2014-03-19

Automatic Readability Detection for Modern Standard Arabic

Jonathan Neil Forsyth
Brigham Young University - Provo

Automatic Readability Prediction for Modern Standard Arabic

Jonathan Neil Forsyth

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Arts

Deryle W. Lonsdale, Chair
Dee I. Gardner
R. Kirk Belnap

Department of Linguistics and English Language

Brigham Young University

March 2014

ABSTRACT

Automatic Readability Prediction for Modern Standard Arabic

Jonathan Neil Forsyth
Department of Linguistics and English Language, BYU
Master of Arts

Research for automatic readability prediction of text has increased in the last decade and has shown that various machine learning methods can effectively address this problem. Many researchers have applied machine learning to readability prediction for English, while Modern Standard Arabic (MSA) has received little attention. Here I describe a system which leverages machine learning to automatically predict the readability of MSA. I gathered a corpus comprising 179 documents that were annotated with the Interagency Language Roundtable (ILR) levels. Then, I extracted lexical and discourse features from each document. Finally, I applied the Tilburg Memory-Based Learning (TiMBL) machine learning system to read these features and predict the ILR level of each document using 10-fold cross validation for both 3-level and 5-level classification tasks and an 80/20 division for a 5-level classification task. I measured performance using the F-score. For 3-level and 5-level classifications my system achieved F-scores of 0.719 and 0.519 respectively. I discuss the implication of these results and the possibility of future development.

Keywords: readability, Modern Standard Arabic, machine learning

ACKNOWLEDGMENTS

I wish to express gratitude to friends, family, and mentors for their positive encouragement. This would not have been possible without them.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this thesis I address the subject of readability. In general, readability is the level of difficulty that a particular document presents to readers. Readers may be adults or children, and may be native or non-native speakers of the language in question. This compounds the difficulty for precisely defining readability, because readers and documents are always different for any given situation. Consequently, researchers in readability have developed many different ways of defining and measuring readability. A common metric for difficulty level or readability is in terms of the years of formal education that is required, on average, to comprehend a given document. For example, a paper published in a scientific journal about a specialized topic will require that a reader be informed about that topic at an advanced level—perhaps several years of study within the given field. In contrast a book written for 7-year-old schoolchildren may require less than a year of formal education for an adult second language learner. In order to narrow my topic I chose to look at readability from the perspective of a second language learner.

Formal second language education utilizes many reading materials. Textbooks alone do not provide enough material in order to learn to read fluently in the second language. Also, it may be safe to assume that students want material that is interesting to them. The challenge, then, is to match each individual learner's unique interests with their reading ability.

One solution to this challenge is to have language instructors select for the students material which is appropriate to their reading ability. This is a practical problem, because the instructor must spend significant time combing through documents to find those at a target readability level. The other problem is that the instructor must choose the reading, and it is impractical for him or her to choose reading for each individual student's topic of interest, if any effort to select texts based on topics of interest is made at all.

A second solution could have the student find appropriate reading materials at a trusted online source. This still may not avoid the problem of finding material that is interesting to the student. The student needs guidance to read specific subject matter because classes are geared toward developing professional language skills, which presupposes certain subject matter. A new and more efficient solution has recently become feasible.

With the onset of computer technology, readability measurement has recently become automated. Automatic readability measurement promises to ease the task of teachers in selecting reading material appropriate for their students' reading level. Furthermore an independent second language learner might find a collection of documents online, and with an automated readability program, determine which documents they should tackle for the most effective learning experience.

I am working with Arabic as a subject of readability. The Arabic language includes a diverse collection of dialects across the Middle East. A standard written form of Arabic used by the educated Arabic world is commonly called Modern Standard Arabic (MSA). My research focuses on MSA readability. MSA offers a special challenge to adults learning it as a second language. For example, MSA is a morphologically complex language that also exhibits very liberal sentence structure.

Some MSA college and university courses in the United States make heavy use of MSA newspaper writing in their curriculum. This is a convenient solution for providing vast amounts of material from which students can learn. However, the linguistic difficulty can render newspaper documents less effective in engaging the student in reading, and learners' competence in the subject area can be problematic. MSA newspaper concepts range from topics of political science and international relations to culture and business, among others.

MSA readability research is in its infancy. Unlike early English readability research, MSA has the advantage of benefiting from computer tools. These are only recently (in the last 10 years) being applied to measure readability for English and other languages. Much research is needed to understand the best practices for determining readability levels of MSA.

The last decade has seen application of computerized methods to the readability problem for English, French, Spanish, English for persons with disabilities, and some preliminary studies for MSA.

1.1 Overview

In Chapter 2 I review the history of readability research beginning with English. I include research for other languages and also show how the research and formulas for calculating readability have evolved over the last hundred years. I also discuss the recent application of machine learning to the readability problem. Then I describe how I used an MSA corpus and custom-developed tools for processing MSA documents, plus a tool for implementing machine learning. Next I describe my methodology, which combines MSA linguistic processing tools, a machine learning tool, an MSA frequency dictionary, and various features from the literature along with novel features to create a system for automatically predicting MSA document readability. I then present and discuss the performance results of my MSA readability system. Finally, I conclude with prospects for further MSA readability research in my system and by others.

Chapter 2

Literature Review

Since the beginning of readability research, English has been the principal language of investigation, although scholars have researched readability in several other languages, including French (François and Watrin, 2011), German (Hancke et al., 2012), and even MSA (Al-Khalifa and Al-Ajlan, 2010). This overview of the literature, therefore, takes English readability as a point of departure. Some researchers have shown that the factors affecting English readability can be useful for readability research in other languages, including MSA. This thesis will show that many of these features, as well as additional novel features, will transfer over well to MSA.

2.1 History of Readability Research

Early on in readability research, Sherman (1893) examined sentence length in his efforts to apply statistical analysis to English documents. Sherman observed that: 1) sentence length tends to be consistent for a given author, 2) sentence length has decreased over time in English writing. He hypothesized that shorter sentences and more concrete terms are easier for readers to understand. He believed that reporting statistical measures of English prose to students could be a useful way to engage them with documents and that these measures objectively indicated document difficulty. His methods of measurement proved to be applicable in subsequent readability research where average sentence length served as a proxy for syntactic complexity. Furthermore, his observation that authors use consistent sentence lengths was useful for early manual calculations of readability because they relied on samples of a document rather than the entire document to determine average sentence length.

Researchers have used various characteristics of a document for calculating readability. These include average sentence length or average word length (Flesch, 1948; Dale and Chall, 1948).

Thorndike's (1921) word frequency list functioned as another feature for predicting readability for English. It lists the 10,000 most frequent words from various English documents, but Thorndike did not specify the base corpus from which he generated this list. Lively and Pressey (1923) used Thorndike's list to rate the vocabulary difficulty of various documents used in elementary, middle school, and college-level U.S. educational institutions. They measured vocabulary burden in terms of the number of different word types, number of words not in Thorndike's list, and the weighted median index of the words from the same list. These metrics correlate fairly well with documents selected from these predetermined education levels. Following Thorndike, Horn (1928) oversaw a national survey of U.S. first grade students resulting in a 2,596-word list that these students knew. Dale (1931) derived a 769-word list by taking words which were common between Horn's list and the first 1,000 words of Thorndike's list. In the same publication, Dale noted a problem with all of these lists—they do not count words by their semantics but by their form only, which ignores that readers may know one meaning of a word form and not another.

McCall and Crabbs (1926) created a more robust criterion through testing student readers on comprehension of various English test passages. These test passages provided a criterion against which to measure subsequent readability formulas which used these tests for validating their formulas. Many researchers created formulas based on these lists and on many features including the surface features: average sentence length and average word length.

As a solution to the impracticality of applying multi-featured formulas, Lorge (1944) authored the Lorge formula. It simplified readability measurement by only including three features that were manually calculated. His features were proxies for syntax and lexical complexity and included the average sentence length, number of prepositional phrases per 100 words, and number of words not in the Dale List. He found these features to correlate well with the test results of McCall and Crabbs's (1926) grade school test experiments.

