A Comparison of Features for Automatic Readability Assessment

Lijun Feng
City University of New York
lijun7.feng@gmail.com

Martin Jansche
Google, Inc.
jansche@acm.org

Matt Huenerfauth
City University of New York
matt@cs.qc.cuny.edu

Noémie Elhadad
Columbia University
noemie@dbmi.columbia.edu

Abstract

Several sets of explanatory variables – including shallow, language modeling, POS, syntactic, and discourse features – are compared and evaluated in terms of their impact on predicting the grade level of reading material for primary school students. We find that features based on in-domain language models have the highest predictive power. Entity-density (a discourse feature) and POS-features, in particular nouns, are individually very useful but highly correlated. Average sentence length (a shallow feature) is more useful – and less expensive to compute – than individual syntactic features. A judicious combination of features examined here results in a significant improvement over the state of the art.

1 Introduction

1.1 Motivation and Method

Readability Assessment quantifies the difficulty with which a reader understands a text. Automatic readability assessment enables the selection of appropriate reading material for readers of varying proficiency. Besides modeling and understanding the linguistic components involved in readability, a readability-prediction algorithm can be leveraged for the task of automatic text simplification: as simplification operators are applied to a text, the readability is assessed to determine whether more simplification is needed or a particular reading level was reached.

Identifying text properties that are strongly correlated with text complexity is itself complex. In this paper, we explore a broad range of text properties at various linguistic levels, ranging from discourse features to language modeling features, part-of-speech-based grammatical features, parsed syntactic features and well studied shallow features, many of which are inspired by previous work.

We use grade levels, which indicate the number of years of education required to completely understand a text, as a proxy for reading difficulty. The corpus in our study consists of texts labeled with grade levels ranging from grade 2 to 5. We treat readability assessment as a classification task and evaluate trained classifiers in terms of their prediction accuracy. To investigate the contributions of various sets of features, we build prediction models and examine how the choice of features influences the model performance.

1.2 Related Work

Many traditional readability metrics are linear models with a few (often two or three) predictor variables based on superficial properties of words, sentences, and documents. These shallow features include the average number of syllables per word, the number of words per sentence, or binned word frequency. For example, the Flesch-Kincaid Grade Level formula uses the average number of words per sentence and the average number of syllables per word to predict the grade level (Flesch, 1979). The Gunning FOG index (Gunning, 1952) uses average sentence length and the percentage of words with at least three syllables. These traditional metrics are easy to compute and use, but they are not reliable, as demonstrated by several recent studies in the field (Si and Callan, 2001; Petersen and Ostendorf, 2006; Feng et al., 2009).

Coling 2008: Poster Volume, pages 276–284,
Beijing, August 2010

With the advancement of natural language processing tools, a wide range of more complex text properties have been explored at various linguistic levels. Si and Callan (2001) used unigram language models to capture content information from scientific web pages. Collins-Thompson and Callan (2004) adopted a similar approach and used a smoothed unigram model to predict the grade levels of short passages and web documents. Heilman et al. (2007) continued using language modeling to predict readability for first and second language texts. Furthermore, they experimented with various statistical models to test their effectiveness at predicting reading difficulty (Heilman et al., 2008). Schwarm/Petersen and Ostendorf (Schwarm and Ostendorf, 2005; Petersen and Ostendorf, 2006) used support vector machines to combine features from traditional reading level measures, statistical language models and automatic parsers to assess reading levels. In addition to lexical and syntactic features, several researchers started to explore discourse level features and examine their usefulness in predicting text readability. Pitler and Nenkova (2008) used the Penn Discourse Treebank (Prasad et al., 2008) to examine discourse relations. We previously used a lexical-chaining tool to extract entities that are connected by certain semantic relations (Feng et al., 2009).

In this study, we systematically evaluate all above-mentioned types of features, as well as a few extensions and variations. A detailed description of the features appears in Section 3. Section 4 discusses results of experiments with classifiers trained on these features. We begin with a description of our data in the following section.

