

## Learning to Predict Readability using Diverse Linguistic Features

Rohit J. Kate<sup>1</sup> Xiaoqiang Luo<sup>2</sup> Siddharth Patwardhan<sup>2</sup> Martin Franz<sup>2</sup>  
Radu Florian<sup>2</sup> Raymond J. Mooney<sup>1</sup> Salim Roukos<sup>2</sup> Chris Welty<sup>2</sup>

<sup>1</sup>Department of Computer Science

The University of Texas at Austin

{rjkate, mooney}@cs.utexas.edu

<sup>2</sup>IBM Watson Research Center

{xiaoluo, spatward, franzm, raduf, roukos, welty}@us.ibm.com

### Abstract

In this paper we consider the problem of building a system to predict readability of natural-language documents. Our system is trained using diverse features based on syntax and language models which are generally indicative of readability. The experimental results on a dataset of documents from a mix of genres show that the predictions of the learned system are more accurate than the predictions of naive human judges when compared against the predictions of linguistically-trained expert human judges. The experiments also compare the performances of different learning algorithms and different types of feature sets when used for predicting readability.

### 1 Introduction

An important aspect of a document is whether it is easily processed and understood by a human reader as intended by its writer, this is termed as the document's *readability*. Readability involves many aspects including grammaticality, conciseness, clarity, and lack of ambiguity. Teachers, journalists, editors, and other professionals routinely make judgements on the readability of documents. We explore the task of learning to automatically judge the readability of natural-language documents.

In a variety of applications it would be useful to be able to automate readability judgements. For example, the results of a web-search can be ordered taking into account the readability of the

retrieved documents thus improving user satisfaction. Readability judgements can also be used for automatically grading essays, selecting instructional reading materials, etc. If documents are generated by machines, such as summarization or machine translation systems, then they are prone to be less readable. In such cases, a readability measure can be used to automatically filter out documents which have poor readability. Even when the intended consumers of text are machines, for example, information extraction or knowledge extraction systems, a readability measure can be used to filter out documents of poor readability so that the machine readers will not extract incorrect information because of ambiguity or lack of clarity in the documents.

As part of the DARPA Machine Reading Program (MRP), an evaluation was designed and conducted for the task of rating documents for readability. In this evaluation, 540 documents were rated for readability by both experts and novice human subjects. Systems were evaluated based on whether they were able to match expert readability ratings better than novice raters. Our system learns to match expert readability ratings by employing regression over a set of diverse linguistic features that were deemed potentially relevant to readability. Our results demonstrate that a rich combination of features from syntactic parsers, language models, as well as lexical statistics all contribute to accurately predicting expert human readability judgements. We have also considered the effect of different genres in predicting readability and how the genre-specific language models can be exploited to improve the readability predictions.

## 2 Related Work

There is a significant amount of published work on a related problem: predicting the reading difficulty of documents, typically, as the school grade-level of the reader from grade 1 to 12. Some early methods measure simple characteristics of documents like average sentence length, average number of syllables per word, etc. and combine them using a linear formula to predict the grade level of a document, for example FOG (Gunning, 1952), SMOG (McLaughlin, 1969) and Flesh-Kincaid (Kincaid et al., 1975) metrics. These methods do not take into account the content of the documents. Some later methods use pre-determined lists of words to determine the grade level of a document, for example the Lexile measure (Stenner et al., 1988), the Fry Short Passage measure (Fry, 1990) and the Revised Dale-Chall formula (Chall and Dale, 1995). The word lists these methods use may be thought of as very simple language models. More recently, language models have been used for predicting the grade level of documents. Si and Callan (2001) and Collins-Thompson and Callan (2004) train unigram language models to predict grade levels of documents. In addition to language models, Heilman et al. (2007) and Schwarm and Ostendorf (2005) also use some syntactic features to estimate the grade level of texts.

Pitler and Nenkova (2008) consider a different task of predicting text quality for an educated adult audience. Their system predicts readability of texts from Wall Street Journal using lexical, syntactic and discourse features. Kanungo and Orr (2009) consider the task of predicting readability of web summary snippets produced by search engines. Using simple surface level features like the number of characters and syllables per word, capitalization, punctuation, ellipses etc. they train a regression model to predict readability values.