Efforts increased dramatically in the 1940s to find ways to improve and formalize readability measurements; prominent among these formulas were the Flesch Reading Ease formula (Flesch, 1948) and the Dale-Chall formula (Dale and Chall, 1948). Both formulas used the criteria from McCall and Crabbs to measure their results, and both used average sentence length. Flesch measured word length in syllables per word; he first tried to measure affixes per word, but found them to be less useful. Dale and Chall used a list of 3,000 words known to 80% of 4th graders surveyed on a larger list.

The Dale-Chall Formula remained popular for decades after this period and Dale and Chall revised it much later (1995). Flesch Reading Ease formula was revised by Kincaid et al. (1975) in a readability study sponsored by the U.S. Navy. This revised version, known as Flesh-Kincaid Grade level is still in wide use today.

Zipf (1949) conducted statistical research of lexical features that are widely applicable to readability. He showed that, for a given corpus, a small percentage of the word types are extremely common and that word rankings (according to a rank by frequency) are inversely related to word frequencies. He also noted that the most frequent words tend to have shorter written forms. This can explain the effectiveness of word rankings, ratio of a document's words found on a frequency list and word length for the formulas previously mentioned.

For a long time, readability prediction methods involved gathering document statistics through manual counts. This has limited the features available to readability formulas as well as their accuracy. Chall and Dale (1995) even promoted hand calculation methods as late as 1995.

Establishing a well-defined standard for readability measurements is difficult in large part because the reading audience and the reading material are different for each study. Readability formulas are not based on a single objective standard of measurement. Stenner (1996) developed the Lexile Framework as a solution to this problem. According to Chall and Dale (1995) the publication of the Lexile Theory (Stenner et al., 1987) restored interest in traditional readability methods and formulas which had waned since the 1970s. Lexile normalizes measurements in relation to the population being measured in order to establish a zero level that can be compared to other populations. To make this possible, Lexile measures individuals from different populations relative to their representative population's mean scores. For features, Lexile makes use of the mean of the raw frequency of a frequency list based on a 5-million word corpus of educational documents compiled by Carroll et al. (1971). For syntactic complexity, Lexile uses the mathematical log of average sentence length.

Lexile and other modern readability efforts use corpora for building readability models and as subjects of readability predictions. Corpora are collections of documents that may be annotated with additional information about the document such as the syntactic structure of sentences or the part-of-speech of words. Researchers use the terms 'type' and 'token' to refer to the items being counted in a corpus which may include words and punctuation. A token is an occurrence of any counted item in a corpus; often, the count of all the words in a given corpus is considered the token count. However, those working with corpora must decide whether to include punctuation, numbers, and other items in the token count. The written or surface form of a token is known as its type. In other words, all tokens with the same surface form are counted as one distinct type regardless of how many tokens of that surface form appear in the corpus. Researchers must also decide whether to count different inflections of the same word as different types, for example whether to consider the plural form of a noun as a separate type from its singular form.

2.1.1 The ILR Levels As A Readability Metric

Starting in the 1950s, U.S. government agencies began development of standard language proficiency levels called the Inter-Agency Language Roundtable (ILR) Scale. The ILR system is used today to objectively determine different skill levels in the four areas of language proficiency—reading, writing, speaking, listening—for government positions requiring foreign language proficiency (Clark and Clifford, 1988; Lowe, 1987). The modern result of these efforts is shown in Table 2.1 taken from the published online rating system.[1] The ILR Scale represents a standard or list of categories to which documents may be classified according to their reading difficulty.

[1] http://www.govtilr.org/Skills/ILRscale4.htm

Reading 0	No Proficiency
Reading 0+	Memorized Proficiency
Reading 1	Elementary Proficiency
Reading 1+	Elementary Proficiency, Plus
Reading 2	Limited Working Proficiency
Reading 2+	Limited Working Proficiency, Plus
Reading 3	General Professional Proficiency
Reading 3+	General Professional Proficiency, Plus
Reading 4	Advanced Professional Proficiency
Reading 4+	Advanced Professional Proficiency, Plus
Reading 5	Functionally Native Proficiency

Table 2.1: *ILR Reading Levels*

2.2 Machine Learning

Looking at readability as a classification problem is useful for purposes of machine learning, which is a method for classifying things automatically. In working with corpora researchers often use machine learning to gather and classify the data into separate categories. For an in-depth understanding of machine learning see Witten et al. (2011).

Machine learning is applicable to many different kinds of statistical problems, both linguistic and non-linguistic and can take two general approaches—supervised or unsupervised learning. In supervised learning the data include distinct entities with preassigned labels, one for each entity, representing their respective class or category. The researcher's goal is to train an algorithm to 'learn' how to predict the correct category on new input through processing a collection of exemplary training entities. In unsupervised learning, the entities are not labeled with a category. A common goal of unsupervised learning is to find patterns in the data which group similar entities together.

Many machine learning specialists choose to split the data such that 80% of the data is used for training, and 20% is used for testing (or evaluation). This is called an 80/20 split, and often allows for a more precise model to be created of the testing data. Both the training and testing sets profit from a greater number of documents—the training set to improve the quality of training, and the testing set to increase the significance of the results. Thus the split that gives more data to one set decreases the effectiveness of the other. Another useful approach is to further split the training set into another 80/20 split and set aside the testing set, called the development test (devtest) set.

Another common way to split the data for training and testing is to use cross validation. In cross validation the researcher divides test data into a number of sections of equal size called folds; often they choose ten folds (10-fold cross validation), but there can be more or less. Then, they train the program on all but one fold. Finally, they run that one fold as the test and record the result. A variation of this is called leave-one-out in which they use all items (e.g. documents in the case of readability) except one for training, and they test the one document which was left out. They repeat this train and test procedure on all but one document until all documents have taken a turn as the test document. The main benefit of cross fold validation is that it maximizes the training set size. It also provides sound test results, based on many outcomes, which are more generalizable.

The performance of the classifications can be measured in various ways, but perhaps the most common is known as the F-score. Two performance measures related to the F-score are precision and recall. Precision is the percentage of predictions that were correct with respect to all predictions for a specific class. Recall is the percentage of documents with a specific preassigned class that were correctly predicted with their class label. The F-score captures both precision and recall in one measure and is also known as the harmonic mean of precision and recall. The F-score is roughly the same as accuracy, though it is more comprehensive. F-scores are calculated per class, then averaged to provide an overall F-score.

Machine learning methods are gaining interest in current research in automated readability. Previous methods—traditional readability formulas—only accounted for a few features because they required human counts. In contrast, a machine learning approach is scalable to a large number of features and can accept features generated automatically from electronic documents. Machine learning performs advanced computations that account for interactions between features, and this approach is optimal for development because one can easily adjust the number of features and retest efficiently. Repeated cycles of adjustments and retests can effectively determine the optimal feature set by comparing the results of different permutations of feature sets. For instance, Vajjala and Meurers (2012, p. 169) reported results of applying a machine learning algorithm to their corpus of 2500 documents and compared the performance of 12 different feature sets containing between 2 to 46 individual features in each set.

Si and Callan (2001) demonstrated that traditional formulas do not scale well to web documents when they reported the results of applying Flesch-Kincaid Grade level on their 91-document web corpus. Traditional formulas like Flesch-Kincaid need more data than they are afforded in many cases by new kinds of non-traditional documents such as web pages.

Additionally, traditional formulas do not have enough features to provide maximal accuracy, and were not intended to do so. Rather they were designed for the purpose of providing human raters a simple approximation of the difficulty of a given document. As an example, Chall and Dale (1995) report the revised version of the 1948 Dale-Chall formula and include a worksheet for a human rater to record counts that he/she gathers manually from samples of a document.