2 Corpus

We contacted the Weekly Reader[1] corporation, an on-line publisher producing magazines for elementary and high school students, and were granted access in October 2008 to an archive of their articles. Among the articles retrieved, only those for elementary school students are labeled with grade levels, which range from 2 to 5. We selected only this portion of articles (1629 in total) for the study.[2] These articles are intended to build children's general knowledge and help them practice reading skills. While pre-processing the texts, we found that many articles, especially those for lower grade levels, consist of only puzzles and quizzes, often in the form of simple multiple-choice questions. We discarded such texts and kept only 1433 full articles. Some distributional statistics of the final corpus are listed in Table 1.

Table 1: Statistics for the Weekly Reader Corpus

Grade	docs.	words/document		words/sentence	
		mean	std. dev.	mean	std. dev.
2	174	128.27	106.03	9.54	2.32
3	289	171.96	106.05	11.39	2.42
4	428	278.03	187.58	13.67	2.65
5	542	335.56	230.25	15.28	3.21

3 Features

3.1 Discourse Features

We implement four subsets of discourse features: entity-density features, lexical-chain features, coreference inference features and entity grid features. The coreference inference features are novel and have not been studied before. We previously studied entity-density features and lexical-chain features for readers with intellectual disabilities (Feng et al., 2009). Entity-grid features have been studied by Barzilay and Lapata (2008) in a stylistic classification task. Pitler and Nenkova (2008) used the same features to evaluate how well a text is written. We replicate this set of features for grade level prediction task.

3.1.1 Entity-Density Features

Conceptual information is often introduced in a text by entities, which consist of general nouns and named entities, e.g. people's names, locations, organizations, etc. These are important in text comprehension, because established entities form basic components of concepts and propositions, on which higher level discourse processing is based. Our prior work illustrated the importance of entities in text comprehension (Feng et al., 2009).

[1] http://www.weeklyreader.com

[2] A corpus of Weekly Reader articles was previously used in work by Schwarm and Ostendorf (2005). However, the two corpora are not identical in size nor content.

Table 2: New Entity-Density Features

1	percentage of named entities per document
2	percentage of named entities per sentences
3	percentage of overlapping nouns removed
4	average number of remaining nouns per sentence
5	percentage of named entities in total entities
6	percentage of remaining nouns in total entities

We hypothesized that the number of entities introduced in a text relates to the working memory burden on their targeted readers – individuals with intellectual disabilities. We defined entities as a union of named entities and general nouns (nouns and proper nouns) contained in a text, with overlapping general nouns removed. Based on this, we implemented four kinds of entity-density features: total number of entity mentions per document, total number of unique entity mentions per document, average number of entity mentions per sentence, and average number of unique entity mentions per sentence.

We believe entity-density features may also relate to the readability of a text for a general audience. In this paper, we conduct a more refined analysis of general nouns and named entities. To collect entities for each document, we used OpenNLP's[3] name-finding tool to extract named entities; general nouns are extracted from the output of Charniak's Parser (see Section 3.3). Based on the set of entities collected for each document, we implement 12 new features. We list several of these features in in Table 2.

3.1.2 Lexical Chain Features

During reading, a more challenging task with entities is not just to keep track of them, but to resolve the semantic relations among them, so that information can be processed, organized and stored in a structured way for comprehension and later retrieval. In earlier work (Feng et al., 2009), we used a lexical-chaining tool developed by Galley and McKeown (2003) to annotate six semantic relations among entities, e.g. synonym, hypernym, hyponym, etc. Entities that are connected by these semantic relations were linked through the text to form lexical chains. Based on these chains, we implemented six features, listed in Table 3, which we use in our current study. The length of a chain is the number of entities contained in the chain, the span of chain is the distance between the index of the first and last entity in a chain. A chain is defined to be active for a word or an entity if this chain passes through its current location.

[3] http://opennlp.sourceforge.net/

Table 3: Lexical Chain Features

1	total number of lexical chains per document
2	avg. lexical chain length
3	avg. lexical chain span
4	num. of lex. chains with span $\geq$ half doc. length
5	num. of active chains per word
6	num. of active chains per entity

Table 4: Coreference Chain Features

1	total number of coreference chains per document
2	avg. num. of coreferences per chain
3	avg. chain span
4	num. of coref. chains with span $\geq$ half doc. length
5	avg. inference distance per chain
6	num. of active coreference chains per word
7	num. of active coreference chains per entity

3.1.3 Coreference Inference Features

Relations among concepts and propositions are often not stated explicitly in a text. Automatically resolving implicit discourse relations is a hard problem. Therefore, we focus on one particular type, referential relations, which are often established through anaphoric devices, e.g. pronominal references. The ability to resolve referential relations is important for text comprehension.

