A Pilot Study on Readability Prediction with Reading Time

Hitoshi Nishikawa, Toshiro Makino and **Yoshihiro Matsuo**
NTT Media Intelligence Laboratories, NTT Corporation
1-1 Hikari-no-oka, Yokosuka-shi, Kanagawa, 239-0847 Japan
{ nishikawa.hitoshi
makino.toshiro, matsuo.yoshihiro } @lab.ntt.co.jp

Abstract

In this paper we report the results of a pilot study of basing readability prediction on training data annotated with reading time. Although reading time is known to be a good metric for predicting readability, previous work has mainly focused on annotating the training data with subjective readability scores usually on a 1 to 5 scale. Instead of the subjective assessments of complexity, we use the more objective measure of reading time. We create and evaluate a predictor using the binary classification problem; the predictor identifies the better of two documents correctly with 68.55% accuracy. We also report a comparison of predictors based on reading time and on readability scores.

1 Introduction

Several recent studies have attempted to predict the readability of documents (Pitler and Nenkova, 2008; Burstein et al., 2010; Nenkova et al., 2010; Pitler et al., 2010; Tanaka-Ishii et al., 2010). Predicting readability has a very important role in the field of computational linguistics and natural language processing:

- Readability prediction can help users retrieve information from the Internet. If the readability of documents can be predicted, search engines can rank the documents according to readability, allowing users to access the information they need more easily (Tanaka-Ishii et al., 2010).

- The predicted readability of a document can be used as an objective function in natural language applications such as machine translation, automatic summarization, and document simplification. Machine translation can use a readability predictor as a part of the objective function to make more fluent translations (Nenkova et al., 2010). The readability predictor can also be used as a part of a summarizer to generate readable summaries (Pitler et al., 2010). Document simplification can help readers understand documents more easily by automatically rewriting documents that are not easy to read (Zhu et al., 2010; Woodsend and Lapata, 2011). This is possible by paraphrasing the sentences so as to maximize document readability.

- Readability prediction can be used for educational purposes (Burstein et al., 2010). It can assess human-generated documents automatically.

Most studies build a predictor that outputs a readability score (generally 1-5 scale) or a classifier or ranker that identifies which of two documents has the better readability. Using textual complexity to rank documents may be adequate for several applications in the fields of information retrieval, machine translation, document simplification, and the assessment of human-written documents. Approaches based on complexity, however, do not well support document summarization.

In the context of automatic summarization, users want concise summaries to understand the important information present in the documents *as rapidly as possible*— to create summaries that can be read as quickly as possible, we need a function that can evaluate the quality of the summary in terms of reading time.

Proceedings of the 2nd Workshop on Predicting and Improving Text Readability for Target Reader Populations, pages 78–84,
Sofia, Bulgaria, August 4-9 2013.

To achieve this goal, in this paper, we show the results of our pilot study on predicting the reading time of documents. Our predictor has two features as follows:

1. Our predictor is trained by documents directly annotated with reading time. While previous work employs subjective assessments of complexity, we directly use the reading time to build a predictor. As a predictor, we adopt Ranking SVM (Joachims, 2002).

2. The predictor predicts the reading time without recourse to features related to document length since our immediate goal is text summarization. A preliminary experiment confirms that document length is effective for readability prediction confirming the work by (Pitler and Nenkova, 2008; Pitler et al., 2010). Summarization demands that the predictor work well regardless of text length.

This is the first report to show that the result of training a predictor with data annotated by reading time is to improve the quality of automatic readability prediction. Furthermore, we report the result of the comparison between our reading time predictor and a conventional complexity-based predictor.

This paper is organized as follows: Section 2 describes related work. Section 3 describes the data used in the experiments. Section 4 describes our model. Section 5 elaborates the features for predicting document readability based on reading time. Section 6 reports our evaluation experiments. We conclude this paper and show future directions in Section 7.