2.3 Machine Learning and Readability

Readability researchers often provide machine learning algorithms data which they derive from statistical pre-processing. A very common pre-processing technique is building N-gram models from a large corpus. N-grams are sequences or chains of words where N represents the length of the chain. N-grams with a length of one or two are called unigrams and bigrams respectively. N-gram models provide the statistical probability of certain N-grams occuring. These probabilities are calculated by the actual rate of occurrence of the N-grams in a model corpus.

Si and Callan (2001) created apparently the first machine learning model to address readability prediction. The data inputs to their model were the average sentence length and a unigram language model. They frame their approach as an attempt to solve the problem of returning web search results which correspond to a certain readability level. For features, they used a unigram language model and average sentence length as input features to a classifier in order to predict 3 different classes K-2, 3-5, 6-8 in terms of U.S. school grade levels. They downloaded their corpus, consisting of 91 documents, from the web. The authors of these web documents provided the grade level of each document. Si and Callan used 10 documents from each of the 3 levels for training. They applied the Flesch-Kincaid formula as a baseline (average sentence length and average word length are the only two features of Flesch-Kincaid). Flesch-Kincaid performed poorly, at 21.3% accuracy, while their unigram language model achieved 75.4% accuracy. However, Si and Callan do point out a problem with applying Flesch-Kincaid to their corpus because this traditional formula has a range of levels beyond their corpus. They explained the poor performance of Flesch-Kincaid as a function of web document writing style which often includes partial sentences—unlike traditional documents—and has insufficient prose length for Flesch-Kincaid to work properly. They also showed that the Flesch-Kincaid formula is not robust enough when applied to new, unconventional web documents which often include fragments of a document and show irregular or non-existent sentence boundaries. Several studies have applied machine learning algorithms to the readability prediction problem since this study as well as addressed the need for web search results to be automatically filtered for readability.

Building on Si and Callan, Collins-Thompson and Callan (2005) used a machine learning classifier and educational web documents for automated readability prediction, and they added smoothing to their unigram language model, which is a process for accounting for words with zero probability (i.e. words that have no occurrences in the corpus used to create the model). Their model made predictions at 12 U.S. school-grade levels—1st through 12th grades—and demonstrated a Pearson correlation of 0.79. Their corpus consisted of 550 documents with 415,330 tokens. The document genres covered fiction, nonfiction, history and science. Accuracy was measured with root mean squared error in order to count close predictions as less accurate than exact predictions rather than simple true/false predictions. The best result they obtained was a root mean square error of 1.92. This means that the predictions differed from the correct level by about 2 grade levels (1.92) on average.

Pitler and Nenkova (2008) showed that discourse connective features perform well in readability scoring and readability ranking for English. Discourse connectives are words and multi-word expressions that connect units of text larger than single words. For example, the English word 'however' is a discourse connective often used to connect two sentence units. The units connected by discourse connectives may be larger or smaller than full sentences. For their study, Pitler and Nenkova use the Wall Street Journal section of the Penn Discourse

Treebank (PDTB), a corpus annotated with discourse connective information (Prasad et al., 2008). They created an automated English readability assessment model that matched human ratings given by college students on various news articles from this corpus. They used a large feature set and found document length and discourse relations between clausal arguments of explicit discourse connectives to be the most significant factors in training a model to match discrete human ratings. Features also significant in their model were vocabulary probability—based on frequency lists that they generated from large corpora of newspaper documents—and the average verb phrases per sentence.

Weekly Reader is an English educational reading website[2] for children and their teachers and has served as a corpus for several machine learning studies in readability in recent years. The advantage of Weekly Reader as with the corpus gathered by Si and Callan is that authors provided human readability ratings, which avoids the expensive process of testing with human readers to establish a standard of measurement. Weekly Reader does not document what criteria the authors follow in production.

In one Weekly Reader study, Feng et al. (2010) used a machine learning program for classifying documents from a Weekly Reader corpus they had compiled previously (Feng et al., 2009) (1433 articles across 4 grade levels). They created language models for their study which were based on the same corpus that they were using to evaluate their model. In order to do this properly they used a 'hold-one-out' approach in which they trained the language model on all the training data and tested on the data that was left out in order to prevent bias. This overcomes the problem that Schwarm and Ostendorf (2005) reported in their Weekly Reader readability study in which performance deteriorated when they split their training/testing set such that there was not enough data for training to produce reliable judgments of classes/levels. Feng et al. used some of the same discourse features from their 2009 study for adults with intellectual disabilities. They also used 5 part-of-speech (POS) feature sets based on the following categories: nouns, verbs, adjectives, adverbs and prepositions. Each set had 5 features among which were percentage of the given POS tokens and percentage of the given POS types per document. Their noun-based POS feature set was the best performing out of all the POS-based feature sets—achieving alone 58.15% accuracy.

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

This was expected, they reported, because noun phrases include entities that a reader must keep in working memory as she is reading a document.

In a study on German web documents, Hancke et al. (2012) use machine learning for making predictions. They gathered electronic documents from 2 German news sites which were written for adults and children aged 8-14, Geo[3] and GeoLino[4] respectively. Their system made binary predictions between these two document sources. They ranked their top ten features in terms of informativeness: the top three were: 1) average word length (traditional feature) 2) ratio of the number of 2nd person verbs to the number of finite verbs (novel feature), and 3) number of syllables per word (traditional feature). They found that morphological features proved very telling of readability in a binary classification task of German online news documents. Their classifier achieved 85.4% classification accuracy based on a set of morphological features alone.

2.4 Arabic Readability

Arabic readability research is in its very early stages; Shen et al. (2013) published a recent example. They used machine learning to create general classifiers for 4 languages: Arabic, English, Dari, and Pashto. Their feature set is small and includes traditional, language-independent features. The 2 categories of features they label are: 1) word usage and 2) shallow length features. The word usage category only includes weighted word frequency from their training corpus. The shallow length category includes average sentence length in words, number of words per document, and average word length in characters. They normalized these three length features using a method to make the scores more comparable. They split their corpus (including the Arabic section) into an 80/20 training/testing split. Their corpus consisted of 1,394 documents across 7 of the 11 ILR levels: 1, 1+, 2, 2+, 3, 3+ and 4. The number of documents was relatively equal across these classes with about 200 documents per class. They reported results in terms of root mean squared error. Their data set is much larger than mine, though it is comparable because they also use a cor-

[3] http://www.geo.de

[4] http://www.geolino.de

pus annotated with ILR levels. They could improve their feature set by modeling linguistic characteristics of Arabic more deeply.

In a preliminary study, Al-Khalifa and Al-Ajlan (2010) collected a document corpus from educational materials for elementary, intermediate, and secondary schools in Saudi Arabia. Their corpus had 150 documents—50 documents from each level—comprising 57,089 tokens. They processed the documents of their corpus to generate 5 features: average sentence length, average word length in letters and syllables, term frequency (ratio of duplicated words), and an N-gram (bigram) language model. They built 3-way machine learning classifiers to classify the documents in this corpus across the 3 education levels. For evaluation they used an 80/20 training/testing split. They compared performance between two different sets of features—their entire feature set and a subset of the best three features: average sentence length, the bigram language model, and term frequency. To measure performance they used the F-score. They averaged the F-scores across the three levels to give an overall accuracy. Table 2.2 summarizes their results. The results set a fairly high preliminary standard for performance in Arabic readability classifiers.

Level	**All Features**	**Feature Subset**
Easy	1.00	1.00
Medium	0.545	0.667
Difficult	0.615	0.667
Average F-score	0.720	0.778

Table 2.2: *F-scores for Classification Using All Features and the Best-Performing Subset*

2.5 Arabic Features

I have described several features that researchers have applied to readability. These features may also be effectively applied to MSA readability. One traditional feature discussed above is word length in terms of syllables, affixes, or characters per word. Because Arabic is so rich in its morphology, the number of morphemes may be an informative readability feature. Habash (2010) explains that in Arabic morphology, multiple affixes and clitics that can be attached to a lexical stem. He uses an example of a verb stem plus multiple attachments shown in Figure 2.1, which I took from Habash (2010).

wasayaktubuwnahA
wa+ sa+ y+ aktub +uwna +hA
and will 3person write masculine-plural it
'and they will write it'

Figure 2.1: *Example of MSA Morphological Complexity*

Figure 2.1 shows optional morphemes including the conjunction word ‘and’, the future marker, and the direct object attached to the end, 'it'. As with verb stems, noun stems can also take several affixes and clitics. These morphemes must be parsed by a reader and thus may add to the difficulty in comprehension.

