounds). From Section 5.8, we know that there are considerable individual differences between teachers, which should be respected, so the goal here is to improve the percentage of useful texts returned between different trials for the same person.

Table 6.1: Pilot study teacher annotation results. Annotator A viewed 151 articles, and Annotator B viewed 135 articles.

	Percentage of articles	
Label	**Annotator A**	**Annotator B**
Good	24%	35%
Low	1%	0%
High	5%	7%
N/A	69%	58%

The annotators viewed the topics in a different order from each other, and neither finished all the articles in the time allotted for the pilot study. Table 6.1 shows the percentage of articles they annotated with each label. Clearly, these results indicate much room for improvement. Most articles are off topic or otherwise inappropriate. While the categories given to the annotators did not allow us to distinguish between these reasons, anecdotal examples suggested that we needed to improve our filtering techniques to remove "junk" pages and increase the quality of the texts submitted to the reading level detectors. We also observed that sorting the web pages by the score of the reading level detector had the unfortunate side effect of returning articles that happened to score well for reading level but were far down the original list of search engine hits and likely to be off topic. In order to increase the topicality of the results, in subsequent experiments the order of pages returned by the search engine is respected. We step through the pages in their original order according to the search engine, returning the first N pages that are classified positively by the reading level detector.

6.2 Filtering Web Text

6.2.1 Heuristics

Based on observations in the pilot study, we made some changes to improve the heuristics used to filter web pages prior to applying the reading level detectors. We continue to filter sentences with an OOV rate greater than 50% but no longer stipulate that all remaining sentences in an article must be from a contiguous chunk. We also filter lines with fewer than 5 words; this removes many leftover formatting and navigation elements of the original web page such as menu bar items, titles, and other web-page-specific artifacts that are not part of the main text of the page.

6.2.2 Naive Bayes Classifier

Unfortunately, the heuristics described above eliminate only a small portion of the non-applicable articles. Many web pages returned by the original Google query are "junk", containing little text to be classified. Inspired by Sethy et al.'s work on web page classification [81], we designed a naive Bayes classifier to distinguish "content" pages from "junk." We select features for this classifier according to their information gain as described in

Chapter 2. All other words that appear in the text are replaced by their part-of-speech tag. After selecting features, a naive Bayes classifier was trained with the *rainbow* program from the Bow toolkit [68].

For our original version of the content vs. junk classifier, we used the annotated web pages from the pilot study as training data. There were 86 negative examples (i.e., "junk" pages) and 36 positive examples (i.e., "content" pages) which we augmented with 25 additional hand-selected pages for a total of 61 positive examples. The feature selection process described above resulted in 404 word features for the original naive Bayes classifier. The number of features increased somewhat in further iterations, described next, but remained in the range of 400 to 500 words.

In order to validate and improve this classifier, we conducted two iterations of annotation and retraining. This technique of selecting additional examples for annotation to improve the classifier is similar to both active learning, as described in the context of SVMs in [94], and relevance feedback, as presented for SVMs in [33]. In the active learning scenario, the examples for which the classifier is most uncertain are presented to the annotator for feedback. In the relevance feedback approach, the user provides feedback on the highest ranked examples. In our work, articles are ranked topically by the search engine, and we choose the first N which are positively classified by the content vs. junk classifier for annotation. These are not necessarily either the most certain or most uncertain examples according to the content vs. junk classifier alone, but they do correspond to the highest ranked examples of the system overall, combining topic and content/junk classification, which is in fact what we want to improve.

Specifically, at each iteration we presented an annotator with a set of articles from each of eight topic categories; the categories were different in each iteration and different from the topics used in the pilot study. The topics were names of animals, countries/continents, and weather phenomena. For the first iteration, the topics were dogs, parrots, sharks, snakes, earthquakes, storms, tornadoes and volcanos. The second-iteration topics were alligators, cobras, whales, zebras, Africa, Egypt, Kenya and Spain. The articles presented to the annotator were the first 30 positively classified articles for each topic according to the content vs. junk classifier. We sought to optimize our use of the annotator's time by only showing them the articles that the classifier believed were "content" and having them confirm or correct this classification. After the first iteration, we added the newly annotated data to the training data and re-ran the feature selection and classifier training steps. We did the same after the second iteration, resulting in the third-iteration content vs. junk classifier used to filter web pages for the reading level detection experiments described in the next section. Table 6.2 shows the number of examples of both content and junk pages used to train each iteration of the content vs. junk classifier. From this data, one can see that an increasing percentage of good content pages are being returned with each iteration, though the numbers are not strictly comparable because the samples are different. In fact, the gain is bigger than indicated in the table, since the actual number of positive content hits in the pilot study was 36, or 30% content. An additional 25 content pages were added by hand to create a more balanced training set for the first iteration.

Table 6.2: Number of training samples for each iteration of the content vs. junk classifier. The training data for iterations 2 and 3 includes the data from the previous iteration.

Iteration	**Num Content**	**Num Junk**	**% Content**
1	61	86	.41
2	217	159	.58
3	361	241	.60

To compare performance across all three iterations of the content vs. junk classifier, we applied these classifiers to a third set of web pages consisting of approximately 60 pages per topic for each of six new topics. The human annotator labeled these pages as either content or junk, resulting in a test set of 348 pages, of which 177 were content and 171 were junk. We applied the three versions of the content vs. junk classifier to this test set. Table 6.3 shows the percentage of articles which were correctly classified in each category. Note that

Table 6.3: Percentage correct classification for content vs. junk classifier at each iteration and for each category.

Iteration	**Content Correct**	**Junk Correct**	**Total Correct**
1	80%	70%	75%
2	94%	55%	75%
3	93%	57%	75%

the total percentage of correctly classified pages is the same across all three iterations, but substantially more content articles are classified correctly by the second and third iteration classifiers. This comes at the expense of more junk articles that are incorrectly classified (i.e., false positives), though some of these may be eliminated in the reading level detection stage. The difference in performance may also reflect the change in the percentage of content pages in the training pool in the second and third iterations.

6.2.3 Reading Level Experiments

We conducted a set of reading level detection experiments on six novel topics (rainforests, hurricanes, tsunami, frogs, leopards and owls) with the same teachers serving as annotators as in the pilot study. We downloaded approximately 1,000 web pages per topic, using the topic word as the Google query. These pages were filtered using the heuristics and the naive Bayes classifier described in the previous section, resulting in between 325 and 450 pages on each topic. The reading level detectors described in Section 5.4 were applied to these articles. The first 10 hits per topic that were classified positively by each grade level detector were presented to the annotators for labeling. Many topics did not have a full 10 hits for grade 2; this is probably due to most web pages being at higher reading levels.

The annotators were asked to choose from the following set of labels for each article. These are similar to the labels used in the pilot study, with slight changes based on observations from that study. "Just links" refers to web pages which provide links to other pages but do not have much content in terms of paragraphs of text to read. These pages are not necessarily off topic, but they are also not the sort of text on which we hope to detect grade level. Annotators were also able to provide free response comments about any page about which they wished to make an additional note.

- Good = Acceptable for this grade level.
- Too low = Too easy for this grade level.
- Too high = Too high for this grade level.
- Just links = A page of links with no significant text. Probably still on topic.
- Off topic.
- Page did not load.

Table 6.4: Summary of annotations for web pages. Percentages do not sum exactly to 100% because in some cases, the annotators marked more than one label, e.g., good reading level but off topic.

	Percentage of articles	
Label	**Annotator A**	**Annotator B**
Good	59%	41%
Low	0%	0%
High	8%	18%
Just links	14%	16%
Off topic	20%	21%
Page did not load	1%	4%

Table 6.4 shows the overall percentage of labels selected by the annotators for all the web pages they viewed. Both annotators labeled a significantly larger percentage of the data as “good” and a much smaller percentage in the not applicable categories: 35-41% vs. 58-69% in the pilot study. In addition, the percentages of pages labeled “just links” and “off topic” are similar for the two annotators. Interestingly, the annotators indicated in free response comments that the off topic articles were off topic but not inappropriate for kids, e.g., a web page for a research study with the acronym OWLS for its name, and a web page about the town of Owlshead both appeared for the query “owls.”

Table 6.5: Percentage of articles of each grade level labeled “good” by each annotator.

Grade	**Annotator A**	**Annotator B**
2	40%	30%
3	37%	23%
4	66%	47%
5	81%	56%

Table 6.5 shows the percentage of web pages marked as “good” for each grade level by the two annotators. Neither annotator thought that any of the articles detected for any grade level were too low for that level. This pattern of detecting articles that are too high but not too low is different from that observed on the “clean” Weekly Reader data, probably because the material on the web is not balanced for reading level and has fewer low-grade-level articles. Annotator B found more articles that were too high than annotator A did; as seen earlier, there are individual differences in judgments about reading level. In general, the reading level detectors are more accurate for the higher grade levels. Also, the overall number of pages returned by the grade 2 detector was low, probably because most web pages are targeted for older audiences. As a result, in the remainder of this chapter, we focus on grades 3-5. This choice also allows us to increase the number of pages per topic/grade pair that are annotated, without creating an unrealistic burden on the annotators.

6.3 Filtering Links

In this section, we described further experiments to improve the filtering techniques used prior to applying the reading level detectors.

6.3.1 Approach

While the filtering techniques in the previous section show large improvements over processing raw web pages, we note that about 15% of the returned pages were labeled by the annotations as "just links", i.e., pages that contain links to other pages but little or no narrative text. The percentages of "just links" pages varies substantially with grade level: 27% for grade 3, 15% for grade 4 and 5% for grade 5, but does not vary between annotators in this experiment. In this section, we use the annotated files from those experiments as training data for a new filter which distinguishes pages that are mostly links from other content pages.

Filtering web pages in this manner is a challenging problem with no solutions that we are aware of described in the literature. We used a heuristic approach, comparing features such as the number of links compared to the total number of words in the file, the word count of the raw HTML file and the word count of the text of the page only. We investigated the following heuristics for this filter:

- Percentage of lines of the raw HTML file which include link tags
- Percentage of words of the raw HTML file which are link tags
- Number of words in the web page after HTML tags are removed divided by the number of words in the raw HTML file

We compared these statistics for pages annotated as "just links" and all other pages in the previous experiment. The first heuristic showed little difference between the two categories of pages. We also recognized that "line" is a very inconsistently-sized unit in raw

HTML files, since line breaks in the raw file are only really included as a convenience for human readers, and different tools for HTML development include different percentages of line breaks. This observation led us to try the second heuristic, but it also showed little difference between the two categories. The third heuristic did show a distinct difference between files that were mostly links and other files, and we were able to empirically select a threshold value based on the annotated data.

6.4 Combined Filtering Experiments

To assess the combined performance of all of the filtering techniques described in this chapter and the reading level detectors, we conducted another set of experiments with teachers to annotate results. Since the annotators from the previous sections were no longer available, we hired three new individuals. All three were graduate students in the College of Education at the University of Washington, and all had teaching experience in elementary school bilingual/ESL classrooms. This set of experiments uses six new topics: butterflies, chimpanzees, comets, herons, thunderstorms, and turtles. This section describes baseline results in which the teachers annotated the first 30 Google hits for each topic, and experiments with the combined filtering techniques and reading level detectors.

6.4.1 Baseline Experiment

To provide a baseline for evaluating the performance of our detectors, the annotators were asked to label the first 30 hits from the search engine Google for each of the six topics. The annotators were asked to choose from the following set of labels:

- Lower = Lower than 3rd grade level.
- 3 = Appropriate for 3rd grade.
- 4 = Appropriate for 4th grade.
- 5 = Appropriate for 5th grade.

- Higher = Higher than 5th grade level.
- Off topic.
- No text = No narrative text (less than a few sentences).
- Broken = Page did not load.

Annotators were instructed to choose as many labels as they felt were appropriate for each page. For the purposes of analysis and presentation, we have grouped the labels in two categories, those related to filtering issues and those related to grade level. Note that in the baseline experiment we do not apply any of the filters described earlier in this chapter; the "filtering errors" in the baseline experiment show the prevalence of these types of pages in the results of a normal Google search. Figure 6.1 shows the number and types of filtering errors compared to pages labeled for grade level only (abbreviated as "good" in the figure). Most pages (52%-62% depending on the annotator) do not have filtering problems. However, 20-35% are marked as not including sufficient narrative text for use in reading or evaluation of reading level (compared to 35-41% for the single filter in the previous section). About 10% of the pages are labeled "off topic", which indicates that Google is doing a reasonable job finding topical pages. A small percentage of pages are labeled "broken", meaning that the page did not load when the annotator attempted to view it. This category is larger than one might normally expect for dead links in a web search, but these errors may be partially explained by some lag time between the day when the pages were originally retrieved and the experiment was set up, and the day when the annotators viewed them. (This category exists with similar percentages in the next section.) In the following section, we will see the effect of our filtering methods in addition to the reading level detectors.