We use OpenNLP to resolve coreferences. Entities and pronominal references that occur across the text and refer to the same person or object are extracted and formed into a coreference chain. Based on the chains extracted, we implement seven features as listed in Table 4. The chain length, chain span and active chains are defined in a similar way to the lexical chain features. Inference distance is the difference between the index of the referent and that of its pronominal reference. If the same referent occurs more than once in a chain, the index of the closest occurrence is used when computing the inference distance.

3.1.4 Entity Grid Features

Coherent texts are easier to read. Several computational models have been developed to represent and

measure discourse coherence (Lapata and Barzilay, 2005; Soricut and Marcu, 2006; Elsner et al., 2007; Barzilay and Lapata, 2008) for NLP tasks such as text ordering and text generation. Although these models are not intended directly for readability research, Barzilay and Lapata (2008) have reported that distributional properties of local entities generated by their grid models are useful in detecting original texts from their simplified versions when combined with well studied lexical and syntactic features. This approach was subsequently pursued by Pitler and Nenkova (2008) in their readability study. Barzilay and Lapata's entity grid model is based on the assumption that the distribution of entities in locally coherent texts exhibits certain regularities. Each text is abstracted into a grid that captures the distribution of entity patterns at the level of sentence-to-sentence transitions. The entity grid is a two-dimensional array, with one dimension corresponding to the salient entities in the text, and the other corresponding to each sentence of the text. Each grid cell contains the grammatical role of the specified entity in the specified sentence: whether it is a subject (S), object (O), neither of the two (X), or absent from the sentence (-).

We use the Brown Coherence Toolkit (v0.2) (Elsner et al., 2007), based on (Lapata and Barzilay, 2005), to generate an entity grid for each text in our corpus. The distribution patterns of entities are traced between each pair of adjacent sentences, resulting in 16 entity transition patterns[4]. We then compute the distribution probability of each entity transition pattern within a text to form 16 entity-grid-based features.

3.2 Language Modeling Features

Our language-modeling-based features are inspired by Schwarm and Ostendorf's (2005) work, a study that is closely related to ours. They used data from the same data – the Weekly Reader – for their study. They trained three language models (unigram, bigram and trigram) on two paired complex/simplified corpora (Britannica and LiteracyNet) using an approach in which words with high information gain are kept and the remaining words are replaced with their parts of speech. These language models were then used to score each text in the Weekly Reader corpus by perplexity. They reported that this approach was more successful than training LMs on text sequences of word labels alone, though without providing supporting statistics.

It's worth pointing out that their LMs were not trained on the Weekly Reader data, but rather on two unrelated paired corpora (Britannica and LiteracyNet). This seems counter-intuitive, because training LMs directly on the Weekly Reader data would provide more class-specific information for the classifiers. They justified this choice by stating that splitting limited Weekly Reader data for training and testing purposes resulted in unsuccessful performance.

We overcome this problem by using a hold-one-out approach to train LMs directly on our Weekly Reader corpus, which contains texts ranging from Grade 2 to 5. We use grade levels to divide the whole corpus into four smaller subsets. In addition to implementing Schwarm and Ostendorf's information-gain approach, we also built LMs based on three other types of text sequences for comparison purposes. These included: word-token-only sequence (i.e., the original text), POS-only sequence, and paired word-POS sequence. For each grade level, we use the SRI Language Modeling Toolkit[5] (with Good-Turing discounting and Katz backoff for smoothing) to train 5 language models (1- to 5-gram) using each of the four text sequences, resulting in $4 \times 5 \times 4 = 80$ perplexity features for each text tested.

3.3 Parsed Syntactic Features

Schwarm and Ostendorf (2005) studied four parse tree features (average parse tree height, average number of SBARs, noun phrases, and verb phrases per sentences). We implemented these and additional features, using the Charniak parser (Charniak, 2000). Our parsed syntactic features focus on clauses (SBAR), noun phrases (NP), verb phrases (VP) and prepositional phrases (PP). For each phrase, we implement four features: total number of the phrases per document, average number of phrases per sentence, and average phrase length

[4]These 16 transition patterns are: "SS", "SO", "SX", "S-", "OS", "OO", "OX", "O-", "XS", "XO", "XX", "X-", "-S", "-O", "-X", "--".