Arabic commonly uses a noun phrase structure called the idafa. An idafa is a chain of two or more syntactically bound nouns. In MSA curriculum the idafa is sometimes called the genitive construct among other terms. According to Ryding, “[t]he noun-noun genitive construct is one of the most basic structures in the Arabic language and occurs with high frequency” (see Ryding, 2005, chapter 8). Idafas have no prepositions between their noun members. They are versatile in use and often translate into English as compound words or phrases expressing possession. Furthermore, idafa chains can become very long, particularly in genres that are scientific, political, or otherwise intended for a highly educated readership. I provide an example of a five-term idafa construction in Figure 2.2 taken from Ryding (2005, p.216):

taTbiiq-u jamii^c-i qaraaraat-i majlis-i l-'amn-i
the application of all of the resolutions of the Security Council

Figure 2.2: *A Five-Term Idafa Noun Construction*

Another complex construction in Arabic is the discourse connective. Al-Batal (1990) has studied these lexical structures extensively and wrote that “MSA seems to have a connecting constraint that requires the writer to signal continuously to the reader, through the use of connectives, the type of link that exists between different parts of the document. This gives the connectives special importance as text-building elements and renders them essential for the reader's processing of text" (p 256). These connectives frequently take the place of punctuation. Furthermore, punctuation is often unreliable and inconsistent for demarcating sentence boundaries in MSA.

Alsaif and Markert (2011) developed the first model for automatic discourse connective identification for MSA. They based the features of their model on a corpus that they created themselves—the Leeds Arabic Discourse Treebank (LADTB; Alsaif and Markert, 2010)—an excerpt of the Penn Arabic Tree Bank (Maamouri and Bies, 2004), which comprises newspaper documents. Alsaif and Markert found through comparing their corpus to the PDTB that discourse connectives for MSA are more ambiguous than their English counterparts. They also report that Arabic writers make use of explicit discourse connectives much more frequently in the newspaper genre than is found in the same genre for English. In comparing the two corpora, they found that 70% of sentence pairings in the LADTB are connected with an explicit discourse connective while the same statistic of the PDTB is only 12%. Even leaving out the most common discourse connective, /wa/ ('and'), the LADTB shows 30% of sentences with an explicit discourse connective (Alsaif and Markert, 2011, pp. 740-741). Therefore, Arabic discourse connectives may be very valuable in matching human-rated readability baseline as Pitler and Nenkova show that they are for English. They may be even more significant for MSA over English given the more frequent use of connectives in the LADTB compared to the PDTB.

Alsaif and Markert published their comprehensive discourse connective list online[5]. Their purpose was to disambiguate the authentic use of discourse forms as functional discourse connectives from the same forms that were not functional as discourse connectives.

I've explained where readability research started and how it has advanced to using computational methods to automatically detect readability level. I also looked at several features that I believe will be useful in deriving readability levels automatically from MSA documents. In the following chapter I will discuss the tools I used for extracting useful

[5] http://www.arabicdiscourse.net/annotation-tool/

linguistic information from MSA documents and for computing accurate predictions for readability levels of the same.

Chapter 3

Resources & Methods

In the present chapter I describe the resources and methods that I employed to build and evaluate an automated readability prediction system for MSA. My resources include a corpus of documents annotated for readability level, a frequency dictionary of MSA, feature extraction tools, and a supervised machine learning system. I explain each and how I applied them to readability prediction of MSA documents.

3.1 The Corpus

In May 2013 I downloaded a corpus from the online curriculum[1] of the Defense Language Institute (DLI) Foreign Language Center, the principal language educational institution for members of the U.S. military. Their site is open to the public and includes several language instruction materials. The DLI corpus contains documents which are based on authentic MSA materials, ranked by the authors according to the Inter-agency Language Round table (ILR) standard levels. As explained earlier, the ILR levels comprise eleven proficiency levels (Clark and Clifford, 1988; Lowe, 1987). The DLI corpus includes five of these proficiency levels: 1, 1+, 2, 2+, and 3 from easiest to most difficult. The corpus has a total of 179 documents and 74,776 tokens. Shen et al. (2013) used a similar corpus, also rated by the ILR standard and also originating from the DLI, in their readability study described above. There is not an even distribution of corpus documents across these levels (See Table 3.1).

I randomly partitioned the DLI Corpus into three sections: training, development test (devtest), and evaluation—a common method in machine learning experiments to prepare for a final experiment. The training and development test sets together comprised 80% of

[1] http://gloss.dliflc.edu/Default.aspx

Level	**# of Texts**	**Train**	**DevTest**	**Evaluation**
1	20	13	3	4
1+	14	9	2	3
2	80	51	13	16
2+	40	26	6	8
3	25	16	4	5
Total	179	115	28	36

Table 3.1: *DLI Corpus Document Levels and Distributions*

the corpus with 58,565 tokens, and the evaluation set comprised the remaining 20% with 16,211 tokens. The training and development test sets are further subdivided by another 80/20 split. The final number of documents in each partition can also be seen in Table 3.1. This allows for development and improvement of the classifier while preserving the integrity of the final results.

3.2 Feature Engineering

Previously, I discussed the use of features in readability research which are derived from different types of word lists. Modern frequency lists are more robust than early ones because they are typically based on multi-million word corpora. For my work I used 'A Frequency Dictionary of Arabic' (Buckwalter and Parkinson, 2011), a published 5000-word frequency dictionary based on a 30-million word corpus. 10% of the base corpus that the authors used to compile the dictionary consists of spontaneous spoken Arabic while 90% is a balanced collection from 5 subcategories: 1) daily newswire; 2) newspaper editorials, opinion essays, regular columns; 3) academic and formal writings; 4) internet discussion forums; 5) literary and fictional publications. Because dialectal Arabic is increasingly used in formal communication contexts which were typically restricted to MSA in the recent past, the authors try to account for the rise in dialectal Arabic. Dialectal words appear in the list of entries where such words are highly frequent and have a wide range across the 5 subcategories. Each dictionary entry represents a base form (or lemma) of a word which usually has several variations on this base form (e.g. the plural form of nouns).

Based on the lemma, part-of-speech, rank, raw frequency, and range provided by the frequency dictionary, I generated features for the corpus documents . Most of my readability features were derived from this dictionary. In order to identify and categorize words in a way which would allow me to compare them to the frequency list, I required an Arabic morphological processing tool.

MSA usage is often ambiguous in its written form. A single written form taken out of context frequently has multiple meanings. This ambiguity owes largely to the fact that authors of MSA rarely include most diacritics in written form. These include short vowels, elongated consonants, and other distinguishing orthography. Two processes are needed to handle morphological ambiguity in MSA. The first—morphological analysis—provides all possible morphological interpretations of a surface form taken out of context. The second—morphological disambiguation—chooses the correct morphological analysis given the context. I utilized a state-of-the-art program that achieves both tasks for MSA known as MADA[2] (Morphological Analysis and Disambiguation of Arabic; Habash et al., 2009). MADA provides full morphological disambiguation, part-of-speech tagging, English glosses and other useful information. MADA can group or separate morphemes in a variety of user specified ways—a process called tokenization. This tokenization is performed after full morphological analysis and disambiguation.