In Figure 6.2, we present the distribution of pages in the subset of the baseline Google hits that were deemed on topic and including text. The percentages are relative to the entire set of returned pages, but this figure only includes "good" pages (those without filtering problems as indicated in Figure 6.1). Annotators were free to mark more than one grade

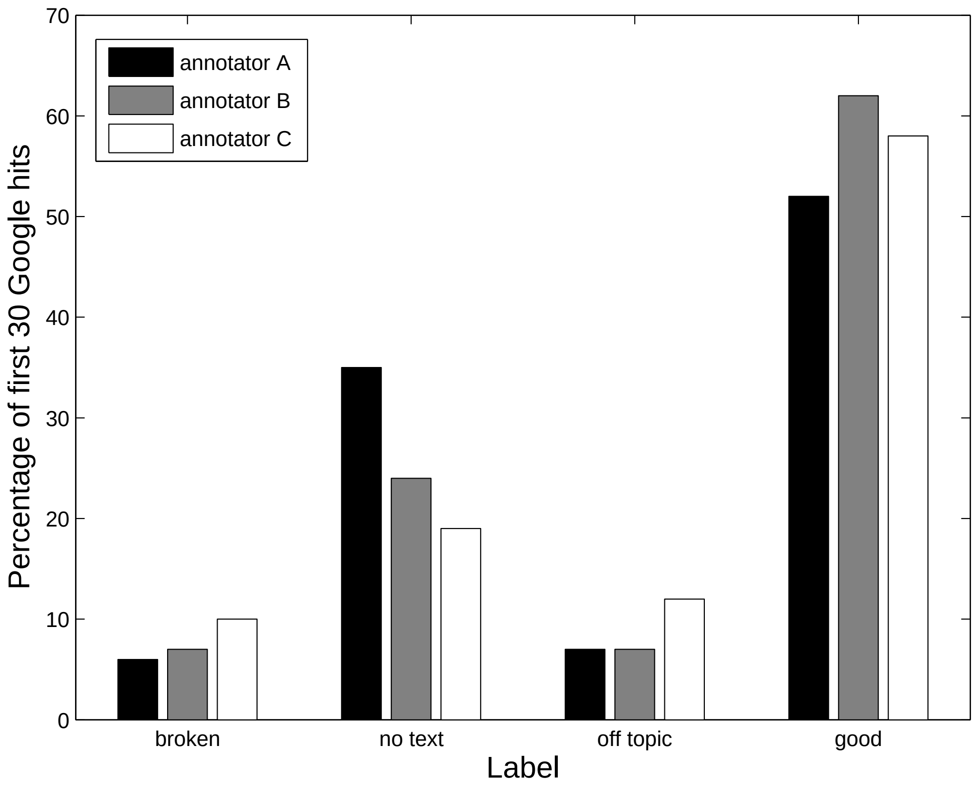

Figure 6.1: Percentage of pages with filtering problems and the percentage of "good" pages, i.e., those which the annotators were able to mark for grade level, in the first 30 web pages returned by Google for each of 6 topics.

level per page; annotators B and C regularly made use of this option, while A did not. Less than 10% of the baseline pages were considered appropriate for third grade by all three annotators, an average of 16% were labeled as good for fourth grade, and 21% were selected as good for fifth grade. The most "optimistic" score in the baseline experiment came from annotator C, who labeled 31% of the pages as good for fifth grade. These results show the need for reading level detection for this task and provide ample room for improvement with our tools.

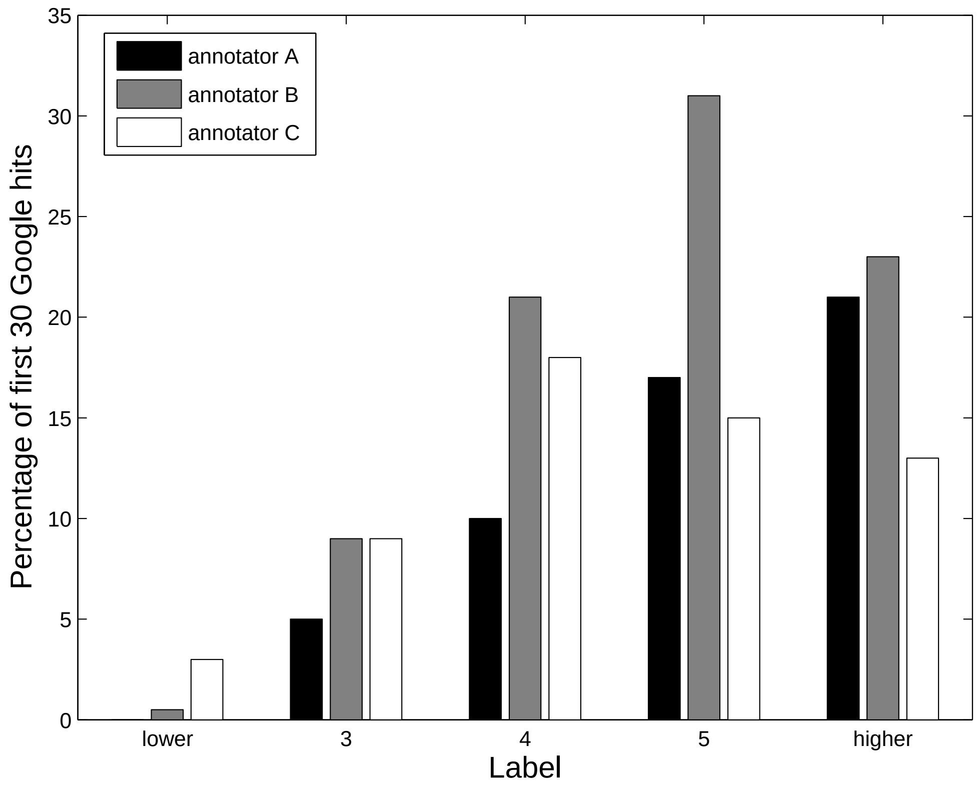

Figure 6.2: Percentage of pages in the first 30 web pages returned by Google for each of 6 topics the annotators marked as appropriate for various grade levels. Percentages are relative to the entire set of annotated pages, although only pages without filtering problems (as indicated in Figure 6.1) are included in this figure. Annotators were free to mark more than one grade level per page.

6.4.2 Filtering and Reading Level

Similar to the experiments in Section 6.2.3, for this section we downloaded approximately 1,000 web pages for each of the same six topics used in the baseline experiment. We applied the heuristic filters, the Naive-Bayes content vs. junk filter, and the links filter sequentially, resulting in a total of 1494 web pages (an average of about 250 per topic) and applied our reading level detectors for grades 3, 4 and 5 to these pages. For each grade, we selected up to the first 20 positively detected pages for each topic[2] to be annotated. The annotators were asked to choose from the following labels:

- Low = Lower than the specified grade level.
- Good = Appropriate for the specified grade.
- High = Higher than the specified grade level.
- Broken = Page did not load.
- No text = No narrative text (less than a few sentences).
- Links = The page is mostly links.
- Off topic.

The annotators were instructed to choose only one of the labels "low", "good" and "high" for each page. They were permitted to select multiple other labels, or a reading level label and a filtering label if they wished, although this was fairly rare in practice. For the purposes of analysis, we separate the filtering errors prior to analyzing reading level errors and in the reading level results we do not include pages with filtering errors that also happened to be labeled for grade level.

Table 6.6 summarizes the results for all grade levels combined for this experiment, and Figure 6.3 shows the filtering results in particular. We note that no pages were labeled "no text", indicating that the filters do an acceptable job of reducing the number of non-useful pages returned, although the "just links" may also cover pages that could previously have been considered "no text". Annotator A appeared to choose a broader interpretation of a page of links, labeling a high number of pages with this category, while annotators

[2]The grade 4 detectors returned fewer than 20 pages for some topics, resulting in 75 total pages for grade 4 and 120 pages each for grades 3 and 5.

Table 6.6: Summary of annotations for web pages in the combined filtering experiment. Percentages do not sum exactly to 100% because in some cases, the annotators marked more than one label, e.g., good reading level but off topic.

	Percentage of articles		
Label	**Annotator A**	**Annotator B**	**Annotator C**
Good	30%	50%	65%
Low	0.3%	6%	3%
High	34%	25%	7%
Just links	27%	12%	14%
Off topic	3%	3%	8%
Broken	4%	4%	3%

B and C marked similar lower percentages of pages. For these two annotators, the link filtering does provide an improvement in grades 3 and 5, but not grade 4. However, the "off topic" percentages are lower across all grade levels compared to the results of the previous experiment, without link filtering.[3] Thus, while the improvements in filtering of "just links" pages appear modest, the filtering provided by this step may be having an additional impact by removing other undesirable pages. Table 6.7 shows the percentage of detected pages at each grade level which do not have filtering errors. In general, the combined set of filtering techniques work better at the higher grade levels, though this could be partially due to the existence of more good pages at higher grade levels on the web in general.

[3]These experiments used different topics and annotators, so the results are not directly comparable. However, since the results of each experiment are an average over six topics, the general trends should be reliable.

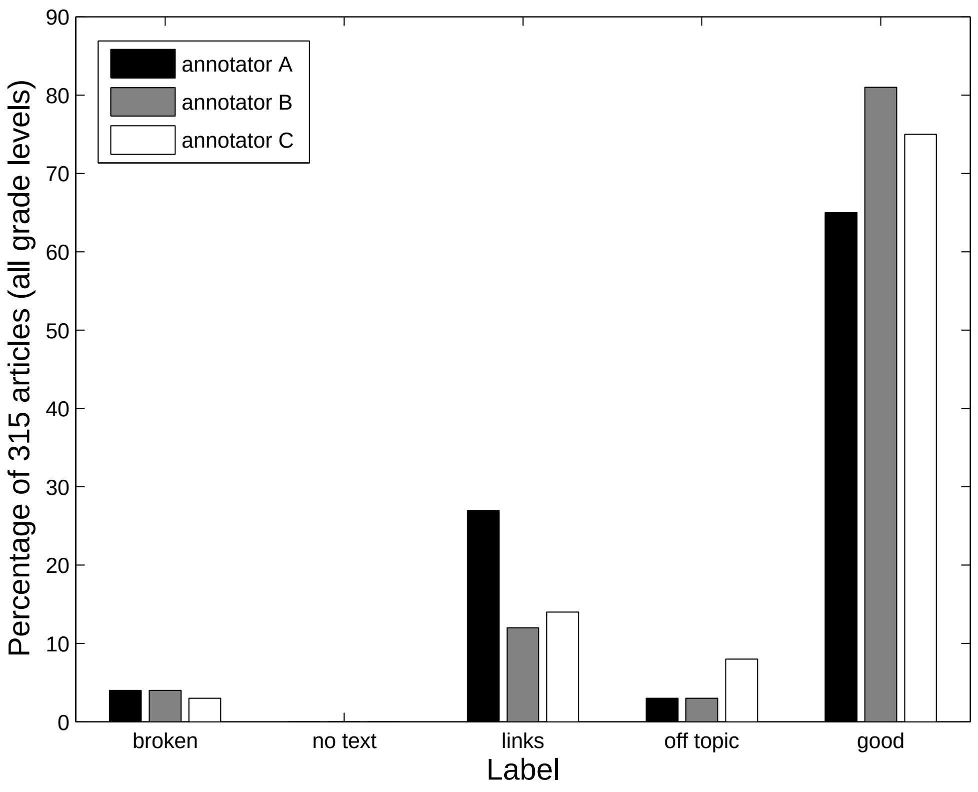

Figure 6.3: Percentage of pages with filtering errors and the percentage of "good" pages, i.e., those containing narrative text in pages returned by the combined filtering and reading level detection system.

Table 6.7: Percentage of pages detected for each grade level which were labeled as not having filtering errors.

	Percentage of articles		
Grade	**Annotator A**	**Annotator B**	**Annotator C**
3	52%	72%	65%
4	57%	77%	71%
5	82%	92%	87%

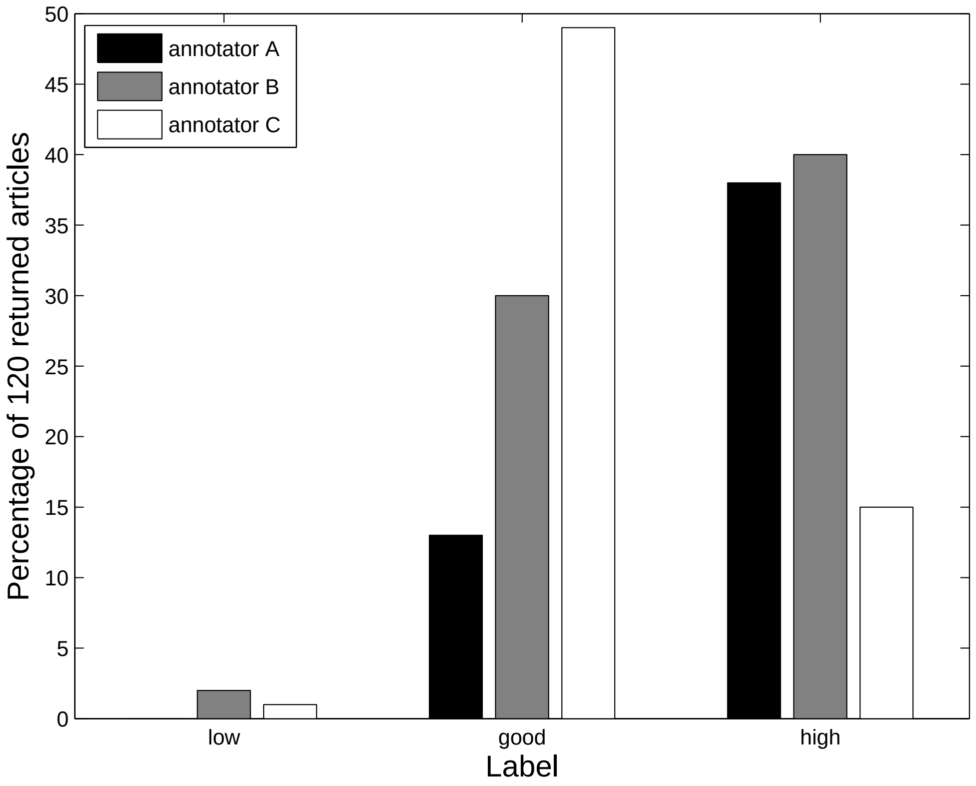

Figure 6.4: Grade level annotations for pages detected by the combined system for grade 3, excluding pages which had filtering errors; i.e., these are the “good” pages from Figure 6.3.

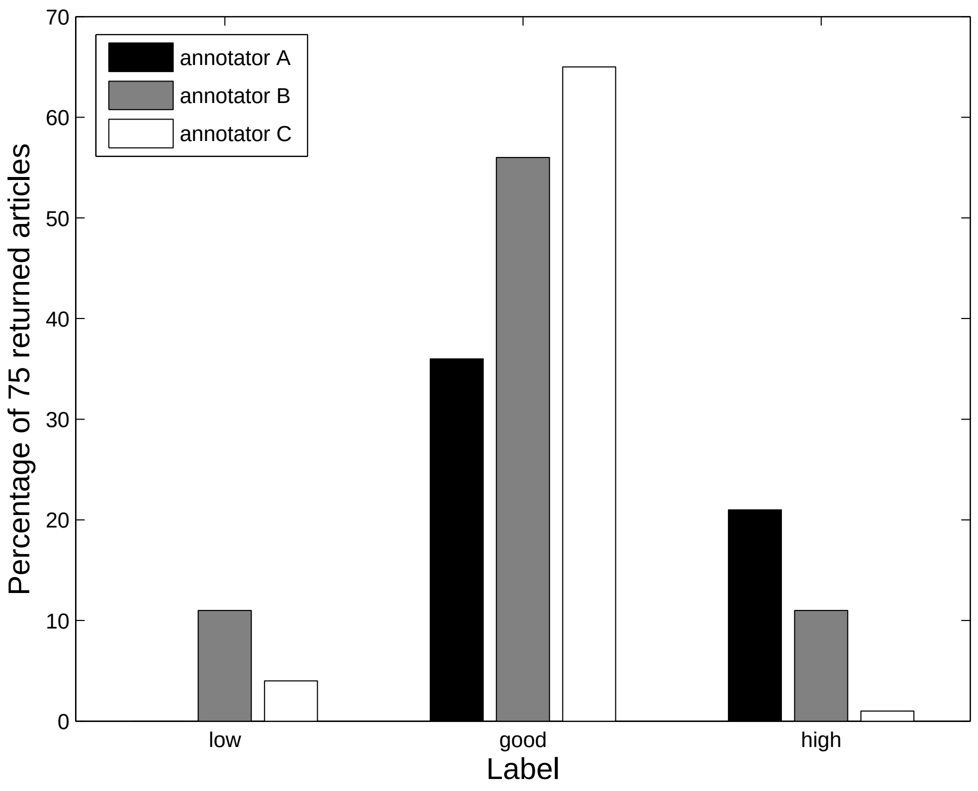

Figure 6.5: Grade level annotations for pages detected by the combined system for grade 4, excluding pages which had filtering errors; i.e., these are the "good" pages from Figure 6.3.