Our work differs from this previous research in several ways. Firstly, the task we have considered is different, we predict the readability of general documents, not their grade level. The documents in our data are also not from any single domain, genre or reader group, which makes our

task more general. The data includes human written as well as machine generated documents. The task and the data has been set this way because it is aimed at filtering out documents of poor quality for later processing, like for extracting machine-processable knowledge from them. Extracting knowledge from openly found text, such as from the internet, is becoming popular but the quality of text found “in the wild”, like found through searching the internet, vary considerably in quality and genre. If the text is of poor readability then it is likely to lead to extraction errors and more problems downstream. If the readers are going to be humans instead of machines, then also it is best to filter out poorly written documents. Hence identifying readability of general text documents coming from various sources and genres is an important task. We are not aware of any other work which has considered such a task.

Secondly, we note that all of the above approaches that use language models train a language model for each difficulty level using the training data for that level. However, since the amount of training data annotated with levels is limited, they can not train higher-order language models, and most just use unigram models. In contrast, we employ more powerful language models trained on large quantities of generic text (which is not from the training data for readability) and use various features obtained from these language models to predict readability. Thirdly, we use a more sophisticated combination of linguistic features derived from various syntactic parsers and language models than any previous work. We also present ablation results for different sets of features. Fourthly, given that the documents in our data are not from a particular genre but from a mix of genres, we also train genre-specific language models and show that including these as features improves readability predictions. Finally, we also show comparison between various machine learning algorithms for predicting readability, none of the previous work compared learning algorithms.

## 3 Readability Data

The readability data was collected and released by LDC. The documents were collected

from the following diverse sources or genres: newswire/newspaper text, weblogs, newsgroup posts, manual transcripts, machine translation output, closed-caption transcripts and Wikipedia articles. Documents for newswire, machine translation and closed captioned genres were collected automatically by first forming a candidate pool from a single collection stream and then randomly selecting documents. Documents for weblogs, newsgroups and manual transcripts were also collected in the same way but were then reviewed by humans to make sure they were not simply spam articles or something objectionable. The Wikipedia articles were collected manually, by searching through a data archive or the live web, using keyword and other search techniques. Note that the information about genres of the documents is not available during testing and hence was not used when training our readability model.

A total of 540 documents were collected in this way which were uniformly distributed across the seven genres. Each document was then judged for its readability by eight expert human judges. These expert judges are native English speakers who are language professionals and who have specialized training in linguistic analysis and annotation, including the machine translation post-editing task. Each document was also judged for its readability by six to ten naive human judges. These non-expert (naive) judges are native English speakers who are not language professionals (e.g. editors, writers, English teachers, linguistic annotators, etc.) and have no specialized language analysis or linguistic annotation training. Both expert and naive judges provided readability judgments using a customized web interface and gave a rating on a 5-point scale to indicate how readable the passage is (where 1 is lowest and 5 is highest readability) where readability is defined as a subjective judgment of how easily a reader can extract the information the writer or speaker intended to convey.

## 4 Readability Model

We want to answer the question whether a machine can accurately estimate readability as judged by a human. Therefore, we built a machine-learning system that predicts the read-

ability of documents by training on expert human judgements of readability. The evaluation was then designed to compare how well machine and naive human judges predict expert human judgements. In order to make the machine's predicted score comparable to a human judge's score (details about our evaluation metrics are in Section 6.1), we also restricted the machine scores to integers. Hence, the task is to predict an integer score from 1 to 5 that measures the readability of the document.

This task could be modeled as a multi-class classification problem treating each integer score as a separate class, as done in some of the previous work (Si and Callan, 2001; Collins-Thompson and Callan, 2004). However, since the classes are numerical and not unrelated (for example, the score 2 is in between scores 1 and 3), we decided to model the task as a regression problem and then round the predicted score to obtain the closest integer value. Preliminary results verified that regression performed better than classification. Heilman et al. (2008) also found that it is better to treat the readability scores as ordinal than as nominal. We take the average of the expert judge scores for each document as its gold-standard score. Regression was also used by Kanungo and Orr (2009), although their evaluation did not constrain machine scores to be integers.

We tested several regression algorithms available in the Weka<sup>1</sup> machine learning package, and in Section 6.2 we report results for several which performed best. The next section describes the numerically-valued features that we used as input for regression.

## 5 Features for Predicting Readability

Good input features are critical to the success of any regression algorithm. We used three main categories of features to predict readability: syntactic features, language-model features, and lexical features, as described below.