MADA's morphological analysis specifies, for each word, up to four proclitics, the lemma, and a possible enclitic. It includes 22 other features such as POS, full diacritization, and distinctive lemma code. In addition to these 22 features, MADA outputs a user customized tokenization of the original input document. I obtained all of my features with the support of MADA's features and tokenization output. I included lexical and discourse features in my model. Lexical features are a strong and consistent feature in all of the forementioned readability research with measurements based on word length, word lists, or N-gram language models. I employed word lengths, a word list, and novel lexical features in my model. I used discourse connective word ratios. I expected that the set of lexical features would perform the best.

[2] I used MADA version 3.2

Alsaif and Markert created a program to disambiguate discourse connective forms between their functional use as discourse connectives and non-functional use as discourse connectives, but their program is not publicly available. Therefore, I used those discourse connective surface forms which they found to be used more than 50% of the time as authentic discourse connectives in their corpus. Coordinated connective pairs, such as the English 'if then' are found in MSA also, but I exclude these from my subset of discourse connectives.

I chose to use full delineation of individual morphemes because these include several prepositions used in discourse connectives which would otherwise be ambiguous as I used a simple search pattern to find multi-word and single word discourse connectives. My use of discourse connective features was inspired by Pitler and Nenkova's application of discourse connectives to English readability (2008). Discourse connectives are very salient and important cohesive lexical items in MSA as mentioned previously.

For the main lexical feature of this model, I counted the number of MADA-lemmatized tokens in the document that appear in the 5000-lemma list of the frequency dictionary (Buckwalter and Parkinson, 2011). I was required to normalize this 5000-lemma list to match MADA's lemma codes. Then, I divided this count by the total number words in the document to obtain the frequency ratio feature. I counted only tokens which have a morphological analysis provided by MADA excluding punctuation. I borrowed a statistic from Lively and Pressey's early vocabulary study (1923) which is the median index from Thorndike's 10,000 frequent word list of the frequent words found in the document. Lexile uses the mean of the log word frequencies from their training corpus because these correlated best with the rankings of their criteria—the Peabody Individual Achievement Test (Dunn and Markwardt, 1970). I used both of these measures.

Buckwalter and Parkinson divided the corpus on which their dictionary is based into 191 logical subsections, and they provide a measure for each lexical entry showing its dispersion across these subsections. I derived various features from this measure because this is not used in previous research. For example, I took the average dispersion for each document's frequent lexical items. (See Appendix B for all dispersion-based features.)

I compared the words in the frequency dictionary with the MADA lexeme code to match against. This is how I counted the frequent tokens in a document. The dictionary has a register value for about half the entries, but I decided this was not enough to be useful as a readability feature. The dictionary entry information which I used includes rank, lemma, POS, gloss, and range.

The type-to-token ratio feature is common to several studies and I included it in my experiments. For English readability research the type-to-token ratio is calculated by counting the number of unique word types, and dividing them by the count of total running words or tokens. I necessarily altered this process of computing the type to token ratio for MSA because its lexical stems usually accept several different affixes and clitics. Even some function words accept clitic attachments. Therefore, two instances of the same stem would be counted as separate items if they had a single difference in their attached morphemes. This would certainly lead to a token counting problem in which similar tokens that differ only in an attached preposition would be counted as separate tokens. The solution I implemented was to count total unique lemmas and omit function words, then divide the lemma type count by the total lemma tokens. This decision excluded clitics and affixes from the type-to-token ratio. A derivative of this feature is the root of the type-to-token ratio which I borrowed from Vajjala and Meurers (2012).

I used the POS-based ratio features inspired by Vajjala and Meurers, but adjusted them to match MSA POS classes provided by MADA. As a reminder, Vajjala and Meurer's POS-based ratio features were the ratio of the individual POS type occurrences to the token count in the document. I believed that the noun type ratio would be the most informative for Arabic because noun features show the best success in previous research including that of Vajjala and Meurers. I included these feature names and the token count of the base POS for each in the table in Appendix A.

I also included average sentence length. I demonstrated above that both modern and traditional readability methods have included sentence length measures with positive results. Sentence length features originated from the traditional formulas. These previous sentence length measures are in terms of the average number of tokens per sentence. I also used average number of tokens per sentence. I only used punctuation to delineate sentences, as the authors of this corpus were consistent in their use of punctuation to mark sentence boundaries.

Another set of lexical features I included, which are novel for readability, are the homograph features. Homographs are words that have the same surface form but have have different semantic interpretations. I identified homographs according to the lemmas in the frequency dictionary that had the same form. This effectively limited the homographs to those that are frequent. I believed this effectively limited the homographs to only those that are likely to present challenges to readers at a high frequency. I accounted for how many entries each homograph had in the dictionary because I believed this would indicate an even more frequently ambiguous homograph. Then, I counted homographs in a document and looked at their form to associate them with their dictionary entry count. I produced five homograph features: the average count of dictionary-based homographs per document, the raw count of the entries, the ratio of homographs to tokens, the ratio of homographs to types, and the ratio of homograph tokens to homograph types.

Altogether I developed 165 features. I've included a full list of the 165 features in Appendix B. Most of these features are derived from the frequency dictionary, while all of them depend on the output of MADA.

I developed a script that composes the features into an appropriate format for machine learning—a feature vector. Each document's feature vector is inserted into a training or testing file for input.

3.3 A Machine Learning Approach—TiMBL

I collected the features above for use in a machine learning system. TiMBL is a machine learning program that uses a strategy known as memory-based learning (Daelemans et al., 2010)[3]. Memory-based learning relies on the theory that humans classify new information using analogy and proximity to the experiences they retain in memory. This study is novel in applying TiMBL to automated readability and, specifically, to MSA documents. Much of the prior research reported above has applied machine learning with other approaches to predict readability. I do not provide here any comparisons between TiMBL and other machine learning methods, but this would be a possibility for future work.

[3] I used TiMBL version 6.4.4.

TiMBL can be applied to a wide range of prediction problems. I specify the application of TiMBL to my task—the readability prediction of documents. For example, where the documentation uses 'item' or 'instance' to represent an entity, I use 'document'. Also, my use of the word 'document' in connection with TiMBL is not the actual Arabic text, but rather it is a numeric representation of the document. This representation is a feature vector, a list of 165 measurements—also called feature values—which I computed for each document. Associated with this list of measurements is the original document's ILR reading level. This is the format that TiMBL requires for machine learning.

A key element of TiMBL's learning phase is its use of nearest neighbors. Nearest neighbors for our purposes are documents stored in memory that are similar to a test document, one that has not yet been seen or processed. TiMBL ranks the similarity of nearest neighbors to a new test document on an ordinal scale of positive integers. The variable k represents how many levels of similarity to include nearest neighbors. The value of k can be set by the user, but the default is 1. Multiple nearest neighbors can be grouped at each level of proximity or distance. For example, if k is set to 2 then the set of nearest neighbors will include the first and second ranked documents in terms of proximity. Nearest neighbors influence TiMBL's ultimate readability level prediction.

To measure the 'neareness' of a nearest neighbor to a test document, TiMBL uses one of several metrics. I chose to use the Overlap metric which compares each feature value of a test feature vector to each corresponding feature value of each feature vector stored in memory. It calculates a distance score for each feature value pair, and sums all distance scores together to derive an overall distance score between two feature vectors.

When calculating the distance scores the features can be weighted; this is beneficial since different features vary in their predictive power of the true reading level of a test document; I used gain ratio, a weighting scheme provided by TiMBL, to assign varying weights to features. It looks at the probability distribution of feature values for each feature with respect to the reading levels and thereby assigns a weight to the feature.

After determining the hierarchy of nearest neighbors, TiMBL takes a vote tally where each nearest neighbor's reading level is a single vote to assign that reading level to the test document. Voting can also be weighted to give a bias to nearer neighbors. I used a vote-weighting bias called inverse linear in which the first nearest neighbors receive a weight of 1 and the weight decreases progressively until the k^{th} nearest neighbors which receive a vote weight of 0. This gave the 9^{th} ranked nearest neighbors a weight of 0 in my experiments. The voting results with inverse linear weighting determined the prediction of each test document's reading level. I set TiMBL to provide evaluation of these predictions using precision, recall, and F-scores.