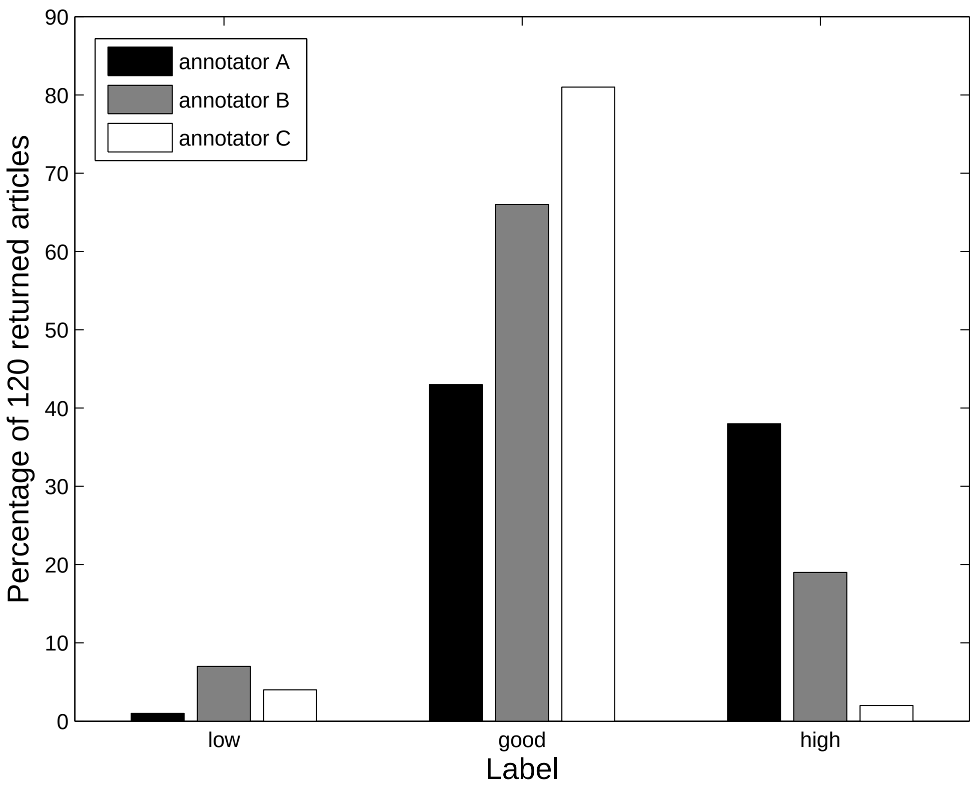

Figure 6.6: Grade level annotations for pages detected by the combined system for grade 5, excluding pages which had filtering errors; i.e., these are the "good" pages from Figure 6.3.

Figures 6.4, 6.5, and 6.6 show grade-level annotation results, separated by grade level, for the web pages which did not have filtering errors. Very few of the pages were considered too low for the selected grade levels, while many of the pages chosen for grade 3 were considered too hard by annotators A and B. In general, the detectors work well for grades 4 and 5, according to all three annotators. As in previous experiments, we see a fairly large amount of variation between individuals. Annotator C had the most “permissive” attitude towards all grade levels, marking more pages as good than too high for grade 3, and very few as too high for grades 4 and 5. This person has more experience teaching students at these grade levels than the other two annotators, so this seems like an example of normal variation and not an anomaly due to lack of annotator experience. Overall, these results show good success for the filtering system and reading level detectors, and also indicate the need for a system that can be adapted to individual users.

In these experiments, we chose to return a maximum of 20 pages per topic and reading level, aiming for higher recall at the possible expense of precision. The average number of pages returned for a particular topic and reading level is 17 since the grade 4 detector did not always return the maximum amount. In order to assess whether higher precision might be obtained by using a more conservative detection threshold we analyzed two subsets of the above results corresponding to lower numbers of returned pages. Two new thresholds were selected, each of which reduced the number of returned articles by approximately 30%, resulting in an average of 12 and 8 articles per topic and grade, respectively. Figure 6.7 shows the change in precision for each grade level for annotator B; these results are representative of the trends seen with all three annotators. While one might expect that returning fewer articles would result in higher precision, the precision is actually fairly stable for these subsets of the results. It would also be interesting to investigate moving the threshold in the opposite direction to increase recall at the expense of precision, but this would be much more time consuming for the teachers and would require more extensive evaluation of the utility for them.

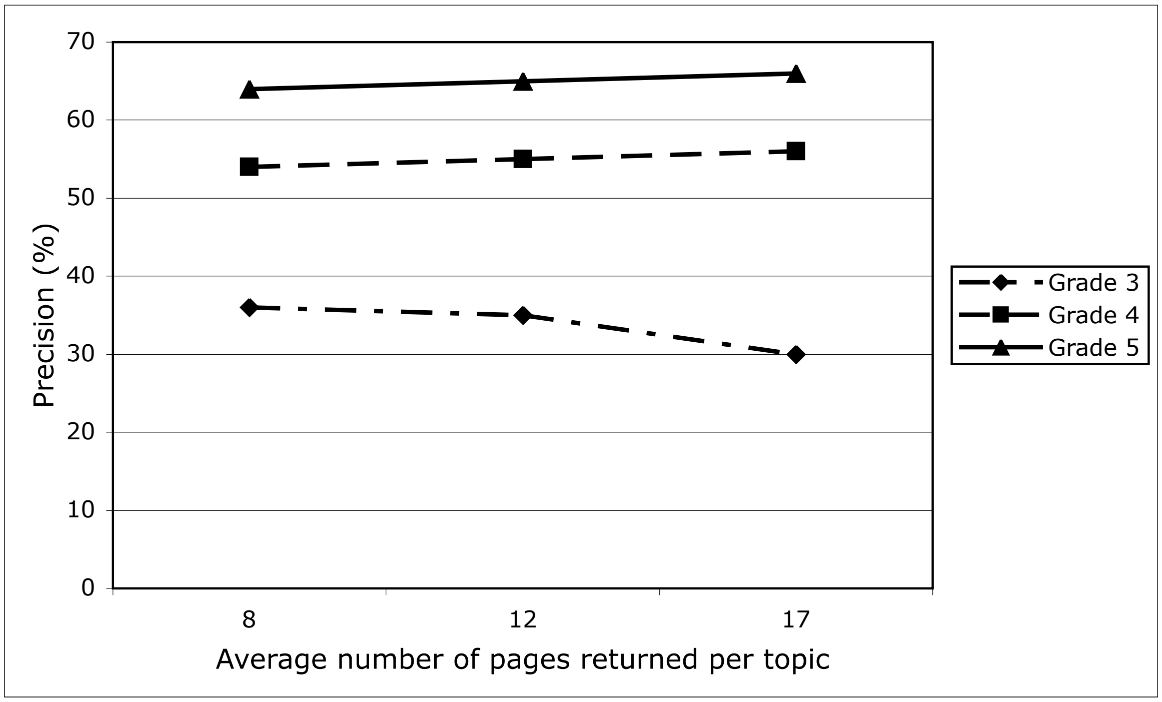

Figure 6.7: Percentage of articles labeled "good" by annotator B for new thresholds which result in fewer returned articles, but stable precision. The original threshold corresponds to an average of 17 articles per topic and grade.

6.5 Adaptation for Individual Users

To adapt our reading level detectors, we draw on results in incremental learning with SVMs by Rüping [76] and SVM adaptation methods developed by Li [61]. Both of these approaches adapt an SVM by using new training examples to adapt the existing support vectors. We focus on Li's regularized adaptation approach because software is available for two variants of this approach in an augmented version of the SVMTorch toolkit. The first adaptation method, which Li refers to simply as "regularized adaptation," keeps the support vectors from the original model and uses the adaptation data to create additional support vectors. Li's second method, "extended regularized adaptation" uses the support vectors from the original model as if they were training examples, adds the adaptation examples, and trains a new SVM model from the combined set of examples. Li tested these adaptation methods on vowel classification and image classification. Both of these tasks involve continuous features. The vowel classification task had a much larger data set than our reading level task. The image data set was comparable in size to our task, but the classes were very well separable, which is not the case for our reading level data. In Li's experiments, regularized adaptation worked better for the vowel classification task, while extended regularized adaptation had better performance on image classification. Since our text classification task is rather different from both of these tasks, we tried both adaptation methods on our task.

Table 6.8: Number of positive and negative training examples for each annotator and grade level.

Grade	**Number of examples**					
	Annotator A		**Annotator B**		**Annotator C**	
	Pos	Neg	Pos	Neg	Pos	Neg
3	16	46	36	50	59	19
4	27	16	42	16	49	4
5	52	47	79	31	97	7

We attempted to adapt each of our SVM detectors for grades 3-5 using results from each annotator in the experiment described in the previous section. We used only the web pages that did not have filtering errors as adaptation data for each grade level. Pages labeled "good" were considered positive adaptation examples and pages labeled "low" or "high" were used as negative examples. Table 6.8 shows the number of positive and negative adaptation examples for each annotator and grade level. For comparison, the original Weekly Reader training set contains about 3k examples, and the original SVM detectors each had about 1k support vectors.

As in Li's experiments, for the adaptation experiments we used the SVM parameters found by cross-validation for the original models. One additional challenge of our task is the need to train an additional threshold, which we did using the Weekly Reader development set for the original models. Since we are not trying to tune to the Weekly Reader set in this case, in the extended regularized adaptation case we used 5-fold cross-validation on the training set (including the adaptation data) to tune thresholds. For the regularized adaptation variant, only the adaptation data is used for training, and cross-validation did not work due to the small amount of adaptation data, so we combined the original development set with the adaptation data in the same cross-validation framework. Unfortunately none of these approaches resulted in thresholds which were effective; unlike the thresholds tuned in the original SVM experiments, these resulted in either very few positive hits or nearly all positive hits.

We speculate that while SVMs worked well for the original classifier, the existing adaptation methods are not geared towards our text-based classification problem. A different classifier or adaptation method is needed for this application with very small amounts of adaptation data.

6.6 Discussion

Our SVM-based reading level detectors trained on Weekly Reader text can be applied successfully to text from web pages. However, filtering the web pages to eliminate "junk" pages and pages which are mostly links before applying the reading level detectors is essential. Currently, the filtering techniques and reading level detectors are too time-consuming to be applied in an online system that works in real time. This could be addressed in future work, either by efforts to streamline and speed up the filters and detectors, or by building a large database of pre-processed web pages which have already been filtered and had features calculated for the reading level detectors. The latter approach is taken in the REAP project [6, 41], although for the system envisioned in our scenario, it would be necessary to consult teachers and build a large list of topics of interest.

Reading level assessment is a subjective problem. Results vary between annotators, as different users have different perceptions of the appropriateness of articles for a particular grade level. We investigated a particular set of algorithms for SVM adaptation which have shown good results on non-text-based classification tasks. While this adaptation method was not successful for this task, different classifiers and adaptation techniques could be tested in the future.

PART II: TEXT SIMPLIFICATION

Part II of the dissertation presents our work on automatic text simplification techniques. As discussed in Chapter 1, language teachers and students would benefit from such techniques, which would save time for teachers and provide more readable and useful texts for students, e.g., on topics related to their science or social studies curriculum.

In Chapter 7 we discuss relevant background on text simplification and related topics. Chapter 8 presents an analysis of the corpus used in our text simplification work, and Chapter 9 presents a decomposition of the simplification problem into subtasks and presents preliminary experimental results on the applicability of existing tools for two of the components.

Chapter 7

BACKGROUND AND RELATED APPROACHES TO TEXT SIMPLIFICATION

Text simplification for second language language learners is a somewhat controversial issue in the education community. Some experts prefer the use of "authentic" texts, i.e., texts which are written for a purpose other than instruction, while simplified texts are also very common in educational use.

In the first section of this chapter, we briefly summarize related education research on authentic and simplified texts. In the second section, we describe text simplification and adaptation systems intended for language learners. The third section presents existing NLP approaches to text simplification which are not specifically targeted at an educational context. The fourth and fifth sections describe approaches to automatic summarization and sentence compression, tasks that alone are not sufficient for simplification but which can be components in a simplification system.

7.1 Simplified Texts in Education

Proponents of authentic texts tout positive effects on student interest and motivation and advantages of exposing students to "real" language and culture. However, authentic materials are often too hard for students who read at lower levels, as they may contain more complex language structures and vocabulary than texts intended for learners [52, 87]. Studies of the lexical and syntactic differences between authentic and simplified texts, and their educational impact, show mixed results [105, 31], indicating that both authentic and simplified texts can have a place in education. In particular, authentic texts are not always available for students whose reading level does not match their intellectual level and interests, and teachers report spending substantial amounts of time adapting texts by hand. Automatic simplification could be a useful tool to help teachers adapt texts for these students.