[5]http://www.speech.sri.com/projects/srilm/

measured by number of words and characters respectively. In addition to average tree height, we implement two non-terminal-node-based features: average number of non-terminal nodes per parse tree, and average number of non-terminal nodes per word (terminal node).

3.4 POS-based Features

Part-of-speech-based grammatical features were shown to be useful in readability prediction (Heilman et al., 2007; Leroy et al., 2008). To extend prior work, we systematically studied a number of common categories of words and investigated to what extent they are related to a text's complexity. We focus primarily on five classes of words (nouns, verbs, adjectives, adverbs, and prepositions) and two broad categories (content words, function words). Content words include nouns, verbs, numerals, adjectives, and adverbs; the remaining types are function words. The part of speech of each word is obtained from examining the leaf node based on the output of Charniak's parser, where each leaf node consists of a word and its part of speech. We group words based on their POS labels. For each class of words, we implement five features. For example, for the adjective class, we implemented the following five features: percent of adjectives (tokens) per document, percent of unique adjectives (types) per document, ratio of unique adjectives per total unique words in a document, average number of adjectives per sentence and average number of unique adjectives per sentence.

3.5 Shallow Features

Shallow features refer to those used by traditional readability metrics, such as Flesch-Kincaid Grade Level (Flesch, 1979), SMOG (McLaughlin, 1969), Gunning FOG (Gunning, 1952), etc. Although recent readability studies have strived to take advantage of NLP techniques, little has been revealed about the predictive power of shallow features. Shallow features, which are limited to superficial text properties, are computationally much less expensive than syntactic or discourse features. To enable a comparison against more advanced features, we implement 8 frequently used shallow features as listed in Table 5.

Table 5: Shallow Features

1	average number of syllables per word
2	percentage of poly-syll. words per doc.
3	average number of poly-syll. words per sent.
4	average number of characters per word
5	Chall-Dale difficult words rate per doc.
6	average number of words per sentence
7	Flesch-Kincaid score
8	total number of words per document

3.6 Other Features

For comparison, we replicated 6 out-of-vocabulary features described in Schwarm and Ostendorf (2005). For each text in the Weekly Reader corpus, these 6 features are computed using the most common 100, 200 and 500 word tokens and types based on texts from Grade 2. We also replicated the 12 perplexity features implemented by Schwarm and Ostendorf (2005) (see Section 3.2).

4 Experiments and Discussion

Previous studies on reading difficulty explored various statistical models, e.g. regression vs. classification, with varying assumptions about the measurement of reading difficulty, e.g. whether labels are ordered or unrelated, to test the predictive power of models (Heilman et al., 2008; Petersen and Ostendorf, 2009; Aluisio et al., 2010). In our research, we have used various models, including linear regression; standard classification (Logistic Regression and SVM), which assumes no relation between grade levels; and ordinal regression/ classification (provided by Weka, with Logistic Regression and SMO as base function), which assumes that the grade levels are ordered. Our experiments show that, measured by mean squared error and classification accuracy, linear regression models perform considerably poorer than classification models. Measured by accuracy and F-measure, ordinal classifiers perform comparable or worse than standard classifiers. In this paper, we present the best results, which are obtained by standard classifiers. We use two machine learning packages known for efficient high-quality multi-class classification: LIBSVM (Chang and Lin, 2001) and the Weka machine learning toolkit (Hall et al., 2009), from which we choose Logistic Regression as classifiers. We train and evaluate various prediction

Table 6: Comparison of discourse features

Feature Set	LIBSVM	Logistic Regress.
Entity-Density	59.63 ± 0.632	57.59 ± 0.375
Lexical Chain	45.86 ± 0.815	42.58 ± 0.241
Coref. Infer.	40.93 ± 0.839	42.19 ± 0.238
Entity Grid	45.92 ± 1.155	42.14 ± 0.457
all combined	60.50 ± 0.990	58.79 ± 0.703