3.4 Evaluation Approach

In order to evaluate my automated readability system, I used two approaches: an 80/20 training-to-testing split and 10-fold cross validation. To apply 10-fold cross validation I divided the documents of each level randomly into 10 partitions or folds, as is commonly done in readability applications. I configured TiMBL to leave one fold out as testing data (10%) and to include the remaining folds as training data (90%). This is repeated every 10 trials where each fold takes a turn as the test data. I also iterated this cross validation procedure 10 times with a random set of documents assigned to each fold each time. This provided an even better sample across the documents than a single 10-fold cross validation iteration could. Finally, I averaged all results over all 10-fold cross validation iterations to report here. I automated this process with computer scripting and programming.

I tuned TiMBL classifiers using 10-fold cross validation before applying them to the evaluation data set. Tuning involved running a trial and adjusting TiMBL settings and the included features based on the results to improve future results. The 80/20 split method was not as useful for tuning the classifiers before the final evaluation because the risk of having a testing set that was a poor sample and consequently performed unusually well or poorly. I found that the results of repeated cross validation trials were less unstable and sporadic than for repeated 80/20 trials and more indicative of performance across samples.

I used the training set to train and test TiMBL as I added features and fine-tuned its settings. Then, I tested TiMBL on the development test set. Next, I combined the training and development test sets to create a new training set and performed 3-way and a 5-way 10-fold cross validation. I then combined all sets to create a superset of all documents—the combination of training, development test, and evaluation sets—and also performed 3-way and 5-way 10-fold cross validation. Lastly, I performed a 3-way and a 5-way 10-fold cross validation over both the training and devtest sets combined and the evaluation set.

In my final results I included an experiment in which I used the evaluation set as the 20% testing data, then I combined it with the new training set and ran 10-fold cross validation experiments for both 3-way and 5-way classifiers. I report details of the final evaluation results in the following section.

Feature Category	**# of Features**
POS-based Frequency Features	96
Type-To-Token POS Ratio Features	23
Token & Type Frequency Features	19
Discourse Connective Features	7
Homographic Features	5
Frequency-based Discourse Connective Features	4
Type-To-Token Features	4
Word Length Features	3
Sentence Length Features	2
Token count Feature	1
Foreign Word Feature	1

Table 3.2: *Features Grouped by Category and the Count of the Features Contained in Each Category (See Appendix B for the Full List of Features.)*

I used 162 of the 165 total features. Table 3.3 lists the features that I excluded. The range feature refers to the range measure provided in the frequency dictionary; it was very sparse because any given document is very likely to have a lexical item with a range of 100—an item found in all subcorpora which comprise the base corpus of the frequency dictionary I used (Buckwalter and Parkinson, 2011). The other two excluded features were both based on ranks of highly frequent lexical items, the preposition fy ‘in’ and the definite article Al ‘the’ which led to virtually no distinction in this measure across document levels.

The TiMBL settings I employed in all of the experiments below were the same. The algorithm was IB1 with the Overlap Metric, Gain Ratio feature weighting, a K-value of 9, and Inverse Linear vote weighting. I chose these settings based on experimental tuning of the parameters. For the K-value I found that a value of 9 worked the best across all

Feature Name	**Description**
maxTypeFreqRange	max frequent word type range
minFreqPrepRank	min frequent preposition rank
minTypeFreqRank	min frequent word type rank

Table 3.3: *Excluded Features for Final Experiment*

the experiments. While some lower and some higher K-values performed slightly better in some preliminary experiments, the results tended to deteriorate when lowering or raising the K-value from 9.

All the cross validation experiments were iterated 10 times with a random division made between the training and testing sets for each iteration. I combined the counts for true positive, false positive, true negative, and false negative predictions across each iteration. Then, I calculated the precision, recall, and F-score for each class using these cumulative counts. Finally, I calculated the average F-score across the class F-Scores which I obtained.

I refer to each applied instance of the TiMBL system as a classifier, consistent with other machine learning research in automated readability. N-way classifiers represent the number of ways/levels that a document can be classified (i.e. the number of levels included in the training set). I used both 3-way and 5-way classifiers which predicted among 3 and 5 levels respectively. A diagram of the readability system is shown in the figure in Appendix C.

Chapter 4

Results

I ran preliminary evaluations in order to provide a context for the final evaluations. These preliminary evaluations treated 80% of the DLI corpus (the train and devtest sets from Table 3.1). I used 3-fold, 5-fold, and 10-fold cross validation, as well as leave-one-out for both 3-way and 5-way classifiers. For the final evaluations I ran 3-fold and 5-fold, but I also added an 80/20 split. In all experiments I used 162 features, leaving out 3 original features that proved ineffective due to their sparseness. I did not try every possible subset of the 162 features generated in these experiments, which would have been impractical even if I discounted the variable TiMBL settings that could be set for each. I do not report results of an 80/20 split for the train and devtest sets because evaluations of these were not calculable since some levels did not have enough documents for adequate training of the TiMBL classifiers. I next discuss features and feature sets that were the most effective in my experiments.

TiMBL provides a ranking of each feature according to its informativeness in predicting the outcome. Table 4.1 shows the top 20 features according to their informativeness rank. All of these features are from the POS-based frequency feature set except for the foreign word feature. All except three of these features are based on pronouns or conjunctions. These particular pronoun and conjunction features are likely to be limited in the variety of measurements across documents. Therefore, they may receive the most weight because of their relative sparseness. This is apparently the case for the other three features as well—ratio of foreign words to tokens, minimum frequency rank of all part* tokens (all tokens whose POS begins with 'part'), and minimum frequency rank of noun_quant tokens. As a group these features are apparently helpful. The pattern of their combined measurements seems to be their value as predictive features.

Feature Name
minimum frequency rank of pron_rel tokens
ratio of frequent pron_dem tokens to all pron_dem tokens
minimum frequency rank of conj_sub tokens
minimum frequency rank of pron_dem tokens
median frequency rank of pron_rel tokens
ratio of frequent pron_rel to all pron_rel tokens
minimum frequency rank of all conj* tokens
minimum frequency rank of all pron* tokens
range of frequency ranks of pron_rel tokens
median frequency rank of pron_dem tokens
range of frequency ranks of pron_dem tokens
maximum frequency rank of pron_rel tokens
maximum frequency rank of pron_dem tokens
maximum frequency rank of conj tokens
minimum frequency rank of all part* tokens
minimum frequency rank of conj tokens
ratio of foreign words to tokens
median frequency rank of conj_sub tokens
minimum frequency rank of noun_quant tokens
median frequency rank of all pron* tokens

Table 4.1: *Top 20 Features Ordered by Informativeness*

The feature sets that were the most informative in making predictions are ranked in Table 4.2. I determined this order of features based on the average informativeness rank of all features within the set. I excluded the foreign word ratio feature, which is the sole feature in its set, because it is too sparse to be helpful when acting alone. The first feature set in this rank is the POS-based frequency feature set. This is not surprising given that the top 20 informative features are all from this set, except for the foreign word ratio. The next two feature sets in the ranking are both based on discourse connectives which supports the argument that discourse connective features are important in Arabic writing. The least informative feature set is the set of word length features. Arabic word lengths widely vary, and do not necessarily add a lot of complexity with increased length.

Feature Set
POS-based Frequency Features
Frequency-based Discourse Connective Features
Discourse Connective Features
Sentence Length Features
Token Count Feature
Type-To-Token POS Ratio Features
Homographic Features
Type-To-Token Features
Token & Type Frequency Features
Word Length Features

Table 4.2: *Performance by Feature Set*

4.1 Preliminary Evaluation

In the preliminary evaluations my system had already 'seen' the data on which it was testing, and I was able to improve the performance by adjusting the features and TiMBL settings incrementally. My experiments demonstrated that 10-fold cross validation produced the best results among all evaluations using the training and devtest sets combined.