Simensen conducted a study of six publishers' policies on adaptation of authentic texts in materials for English as a foreign language learners [86]. She collected guidelines provided by publishers to writers and generalized these guidelines into three principles of adaption: control of information, control of language, and control of discourse. Information control includes reduction of "marginal" or "irrelevant" information by eliminating longer descriptive passages and subplots. Supply or substitution of new information is also part of control of information. Control of language is accomplished by lists of acceptable vocabulary and grammatical structures at different levels, but is also often performed based on the writer's intuition. Control of discourse is only mentioned by two publishers in the study, who suggest that choppy, disjointed sentences in an adapted text can be hard to read. Simensen observed that adaptation is often performed on an intuitive basis, based on assumptions about what is harder or easier for language students. In Chapter 8, we present a data-driven analysis of a corpus of original and simplified texts to gain more insight into specific grammatical characteristics and other features of simplified texts. Although authors may make many of these changes based on their own intuition, we believe that we can discover regularities that can be used by an automatic system.

7.2 Automatic Text Simplification for Educational Purposes

Inui *et al.* address the needs of deaf learners of written English and Japanese by paraphrasing texts to remove syntactic structures known to be difficult for this group of learners [47]. The first part of their system is a readability classifier trained on paraphrases of individual sentences which were hand-labeled for readability by teachers of the deaf. The second phase of the system performs simplification using several thousand hand-built tree transformation rules. Although their work focuses on improving the speed and ease of the rule-creation process, the need to manually generate a large number of rules is cumbersome.

The Practical Simplification of English Text (PSET) project's goal is to paraphrase newspaper texts for people with aphasia[1] [11, 13, 14]. The PSET system follows a similar "analysis and transformation" approach to that of Chandrasekar and Srinivas [17] (see Section 7.3). This approach begins with lexical, morphological and parse analysis of the text and performs the following syntactic simplifications: splitting of compound sentences, converting seven types of passive clauses to active voice, and resolving and replacing eight common anaphoric pronouns. Despite the complexity of the analysis phase of this system, the scope of the resulting transformations is limited to simplification of specific types of phrases known to be challenging to aphasics.

A related ongoing project by Max and colleagues at LIMSI-CNRS in France also targets language-impaired readers such as those with aphasia [66, 67]. They are developing an interactive text simplification system which is built into a word processor. Based on the observation that automatic transformations are likely to change meaning, they choose to integrate simplification in the writing process to allow the writer to maintain control over content and meaning. The writer requests simplification suggestions for an individual sentence; an automatic parser generates analysis of an individual sentence; and the system applies rewrite rules which were handwritten by a linguist. The resulting suggested simplifications are ranked by a score of syntactic complexity and potential meaning change. The writer then chooses their preferred simplification. This system ensures accurate output, but requires human intervention at every step and is more useful for creation of new texts than for adaptation of existing texts.

The Education Testing Service (ETS) has developed the Automated Text Adaptation Tool [8, 9]. This tool uses existing NLP tools to provide text adaptations in English and/or Spanish that are displayed together with the original text. The adaptations which are provided include vocabulary support, marginal notes, and text-to-speech. The vocabulary support includes synonyms for lower-frequency words, antonyms, and cognates in English and Spanish. The marginal notes are provided by an automatic summarizer, and the user can control the amount of notes provided by selecting the summarization percentage. The

[1]Aphasia is an impairment in the ability to process language due to a brain injury or lesion.

marginal notes are simply extracts of the original text as selected by the summarization system; they do not include additional information. An automatic text-to-speech system provides machine-read versions of the text in either English or Spanish.

Lal and Ruger's summarization system [58], described in Section 7.4, also includes a simplification module intended to reduce the complexity of the summarized text for target audiences such as schoolchildren, who may have limited background knowledge or reading ability. The system performs lexical simplification by finding synonyms in WordNet for "difficult" words (selected based on syllable count and corpus frequency). Results of this part of the system show some promise but have problems due to word sense ambiguity and synonym choices that do not sound right in the target context. The system also attempts to provide background information for named entities by querying a biographical database for names of people and an image search engine for names of places. The authors report that this part of the system had problems due to multiple results for the same query with no disambiguation mechanism.

7.3 NLP Approaches to Text Simplification

Other NLP research efforts are aimed at simplifying text to improve further processing by other NLP tools, and not necessarily for the benefit of human readers. Chandrasekar and Srinivas developed a system that simplifies sentences with the goal of improving the performance of other NLP tools such as parsers and machine translation systems [17, 18]. Their approach consists of analysis followed by transformation. The analysis phase uses a dependency-based approach, not full parsing, using punctuation and Supertags, which are based on n-gram statistics rather than parses. The system learns transformation rules from an annotated and aligned corpus of complex sentences with manual simplifications. Sentences are processed one at a time and each sentence is "maximally simplified" by the system. Our analysis and proposed approach, described in Chapters 8 and 9, borrows some of the same underlying assumptions, including learning from an aligned corpus, but our goal is to provide more readable text for people, not for further automatic processing. We start with a corpus of paired articles in which each original sentence does not necessarily have a corresponding simplified sentence.

Like Chandrasekar and Srinivas's approach, Siddharthan's work on syntactic simplification and text cohesion has the goal of making individual sentences simpler but not to make the text itself shorter. His approach uses a three-stage pipelined architecture for text simplification: analysis, transformation and regeneration, paying particular attention to discourse-level issues (i.e., interaction between sentences in a text, not just individual sentences) since these issues have not been addressed in previous work [83, 85, 84]. In this approach, POS-tagging and noun chunking are used in the analysis stage, followed by pattern-matching for known templates which can be simplified. The transformation stage sequentially applies handcrafted rules for simplifying conjunctions, relative clauses and appositives. The regeneration stage of the system fixes mistakes introduced by the previous phase by generating referring expressions, selecting determiners, and generally preserving discourse structure with the goal of improving cohesiveness of the resulting text. Siddharthan's system is designed and tuned for British English news text, but we expect that the overall approach will still apply to our corpus of American English text. We will evaluate its effectiveness in Chapter 9.

Klebanov, Knight and Marcu developed a system to simplify text for information-seeking systems, e.g., information retrieval or question answering applications [54]. Unlike the work described above, the goal is not to produce a coherent text that is pleasant for humans to read. Instead, the goal is to produce as many "Easy Access Sentences" as possible. An Easy Access Sentence based on a text T has exactly one finite verb, does not make any claims not present in the original text T, and contains as many named entities as possible, i.e., pronouns are replaced with their referents. This system produces extremely choppy, repetitive texts which are ideal for information retrieval systems but are not suited to our target audience of student readers.

7.4 Related Summarization Techniques

Most simplified texts are shorter than the original text. Applying automatic summarization techniques is not a complete solution to simplification since a summarizer could choose long, difficult sentences, resulting in a shorter text that is still too challenging. However, summarization is an obvious step to consider in the simplification process. In this section we focus on extractive summarization, in which the system selects sentences or parts of sentences to form a summary. Other approaches to summarization merge and reorder sentences (especially if multiple documents are used to form a single summary) but these additional challenges are less relevant to the task of simplification.

In Chapter 9, we evaluate the performance of Marcu's extractive summarizer [64] as one possible step in a simplification system. The ATA project uses a similar summarizer to extract marginal notes [9]. This system uses rhetorical parsing to include discourse features in the summarization process in order to achieve improved results. Position features indicating the location of each sentence in the document and within the paragraph containing it are also used.

Barzilay and Lee use HMM-based content models to summarize a document by tagging sentences with topic-model states and finding the probability that a given state generates sentences that belong in the summary [3]. This particular approach relies on a collection of documents on the same topic, e.g., news reports about earthquakes, to build the topic model, so it is not directly applicable to our task. However, they compare their results to a version of the summarization system developed by Kupiec *et al.* which uses topic-independent features in a Bayesian classification function [57]. Features of this model include paragraph position features, an uppercase word indicator feature, and fixed-phrase features. The phrase features indicate certain phrases such as "in conclusion", which are likely indicators of sentences which should appear in a summary in their target domain, creation of abstracts for journal papers.

Other summarization systems follow Kupiec *et al.*'s general approach, including Lal and Ruger [58], who use features for sentence length, position of the sentence within its paragraph and the paragraph within the document, and XML labels from the Text Retrieval Conference (TREC) corpus which include tags such as "headline" and "lead paragraph." Sekine and Nobata also use a similar approach, including position-based features, named entity features, and domain-specific pattern features [80].

Hori, Furui *et al.*'s work on speech summarization employs a two-stage summarization method consisting of important sentence extraction and sentence compaction [51, 43, 20]. The sentence extraction phase combines a linguistic score based on n-gram probability, a significance score comparing word frequencies in the sentence to the entire corpus, and a confidence score from the speech recognizer. The sentence compaction phase combined the above three scores with an additional word concatenation score [51]. Early work on this project compacted all sentences and then selected some [43]. Further research added topic and stylistic adaptation by interpolating multiple language models to calculate the linguistic score [20]. While this approach addresses additional challenges due to speech recognition errors, the overall strategy of extracting and compacting sentences as separate processes would also serve the needs of simplification, based on our empirical observations described in Chapter 8 that both dropping and shortening of sentences occurs in simplified text.

7.5 Related Sentence Compression Techniques

Like summarization, sentence compression alone is not sufficient for simplification, but it is an obvious component of a simplification system. There are two main approaches in recent work on sentence compression. The first is constituent-based compression, in which parse tree constituents are selected for deletion. Knight and Marcu's decision-based model is an example of this type of compression; they use tree re-writing rules to shorten sentences [55]. This model is trained and tested on pairs of original and compressed sentences from the Ziff-Davis corpus, a collection of newspaper articles about computer products. The second approach to sentence compression is word-based, where a decision is made about whether or not to drop each word in the sentence. Furui *et al.*'s summarization system described above uses this style of compression [43].

Clarke and Lapata conducted a study of the constituent-based and word-based sentence compression systems just described on the Ziff-Davis corpus and on a new Broadcast News corpus which they developed [23]. Their corpus consists of 1370 sentences (50 stories) from Broadcast News transcripts which were manually compressed by three annotators. In their experiments, they found that the constituent-based compression approach was more sensitive to the style of the training data. Although Broadcast News is a speech corpus, much of it is planned speech read by newscasters, and the transcripts selected by Clarke and Lapata are quite clean. This corpus includes news stories on a variety of topics, making it more relevant to our simplification task. Their sentence compression system based on this corpus is not currently publicly available, but in the future it or a similar tool could be used as one stage of a simplification system.

Chapter 8
TEXT SIMPLIFICATION CORPUS AND ANALYSIS

When creating simplified or abridged texts, authors may drop sentences or phrases, split long sentences into multiple sentences, modify vocabulary, shorten long descriptive phrases, etc. Here, we do not address changes to vocabulary; Burstein *et al.*'s approach to choosing synonyms for challenging words could be used to simplify vocabulary items [8]. Instead, this chapter presents an analysis of a corpus of original and manually simplified news articles with the goal of gaining insight into what people most often do to simplify text in order to develop better automatic tools. We focus on the following research questions:

- What differences in part-of-speech usage and phrase types are found in original and simplified sentences?
- What are the characteristics of sentences which are dropped when an article is simplified?
- What are the characteristics of sentences which are split when an article is simplified?

We are interested in a data-driven approach to simplification like that of Chandrasekar and Srinivas [18]. However, unlike their work, which is based on a corpus of original and manually simplified *sentences*, we study a corpus of paired *articles* in which each original sentence does not necessarily have a corresponding simplified sentence. Our corpus makes it possible to learn where rewriters have dropped as well as simplified sentences. These findings should help with the development of simplification tools that can be trained from text in a variety of domains.

In the following sections, we describe the sentence-aligned corpus used here and an analysis of differences between the original and abridged articles. We conclude with a summary of our findings and suggestions for future work.

Table 8.1: Total sentences and words and average sentence length for corpus of 104 pairs of original and abridged articles.

	Original	**Abridged**
Total sentences	2539	2459
Total words	41982	29584
Average sentence length (words)	16.5	12.0

8.1 Aligned Corpus of News Articles

This work is based on a corpus of 104 original news articles with corresponding abridged versions developed by Literacyworks as part of an literacy website for learners and instructors [100]. The target audience for these articles is adult literacy learners (i.e., native speakers with poor reading skills), but the site creators say that they believe that the abridged articles can be used by instructors and learners of all ages. In this thesis, we will refer to “original” and “abridged” texts, the terms used by the authors of this corpus. This section presents characteristics of the corpus, including features of sentences in the original and abridged articles, and manual alignment of original to abridged sentences.

8.1.1 Overall Corpus Statistics

Table 8.1 lists the total number of sentences and words and the average sentence length in words for the original and abridged portions of the corpus. There are nearly as many abridged sentence as original sentences, but there are 30% fewer words in the set of abridged articles, and the average sentence length is 27% shorter in this set.

To explore other differences between the original and abridged sentences we used an automatic parser [19] to get parses and part-of-speech tags for all sentences. Table 8.2 shows, for selected POS tags, the average number of that tag in the original and abridged sentences,

Table 8.2: Average frequency of selected part-of-speech tags in original and abridged sentences. On average, abridged sentences contain 27% fewer words than original sentences.

Tag	**Original**	**Abridged**	**Difference**
Adjective	1.2	0.8	33%
Adverb	1.0	0.6	40%
CC	0.5	0.3	40%
Determiner	1.9	1.4	26%
IN	1.8	1.3	28%
Noun	3.6	2.8	22%
Proper Noun	1.4	1.0	28%
Pronoun	1.2	0.8	33%
Verb	2.1	1.6	24%

Table 8.3: Average frequency and length in words of selected phrases in original and abridged sentences.

Phrase tag	**Avg phrases per sentence**		**Avg words per phrase**	
	Original	**Abridged**	**Original**	**Abridged**
S	2.6	2.0	13.7	10.8
SBAR	0.8	0.5	11.3	8.5
NP	6.3	4.5	3.4	2.8
VP	3.6	2.8	9.3	7.2
PP	1.7	1.2	5.3	4.3

and the percent difference between the two.[1] Since abridged sentences are on average 27% shorter, we expect fewer words and therefore fewer POS tags per sentence. However, we note that the percentage decrease in average frequency is greater for adjectives, adverbs and coordinating conjunctions, i.e., abridged sentences have fewer of these words. The percent decrease in nouns is only 22%, compared with 33% for pronouns, indicating that nouns are deleted less often than the average and it is unlikely that nouns are often replaced with pronouns. The frequency difference for determiners, IN (prepositions and subordinating conjunctions), proper nouns, and verbs is near 27%, indicating no difference other than that expected for the shorter abridged sentences. Table 8.3 shows average frequencies and average lengths of selected types of phrases. There are fewer phrases per sentence in the abridges sentences, and the phrases are shorter.