Table 7: Comparison of lang. modeling features

Feature Set	LIBSVM	Logistic Regress.
IG	62.52 ± 1.202	62.14 ± 0.510
Text-only	60.17 ± 1.206	60.31 ± 0.559
POS-only	56.21 ± 2.354	57.64 ± 0.391
Word/POS pair	60.38 ± 0.820	59.00 ± 0.367
all combined	68.38 ± 0.929	66.82 ± 0.448
IG by Schwarm	52.21 ± 0.832	51.89 ± 0.405

Table 8: Comparison of parsed syntactic features

Feature Set	# Feat.	LIBSVM
Original features	4	50.68 ± 0.812
Expanded features	21	57.79 ± 1.023

models using the features described in Section 3. We evaluate classification accuracy using repeated 10-fold cross-validation on the Weekly Reader corpus. Classification accuracy is defined as the percentage of texts predicted with correct grade levels. We repeat each experiment 10 times and report the mean accuracy and its standard deviation.

4.1 Discourse Features

We first discuss the improvement made by extending our earlier entity-density features (Feng et al., 2009). We used LIBSVM to train and test models on the Weekly Reader corpus with our earlier features and our new features respectively. With earlier features only, the model achieves 53.66% accuracy. With our new features added, the model performance is 59.63%.

Table 6 presents the classification accuracy of models trained with discourse features. We see that, among four subsets of discourse features, entity-density features perform significantly better than the other three feature sets and generate the highest classification accuracy (LIBSVM: 59.63%, Logistic Regression: 57.59%). While Logistic Regression results show that there is not much performance difference among lexical chain, coreference inference, and entity grid features, classification accuracy of LIBSVM models indicates that lexical chain features and entity grid features are better in predicting text readability than coreference inference features. Combining all discourse features together does not significantly improve accuracy compared with models trained only with entity-density features.

4.2 Language Modeling Features

Table 7 compares the performance of models generated using our approach and our replication of Schwarm and Ostendorf's (2005) approach. In our approach, features were obtained from language models trained on the Weekly Reader corpus. Not surprisingly, these are more effective than LMs trained on the Britannica and LiteracyNet corpora, in Schwarm and Ostendorf's approach. Our results support their claim that LMs trained with information gain outperform LMs trained with POS labels. However, we also notice that training LMs on word labels alone or paired word/POS sequences achieved similar classification accuracy to the IG approach, while avoiding the complicated feature selection of the IG approach.

4.3 Parsed Syntactic Features

Table 8 compares a classifier trained on the four parse features of Schwarm and Ostendorf (2005) to a classifier trained on our expanded set of parse features. The LIBSVM classifier with the expanded feature set scored 7 points higher than the one trained on only the original four features, improving from 50.68% to 57.79%. Table 9 shows a detailed comparison of particular parsed syntactic features. The two non-terminal-node-based features (average number of non-terminal nodes per tree and average number of non-terminal nodes per word) have higher discriminative power than average tree height. Among SBARs, NPs, VPs and PPs, our experiments show that VPs and NPs are the best predictors.

4.4 POS-based Features

The classification accuracy generated by models trained with various POS features is presented in Table 10. We find that, among the five word classes investigated, noun-based features gener-

Table 9: Detailed comp. of syntactic features

Feature Set	LIBSVM	Logistic Regress.
Non-term.-node ratios	53.02 ± 0.571	51.80 ± 0.171
Average tree height	44.26 ± 0.914	43.45 ± 0.269
SBARs	44.42 ± 1.074	43.50 ± 0.386
NPs	51.56 ± 1.054	48.14 ± 0.408
VPs	53.07 ± 0.597	48.67 ± 0.484
PPs	49.36 ± 1.277	46.47 ± 0.374
all combined	57.79 ± 1.023	54.11 ± 0.473

Table 10: Comparison of POS features

Feature Set	LIBSVM	Logistic Regress.
Nouns	58.15 ± 0.862	57.01 ± 0.256
Verbs	54.40 ± 1.029	55.10 ± 0.291
Adjectives	53.87 ± 1.128	52.75 ± 0.427
Adverbs	52.66 ± 0.970	50.54 ± 0.327
Prepositions	56.77 ± 1.278	54.13 ± 0.312
Content words	56.84 ± 1.072	56.18 ± 0.213
Function words	52.19 ± 1.494	50.95 ± 0.298
all combined	59.82 ± 1.235	57.86 ± 0.547

ate the highest classification accuracy, which is consistent with what we have observed earlier about entity-density features. Another notable observation is that prepositions demonstrate higher discriminative power than adjectives and adverbs. Models trained with preposition-based features perform close to those trained with noun-based features. Among the two broader categories, content words (which include nouns) demonstrate higher predictive power than function words (which include prepositions).