Each of the preliminary results shown in Tables 4.3 and 4.4 represents the highest average F-scores I achieved with various features and TiMBL settings. These tables show that the 3-way classifier performed the best, as expected because there are fewer levels to choose between. Results are similar for all levels. The 5-way classifier showed more variance between the individual levels. Notably, it performed very poorly on level 2+. In fact this performance is barely above a baseline of choosing level 2 for each prediction which would achieve 44.7%. The reason for this unimpressive performance may be that level 2+ documents were very similar to the neighboring levels. This may be due to the vagueness for the ILR plus levels.[1]

4.2 Final Evaluation

I did two types of experiments for the final evaluation. First, I ran an 80/20 split experiment using the 143 documents of the training and devtest sets combined for the 80%

[1] Troy Cox, personal communication, March 4, 2014

Level	Precision	Recall	F-Score
1 & 1+	0.840	0.777	0.807
2	0.724	0.810	0.765
2+ & 3	0.825	0.736	0.778
		Average F-Score	0.783

Table 4.3: *10-Fold Cross Validation 3-Way Classifier—Preliminary Evaluation.*

Level	Precision	Recall	F-Score
1	0.764	0.487	0.595
1+	0.602	0.427	0.500
2	0.558	0.760	0.644
2+	0.292	0.184	0.226
3	0.698	0.615	0.654
		Average F-Score	0.523

Table 4.4: *10-Fold Cross Validation 5-Way Classifier—Preliminary Evaluation.*

training portion and the 36 unseen documents of the evaluation set for the 20% testing portion. Table 4.5 shows results for this experiment.

Table 4.5 shows no valid precision or F-score level 1+ because there were no predictions made at this level. This is a function of having a small corpus that is unbalanced in the number of documents per level. In fact, level 1+ had the lowest representation with only 16 training and 4 testing documents. Therefore, the F-score is only averaged over the 4 levels with calculable results.

Of the other levels, level 2 performs the best with an F-score of 0.650. I attribute this better performance to the unbalance of the corpus—level 2 had the most documents in its training partition and testing partition with 64 and 16 documents in each respectively. The training process had an advantage of many more level 2 documents than the other levels. Level 1 performed the best with an F-score of 0.857, while level 3 performed the worst (not considering level 1+) with an F-score of 0.222.

I also conducted 10-fold cross validation tests on the data to overcome the problem of data sparseness across levels. I used 3-way and 5-way classifiers for the cross validation; results are shown in Tables 4.6 and 4.7 respectively.

Level	Precision	Recall	F-Score
1	1.00	0.750	0.857
1+	-	0.000	-
2	0.541	0.812	0.650
2+	0.400	0.250	0.307
3	0.250	0.200	0.222
		Average F-Score	0.509

Table 4.5: *80/20 Split 5-Way Classifier*

Level	Precision	Recall	F-Score
1 & 1+	0.818	0.582	0.680
2	0.663	0.795	0.723
2 & 2+	0.791	0.718	0.753
		Average F-Score	0.719

Table 4.6: *10-Fold Cross Validation 3-Way Classifier*

Level	Precision	Recall	F-Score
1	0.813	0.610	0.697
1+	0.724	0.150	0.248
2	0.653	0.856	0.740
2+	0.460	0.290	0.355
3	0.503	0.624	0.557
		Average F-Score	0.519

Table 4.7: *10-Fold Cross Validation 5-Way Classifier*

The individual level statistics in Table 4.7 show that TiMBL predicted levels 1, 2 and 3 much more accurately than levels 1+ and 2+. The advantage seems to be partly that levels 1+ and 2+, being plus levels are more vague. The better performance of level 2 over the non-plus levels is likely because of abundant training examples. Level 2 has the most documents in training.

The difference in performance between the preliminary and final evaluations is such that the 3-way classifiers' F-scores decreased from 0.783 to 0.719. This is to be expected because the final evaluation was on unseen data. Likewise, the 5-way classifiers' F-score decreased from 0.523 to 0.519.

My automated readability prediction system achieved the best results with 3-way classifiers and using all but 3 features which were found to be sparse in the training set. This is slightly lower than the results achieved by Al-Khalifa and Al-Ajlan (2010) in their preliminary study for their 3-way classifier—0.778 F-score. These two results, though, are not entirely comparable because they derive from different corpora. The DLI Corpus is written for adult readers of Arabic as a second language, while Al-Khalifa and Al-Ajlan used a grade-school corpus.

Chapter 5

Conclusion

Readability has been an important research problem for nearly a century, though research into readability for Arabic (MSA) is in the very early stages. My thesis is an important contribution to the state-of-the-art for automated Arabic readability prediction. It is the first study to employ TiMBL in readability prediction for any language and shows that this can be done to good effect in Arabic.

A limitation of my study was that I used a very small corpus. Larger corpora are advantageous because they provide larger training sets for machine learning classifiers. The classifiers can be improved thereby since a larger training set is more representative of a set as a whole. I anticipate that, as for other languages, more MSA corpora annotated for readability will be made available in the future.

My system used lexical and discourse features, but omitted syntactic features. MSA has very rich syntax that reflects a wide range of complexity. I excluded syntactic features because of the difficulty in applying syntactic parsing to my data set. Syntactic features have shown positive contributions in previous readability research. Ongoing work on Arabic parser development will render them more robust to data like mine.

Other research has employed N-grams to model readability. I substituted a frequency dictionary for this method. N-gram models are more robust with larger corpora than the DLI corpus.

I only applied my classifiers to data rated by the ILR system. Thus, I only target ILR readability levels. As other scales become available the system could be retrained. Furthermore, of the 11 ILR reading levels, only 5 are represented in the DLI corpus, and thus available for my classification system.

A possibility of future work is to compare TiMBL's performance with other machine learning programs using the same features. Other systems might perform better with the same features.

Other features known to be useful for multiple languages are yet untried for MSA readability. Noun-based features have been especially useful for English readability. Since MSA has a high noun density due to frequent multi-noun constructions this may be a fruitful source of features. I deferred to future work the use of some noun-based features that may be useful for readability in MSA but difficult to implement.

A very helpful application for future work would be a graphical user interface for inputting Arabic documents, to produce a readability score. This could be made widely available to students and teachers in web-based context. Similarly, web search engines could build upon this research to return results according to specific readability levels. Applications like this could use crowdsourcing to evaluate their performance.

A final area of further development may be to build domain-specific classifiers. Such classifiers would be more accurate by leveraging vocabulary information and other unique information of a given domain. Separate domain-based classifiers could be bundled together and invoked according to the need for readability prediction in their respective domains, thus improving readability for many domains.

In conclusion, I developed the first TiMBL-based system for distinguishing MSA documents annotated with ILR readability levels. I used some novel features, most of which were derived from a frequency dictionary. This is the only modern MSA readability prediction system that uses a frequency dictionary. Using standard machine learning evaluation techniques, I was able to show that the system produced state-of-the-art results, even with a smaller corpus than used in comparable studies. My study has shown that the combination of several lexical, discourse, and traditional features are effective indicators of MSA readability and that further research to improve MSA readability is worthwhile.

Bibliography

Al-Batal, M. (1990). Connectives as cohesive elements in a modern expository Arabic text. In Eid, M. and McCarthy, J., editors, *Perspectives on Arabic Linguistics II*. John Benjamins, Amsterdam/Philadelphia.

Al-Khalifa, H. S. and Al-Ajlan, A. A. (2010). Automatic readability measurements of the Arabic text: An exploratory study. *The Arabian Journal for Science and Engineering*, 35(2C):103–124.

Alsaif, A. and Markert, K. (2010). The Leeds Arabic discourse treebank: Annotating discourse connectives for Arabic. In *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC)*, pages 2046–2053. European Language Resources Association.

Alsaif, A. and Markert, K. (2011). Modeling discourse relations for Arabic. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 736–747. Association for Computational Linguistics.