2 Related Work

Recent work formulates readability prediction as an instance of a classification, regression, or ranking problem. A document is regarded as a mixture of complex features and its readability is predicted by the use of machine learning (Pitler and Nenkova, 2008; Pitler et al., 2010; Tanaka-Ishii et al., 2010). Pitler and Nenkova (2008) built a classifier that employs various features extracted from a document and newswire documents annotated with a readability score on a 1 to 5 scale. They integrated complex features by using SVM and identified the better document correctly with 88.88% accuracy. They reported that the log likelihood of a document based on its discourse relations, the log likelihood of a document based on n-gram, the average number of verb phrases in sentences, the number of words in the document were good indicators on which to base readability prediction. Pitler et al., (2010) used the same framework to predict the linguistic quality of a summary. In the field of automatic summarization, linguistic quality has been assessed manually and hence to automate the assessment is an important research problem (Pitler et al., 2010). A ranker based on Ranking SVM has been constructed (Joachims, 2002) and identified the better of two summaries correctly with an accuracy of around 90%. Tanaka-Ishii et al., (2010) also built a ranker to predict the rank of documents according to readability. While Tanaka-Ishii et al. used word-level features for the prediction, Pitler and Nenkova (2008) and Pitler et al., (2010) also leveraged sentence-level features and document-level features. In this paper, we extend their findings to predict readability. We elaborate our feature set in Section 5. While all of them either classify or rank the documents by assigning a readability score on a 1-5 scale, our research goal is to build a predictor that can also estimate the reading time.

In the context of multi-document summarization, the linguistic quality of a summary is predicted to order the sentences extracted from the original documents (Barzilay and Lapata, 2005; Lapata, 2006; Barzilay and Lapata, 2008). In multi-document summarization, since sentences are extracted from the original documents without regard for context, they must be ordered in some way to make the summary coherent. One of the most important features for ordering sentences is the entity grid suggested by Barzilay and Lapata (2005; 2008). It captures transitions in the semantic roles of the noun phrases in a document, and can predict the quality of an order of the sentences with high accuracy. It was also used as an important feature in the work by Pitler and Nenkova (2008) and Piter et al., (2010) to predict the readability of a document. Burstein et al., (2010) used it for an educational purpose, and used it to predict

the readability of essays. Lapata (Lapata, 2006) suggested the use of Kendall's Tau as an indicator of the quality of a set of sentences in particular order; she also reported that self-paced reading time is a good indicator of quality. While Lapata focuses on sentence ordering, our research goal is to predict the overall quality of a document in terms of reading time.

3 Data

To build a predictor that can estimate the reading time of a document, we made a collection of documents and annotated each with its reading time and readability score. We randomly selected 400 articles from Kyoto Text Corpus 4.0 [1]. The corpus consists of newswire articles written in Japanese and annotated with word boundaries, part-of-speech tags and syntactic structures. We developed an experimental system that showed articles for each subject and gathered reading times. Each article was read by 4 subjects. All subjects are native speakers of Japanese.

Basically, we designed our experiment following Pitler and Nenkova (2008). The subjects were asked to use the system to read the articles. They could read each document without a time limit, the only requirement being that they were to understand the content of the document. While the subjects were reading the article, the reading time was recorded by the system. We didn't tell the subjects that the time was being recorded.

To prevent the subjects from only partially reading the document and raise the reliability of the results, we made a multiple-choice question for each document; the answer was to be found in the document. This was used to weed out unreliable results.

After the subjects read the document, they were asked to answer the question.

Finally, the subjects were asked questions related to readability as follows:

1. How well-written is this article?
2. How easy was it to understand?
3. How interesting is this article?

Following the work by Pitler and Nenkova (2008), the subjects answered by selecting a value between 1 and 5, with 5 being the best and 1 being the worst and we used only the answer to the first question (How well-written is this article?) as the readability score. We dropped the results in which the subjects gave the wrong answer to the multiple-choice question. Finally, we had 683 tuples of documents, reading times, and readability scores.

[1] http://nlp.ist.i.kyoto-u.ac.jp/EN/

4 Model

To predict the readability of a document according to reading time, we use Ranking SVM (Joachims, 2002). A target document is converted to a feature vector as explained in Section 5, then the predictor ranks two documents. The predictor assigns a real number to a document as its score; ranking is done according to score. In this paper, a higher score means better readability, i.e., shorter reading time.

5 Features

In this section we elaborate the features used to predict the reading time. While most of them were introduced in previous work, see Section 3, the word level features are introduced here.