4.5 Shallow Features

We present some notable findings on shallow features in Table 11. Experimental results generated by models trained with Logistic Regression show that average sentence length has dominating predictive power over all other shallow features. Features based on syllable counting perform much worse. The Flesch-Kincaid Grade Level score uses a fixed linear combination of average words per sentence and average syllables per word. Combining those two features (without fixed coefficients) results in the best overall accuracy, while using the Flesch-Kincaid score as a single feature is significantly worse.

Table 11: Comparison of shallow features

Feature Set	Logistic Regress.
Avg. words per sent.	52.17 ± 0.193
Avg. syll. per word	42.51 ± 0.264
above two combined	53.04 ± 0.514
Flesch-Kincaid score	50.83 ± 0.144
Avg. poly-syll. words per sent.	45.70 ± 0.306
all 8 features combined	52.34 ± 0.242

4.6 Comparison with Previous Studies

A trivial baseline of predicting the most frequent grade level (grade 5) predicts 542 out of 1433 texts (or 37.8%) correctly. With this in mind, we first compare our study with the widely-used Flesch-Kincaid Grade Level formula, which is a linear function of average words per sentence and average syllables per word that aims to predict the grade level of a text directly. Since this is a fixed formula with known coefficients, we evaluated it directly on our entire Weekly Reader corpus without cross-validation. We obtain the predicted grade level of a text by rounding the Flesch-Kincaid score to the nearest integer. For only 20 out of 1433 texts the predicted and labeled grade levels agree, resulting in a poor accuracy of 1.4%. By contrast, using the Flesch-Kincaid score as a feature of a simple logistic regression model achieves above 50% accuracy, as discussed in Section 4.5.

The most closely related previous study is the work of Schwarm and Ostendorf (2005). However, because their experiment design (85/15 training/test data split) and machine learning tool (SVM^{light}) differ from ours, their results are not directly comparable to ours. To make a comparison, we replicated all the features used in their study and then use LIBSVM and Weka's Logistic Regression to train two models with the replicated features and evaluate them on our Weekly Reader corpus using 10-fold cross-validation.

Using the same experiment design, we train classifiers with three combinations of our features as listed in Table 12. "All features" refers to a naive combination of all features. "AddOneBest" refers to a subset of features selected by a group-wise add-one-best greedy feature selection. "WekaFS" refers to a subset of features chosen by Weka's feature selection filter.

"WekaFS" consists of 28 features selected au-

Table 12: Comparison with previous work

baseline accuracy (majority class)		37.8	
Flesch-Kincaid Grade Level		1.4	
Feature Set	# Feat.	LIBSVM	Logistic Reg.
Schwarm	25	63.18 ± 1.664	60.50 ± 0.477
All features	273	72.21 ± 0.821	63.71 ± 0.576
AddOneBest	122	74.01 ± 0.847	69.22 ± 0.411
WekaFS	28	70.06 ± 0.777	65.46 ± 0.336

tomatically by Weka's feature selection filter using a best-first search method. The 28 features include language modeling features, syntactic features, POS features, shallow features and out-of-vocabulary features. Aside from 4 shallow features and 5 out-of-vocabulary features, the other 19 features are novel features we have implemented for this paper.

As Table 12 shows, a naive combination of all features results in classification accuracy of 72%, which is much higher than the current state of the art (63%). This is not very surprising, since we are considering a greater variety of features than any previous individual study. Our WekaFS classifier uses roughly the same number of features as the best published result, yet it has a higher accuracy (70.06%). Our best results were obtained by group-wise add-one-best feature selection, resulting in 74% classification accuracy, a big improvement over the state of the art.

5 Conclusions

We examined the usefulness of features at various linguistic levels for predicting text readability in terms of assigning texts to elementary school grade levels. We implemented a set of discourse features, enriched previous work by creating several new features, and systematically tested and analyzed the impact of these features.

We observed that POS features, in particular nouns, have significant predictive power. The high discriminative power of nouns in turn explains the good performance of entity-density features, based primarily on nouns. In general, our selected POS features appear to be more correlated to text complexity than syntactic features, shallow features and most discourse features.