### 5.1 Features Based on Syntax

Many times, a document is found to be unreadable due to unusual linguistic constructs or ungram-

---

<sup>1</sup><http://www.cs.waikato.ac.nz/ml/weka/>

matical language that tend to manifest themselves in the syntactic properties of the text. Therefore, syntactic features have been previously used (Bernth, 1997) to gauge the “clarity” of written text, with the goal of helping writers improve their writing skills. Here too, we use several features based on syntactic analyses. Syntactic analyses are obtained from the Sundance shallow parser (Riloff and Phillips, 2004) and from the English Slot Grammar (ESG) (McCord, 1989).

**Sundance features:** The Sundance system is a rule-based system that performs a shallow syntactic analysis of text. We expect that this analysis over readable text would be “well-formed”, adhering to grammatical rules of the English language. Deviations from these rules can be indications of unreadable text. We attempt to capture such deviations from grammatical rules through the following Sundance features computed for each text document: proportion of sentences with no verb phrases, average number of clauses per sentence, average sentence length in tokens, average number of noun phrases per sentence, average number of verb phrases per sentence, average number of prepositional phrases per sentence, average number of phrases (all types) per sentence and average number of phrases (all types) per clause.

**ESG features:** ESG uses slot grammar rules to perform a deeper linguistic analysis of sentences than the Sundance system. ESG may consider several different interpretations of a sentence, before deciding to choose one over the other interpretations. Sometimes ESG’s grammar rules fail to produce a single complete interpretation of a sentence, in which case it generates partial parses. This typically happens in cases when sentences are ungrammatical, and possibly, less readable. Thus, we use the proportion of such incomplete parses within a document as a readability feature. In case of extremely short documents, this proportion of incomplete parses can be misleading. To account for such short documents, we introduce a variation of the above incomplete parse feature, by weighting it with a log factor as was done in (Riloff, 1996; Thelen and Riloff, 2002).

We also experimented with some other syntactic features such as average sentence parse scores from Stanford parser and an in-house maxi-

mum entropy statistical parser, average constituent scores etc., however, they slightly degraded the performance in combination with the rest of the features and hence we did not include them in the final set. One possible explanation could be that averaging diminishes the effect of low scores caused by ungrammaticality.

## 5.2 Features Based on Language Models

A probabilistic language model provides a prediction of how likely a given sentence was generated by the same underlying process that generated a corpus of training documents. In addition to a general n-gram language model trained on a large body of text, we also exploit language models trained to recognize specific “genres” of text. If a document is translated by a machine, or casually produced by humans for a weblog or newsgroup, it exhibits a character that is distinct from documents that go through a dedicated editing process (e.g., newswire and Wikipedia articles). Below we describe features based on generic as well as genre-specific language models.

**Normalized document probability:** One obvious proxy for readability is the score assigned to a document by a generic language model (LM). Since the language model is trained on well-written English text, it penalizes documents deviating from the statistics collected from the LM training documents. Due to variable document lengths, we normalize the document-level LM score by the number of words and compute the normalized document probability  $NP(\mathcal{D})$  for a document  $\mathcal{D}$  as follows:

$$NP(\mathcal{D}) = (P(\mathcal{D}|\mathcal{M}))^{\frac{1}{|\mathcal{D}|}}, \quad (1)$$

where  $\mathcal{M}$  is a general-purpose language model trained on clean English text, and  $|\mathcal{D}|$  is the number of words in the document  $\mathcal{D}$ .

**Perplexities from genre-specific language models:** The usefulness of LM-based features in categorizing text (McCallum and Nigam, 1998; Yang and Liu, 1999) and evaluating readability (Collins-Thompson and Callan, 2004; Heilman et al., 2007) has been investigated in previous work. In our experiments, however, since documents were acquired through several different channels, such as machine translation or web logs,

we also build models that try to predict the genre of a document. Since the genre information for many English documents is readily available, we trained a series of genre-specific 5-gram LMs using the modified Kneser-Ney smoothing (Kneser and Ney, 1995; Stanley and Goodman, 1996). Table 1 contains a list of a base LM and genre-specific LMs.