5.1 Word-level Features

Character Type (CT)

Japanese sentences consist of several types of characters: kanji, hiragana, katakana, and Roman letters. We use the ratio of the number of kanji to the number of hiragana as a feature of the document.

Word Familiarity (WF)

Amano and Kondo (2007) developed a list of words annotated with word familiarity; it indicates how familiar a word is to Japanese native speakers. The list is the result of a psycholinguistic experiment and the familiarity ranges from 1 to 7, with 7 being the most familiar and 1 being the least familiar. We used the average familiarity of words in the document as a feature.

5.2 Sentence-level Features

Language Likelihood (LL)

Language likelihood based on an n-gram language model is widely used to generate natural sentences. Intuitively, a sentence whose language likelihood is high will have good readability. We

made a trigram language model from 17 years (1991-2007) of Mainichi Shinbun Newspapers by using SRILM Toolkit. Since the language model assigns high probability to shorter documents, we normalized the probability by the number of words in a document.

Syntactic Complexity (TH/NB/NC/NP)

Schwarm and Ostendorf (2005) suggested that syntactic complexity of a sentence can be used as a feature for reading level assessment. We use the following features as indicators of syntactic complexity:

- The height of the syntax tree (TH): we use the height of the syntax tree as an indicator of the syntactic complexity of a sentence. Complex syntactic structures demand that readers make an effort to interpret them. We use the average, maximum and minimum heights of syntax trees in a document as a feature.

- The number of bunsetsu (NB): in Japanese dependency parsing, syntactic relations are defined between *bunsetsu*; they are almost the same as Base-NP (Veenstra, 1998) with postpositions. If a sentence has a lot of bunsetsu, it can have a complex syntactic structure. We use the average, maximum and minimum number of them as a feature.

- The number of commas (NC): a comma suggests a complex syntax structure such as subordinate and coordinate clauses. We use the average, maximum and minimum number of them as a feature.

- The number of predicates (NP): intuitively, a sentence can be syntactically complex if it has a lot of predicates. We use the average, maximum and minimum number of them as a feature.

5.3 Document-level Features

Discourse Relations (DR)

Pitler and Nenkova (2008) used discourse relations of the Penn Discourse Treebank (Prasad et al., 2008) as a feature. Since our corpus doesn't have human-annotated discourse relations between the sentences, we use the average number of connectives per sentence as a feature. Intuitively, the explicit discourse relations indicated by the connectives will yield better readability.

Entity Grid (EG)

Along with the previous work (Pitler and Nenkova, 2008; Pitler et al., 2010), we use entity grid (Barzilay and Lapata, 2005; Barzilay and Lapata, 2008) as a feature. We make a vector whose element is the transition probability between syntactic roles (i.e. subject, object and other) of the noun phrases in a document. Since our corpus consists of Japanese documents, we use postpositions to recognize the syntactic role of a noun phrase. Noun phrases with postpositions "Ha" and "Ga" are recognized as subjects. Noun phrases with postpositions "Wo" and "Ni" are recognized as objects. Other noun phrases are marked as other. We combine the entity grid vector to form a final feature vector for predicting reading time.

Lexical Cohesion (LC)

Lexical cohesion is one of the strongest features for predicting the linguistic quality of a summary (Pitler et al., 2010). Following their work, we leverage the cosine similarity of adjacent sentences as a feature. To calculate it, we make a word vector by extracting the content words (nouns, verbs and adjectives) from a sentence. The frequency of each word in the sentence is used as the value of the sentence vector. We use the average, maximum and minimum cosine similarity of the sentences as a feature.