For parsed syntactic features, we found that verb phrases appear to be more closely correlated with text complexity than other types of phrases. While SBARs are commonly perceived as good predictors for syntactic complexity, they did not prove very useful for predicting grade levels of texts in this study. In future work, we plan to examine this result in more detail.

Among the 8 shallow features, which are used in various traditional readability formulas, we identified that average sentence length has dominating predictive power over all other lexical or syllable-based features.

Not surprisingly, among language modeling features, combined features obtained from LMs trained directly on the Weekly Reader corpus show high discriminative power, compared with features from LMs trained on unrelated corpora.

Discourse features do not seem to be very useful in building an accurate readability metric. The reason could lie in the fact that the texts in the corpus we studied exhibit relatively low complexity, since they are aimed at primary-school students. In future work, we plan to investigate whether these discourse features exhibit different discriminative power for texts at higher grade levels.

A judicious combination of features examined here results in a significant improvement over the state of the art.

References

Sandra Aluisio, Lucia Specia, Caroline Gasperin, and Carolina Scarton. 2010. Readability assessment for text simplification. In *NAACL-HLT 2010: The 5th Workshop on Innovative Use of NLP for Building Educational Applications*.

Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34.

Chih-Chung Chang and Chih-Jen Lin. 2001. *LIBSVM: A Library for Support Vector Machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st Conference of the North American Chapter of the ACL*, pages 132–139.

Kevyn Collins-Thompson and Jamie Callan. 2004. A language modeling approach to predicting reading difficulty. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2004)*.

Micha Elsner, Joseph Austerweil, and Eugene Charniak. 2007. A unified local and global model for discourse coherence. In *Proceedings of the Conference on Human Language Technology and North American chapter of the Association for Computational Linguistics (HLT-NAACL 2007)*.

Lijun Feng, Noémie Elhadad, and Matt Huenerfauth. 2009. Cognitively motivated features for readability assessment. In *The 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2009)*.

Rudolf Flesch. 1979. *How to write plain English*. Harper and Brothers, New York.

Michel Galley and Kathleen McKeown. 2003. Improving word sense disambiguation in lexical chaining. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*.

Robert Gunning. 1952. *The Technique of Clear Writing*. McGraw-Hill.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: An update. *SIGKDD Explorations*, 11(1):10–18.

Michael J. Heilman, Kevyn Collins-Thompson, Jamie Callan, and Maxine Eskenazi. 2007. Combining lexical and grammatical features to improve readability measures for first and second language texts. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*.

Michael J. Heilman, Kevyn Collins-Thompson, and Maxine Eskenazi. 2008. An analysis of statistical models and features for reading difficulty prediction. In *ACL 2008: The 3rd Workshop on Innovative Use of NLP for Building Educational Applications*.

Mirella Lapata and Regina Barzilay. 2005. Automatic evaluation of text coherence: Models and representations. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'05)*, pages 1085–1090.

Gondy Leroy, Stephen Helmreich, James R. Cowie, Trudi Miller, and Wei Zheng. 2008. Evaluating online health information: Beyond readability formulas. In *AMIA 2008 Symposium Proceedings*.

G. Harry McLaughlin. 1969. Smog grading a new readability formula. *Journal of Reading*, 12(8):639–646.

Sarah E. Petersen and Mari Ostendorf. 2006. A machine learning approach to reading level assessment. Technical report, University of Washington CSE Technical Report.

Sarah E. Petersen and Mari Ostendorf. 2009. A machine learning approach to reading level assessment. *Computer Speech and Language*, 23:89–106.

Emily Pitler and Ani Nenkova. 2008. Revisiting readability: A unified framework for predicting text quality. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*.

R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. Joshi, and B. Webber. 2008. The Penn discourse treebank. In *The Sixth International Conference on Language Resources and Evaluation (LREC'08)*.

Sarah E. Schwarm and Mari Ostendorf. 2005. Reading level assessment using support vector machines and statistical language models. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*.

Luo Si and Jamie Callan. 2001. A statistical model for scientific readability. In *Proceedings of the Tenth International Conference on Information and Knowledge Management*.

Radu Soricut and Daniel Marcu. 2006. Discourse generation using utility-trained coherence models. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*.