8.1.2 Alignment Methodology

In order to understand the techniques the authors used when editing each original article to create the abridged article, we hand-align the sentences in each pair of articles. This alignment was done by a native English speaker using the instructions used by Barzilay and Elhadad in their work on alignment of comparable corpora [2]. These instructions direct the annotator to mark sentences in the original and abridged versions which convey the same information in at least one clause. Since nearly all abridged sentences have a corresponding original sentence but some original sentences are dropped, we asked the annotator to align all the sentences in each abridged article to a corresponding sentence or sentences in the original file and automatically reversed the hand alignments to get the alignments of original to abridged sentences.

[1]The parser outputs the standard Penn Treebank tag set. For the purposes of this analysis, we aggregate tags for adjectives, adverbs, determiners, nouns, proper nouns, pronouns and verbs. For example, our adjective category includes JJ, JJR and JJS.

8.1.3 Original and Aligned Sentences

Table 8.4 shows the distribution of original sentences into categories based on the alignment described above.[2] Sentences can be dropped (no corresponding abridged sentence) or aligned to one or more abridged sentences. For sentences which are aligned to exactly one other sentence, we calculate whether the abridged sentence is more than 20% shorter or longer, or approximately the same length as the original sentence. A sentence which is aligned to more than one abridged sentence is hypothesized to be split. Likewise, sentences which are aligned to a single shorter sentence are hypothesized to be split with one part dropped. Note that the average sentence length in these categories is longer than the other categories. We will further investigate the alignment of these hypothesized split sentences in the next subsection.

The last two categories in the table are rare and specific to the abilities of human authors. An original sentence aligned to a longer abridged sentence indicates that the author added some material, perhaps to explain a difficult point. The average length of these original sentences is shorter than the other categories. In the case of merged sentences, two sentences are aligned to one abridged sentence. Both of these cases are much more difficult to handle, but relatively infrequent, so we will focus the remainder of our analysis on the other categories.

Table 8.5 shows the distribution of sentences in the aligned part of the corpus. Nearly half of the aligned sentences are in one-to-one alignment with original sentences. Of the remaining sentences, 43% are aligned to an original sentence that is aligned to more than one abridged sentence, i.e., a sentence which is possibly split. The remaining categories are small: merged sentences as described above, and unaligned sentences which probably contain new information added by the author who did the adaptation.

[2]Since multiple sentences in an article could be considered to contain the same information per the alignment instructions, there are infrequent overlaps between categories which cause the percentages to add up to slightly more than 100%, e.g., a sentence could be both split and merged.

Table 8.4: Alignment of original to abridged sentences, showing the number of original sentences and average length of original sentences in words for each category.

Category	**Number of Sentences**	**Avg length**
Total	2539 (100%)	16.5
1 to 0 (dropped)	763 (30%)	14.1
1 to >=2 (split)	470 (19%)	24.6
1 to 1 (total)	1188 (47%)	15.8
1 to 1 (shorter abr.)	350 (14%)	21.0
1 to 1 (same length abr.)	725 (29%)	14.4
1 to 1 (longer abr.)	113 (4%)	9.1
2 to 1 (merged)	167 (7%)	14.6

Table 8.5: Alignment of abridged sentences showing where they come from.

Category	**Number (percent)**
Total	2459 (100%)
1 to 1	1188 (48%)
1 to >=2 (original sentence split)	1066 (43%)
2 to 1 (original sentences merged)	79 (3%)
Unaligned	128 (5%)

Table 8.6: Examples of “split”, “edited”, and “different” aligned sentences. The split point in the first sentence is indicated by ***.

Split	
Original	Keith Johnson is the Makah Tribe Spokesman, *** and he comments, ”We made history today.
Abridged	Keith Johnson is the Makah Tribe Spokesman. He said, ”We made history today.
Edited	
Original	Congress gave Yosemite the money to repair damage from the 1997 flood.
Abridged	Congress gave the money after the 1997 Flood.
Different	
Original	The park service says the solution is money.
Abridged	Why hasn’t the National Park Service kept up the park repairs? There is a lack of money.

Table 8.7: Distribution of hypothesized split sentences.

Category	**Number of Sentences**	
	One to Many	**One to One**
Total	470 (100%)	350 (100%)
True split	368 (78%)	202 (58%)
Edited	17 (4%)	145 (41%)
Different	85 (18%)	3 (1%)

8.1.4 Annotating True Split Sentences

Nearly 20% of the original sentences are aligned to more than one abridged sentence. It is reasonable to expect that in most cases, these are long sentences containing multiple ideas which the author chose to split into shorter sentences. We also hypothesize that sentences which are aligned to shorter abridged sentences could also be split, with one part dropped. To validate these assumptions, we asked our annotator to mark split points in these sentences corresponding to their alignment to the previously-aligned abridged sentences. Some sentences did not have split points and were categorized as either "edited", indicating that the sentence has minor changes but no obvious split point, or "different," indicating that while the sentences convey the same information, they are lexically very different. Table 8.6 shows examples of these three categories. Table 8.7 shows the distribution of original sentences in these three categories for the hypothesized one-to-many and one-to-one splits. In the one-to-many case, 78% are true splits, with 14% split into more than two pieces, with one dropped. In the one-to-one case, all true splits consist of a piece that is kept and a piece that is dropped. Of the sentences that are not true splits, most are different in the one-to-many case, and edited in the one-to-one case. This is not surprising, since editing one sentence to make a new slightly shorter sentence seems more plausible than changing one sentence into two new sentences without an obvious split point.

In the following section we will analyze the differences between the "true split" sentences identified in this section and other sentences in the corpus.

8.2 Analysis of Split vs. Unsplit Sentences

A step in automatic simplification is choosing sentences to split. We expect that long sentences would be selected for splitting, but other characteristics are likely to be considered, too. In this section we analyze the 570 sentences identified as "true splits" compared to 1205 unsplit sentences. For the purpose of this analysis, the "dropped" sentences are not included in the unsplit category, since some of them might have characteristics of sentences that should be split if they were kept. Other sentence categories from Table 8.4 and the edited and different sentences from the hypothesized split sentences are considered unsplit. As expected, split sentences are longer, with an average sentence length of 23.3 words compared to 15.0 words for unsplit sentences. The average number of phrases identified by the parser (S, NP, etc.) and the length of these phrases is also longer on average for split sentences.

In addition to length, we hypothesize that the decision to split a sentence is based on syntactic features, since splitting a sentence reduces the syntactic complexity of the resulting sentences while retaining the same information. To investigate which features are most important for splitting sentences, we used the C4.5 decision tree learner [73] to build a classifier for split and unsplit sentences. We chose C4.5 and its rule generator because the results are easily interpreted, and the focus of this chapter is on analysis rather than classification. We use the following features for each sentence:

- Sentence length in words.
- POS: number of adjectives, adverbs, CC, IN, determiners, nouns, proper nouns, pronouns, and verbs.
- Phrase: number and average length of S, SBAR, NP, PP, and VP.

The POS tags are aggregated as described in Section 8.1.1 and the phrase types are chosen because they show differences in average frequency and length in the original and abridged sentences.

Using a pruning confidence level chosen by 10-fold cross-validation, we trained a tree and its accompanying rules on the entire split vs. unsplit dataset.[3] As expected, length is the most important feature in the tree and results in the two rules with highest usage: sentences with length less than 19 tend not to be split, and sentences with length more than 24 tend to be split. When evaluated on the training set, these rules apply to 907 and 329 sentences, respectively, with error rates of 17% and 33%. The next most frequently

[3]Average error rate in the cross-validation experiments is 29%; these classes are not easily separable.

applied rule for unsplit sentences uses features for length < 24 and NP average length $<= 1.4$, i.e., moderately long sentences with short noun phrases, perhaps indicating that longer noun phrases are something that would be split. Commonly used features in other rules for split sentences include the number of nouns, pronouns, verbs, determiners and VPs. Surprisingly, the number and average length of S and SBAR constituents were not commonly used features.

8.3 Analysis of Dropped Sentences

Most abridged or simplified texts are shorter than the original text, so another step to consider is choosing sentences to omit. In this section, we compare the dropped sentences with the rest of the original sentences. As in the previous section, we use the C4.5 rule generator to see which features are most important.

In the case of dropped sentences, the rationale for picking sentences to drop is more likely to be content-based than syntactic, thus we consider a different set of features in this section. Intuitively, we expect that sentences which are somewhat redundant may be dropped. However, it is also possible that some repetition is a good thing since familiar content may be easier to read. To explore these options, we use features for the percentage of the content words in a sentence that have already occurred in previous sentences in the document. Position in the original document may also be a factor. The complete set of features for each sentence is:

- Position in the document: sentence number, percentage.
- Paragraph number, first or last sentence in paragraph.
- Does this sentence contain a direct quotation?
- Percentage of words that are stop words.[4]

[4]We use the list of 612 stop words provided with WordNet 1.6 [35].

- Percentage of content words which have already occurred one/two/three/four/more times in the document.

As in the previous section, we chose a pruning threshold using cross-validation and trained a classifier on the entire dataset. The majority class is "not dropped", and this class is identified by the rule with highest applicability (895 sentences) and lowest error rate (15%):

> position $\leq$ 12, stop words $\leq$ 70%, content words seen once $\leq$ 40%, no content words seen $>$ 5 times

This rule indicates that sentences early in the document with low redundancy are not dropped. Rules for dropped sentences have lower applicability and higher error rates. The quote feature is used several times; we have observed that quotes are often removed from the abridged documents in this corpus, and the rules indicate that this is particularly true if the quote consists of more than 70% stop words or is past the 12th sentence in the document. An example of a sentence with mostly stop words is "The judge says, 'I am going to go through a few forms' ". In this sentence, "judge" and "forms" are the only non-stop words according to the Wordnet list of 612 stop words. Another rule for dropped sentences is that later sentences (position $>$ 35) are dropped; this rule applies to to 91 sentences with 33% error. The redundancy features are also used, though these rules have higher error rates.

8.3.1 Dropped Phrases

In addition to dropping entire sentences, the authors of the abridged articles sometimes split a sentence and dropped part of it, as introduced in Section 8.1.4. Table 8.8 shows how many phrases in the original sentences which were categorized as "true split" sentences in Table 8.7 were aligned to abridged sentences and how many were dropped. In this analysis, "phrases" are defined by the sentence boundaries and the manually annotated split points; they are not required to have any particular syntactic characteristics. Most phrases in the one-to-many split sentences were retained, which is not surprising since these original

Table 8.8: Distribution of phrases in sentences with true splits. "Phrases" are defined by manually annotated split points.

Category	**Number of Phrases**	
	One to Many	**One to One**
Total	901 (100%)	451 (100%)
Aligned	827 (92%)	230 (51%)
Dropped	74 (8%)	221 (49%)

sentences are aligned to more than one abridged sentence, each of which gets one phrase. As expected, about half the phrases in the one-to-one split sentences are kept and half are dropped.

This is a smaller set of data than the dropped vs. not-dropped sentences studied earlier in this section, and many of the phrases are quite short. However, we expect that these phrases may still share characteristics with the sentences that are and are not dropped. In fact, applying the decision tree classifier trained above for dropped sentences to this phrase data set yields an error rate of 27%, which is actually lower than the average error rate of 30% achieved by in 10-fold cross-validation training for the dropped sentence classifier. This may indicate that dropped phrases are actually easier to identify than dropped sentences, although direct comparison is not possible since the training set is not exactly the same for the full classifier vs. the classifiers trained by cross-validation. It is encouraging to see that the dropped and not dropped categories share characteristics at both the sentence and phrase level.

8.4 Summary

We have described a corpus of original and abridged news articles, observing that the simplified articles contain fewer adjectives, adverbs, coordinating conjunctions and pronouns, but more nouns. These sentences also contain fewer and shorter phrases such as NPs and

VPs. Our analysis of sentences which are split or dropped by the authors of the abridged articles shows the importance of syntactic features in addition to sentence length for decisions about sentence splitting, and the use of position and redundancy information in decisions about which sentences to keep and which to drop. These features may also be valuable for determining which phrases to keep or drop when sentences are split. These insights will be useful in work on automating the simplification process, but may also be useful for improving reading level detection.

Chapter 9
TEXT SIMPLIFICATION FOR EDUCATION

Based on the review of prior work in Chapter 7 and data analysis in Chapter 8, we decompose the problem of simplification into four component problems: sentence selection, sentence splitting, sentence compression, and lexical replacement. The decoupling of these processes is useful for understanding their role in simplification, and for leveraging existing text processing tools. It is of particular interest to explore general purpose tools for simplification of text for language learners since the target task is unconstrained in domain (teachers are interested in a broad range of topics for their students), and parallel training data with original and simplified articles is very limited.

Sentence selection is a key, and often only, component of summarization systems, and much research addresses this problem. Therefore, it is interesting to assess the effect of sentence selection alone in the simplification task, since the goal here is somewhat different from summarization. Summarization systems aim to extract the most important information from an article but may select longer, more complicated sentences because those sentences convey more information. Most research on summarization focuses on relatively high compression rates, e.g., creating abstracts from journal articles [57]. In contrast, the simplified articles in the Literacyworks corpus analyzed in Chapter 8 are, on average, 70% as long as the original articles. Differences aside, the similarities between summarization and simplification make it possible to use summarization evaluation tools for assessing simplification, especially sentence selection. We will take advantage of this in our experiments in Section 9.3.2.

In Chapter 8 we observed that approximately 22% of sentences in the original articles are split, including sentences which have a part dropped after being split. Previous work on sentence splitting includes Siddharthan's syntactic simplification system [85] described in Chapter 7. We will test the effectiveness of this tool for our task, and also investigate a sentence splitting classifier trained on the Literacyworks corpus.

Sentence compression is a separate area of research, which is also increasingly being leveraged in summarization work. However, most previous work tends to focus on a single domain, e.g., Barzilay and Lee's content-model summarizer [3] is designed for constrained topics such as news articles about earthquakes, and Knight and Marcu's sentence compression system [55] is trained on a corpus of news articles announcing computer products. Hence, existing software and associated corpora for learning are not well suited to the language learning context, and we do not explore sentence compression in this work. Similarly, though there has been work on replacing lexical items with synonyms and pronouns with nouns, e.g., [58], we are not aware of available software tools or corpora that would be applicable to our target domain. For these reasons, in this chapter, we explore only sentence selection and splitting.