6 Experiments

This section explains the setting of our experiment. As mentioned above, we adopted Ranking SVM as a predictor. Since we had 683 tuples (documents, reading time and readability scores), we made ${}_{683}C_2 = 232,903$ pairs of documents for Ranking SVM. Each pair consists of two documents where one has a shorter reading time than the other. The predictor learned which parameters were better at predicting which document would have the shorter reading time, i.e. higher score. We performed a 10-fold cross validation on the pairs consisting of the reading time explained in Section 3 and the features explained in Section 5. In order to analyze the contribution of each feature

Features	Accuracy
ALL	68.45
TH + EG + LC	**68.55**
Character Type (CT)	52.14
Word Familiarity (WF)	51.30
Language Likelihood (LL)	50.40
Height of Syntax Tree (TH)	61.86
Number of Bunsetsu (NB)	51.54
Number of Commas (NC)	47.07
Number of Predicates (NP)	52.82
Discourse Relations (DR)	48.04
Entity Grid (EG)	67.74
Lexical Cohesion (LC)	61.63
Document Length	69.40
Baseline	50.00

Table 1: Results of proposed reading time predictor.

to prediction accuracy, we adopted a linear kernel. The range of the value of each feature was normalized to lie between -1 and 1.

6.1 Classification based on reading time

Table 1 shows the results yielded by the reading time predictor. ALL indicates the accuracy achieved by the classifier with all features explained in Section 5. At the bottom of Table 1, Baseline shows the accuracy of random classification. As shown in Table 1, since the height of syntax tree, entity grid and lexical cohesion are good indicators for the prediction, we combined these features. TH + EG + LC indicates that this combination achieves the best performance.

As to individual features, most of them couldn't distinguish a better document from a worse one. CT, WF and LL show similar performance to Baseline. The reason why these features failed to clearer identify the better of the pair could be because the documents are newswire articles. The ratio between kanji and hiragana, CT, is similar in most of the articles and hence it couldn't identify the better document. Similarly, there isn't so much of a difference among the documents in terms of word familiarity, WF. The language model used, LL, was not effective against the documents tested but it is expected that it would useful if the target documents came from different fields.

Among the syntactic complexity features, TH offers the best performance. Since its learned feature weight is negative, the result shows that a higher syntax tree causes longer reading time. While TH has shows good performance, NB, NC and NP fail to offer any significant advantage. As with the word-level features, there isn't so much of a difference among the documents in terms of the values of these features. This is likely because most of the newswire articles are written by experts for a restricted field.

Among the document-level features, EG and LC show good performance. While Pitler and Nenkova (2008) have shown that the discourse relation feature is strongest at predicting the linguistic quality of a document, DR shows poor performance. Whereas they modeled the discourse relations by a multinomial distribution using human-annotated labels, DR was simply the number of connectives in the document. A more sophisticated approach will be needed to model discourse.

EG and LC show the best prediction performance of the single features, which agrees with previous work (Pitler and Nenkova, 2008; Pitler et al., 2010). While, as shown above, most of the sentence-level features don't have good discriminative performance, EG and LC work well. Since these features can work well in homogeneous documents like newswire articles, it is reasonable to expect that they will also work well in heterogeneous documents from various domains.

We also show the classification result achieved with document length. Piter and Nenkova (2008) have shown that document length is a strong indicator for readability prediction. We measure document length by three criteria: the number of characters, the number of words and the number of sentences in the document. We used these values as features and built a predictor. While the document length has the strongest classification performance, the predictor with TH + EG + LC shows equivalent performance.

6.2 Classification based on readability score

We also report that the result of the classification based on the readability score in Table 2. Along with the result of the reading time, we tested ALL and TH + EG + LC, and the single features. While DR shows poor classification performance in terms of reading time, it shows the best classi-

Features	Accuracy
ALL	57.25
TH + DR + EG + LC	56.51
TH + EG + LC	56.50
Character Type (CT)	51.96
Word Familiarity (WF)	51.50
Language Likelihood (LL)	50.68
Height of Syntax Tree (TH)	55.77
Number of Bunsetsu (NB)	52.99
Number of Commas (NC)	51.50
Number of Predicates (NP)	52.56
Discourse Relations (DR)	**58.14**
Entity Grid (EG)	56.14
Lexical Cohesion (LC)	55.77
Document Length	56.83
Baseline	50.00

Table 2: A result of classification based on readability score.

	Cor. coef.
Reading Time	0.822
Readability Score	0.445

Table 3: Correlation coefficients of the reading time and readability score between the subjects. We calculated the coefficient for each pair of subjects and then averaged them.

fication performance as regards readability score. Hence we add the result of TH + DR + EG + LC. It agrees with the findings showed by Pitler and Nenkova (2008) in which they have shown discourse relation is the best feature for predicting the readability score.