Given a document  $\mathcal{D}$  consisting of tokenized word sequence  $\{w_i : i = 1, 2, \dots, |\mathcal{D}|\}$ , its perplexity  $L(\mathcal{D}|\mathcal{M}_j)$  with respect to a LM  $\mathcal{M}_j$  is computed as:

$$L(\mathcal{D}|\mathcal{M}_j) = e^{\left(-\frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \log P(w_i|h_i;\mathcal{M}_j)\right)}, \quad (2)$$

where  $|\mathcal{D}|$  is the number of words in  $\mathcal{D}$  and  $h_i$  are the history words for  $w_i$ , and  $P(w_i|h_i;\mathcal{M}_j)$  is the probability  $\mathcal{M}_j$  assigns to  $w_i$ , when it follows the history words  $h_i$ .

**Posterior perplexities from genre-specific language models:** While perplexities computed from genre-specific LMs reflect the absolute probability that a document was generated by a specific model, a model’s *relative* probability compared to other models may be a more useful feature. To this end, we also compute the posterior perplexity defined as follows. Let  $\mathcal{D}$  be a document,  $\{\mathcal{M}_i\}_{i=1}^G$  be  $G$  genre-specific LMs, and  $L(\mathcal{D}|\mathcal{M}_i)$  be the perplexity of the document  $\mathcal{D}$  with respect to  $\mathcal{M}_i$ , then the posterior perplexity,  $R(\mathcal{M}_i|\mathcal{D})$ , is defined as:

$$R(\mathcal{M}_i|\mathcal{D}) = \frac{L(\mathcal{D}|\mathcal{M}_i)}{\sum_{j=1}^G L(\mathcal{D}|\mathcal{M}_j)}. \quad (3)$$

We use the term “posterior” because if a uniform prior is adopted for  $\{\mathcal{M}_i\}_{i=1}^G$ ,  $R(\mathcal{M}_i|\mathcal{D})$  can be interpreted as the posterior probability of the genre LM  $\mathcal{M}_i$  given the document  $\mathcal{D}$ .

### 5.3 Lexical Features

The final set of features involve various lexical statistics as described below.

**Out-of-vocabulary (OOV) rates:** We conjecture that documents containing typographical errors (e.g., for closed-caption and web log documents) may receive low readability ratings. Therefore, we compute the OOV rates of a document with respect to the various LMs shown in Table 1. Since

modern LMs often have a very large vocabulary, to get meaningful OOV rates, we truncate the vocabularies to the top (i.e., most frequent) 3000 words. For the purpose of OOV computation, a document  $\mathcal{D}$  is treated as a sequence of tokenized words  $\{w_i : i = 1, 2, \dots, |\mathcal{D}|\}$ . Its OOV rate with respect to a (truncated) vocabulary  $\mathcal{V}$  is then:

$$OOV(\mathcal{D}|\mathcal{V}) = \frac{\sum_{i=1}^{|\mathcal{D}|} I(w_i \notin \mathcal{V})}{|\mathcal{D}|}, \quad (4)$$

where  $I(w_i \notin \mathcal{V})$  is an indicator function taking value 1 if  $w_i$  is not in  $\mathcal{V}$ , and 0 otherwise.

**Ratio of function words:** A characteristic of documents generated by foreign speakers and machine translation is a failure to produce certain function words, such as “the,” or “of.” So we predefine a small set of function words (mainly English articles and frequent prepositions) and compute the ratio of function words over the total number words in a document:

$$RF(\mathcal{D}) = \frac{\sum_{i=1}^{|\mathcal{D}|} I(w_i \in \mathcal{F})}{|\mathcal{D}|}, \quad (5)$$

where  $I(w_i \in \mathcal{F})$  is 1 if  $w_i$  is in the set of function words  $\mathcal{F}$ , and 0 otherwise.

**Ratio of pronouns:** Many foreign languages that are source languages of machine-translated documents are pronoun-drop languages, such as Arabic, Chinese, and romance languages. We conjecture that the pronoun ratio may be a good indicator whether a document is translated by machine or produced by humans, and for each document, we first run a POS tagger, and then compute the ratio of pronouns over the number of words in the document:

$$RP(\mathcal{D}) = \frac{\sum_{i=1}^{|\mathcal{D}|} I(POS(w_i) \in \mathcal{P})}{|\mathcal{D}|}, \quad (6)$$

where  $I(POS(w_i) \in \mathcal{P})$  is 1 if the POS tag of  $w_i$  is in the set of pronouns,  $\mathcal{P}$ , and 0 otherwise.

**Fraction of known words:** This feature measures the fraction of words in a document that occur either in an English dictionary or a gazetteer of names of people and locations.