In Section 9.1, we describe the corpora used in this chapter. Sections 9.2 and 9.3 present approaches and experimental results for sentence selection and sentence splitting, respectively. Section 9.4 summarizes the chapter and discusses opportunities for future work.

9.1 Corpora

In this chapter, we use the hand-aligned Literacyworks corpus to train models for sentence selection and splitting. For evaluation of these models and existing language processing tools, we use a development set selected from the Literacyworks corpus and two test sets selected from two sources on the web. The test sets are small due to the time expense of human evaluation, but we believe they are representative of our target task. Evaluation of automatic simplification is similar to the closely related summarization and compression tasks and typically involves comparison to a handwritten gold standard or human judgment of the quality of the results; we use both of these methods in this chapter.

We selected five articles from the Literacyworks corpus to use as a development set and

Table 9.1: Average length of articles in sentences and words and average sentence length in words for the original and abridged (simplified) articles in the development set.

Development Set	**Avg Num Sents**	**Avg Num Words**	**Avg Sent Length**
Original	22	384	17.8
Abridged	21	277	13.1

used the rest of the corpus as a training set for the experiments in this chapter. The articles were selected based on being representative of the corpus in terms of the average number of sentences dropped and split. We selected articles with average sentence lengths that were slightly higher than the average for the whole corpus with the intent of providing more room for simplification to occur. The development set allows us to compare the automatically simplified articles with the abridged versions written by the authors of the Literacyworks corpus. While there is only one abridged version per article in this set, the split sentences are explicitly indicated so this set is useful for testing the sentence splitting module.

We created two additional test sets. The first consists of five articles from the online Principles of Aeronautics textbook written by Cislunar Aerospace, Inc. [22]. This online textbook contains articles about many topics related to flight and aerodynamics which are written in several versions for students between kindergarten and 8th grade. We selected articles from the "Advanced" version; the exact grade level is not specified, but it is probably around 7th to 8th grade level. These articles contain sentences that are on average the same length or a little shorter than the average sentence length of the original articles in the Literacyworks corpus, although many of the articles are longer. This test set will allow us to see how these tools perform on articles in a different topic domain.

The second test set is from Wikinews [101], the current events portion of the Wikipedia website. We selected articles of about the same length as the Literacyworks articles, and noted that the average sentence length is much longer for this corpus. Although these articles are in the same news domain as the Literacyworks corpus, the longer average sentence length will provide a different challenge for simplification. Table 9.1 shows the average article length and average sentence length for the development and test sets.

For the Aeronautics and Wikinews articles, we asked three teachers of ESL or bilingual education[1] to create a simplified version of each article, resulting in three simplified reference versions of each test set article. (Although lower-level versions of the Aeronautics articles exist, they are substantially different from the test set articles rather than directly simplified versions, and are not suitable references for this task.) The teachers were provided with three examples of original and simplified articles from the Literacyworks training set, and the instructions to simplify the test articles to approximately the same level as the simplified Literacyworks articles. (There is no grade level provided for that corpus, so we did not specify a specific target level to the teachers, either.) The teachers were told that suggested simplifications observed in the training data included dropping sentences, splitting sentences, and rephrasing, but we also urged them to use their own judgment and professional experience.

Table 9.2 shows the average size of the resulting articles. There is variation between results for the individual teachers, although the average sentence length for the Aeronautics articles and the average number of sentences in the Wikinews articles are quite consistent for all three people. For comparison, the average length in words of the abridged articles in the development set is 72% of the length of the original articles, and the average sentence length is 74% of the original. In the Aeronautics test set, the average word length of the simplified versions ranges from 60-98% of the original, and the sentences average 85-86% as long as the sentences in the original articles. For the simplified Wikinews articles, the average word length is 53-85% of that of the original articles, and the sentences are 48-69% as long as the original sentences.

[1]Two of these teachers were also annotators for the results in Section 6.4; the third was unavailable and was replaced by another individual with related teaching experience.

Table 9.2: Average length of articles in sentences and words and average sentence length in words for the original test set articles and the hand-simplified versions by three authors.

Author/Set	**Avg Num Sents**	**Avg Num Words**	**Avg Sent Length**
Original			
Aeronautics	59	877	14.9
Wikinews	18	426	24.2
A			
Aeronautics	62.6	860	12.8
Wikinews	21.8	362	16.6
B			
Aeronautics	51	641	12.6
Wikinews	20	296	14.8
C			
Aeronautics	41.6	523	12.6
Wikinews	19.4	227	11.7

9.2 Sentence Selection

This step is similar to extractive summarization, although simplification does not require the level of compression that is often expected for summarization tasks. In Section 9.2.1, we explore the development of a classifier for sentence selection which is tuned to the simplification task. However, the results to date are not successful, so we only conduct limited evaluation of this method. In Section 9.2.2 we evaluate an existing extractive summarizer [64] applied to this task.

9.2.1 Decision Tree Selection

For our sentence-selection classifier, we use an expanded version of the features used in the analysis of dropped sentences in Chapter 8, many of which are typically used in extractive summarization systems. The features can be viewed as belonging to three general categories: position, quotation, and content/redundancy. The complete list of features is:

- Position of the sentence in the document: sentence number, percentage.
- Position of the enclosing paragraph in the document: paragraph number, percentage.
- Position of the sentence in its enclosing paragraph: percentage, flags for first and last sentence.
- Does this sentence contain a direct quotation?
- Percentage of words that are stop words.[2]
- Percentage of content words which have already occurred one/two/three/four/more times in the document.

[2]We use the list of 612 stop words provided with WordNet 1.6 [35].

Position-based features are used in all the text summarization systems described in Chapter 7. We use features for absolute and relative (percentage) sentence and paragraph position in the document, and absolute and relative position of each sentence in its paragraph. However, we observe that articles in this corpus contain many very short paragraphs, so the features based on position within paragraphs may not be as informative for this particular corpus.

The quotation feature is not commonly used by other systems but it appears useful for this corpus, and we believe it may be useful in general for this application. The content/redundancy features serve a similar purpose to the tf-idf features used by Sekine and Nobata [80], but our features provide information about content at the level of a single document instead of requiring a corpus of related documents. Other features commonly used in extractive summarization but which are not relevant to our unconstrained domain are related to titles and headlines [64, 58] or domain-specific fixed phrases such as "in this letter" [57].

Using the IND decision tree package [7], we train a classifier with these features based on the dropped and not dropped sentences in the annotated corpus. We use IND instead of C4.5 as in the previous chapter because this package provides probability estimates for its classifications. Due to the imbalanced nature of the training set (734 sentences dropped vs. 1696 not dropped) we use bagging: 50 iterations of random sampling were conducted, choosing 600 examples of each class for each iteration. For each iteration, decision trees are trained using 10-fold cross-validation and cost-complexity pruning. Finally, to apply the classifier to a new feature vector x for a sentence, all B trees T_i are applied to the vector and the resulting posterior probabilities are averaged and then normalized to account for the resampling, i.e.,

$$p(drop|x) = \frac{q_d}{0.5B} \sum_{i=1}^{B} p_i(drop|T_i(x))$$

where here $B = 50$ and $q_d = 0.3$ is the prior probability of the "dropped" class.

Applying the classifier to the development set yields performance that is no better than chance, so we do not pursue further evaluation of this classifier on the test set. We will compare the development set results with the summarizer results in the next section.

Manual inspection of some of the decision trees indicates that all of the feature types are used, with the "quote" feature often occurring near the top of the tree. Both the position and redundancy feature categories appear important, and it is not clear that either category is consistently used before the other.

The poor performance of this classifier may be due to the fact that authors make a variety of choices when simplifying text. The choice to drop sentences can be based on novelty or redundancy of content, location, or perhaps other factors such as the author's opinion of the appropriateness of a particular piece of information for the audience. The same features may not be important for all sentences. There may also be relationships between decisions made about sentences in a document, so a system that does not consider each sentence in isolation could be more successful. Our redundancy features do take into account content relationships between sentences in the original document, but we do not consider the impact of dropping (or not dropping) an earlier sentence on decisions about later sentences. Also, our classifier does not incorporate discourse cues, a feature used in the summarizer tested in the next section.

9.2.2 Extractive Summarization

In the absence of a classifier tuned specifically for the simplification task, we turn to existing extractive summarization techniques. Here, we evaluate the performance of Marcu's extractive summarizer [64], which uses discourse and position features to create a summary by extracting sentences and/or partial sentences. The length of the summary is a specified percentage of the length of the original.

We compare the results of Marcu's discourse-based extractive summarizer [64] and the standard "lead baseline" often used in summarization evaluation. The lead baseline consists of a selected percentage of text from the beginning of the article. Particularly for news stories, this baseline often performs well. This is also a reasonable baseline for the text simplification task, since a student reading an article that is too long or too difficult is likely to start at the beginning and just quit somewhere before the end. For both the baseline and the summarizer, a summarization percentage of 70% is chosen, based on the average difference in article length between the original and abridged articles in the Literacyworks corpus. This percentage is higher than is typical for many summarization tasks.

The existence of the handwritten reference articles allow us to use the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) summarization evaluation tool [62] to compare the handwritten references to the baseline and summarizer output. ROUGE measures similarity between a candidate text and one or more reference texts. There are several variations of ROUGE; we use two that show good correlation with human judgment for moderate-length single document summaries, which is the closest to our task. ROUGE-N is based on n-gram co-occurrence, counting the number of n-grams of a specified order in each reference that match the candidate text to be evaluated, divided by the total number of n-grams in the references. ROUGE-2 (bigram co-occurrence) performs the best in [62]; we show results for ROUGE-1, ROUGE-2 and ROUGE-3. ROUGE-L is similar to ROUGE-N, but counts the longest common subsequence (LCS) for each reference sentence compared to the sentences in the candidate. The standard ROUGE scored based on the specified co-occurrence statistic (i.e., n-grams or LCS) is recall, a number between 0 and 1, where larger numbers are better. The ROUGE tools also provide precision (dividing by the length of the candidate rather than the references), and F-measure, but for simplicity we report only recall here. (Recall is the measure most commonly used in summarization evaluation, and in our case the trends are the same for recall, precision, and F-measure.)

Table 9.3 shows ROUGE results for the baseline and summarizer conditions for the two test sets. The baseline slightly outperforms the summarizer. However, ROUGE also provides a 95% confidence interval, and scores for any baseline/summarizer pair are well within this window. (The width of the confidence intervals ranges from 0.05 to 0.13, and the largest absolute different in the table is 0.04.) Also, the lead baseline is known to be hard to beat in summarization, especially of news articles, so it is not too surprising that this baseline is also good for simplification.

Table 9.3: ROUGE-N and ROUGE-L recall scores for both test sets for baseline and extractive summarizer. Scores are calculated relative to 3 sets of handwritten references.

Measure	**Aeronautics**		**Wikinews**	
	Baseline	Extr. Summ.	Baseline	Extr. Summ.
ROUGE-1	0.68	0.65	0.65	0.64
ROUGE-2	0.51	0.48	0.48	0.46
ROUGE-3	0.43	0.39	0.41	0.38
ROUGE-L	0.67	0.64	0.65	0.63

Table 9.4: ROUGE-N and ROUGE-L recall scores for the Literacyworks development set for baseline, extractive summarizer, sentence selection decision tree, and oracle. Scores are calculated relative to one reference.

Measure	**Baseline**	**Extr. Summ.**	**Sent. Select**	**Oracle**
ROUGE-1	0.58	0.52	0.64	0.68
ROUGE-2	0.29	0.23	0.31	0.38
ROUGE-3	0.20	0.14	0.19	0.26
ROUGE-L	0.56	0.50	.62	0.66

Table 9.4 shows ROUGE results for the baseline, extractive summarizer, and sentence selection decision tree classifier from Section 9.2.1 run on the development set, compared to an oracle system which chooses all sentences not marked “dropped” in the annotated corpus. Again, the baseline outperforms the summarizer, but each set of scores is within the 95% confidence interval for the other system. The decision tree classifier outperforms both other systems except in the case of ROUGE-3, but again, the scores are all within the 95% confidence interval. In this case, for each article there is only a single reference, the abridged version from the Literacyworks corpus, so the confidence intervals are wider in this experiment and these results are only indicative of the overall trend. Also, the sentence selection classifier performance is near chance, so the other systems may also be near chance. The oracle outperforms the other systems by about 10%, but the oracle scores themselves are not very high. These results indicate that standard summarization techniques are not an unreasonable choice for this stage of the simplification process, but sentence selection alone is not enough.

9.3 Sentence Splitting

In Chapter 8, we analyzed the characteristics of sentences which are split vs. not split in the Literacyworks corpus. However, in addition to choosing sentences to split, an automatic simplification system needs to also choose *where* to split them and, in most cases, perform some rewriting of the resulting pieces. In the following sections we discuss potential features for choosing split points and evaluate Siddharthan’s syntactic simplification system [85] applied to this task.

9.3.1 Approach

Using the Literacyworks data for training, we viewed the boundary between each pair of words in a sentence as a potential split point. We considered features including POS tags and parser-generated phrase features (NP, VP, PP, etc.) for the words on either side of this potential split point, distance from either end of the sentence, and the presence of punctuation or quotations before or after the potential split point. While we believe that this approach could be successful for automatically choosing which sentences to split, some of these sentences are difficult to automatically rewrite. For the purpose of having higher quality output for this component of the system, we chose to split fewer, simpler cases for which the approach to rewriting is well understood. To do this, we use Siddharthan's syntactic simplification system [85], described in the previous chapter. This system is designed for news text, which is a good match for our target domain. The system chooses sentences to split based on a set of handwritten rules and uses a regeneration approach to rewrite the resulting sentences to make them semantically and grammatically correct. The results are not perfect, but syntactic simplification is a challenging task and this system represents the state of the art in this area.

9.3.2 Syntactic Simplification Experiments

We ran Siddharthan's syntactic simplifier on the original articles in the entire Literacyworks corpus and counted the number of sentences per article that the system split. We also counted the number that were marked as split by the human annotator. Figure 9.1 shows histograms of these counts. The authors of the abridged articles split somewhat more sentences than the system did, which is not surprising since the system only chooses splits based on a small set of specific rules while human authors have greater freedom to split and rewrite sentences.