In general, the same features used for classification based on the reading time work well for predicting the readability score. TH and EG, LC have good prediction performance.

6.3 Variation in reading time vs. variation in readability score

We show the correlation between the subjects in terms of the variation in reading time and readability score in Table 3. As shown, the reading time shows much higher correlation (less variation) than the readability score. This agrees with the findings shown by Lapata (2006) in which the reading time is a better indicator for readability prediction. Since the readability score varies widely among the subjects, training becomes problematic with lowers predictor performance.

The biggest difference between the prediction of the reading time and readability score is the effect of feature DR. One hypothesis that could explain the difference is that the use of connectives works as a strong sign that the document has a good readability score—it doesn't necessarily imply that the document has good *readability*—for the subjects. That is, the subjects perceived the documents with more connectives as readable, however, those connectives contribute to the reading time. Of course, our feature about discourse relations is just based on their usage frequency and hence more precise modeling could improve performance.

7 Conclusion and Future Work

This paper has described our pilot study of readability prediction based on reading time. With automatic summarization in mind, we built a predictor that can predict the reading time, and readability, of a document. Our predictor identified the better of two documents with 68.55% accuracy without using features related to document length.

The following findings can be extracted from the results described above:

- The time taken to read documents can be predicted through existing machine learning technique and the features extracted from training data annotated with reading time (Pitler and Nenkova, 2008; Pitler et al., 2010).

- As Lapata (2006) has shown, reading time is a highly effective indicator of readability. In our experiment, reading time showed good agreement among the subjects and hence more coherent prediction results can be expected.

Future work must proceed in many directions:

1. Measuring more precise reading time is one important problem. One solution is to use an eye tracker; it can measure the reading time more accurately because it can capture when

the subject finishes reading a document. In order to prepare the data used in this paper, we set questions so as to identify and drop unreliable data. The eye tracker could alleviate this effort.

2. Testing the predictor in another domain is necessary for creating practical applications. We tested the predictor only in the domain of newswire articles, as described earlier, and different results might be recorded in domains other than newswire articles.

3. Improving the accuracy of the predictor is also important. There could be other features associated with readability prediction. We plan to explore other features.

4. Applying the predictor to natural language generation tasks is particularly important. We plan to integrate our predictor into a summarizer and evaluate its performance.

References

Shigeaki Amano and Tadahisa Kondo. 2007. Reliability of familiarity rating of ordinary japanese words for different years and places. *Behavior Research Methods*, 39(4):1008–1011.

Regina Barzilay and Mirella Lapata. 2005. Modeling local coherence: an entity-based approach. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL)*, pages 141–148.

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

Jill Burstein, Joel Tetreault, and Slava Andreyev. 2010. Using entity-based features to model coherence in student essays. In *Proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 681–684.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*, pages 133–142.

Mirella Lapata. 2006. Automatic evaluation of information ordering: Kendall's tau. *Computational Linguistics*, 32(4):471–484.

Ani Nenkova, Jieun Chae, Annie Louis, and Emily Pitler. 2010. Structural features for predicting the linguistic quality of text: Applications to machine translation, automatic summarization and human-authored text. In Emiel Krahmer and Theunem Mariet, editors, *Empirical Methods in Natural Language Generation: Data-oriented Methods and Empirical Evaluation*, pages 222–241. Springer.

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*, pages 186–195.

Emily Pitler, Annie Louis, and Ani Nenkova. 2010. Automatic evaluation of linguistic quality in multi-document summarization. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 544–554.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The penn discourse treebank 2.0. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC)*.

Sarah 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 (ACL)*, pages 523–530.

Kumiko Tanaka-Ishii, Satoshi Tezuka, and Hiroshi Terada. 2010. Sorting by readability. *Computational Linguistics*, 36(2):203–227.

Jorn Veenstra. 1998. Fast np chunking using memory-based learning techniques. In *Proceedings of the 8th Belgian-Dutch Conference on Machine Learning (Benelearn)*, pages 71–78.

Kristian Woodsend and Mirella Lapata. 2011. Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 409–420.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1353–1361.