## 6 Experiments

This section describes the evaluation methodology and metrics and presents and discusses our

Genre	Training Size(M tokens)	Data Sources
base	5136.8	mostly LDC’s GigaWord set
NW	143.2	newswire subset of base
NG	218.6	newsgroup subset of base
WL	18.5	weblog subset of base
BC	1.6	broadcast conversation subset of base
BN	1.1	broadcast news subset of base
wikipedia	2264.6	Wikipedia text
CC	0.1	closed caption
ZhEn	79.6	output of Chinese to English Machine Translation
ArEn	126.8	output of Arabic to English Machine Translation

Table 1: Genre-specific LMs: the second column contains the number of tokens in LM training data (in million tokens).

experimental results. The results of the official evaluation task are also reported.

### 6.1 Evaluation Metric

The evaluation process for the DARPA MRP readability test was designed by the evaluation team led by SAIC. In order to compare a machine’s predicted readability score to those assigned by the expert judges, the Pearson correlation coefficient was computed. The mean of the expert-judge scores was taken as the gold-standard score for a document.

To determine whether the machine predicts scores closer to the expert judges’ scores than what an average naive judge would predict, a sampling distribution representing the underlying novice performance was computed. This was obtained by choosing a random naive judge for every document, calculating the Pearson correlation coefficient with the expert gold-standard scores and then repeating this procedure a sufficient number of times (5000). The upper critical value was set at 97.5% confidence, meaning that if the machine performs better than the upper critical value then we reject the null hypothesis that machine scores and naive scores come from the same distribution and conclude that the machine performs significantly better than naive judges in matching the expert judges.

### 6.2 Results and Discussion

We evaluated our readability system on the dataset of 390 documents which was released earlier during the training phase of the evaluation task. We

Algorithm	Correlation
Bagged Decision Trees	0.8173
Decision Trees	0.7260
Linear Regression	0.7984
SVM Regression	0.7915
Gaussian Process Regression	0.7562
Naive Judges	
Upper Critical Value	0.7015
Distribution Mean	0.6517
Baselines	
Uniform Random	0.0157
Proportional Random	-0.0834

Table 2: Comparing different algorithms on the readability task using 13-fold cross-validation on the 390 documents using all the features. Exceeding the upper critical value of the naive judges’ distribution indicates statistically significantly better predictions than the naive judges.

used stratified 13-fold cross-validation in which the documents from various genres in each fold was distributed in roughly the same proportion as in the overall dataset. We first conducted experiments to test different regression algorithms using all the available features. Next, we ablated various feature sets to determine how much each feature set was contributing to making accurate readability judgements. These experiments are described in the following subsections.

#### 6.2.1 Regression Algorithms

We used several regression algorithms available in the Weka machine learning package and Table 2 shows the results obtained. The default values

Feature Set	Correlation
Lexical	0.5760
Syntactic	0.7010
Lexical + Syntactic	0.7274
Language Model based	0.7864
All	0.8173

Table 3: Comparison of different linguistic feature sets.

in Weka were used for all parameters, changing these values did not show any improvement. We used decision tree (reduced error pruning (Quinlan, 1987)) regression, decision tree regression with bagging (Breiman, 1996), support vector regression (Smola and Scholkopf, 1998) using polynomial kernel of degree two,<sup>2</sup> linear regression and Gaussian process regression (Rasmussen and Williams, 2006). The distribution mean and the upper critical values of the correlation coefficient distribution for the naive judges are also shown in the table.

Since they are above the upper critical value, all algorithms predicted expert readability scores significantly more accurately than the naive judges. Bagged decision trees performed slightly better than other methods. As shown in the following section, ablating features affects predictive accuracy much more than changing the regression algorithm. Therefore, on this task, the choice of regression algorithm was not very critical once good readability features are used. We also tested two simple baseline strategies: predicting a score uniformly at random, and predicting a score proportional to its frequency in the training data. As shown in the last two rows of Table 2, these baselines perform very poorly, verifying that predicting readability on this dataset as evaluated by our evaluation metric is not trivial.

### 6.2.2 Ablations with Feature Sets

We evaluated the contributions of different feature sets through ablation experiments. Bagged decision-tree was used as the regression algorithm in all of these experiments. First we compared syntactic, lexical and language-model based features as described in Section 5, and Table 3 shows

<sup>2</sup>Polynomial kernels with other degrees and RBF kernel performed worse.

the results. The language-model feature set performs the best, but performance improves when it is combined with the remaining features. The lexical feature set by itself performs the worst, even below the naive distribution mean (shown in Table 2); however, when combined with syntactic features it performs well.