For the development and test sets, the quality of the sentences that were split and regenerated by the syntactic simplification system was evaluated by the annotator who did the sentence alignment and splitting for the Literacyworks corpus. The annotator was asked to answer two questions about each split. First, is this choice of split point good? (Yes or no.) Second, are the resulting sentences grammatically correct? This question is answered on a 0-2 scale where 0 means incorrect grammar, 2 means correct, and 1 indicates grammar that is awkward but still comprehensible. Table 9.5 shows the results of this evaluation. The annotator indicated that most split points were chosen well, and most of the results

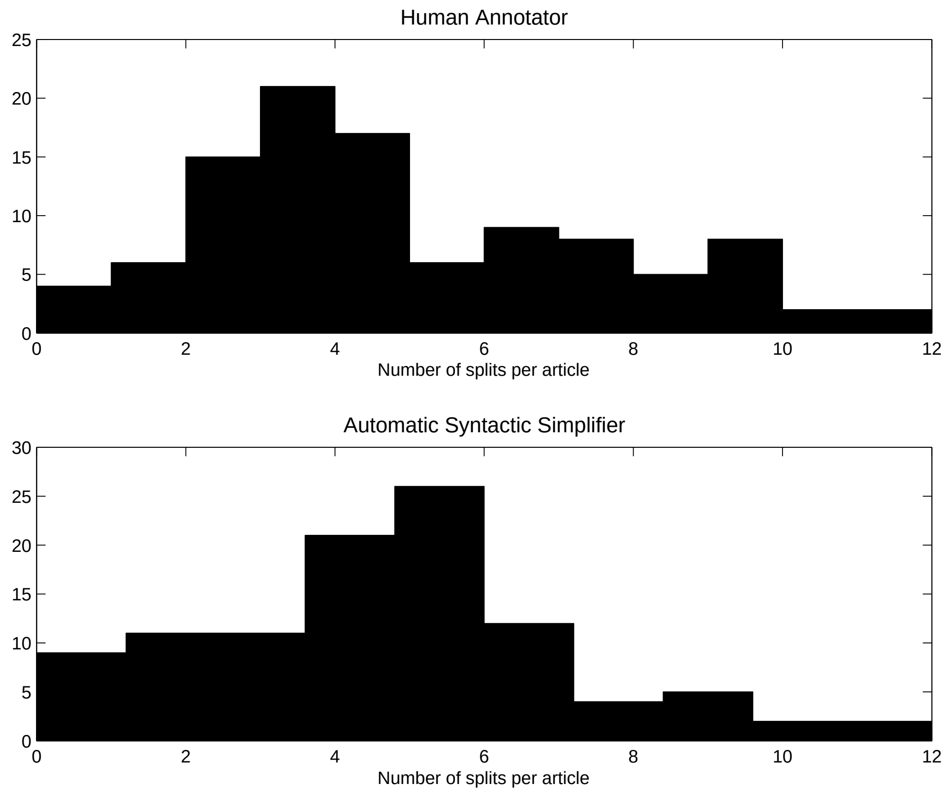

Figure 9.1: Histograms of the number of sentences split per article in the Literacyworks corpus by the automatic syntactic simplifier vs. by the authors of the abridged articles.

were reasonably grammatically correct.

We can also calculate precision and recall for the automatically split sentences relative to the manual alignments of original and abridged articles in the development set. This results in precision of 59% and recall of 37%. The precision result is encouraging, and the low recall is not surprising, since as previously noted, the automatic syntactic simplifier chooses splits according to a limited set of rules.

Table 9.5: Annotator evaluation of sentences split by the automatic syntactic simplifier showing the percentage of sentences from each test set in each category.

Category	**Development**	**Test (Aeronautics)**	**Test (Wikinews)**
Number of split sentences	17	37	31
% good split point	88%	84%	90%
Average grammar score	1.4	1.3	1.3

9.4 Summary

These results indicate that existing summarization and syntactic simplification systems can play a role in automatic simplification, but there is ample room for future work in this area. Obvious avenues for additional research include further study of the applicability of summarization or other classification techniques to the task of simplification, expanded sentence-splitting tools, sentence compression techniques tailored to this task, and lexical replacement tools. More challenging areas for future work could be based on more sophisticated simplification techniques employed by teachers. One of the teachers who simplified the test set articles provided a list of the types of changes she made, which included changing vocabulary, making implicit ideas more explicit, using more action verbs, repeating key concepts and sometimes changing the order of information presented [77]. The latter two changes are perhaps more particular to the educational context of text simplification. Changing the order of points in the text requires a higher-level view of the topic than the tools we explored in this chapter; repeating concepts also requires the ability to identify a small number of key ideas.

Chapter 10
SUMMARY AND FUTURE DIRECTIONS

The purpose of this dissertation was to apply and extend existing NLP technology to problems faced by teachers and students in the context of bilingual education, advancing the state of the art in the relevant NLP areas. We developed tools whose goal is to assist teachers in finding appropriate-level, topical texts for their students using new methods of reading level assessment. We explored characteristics of abridged text and assessed component tools for automatic text simplification. This chapter summarizes the specific contributions of this dissertation and presents opportunities for future work.

10.1 Contributions

The contributions of this dissertation in the area of reading level assessment include the development of reading level detectors for clean text and extending them to the more varied text found on the World Wide Web. We found that SVM classifiers combining traditional grade level features, n-gram language model scores, and parser-based features outperformed classifiers based on n-gram LMs alone. On the Weekly Reader corpus, our detectors significantly outperform the traditional Flesch-Kincaid and Lexile reading level measures, with F-measures of roughly 0.7 for grades 3-5 on data where human annotators agreed within one level of the reference. On the same data, the other methods have F-measures in the range of 0.25-0.45 depending on grade level. These results demonstrate the power of data-driven learning for this task. With the addition of adult-level news text as negative training examples, we reduce the number of false positives on articles at reading levels above grade 5, rejecting 90% of the 30 cases that we tested. This change does not seriously reduce performance on the test set for the target grade levels.

The web provides a much larger collection of text resources that could be useful to teachers and students, but the quality of the pages is mixed. In a pilot study, we found that the detectors trained on clean text were simply not able to handle the messiness of the web without additional processing. We augmented our tools to first filter out pages which lacked narrative text or consisted mostly of links, advertisements or other unwanted content, using heuristics and a Naive Bayes classifier. The Naive Bayes classifier was trained using several iterations of human-in-the-loop annotation and retraining, in the style of active learning or relevance feedback. This was a quick and effective way to increase the amount of labeled training data for this task which is simple for people but challenging for computers. These filtering steps reduced the amount of unwanted pages from more than 60% of the returned pages to less than 30%, on average.

Different annotators had different perceptions of the appropriateness of articles for a particular grade level, in part reflecting variability between individual students. We studied the applicability of existing SVM adaption techniques to this task, using annotations from each teacher to adapt the reading level detectors. We found that SVM adaptation techniques developed for non-text-based tasks with well-separable categories do not necessarily apply to this task, but we are optimistic that other classifier/adaptation combinations will provide better results in the future.

Using the Literacyworks corpus of news articles with manually abridged versions, we studied the characteristics of these texts to gain insight into what people typically do when performing this type of text adaptation. The sentences were hand-aligned to show how sentences in the original and abridged versions of each article relate to each other, and split points were marked where one original sentence mapped to two abridged sentences. The resulting corpus with these annotations is a contribution to the field of NLP research; it could be used for research on topics including summarization, simplification, and sentence alignment.

Our analysis of the characteristics of the aligned Literacyworks corpus showed that simplified articles contain fewer adjectives, adverbs, coordinating conjunctions and pronouns, but more nouns. These sentences also contain fewer and shorter phrases such as NPs and

VPs. Syntactic features as well as sentence length are important in determining which sentences should be split, and decisions about dropping sentences are based on position and redundancy features. These observations helped inform our approach to automatic simplification.

Based on what we learned in our corpus analysis, we explored components of text simplification system for the context of language learning. We suggest four steps: choose sentences to keep, split selected sentences, simplify those sentences (e.g., remove adjectives) as needed, and perform lexical replacement. In this dissertation, we focused on the first two steps and showed that summarization and syntactic simplification tools can be applied to this task, although there is room for improvement with tools tailored more specifically to the challenges of simplification.

10.2 Future Work

There are several possible avenues for future work based on this dissertation. In this section, we discuss four main directions: application to other languages, creation of a system for interactive use by teachers and students, adaptation, and further work on simplification.

We focused on developing tools for English, but it would be useful for teachers and students to find and/or simplify texts in other languages, too. Students learning a second language benefit from education in both their first language and the new language [24, 56]. Some schools do have bilingual education programs, but many schools lack the resources to provide instruction in more than one language. Seventy-three percent of LEP students in Washington receive little or no instruction in their native language, despite studies that show that students spend less time in LEP programs and have higher long-term academic achievement if they receive more instruction in their native language [93]. Extending our tools to apply to other languages could help address this problem by allowing teachers to more easily find supplemental material in students' native languages. Even if only one additional language were supported, this could have a large impact. Nationally, the majority (75%) of LEP students are Spanish speakers; about 61% of Washington's LEP students speak Spanish as their first language [10].

While our approach to reading level assessment is general and should apply to other languages, some resources are required. In addition to a labeled training corpus, a POS tagger and parser are necessary. These resources do exist for many languages, e.g., Schimd's TreeTagger is trainable for multiple languages, and parameter files for Spanish already exist [78]. Cowan and Collins have developed a Spanish parser [29], and parsers for other important world languages also exist, such as Bikel's parser for Chinese [4]. Languages other than English may also benefit from the use of additional features, e.g., to capture richer morphology.

Another area for future work is to extend and/or modify the tools developed in this dissertation to create a system that works in real time. Currently, the tools require off-line processing that takes too long for an interactive system. This is a challenge for other similar systems, e.g., REAP [6], which does off-line processing of web pages to build a large database of articles which can later be retrieved in real time. This approach is effective, but obviously limits query results to documents in the database. Given sufficient computer resources, such as in a commercial context, this would be a satisfactory approach. Another possibility would be to explore other features or other ways of extracting similar features which are less computationally expensive. For example, shallow parsing or chunking could be used instead of full parsing.

Development of a user interface and studies of how users interact with the system are other areas for future research. The reading level assessment module could conceivably be integrated with an already-familiar search engine user interface, but the simplification tool raises more questions. Would users want control over the level of simplification? Should they be able to view the original article side by side with the simplified version? These and other questions could be explored in hands-on studies with teachers and students.

A user interface that allows interactive use of the simplification tool would also provide the possibility for adaptation of the system. If the user could indicate that a simplified article was too hard or easy, or too long or short, this data would allow thresholds to be tuned or the sentence-selection and transformation components to be retrained based on the user's feedback. As discussed in Chapter 6, there is also opportunity for future research on adapting the reading level detectors to individual users, including exploring new adaptation techniques and other models of reading level. The metaclassifer configuration introduced in Section 5.6 could provide new opportunities for adaptation, since only the lower-dimensionality metaclassifier would need to be adapted.

Chapter 9 raises several possibilities for future work on automatic simplification, including further development of sentence selection, splitting, and compression tools tailored to this task. Additionally, two possible directions for lexical replacement are replacing words with more common synonyms or replacing pronouns with nouns, perhaps by leveraging existing research on anaphora resolution in the latter case. Teachers also employ more sophisticated simplification techniques, such as changing the order of topics or providing background information; automating these tasks would provide opportunities for research in natural language understanding and artificial intelligence.

BIBLIOGRAPHY

[1] C. Baker. *Foundations of Bilingual Education and Bilingualism.* Multilingual Matters, Ltd, Clevendon, UK, third edition, 2001.

[2] R. Barzilay and N. Elhadad. Sentence alignment for monolingual comparable corpora. In *Proc. of EMNLP*, pages 25–32, 2003.

[3] R. Barzilay and L. Lee. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *Proc. of HLT-NAACL*, pages 113–120, 2004.

[4] D. Bikel and D. Chiang. Two statistical parsing models applied to the chinese treebank. In *Proc. of the Second Chinese Language Processing Workshop*, pages 1–6, 2000.

[5] C. Boulis and M. Ostendorf. Text classification by augmenting the bag-of-words representation with redundancy-compensated bigrams. In *Workshop on Feature Selection in Data Mining, in conjunction with SIAM conference on Data Mining*, 2005.

[6] J. Brown and M. Eskenazi. Retrieval of authentic documents for reader-specific lexical practice. In *Proc. of INSTIL*, 2004.

[7] W. Buntine and R. Caruana. Introduction to IND version 2.1 and recursive partitioning, 1992.

[8] J. Burstein, J. Shore, J. Sabatini, Y. Lee, and M. Ventura. The automated text adaptation tool. In *Demo Proc. of the the annual conference of the North American chapter of the Association for Computational Linguistics (NAACL-HLT)*, 2007.

[9] J. Burstein, J. Shore, J. Sabatini, Y. Lee, and M. Ventura. The automated text adaptation tool v1.0: Reading comprehension support for english language learners. Final report to the ELLA Initiative (Draft), 2007.

[10] P. Bylsma, L. Ireland, and H. Malagon. *Educating English Language Learners in Washington State.* Office of the Superintendent of Public Instruction, Olympia, WA, 2003.

[11] Y. Canning, J.I. Tait, J. Archibald, and R. Crawley. Cohesive regeneration of syntactically simplified newspaper text. In *Proc. of the First Workshop on Robust Methods in Analysis of Nat ural Language Data*, pages 3–16, 2000.

[12] J. Carletta. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22(2):249–256, 1996.

[13] J. Carroll, G. Minnen, Y. Canning, S. Devlin, and J.I. Tait. Practical simplification of English newspaper text to assist aphasic readers. In *Proc. of AAAI*, pages 7–10, 1998.

[14] J. Carroll, G. Minnen, D. Pearce, Y. Canning, S. Devlin, and J.I. Tait. Simplifying text for language-impaired readers. In *Proc. of EACL*, pages 269–270, 1999.

[15] W.B. Cavnar and J.M. Trenkle. N-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, 1994.

[16] J.S. Chall and E. Dale. *Readability revisited: the new Dale-Chall readability formula.* Brookline Books, Cambridge, Mass., 1995.

[17] R. Chandrasekar, C. Doran, and B. Srinivas. Motivations and methods for text simplification. In *Proc. of COLING96*, pages 1041–1044., 1996.

[18] R. Chandrasekar and B. Srinivas. Automatic induction of rules for text simplification. *Knowledge Based Systems*, 10:183–190, 1997.