In our second ablation experiment, we compared the performance of genre-independent and genre-based features. Since the genre-based features exploit knowledge of the genres of text used in the MRP readability corpus, their utility is somewhat tailored to this specific corpus. Therefore, it is useful to evaluate the performance of the system when genre information is not exploited. Of the lexical features described in subsection 5.3, the ratio of function words, ratio of pronoun words and all of the out-of-vocabulary rates except for the base language model are genre-based features. Out of the language model features described in the Subsection 5.2, all of the perplexities except for the base language model and all of the posterior perplexities<sup>3</sup> are genre-based features. All of the remaining features are genre-independent. Table 4 shows the results comparing these two feature sets. The genre-based features do well by themselves but the rest of the features help further improve the performance. While the genre-independent features by themselves do not exceed the upper critical value of the naive judges' distribution, they are very close to it and still outperform its mean value. These results show that for a dataset like ours, which is composed of a mix of genres that themselves are indicative of readability, features that help identify the genre of a text improve performance significantly.<sup>4</sup> For applications mentioned in the introduction and related work sections, such as filtering less readable documents from web-search, many of the input documents could come from some of the common genres considered in our dataset.

In our final ablation experiment, we evaluated

<sup>3</sup>Base model for posterior perplexities is computed using other genre-based LMs (equation 3) hence it can not be considered genre-independent.

<sup>4</sup>We note that none of the genre-based features were trained on supervised readability data, but were trained on readily-available large unannotated corpora as shown in Table 1.

Feature Set	Correlation
Genre-independent	0.6978
Genre-based	0.7749
All	0.8173

Table 4: Comparison of genre-independent and genre-based feature sets.

Feature Set	By itself	Ablated from All
Sundance features	0.5417	0.7993
ESG features	0.5841	0.8118
Perplexities	0.7092	0.8081
Posterior perplexities	0.7832	0.7439
Out-of-vocabulary rates	0.3574	0.8125
All	0.8173	-

Table 5: Ablations with some individual feature sets.

the contribution of various individual feature sets. Table 5 shows that posterior perplexities perform the strongest on their own, but without them, the remaining features also do well. When used by themselves, some feature sets perform below the naive judges’ distribution mean, however, removing them from the rest of the feature sets degrades the performance. This shows that no individual feature set is critical for good performance but each further improves the performance when added to the rest of the feature sets.

### 6.3 Official Evaluation Results

An official evaluation was conducted by the evaluation team SAIC on behalf of DARPA in which three teams participated including ours. The evaluation task required predicting the readability of 150 test documents using the 390 training documents. Besides the correlation metric, two additional metrics were used. One of them computed for a document the difference between the average absolute difference of the naive judge scores from the mean expert score and the absolute difference of the machine’s score from the mean expert score. This was then averaged over all the documents. The other one was “target hits” which measured if the predicted score for a document fell within the width of the lowest and the highest expert scores for that document, and if so, com-

System	Correl.	Avg. Diff.	Target Hits
Our (A)	0.8127	0.4844	0.4619
System B	0.6904	0.3916	0.4530
System C	0.8501	0.5177	0.4641
Upper CV	0.7423	0.0960	0.3713

Table 6: Results of the systems that participated in the DARPA’s readability evaluation task. The three metrics used were correlation, average absolute difference and target hits measured against the expert readability scores. The upper critical values are for the score distributions of naive judges.

puted a score inversely proportional to that width. The final target hits score was then computed by averaging it across all the documents. The upper critical values for these metrics were computed in a way analogous to that for the correlation metric which was described before. Higher score is better for all the three metrics. Table 6 shows the results of the evaluation. Our system performed favorably and always scored better than the upper critical value on each of the metrics. Its performance was in between the performance of the other two systems. The performances of the systems show that the correlation metric was the most difficult of the three metrics.

## 7 Conclusions

Using regression over a diverse combination of syntactic, lexical and language-model based features, we built a system for predicting the readability of natural-language documents. The system accurately predicts readability as judged by linguistically-trained expert human judges and exceeds the accuracy of naive human judges. Language-model based features were found to be most useful for this task, but syntactic and lexical features were also helpful. We also found that for a corpus consisting of documents from a diverse mix of genres, using features that are indicative of the genre significantly improve the accuracy of readability predictions. Such a system could be used to filter out less readable documents for machine or human processing.