[19] E. Charniak. A maximum-entropy-inspired parser. In *Proc. of NAACL*, pages 132–139, 2000.

[20] P. Chatain, E. W. D. Whittaker, J. A. Mrozinshi, and S. Furui. Topic and stylistic adaptation for speech summarization. In *Proc. of ICASSP*, volume 1, pages 977–980, 2006.

[21] S. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 13(4):359–393, 1999.

[22] Inc. Cislunar Aerospace. The k-8 internet aeronautics textbook: Principles of aeronautics. `http://wings.avkids.com/`, 2002. Accessed May 30, 2007.

[23] J. Clarke and M. Lapata. Models for sentence compression: a comparison across domains, training requirements and evaluation measures. In *Proc. ACL*, pages 377–384, 2006.

[24] V. Collier. Acquiring a second language for school. *Directions in Language and Education*, 1(4), 1995. `http://www.ncela.gwu.edu/pubs/directions/04.htm`.

[25] K. Collins-Thompson and J. Callan. A language modeling approach to predicting reading difficulty. In *Proc. of HLT/NAACL*, pages 193–200, 2004.

[26] K. Collins-Thompson and J. Callan. Predicting reading difficulty with statistical language models. *Journal of the American Society for Information Science and Technology*, 56(13):448–1462, 2005.

[27] R. Collobert, S. Bengio, and J. Mariethoz. Torch: a modular machine learning software library. Technical report IDIAP-RR 02-46, IDIAP, 2002.

[28] Ronan Collobert and Samy Bengio. SVMTorch: Support vector machines for large-scale regression problems. *Journal of Machine Learning Research*, 1:143–160, 2001.

[29] B. Cowan and M. Collins. Morphology and reranking for the statistical parsing of spanish. In *Proc. of HLT*, pages 795–802, 2005.

[30] A. Coxhead. A new academic word list. *TESOL Quarterly*, 34(2):213–238, 2000.

[31] S. A. Crossley, M. M. Louwerse, P. M. McCarthy, and D. S. Macnamara. A linguistic analysis of simplified and authentic texts. *The Modern Language Journal*, 91(1):15–30, 2007.

[32] M. Damashek. Gauging similarity with n-grams: Language-independent categorization of text. *Science*, 267(5199):843–848, 1995.

[33] H. Drucker, B. Shahraray, and D. C. Gibbon. Relevance feedback using support vector machines. In *Proc. of ICML*, pages 122–129, 2001.

[34] A. Farahat, G. Nunberg, F. Chen, and C. Mathis. AuGEAS: authoritativeness grading, estimation, and sorting. In *Proc. of the Eleventh International Conference on Information and Knowledge Management*, pages 194–202, 2002.

[35] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database.* MIT Press, Cambridge, MA, 1998.

[36] W.H. Fletcher. Concordancing the Web with KWiCFinder. `http://www.kwicfinder.com/FletcherCLLT2001.pdf`, 2003. Accessed March 29, 2004.

[37] I. C. Fountas and G. S. Pinnell. *Matching Books to Readers: Using Leveled Books in Guided Reading, K-3.* Heinemann, Portsmouth, NH, 1999.

[38] S. Ghadirian. Providing controlled exposure to target vocabulary through the screening and arranging of texts. *Language Learning and Technology*, 6(2):147–164, 2002.

[39] R. Gunning. *The technique of clear writing.* McGraw-Hill, New York, 1952.

[40] D. Harman and M. Liberman. TIPSTER complete. Linguistic Data Consortium, catalog number LDC93T3A and ISBN: 1-58563-020-9, 1993.

[41] M. Heilman, K. Collins-Thompson, J. Callan, and M. Eskenazi. Classroom success of an intelligent tutoring system for lexical practice and reading comprehension. In *Proc. of ICSLP*, 2006.

[42] M. Heilman, K. Collins-Thompson, J. Callan, and M. Eskenazi. Combining lexical and grammatical features to improve readability measures for first and second language texts. In *Proc. of HLT*, 2007.

[43] C. Hori, S. Furui, R. Malkin, H. Yu, and A. Waibel. A statistical approach to automatic speech summarization. *EURASIP Journal on Applied Signal Processing*, 2:128–139, 2003.

[44] M. Horst, T. Cobb, and P.M. Meara. Beyond a Clockwork Orange; acquiring second language vocabulary through reading. *Reading in a Foreign Language*, 11:207–223, 1998.

[45] C.-W. Hsu, C.-C. Chang, and C.-J. Lin. A practical guide to support vector classification. `http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf`, 2003.

[46] S. Huffman. Acquaintance: Language-independent document categorization by n-grams. In *Proc. of TREC-4, 4th Text Retrieval Conference*, pages 359–371, 1995.

[47] K. Inui, A. Fujiti, T. Takahashi, R. Iida, and T. Iwakura. Text simplification for reading assistance: A project note. In *Second International Workshop on Paraphrasing: Paraphrase Acquisition and Applications (IWP2003)*, pages 9–16, 2003.

[48] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In *Proc. of the European Conference on Machine Learning*, pages 137–142, 1998.

[49] T. Joachims. Making large-scale support vector machine learning practical. In A. Smola B. Schölkopf, C. Burges, editor, *Advances in Kernel Methods:Support Vector Machines*. MIT Press, Cambridge, MA, 1999.

[50] B. Kessler, G. Nunberg, and H. Schuetze. Automatic detection of text genre. In *Proc. of ACL/EACL*, pages 32–38. Association for Computational Linguistics, 1997.

[51] T. Kikuchi, S. Furui, and C. Hori. Automatic speech summarization based on sentence extraction and compatcion. In *Proc. of ICASSP*, volume 1, pages 384–387, 2003.

[52] F. Kilickaya. Authentic materials and cultural content in efl classrooms. *The Internet TESL Journal*, X(7), 2004.

[53] J.P. Kincaid, Jr. R.P. Fishburne, R.L. Rodgers, and B.S. Chisson. Derivation of new readability formulas for Navy enlisted personnel. Research Branch Report 8-75, U.S. Naval Air Station, Memphis, 1975.

[54] B. Beigman Klebanov, K. Knight, and D. Marcu. Text simplification for information-seeking applications. In R. Meersman and Z. Tari, editors, *On the Move to Meaningful Internet Systems*, Lecture Notes in Computer Science, pages 735–747. Springer Verlag, 2004.

[55] K. Knight and D. Marcu. Statistics-based summarization - step one: Sentence compression. In *Proc. of AAAI/IAAI*, pages 703–710, 2000.

[56] S. Krashen. Arguments for and (bogus) arguments against bilingual education. Georgetown University Roundtable on Languages and Linguistics, 1999.

[57] J. Kupiec, J. Pedersen, and F. Chen. A trainable document summarizer. In *Proc. of SIGIR*, pages 68–73, 1995.

[58] P. Lal and S. Ruger. Extract-based summarization with simplification. In *Proc. of DUC*, 2002.

[59] Y.-B. Lee and S.H. Myaeng. Text genre classification with genre-revealing and subject-revealing features. In *Proc. of SIGIR*, pages 145–150, 2002.

[60] The Lexile framework for reading. `http://www.lexile.com`, 2005. Accessed April 15, 2005.

[61] X. Li. *Regularized Adaptation: Theory, Algorithms and Applications*. PhD thesis, University of Washington, Seattle, WA, 2007.

[62] C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. In *Proc. of Text Summarization Branches out, ACL Post-Conference Workshop*, 2004.

[63] T.-Y. Liu, Y. Yang, H. Wan, H.-J. Zeng, Z. Chen, and W.-Y. Ma. Support vector machines classification with a very large-scale taxonomy. *SIGKDD Explorations*, 7(1):36–43, 2005.

[64] D. Marcu. Improving summarization through rhetorical parsing tuning. In *The Sixth Workshop on Very Large Corpora*, pages 206–215, 1998.

[65] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki. The DET curve in assessment of detection task performance. In *Proc. of Eurospeech*, volume 4, pages 1895–1898, 1997.

[66] A. Max. Simplification interactive pour la production de textes adaptés aux personnes souffrant de troubles de la compréhension. In *Proc. of Traitement Automatique des Langues Naturelles (TALN)*, 2005.

[67] A. Max. Writing for language-impaired readers. In *Proc. of Computational Linguistics and Intelligent Text Processing (CICLing)*, volume LNCS 3878, pages 567–570, 2006.

[68] A. K. McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. `http://www.cs.cmu.edu/~mccallum/bow`, 1996.

[69] F. Peng, D. Schuurmans, and S. Wang. Augmenting naive Bayes classifiers with statistical language models. *Information Retrieval*, 7(3-4):317–345, 2004.

[70] S. E. Petersen and M. Ostendorf. Assessing the reading level of web pages. In *Proc. of ICSLP*, pages 833–836, 2006.

[71] S. E. Petersen and M. Ostendorf. A machine learning approach to reading level assessment. Technical Report 2006-06-06, University of Washington, Department of Computer Science and Engineering, 2006.

[72] S. E. Petersen and M. Ostendorf. Text simplification for language learners: a corpus analysis. In *Proc. of Workshop on Speech and Language Technology for Education*, 2007.

[73] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.

[74] A. Ratnaparkhi. A maximum entropy part-of-speech tagger. In *Proc. of EMNLP*, pages 133–141, 1996.

[75] J.W. Rosenthal. *Teaching Science to Language Minority Students*. Multilingual Matters, Ltd, Clevendon, UK, 1996.

[76] S. Rüping. Incremental learning with support vector machines. In *Proc. of the IEEE International Conference on Data Mining*, 2001.

[77] M. Sacks. personal communcation, 2007.

[78] H. Schmid. Probabilistic part-of-speech tagging using decision trees. In *Proc. of the International Conference on New Methods in Language Processing*, Manchester, UK, 1994.

[79] S. E. Schwarm and M. Ostendorf. Reading level assessment using support vector machines and statistical language models. In *Proc. of ACL*, pages 523–530, 2005.

[80] S. Sekine and C. Nobata. Sentence extraction with information extraction technique. In *Proc. of DUC*, 2001.

[81] A. Sethy, P. G. Georgiou, and S. Narayanan. Building topic specific language models from webdata using competitive models. In *Proc. INTERSPEECH*, pages 1293–1296, 2005.

[82] L. Si and J.P. Callan. A statistical model for scientific readability. In *Proc. of CIKM*, pages 574–576, 2001.

[83] A. Siddharthan. An architecture for a text simplification system. In *LEC '02: Proceedings of the Language Engineering Conference (LEC'02)*, pages 64–71, 2002.

[84] A. Siddharthan. *Syntactic Simplification and Text Cohesion*. PhD thesis, University of Cambridge, Cambridge, UK, 2003.

[85] A. Siddharthan. Syntactic simplification and text cohesion. *Research on Language and Computation*, 4(1):77–109, 2006.

[86] A. M. Simensen. Adapted readers: How are they adapted? *Reading in a Foreign Language*, 4(1):41–57, 1987.

[87] S. Sonmez. An overview of the studies on the use of authentic texts in language classrooms. In *Proc. of the Third International Online Conference on Second and Foreign Language Teaching and Research*, pages 51–62, 2007.

[88] E. Stamatatos. Text genre detection using common word frequencies. In *Proc. of the 18th International Conference on COLING2000*, pages 808–814, 2000.

[89] A.J. Stenner. Measuring reading comprehension with the Lexile framework. Presented at the Fourth North American Conference on Adolescent/Adult Literacy, 1996.

[90] A. Stolcke. SRILM – an extensible language modeling toolkit. In *Proc. of Intl. Conf. on Spoken Language Processing*, volume 2, pages 901–904, 2002.

[91] A. Sun, E.-P. Lim, and W.-K. Ng. Web classification using support vector machine. In *Proc. of the 4th Int. Workshop on Web Information and Data Management*, pages 96–99, 2002.

[92] The Washington Post. `http://www.washingtonpost.com`, 2005. Accessed April 20, 2005.

[93] W.P. Thomas and V. Collier. School effectiveness for language minority students. National Clearinghouse for English Language Acquisition, `http://www.ncela.gwu.edu/pubs/resource/effectiveness/`, 1997. Accessed June 23, 2004.

[94] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66, 2001.

[95] U.S. Department of Education, National Center for Educational Statistics. NCES fast facts: Bilingual education/Limited English Proficient students. `http://nces.ed.gov/fastfacts/display.asp?id=96`, 2004. Accessed March 12, 2007.

[96] U.S. Department of Education, National Center for Educational Statistics. The condition of education 2006. `http://nces.ed.gov/pubs2006/2006071.pdf`, 2006. Accessed March 12, 2007.

[97] G. van Noord. TextCat. `http://odur.let.rug.nl/~vannoord/TextCat`, 1999. Accessed June 15, 2004.

[98] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[99] Weekly Reader. `http://www.weeklyreader.com/`, 2004. Accessed July, 2004.

[100] Western/Pacific Literacy Network / Literacyworks. CNN SF learning resources. `http://literacynet.org/cnnsf/`, 2004. Accessed June 15, 2004.

[101] Wikipedia. Wikinews. `http://en.wikinews.org/wiki/Main_Page`, 2007. Accessed May 30, 2007.

[102] J.P. Yamron, I. Carp, L. Gillick, S. Lowe, and P. van Mulbregt. Topic tracking in a news stream. In *Proc. of the DARPA Broadcast News Workshop*, 1999. `http://www.nist.gov/speech/publications/darpa99/pdf/tdt250.pdf`.

[103] Y. Yang. A study on thresholding strategies for text categorization. In *Proc. of SIGIR*, pages 137–145, 2001.

[104] Y. Yang and J. Pedersen. A comparative study on feature selection in text categorization. In *Proc. ICML*, pages 412–420, 1997.

[105] D. J. Young. Linguistic simplification of sl reading material: Effective instructional practice? *The Modern Language Journal*, 83(3):350–366, 1999.

VITA

Sarah Elizabeth Petersen née Schwarm grew up in Virginia and received her B.A. in Cognitive Science and French from the University of Virginia in 1999. She continued her studies at the University of Washington in Seattle, focusing on language modeling for automatic speech recognition and earning an M.S. in Computer Science and Engineering in 2001. She spent 6 months in 2002-2003 at an internship at the Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur (LIMSI/CNRS) in Paris. Upon returning to Seattle, she worked briefly on edit detection in conversational speech and then turned her attention to her dissertation research on educational applications of natural language processing. She received the Ph.D. degree in August 2007.