### Acknowledgment

This research was funded by Air Force Contract FA8750-09-C-0172 under the DARPA Machine Reading Program.



## References

- Bernth, Arendse. 1997. Easyenglish: A tool for improving document quality. In *Proceedings of the fifth conference on Applied Natural Language Processing*, pages 159–165, Washington DC, April.
- Breiman, Leo. 1996. Bagging predictors. *Machine Learning*, 24(2):123–140.
- Chall, J.S. and E. Dale. 1995. *Readability Revisited: The New Dale-Chall Readability Formula*. Brookline Books, Cambridge, MA.
- Collins-Thompson, Kevyn and James P. Callan. 2004. A language modeling approach to predicting reading difficulty. In *Proc. of HLT-NAACL 2004*, pages 193–200.
- Fry, E. 1990. A readability formula for short passages. *Journal of Reading*, 33(8):594–597.
- Gunning, R. 1952. *The Technique of Clear Writing*. McGraw-Hill, Cambridge, MA.
- Heilman, Michael, 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 *Proc. of NAACL-HLT 2007*, pages 460–467, Rochester, New York, April.
- Heilman, Michael, Kevyn Collins-Thompson, and Maxine Eskenazi. 2008. An analysis of statistical models and features for reading difficulty prediction. In *Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications*, pages 71–79, Columbus, Ohio, June. Association for Computational Linguistics.
- Kanungo, Tapas and David Orr. 2009. Predicting the readability of short web summaries. In *Proc. of WSDM 2009*, pages 202–211, Barcelona, Spain, February.
- Kincaid, J. P., R. P. Fishburne, R. L. Rogers, and B.S. Chissom. 1975. Derivation of new readability formulas for navy enlisted personnel. Technical Report Research Branch Report 8-75, Millington, TN: Naval Air Station.
- Kneser, Reinhard and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *Proc. of ICASSP-95*, pages 181–184.
- McCallum, Andrew and Kamal Nigam. 1998. A comparison of event models for naive Bayes text classification. In *Papers from the AAAI-98 Workshop on Text Categorization*, pages 41–48, Madison, WI, July.
- McCord, Michael C. 1989. Slot grammar: A system for simpler construction of practical natural language grammars. In *Proceedings of the International Symposium on Natural Language and Logic*, pages 118–145, May.
- McLaughlin, G. H. 1969. Smog: Grading: A new readability formula. *Journal of Reading*, 12:639–646.
- Pitler, Emily and Ani Nenkova. 2008. Revisiting readability: A unified framework for predicting text quality. In *Proc. of EMNLP 2008*, pages 186–195, Waikiki, Honolulu, Hawaii, October.
- Quinlan, J. R. 1987. Simplifying decision trees. *International Journal of Man-Machine Studies*, 27:221–234.
- Rasmussen, Carl and Christopher Williams. 2006. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA.
- Riloff, E. and W. Phillips. 2004. An introduction to the Sundance and Autoslog systems. Technical Report UUCS-04-015, University of Utah School of Computing.
- Riloff, Ellen. 1996. Automatically generating extraction patterns from untagged text. In *Proc. of 13th Natl. Conf. on Artificial Intelligence (AAAI-96)*, pages 1044–1049, Portland, OR.
- Schwarm, Sarah E. and Mari Ostendorf. 2005. Reading level assessment using support vector machines and statistical language models. In *Proc. of ACL 2005*, pages 523–530, Ann Arbor, Michigan.
- Si, Luo and James P. Callan. 2001. A statistical model for scientific readability. In *Proc. of CIKM 2001*, pages 574–576.
- Smola, Alex J. and Bernhard Scholkopf. 1998. A tutorial on support vector regression. Technical Report NC2-TR-1998-030, NeuroCOLT2.
- Stanley, Chen and Joshua Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proc. of the 34th Annual Meeting of the Association for Computational Linguistics (ACL-96)*, pages 310–318.
- Stenner, A. J., I. Horabin, D. R. Smith, and M. Smith. 1988. *The Lexile Framework*. Durham, NC: MetaMetrics.
- Thelen, M. and E. Riloff. 2002. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proc. of EMNLP 2002*, Philadelphia, PA, July.
- Yang, Yiming and Xin Liu. 1999. A re-examination of text categorization methods. In *Proc. of 22nd Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 42–48, Berkeley, CA.