Buckwalter, T. and Parkinson, D. (2011). *A Frequency Dictionary of Arabic: Core Vocabulary for Learners*. Routledge Frequency Dictionaries.

Carroll, J. B., Davies, P., and Richman, B. (1971). *Word Frequency Book*. Houghton Mifflin, Boston.

Chall, J. S. and Dale, E. (1995). *Readability Revisited: The New Dale-Chall Readability Formula*. Brookline Books.

Clark, J. L. and Clifford, R. T. (1988). The FSI/ILR/ACTFL proficiency scales and testing techniques: Development, current status, and needed research. *Studies in Second Language Acquisition*, 10(2):129–147.

Collins-Thompson, K. and Callan, J. (2005). Predicting reading difficulty with statistical language models. *Journal of the American Society for Information Science and Technology*, 56(13):1448–1462.

Daelemans, W., Zavrel, J., van der Sloot, K., and van den Bosch, A. (2010). TiMBL: Tilburg Memory Based Learner, version 6.3, reference guide. Technical Report version 6.3, ILK Research Group Technical Report.

Dale, E. (1931). A comparison of two word lists. *Educational Research Bulletin*, 10(18):484–489.

Dale, E. and Chall, J. S. (1948). A formula for predicting readability. *Educational Research Bulletin*, 27(1):11–28.

Dunn, L. M. and Markwardt, F. C. (1970). *Peabody Individual Achievement Test.* American Guidance Service.

Feng, L., Elhadad, N., and Huenerfauth, M. (2009). Cognitively motivated features for readability assessment. In *Proceedings of The 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2009)*, pages 229–237. Association for Computational Linguistics.

Feng, L., Jansche, M., Huenerfauth, M., and Elhadad, N. (2010). A comparison of features for automatic readability assessment. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING): Posters*, pages 276–284. Association for Computational Linguistics.

Flesch, R. (1948). A new readability yardstick. *Journal of Applied Psychology*, 32(3):221–233.

François, T. and Watrin, P. (2011). On the contribution of MWE-based features to a readability formula for French as a foreign language. In *Proceedings of Recent Advances in Natural Language Processing (RANLP)*, pages 441–447. Association for Computational Linguistics.

Habash, N. (2010). *Introduction to Arabic Natural Language Processing.* Morgan & Claypool.

Habash, N., Rambow, O., and Roth, R. (2009). MADA+TOKAN: A toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization. In *Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR)*, pages 102–109. The MEDAR Consortium.

Hancke, J., Vajjala, S., and Meurers, D. (2012). Readability classification for German using lexical, syntactic, and morphological features. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING): Technical Papers*, pages 1063–1080. Association for Computational Linguistics.

Horn, M. D. (1928). *A study of the vocabulary of children before entering the first grade.* Association for Childhood Education, Washington.

Kincaid, P. J., Fishburne Jr., R. P., Rogers, R. L., and Chissom, B. S. (1975). *Derivation of new readability formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy enlisted personnel.* Naval Technical Training, U.S. Naval Air Station.

Lively, B. A. and Pressey, S. L. (1923). A method for measuring the 'vocabulary burden' of textbooks. *Educational Administration and Supervision*, 9(7):389–398.

Lorge, I. (1944). Predicting readability. *Teachers College*, 45(6):404–419.

Lowe, P. (1987). Revising the ACTFL/ETS scales for a new purpose: Rating skill in translating. *Translation Excellence: Assessment, Achievement, Maintenance*, 1.

Maamouri, M. and Bies, A. (2004). Developing an Arabic treebank: Methods, guidelines, procedures, and tools. In *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages (COLING)*. Association for Computational Linguistics.

McCall, W. A. and Crabbs, L. M. (1926). *Standard Test Lessons in Reading. Books II, III, IV, and V*. Bureau of Publications.

Pitler, E. and Nenkova, A. (2008). Revisiting readability: A unified framework for predicting text quality. In *Proceedings of the Conference on Empirical Linguistics*, pages 186–195. Association for Computational Linguistics.

Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A., and Webber, B. (2008). The Penn Discourse Treebank 2.0. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC)*, pages 2961–2968. Association for Computational Linguistics.

Ryding, K. C. (2005). *A Reference Grammar of Modern Standard Arabic*. Cambridge University Press, Cambridge, MA.

Schwarm, S. E. and Ostendorf, M. (2005). Reading level assessment using support vector machines and statistical language models. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL'05)*, pages 523–530. Association for Computational Linguistics.

Shen, W., Williams, J., Marius, T., and Salesky, E. (2013). A language-independent approach to automatic text difficulty assessment for second-language learners. In *Proceedings of the 2nd Workshop on Predicting and Improving Text Readability for Target Reader Populations*, pages 30–38. Association for Computational Linguistics.

Sherman, L. A. (1893). *Analytics of Literature: A Manual for the Objective Study of English Prose and Poetry*. Ginn and Company, Boston.

Si, L. and Callan, J. (2001). A statistical model for scientific readability. In *Proceedings of the 10th International Conference on Information and Knowledge Management (CIKM)*, pages 574–576. Association for Computing Machinery.

Stenner, A. J. (1996). *Measuring Reading Comprehension with the Lexile Framework*. MetaMetrics Inc., Durham, NC.

Stenner, A. J., Smith, D. R., Horabin, I., and Smith, M. (1987). *Fit of the Lexile Theory to Item Difficulties on Fourteen Standardized Reading Comprehension Tests*. MetaMetrics, Durham, NC.

Thorndike, E. L. (1921). The teacher's word book. *Teachers College, Columbia University*, 134.

Vajjala, S. and Meurers, D. (2012). On improving the accuracy of readability classification using insights from second language acquisition. In *Proceedings of The 7th Workshop on the Innovative Use of NLP for Building Educational Applications*, pages 163–173. Association for Computational Linguistics.

Witten, I. H., Frank, E., and Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier Inc., Burlington, MA.

Zipf, G. K. (1949). *Human Behavior and The Principle of Least Effort*. Addison-Wesley, Cambridge, MA.

Appendix A

POS Ratio Features

Feature Name	**Count**
adj-to-token ratio (Adjective)	8,365
adj_comp-to-token ratio (Comparative Adjective)	464
adj_num-to-token ratio (Numeric Adjective)	385
adv-to-token ratio (Adverb)	235
adv_rel-to-token ratio (Relational Adverb)	147
conj-to-token ratio (Conjunction)	1,182
conj_sub-to-token ratio (Subordinating Conjunction)	1,949
noun-to-token ratio (Noun)	29,921
noun_num-to-token ratio (Numeric Noun)	1,453
noun_prop-to-token ratio (Proper Noun)	2,888
noun_quant-to-token ratio (Quantitative Noun)	685
part-to-token ratio (Particle)	57
part_focus-to-token ratio (Focus Particle)	35
part_interrog-to-token ratio (Interrogative Particle)	95
part_neg-to-token ratio (Negative Particle)	579
part_verb-to-token ratio (Verb Particle)	201
prep-to-token ratio (Preposition)	7,718
pron-to-token ratio (Pronoun)	494
pron_dem-to-token ratio (Demonstrative Pronoun)	981
pron_rel-to-token ratio (Relative Pronoun)	1,308
punc-to-token ratio (Punctuation)	7,244
verb-to-token ratio (Verb)	7,707
verb_pseudo-to-token ratio (Pseudo Verb)	372

Table A.1: *POS Features and Count of Tokens with the Base POS in the DLI Corpus*

Appendix B

Features for Readability Prediction

POS-Based Frequency Features
ratio frequent adj tokens to all adj tokens
ratio frequent adj* tokens to all adj* tokens
ratio frequent conj_sub tokens to all conj_sub tokens
ratio frequent conj tokens to all conj tokens
ratio frequent conj* tokens to all conj* tokens
ratio frequent noun_num tokens to all noun_num tokens
ratio frequent noun_prop tokens to all noun_prop tokens
ratio frequent noun_quant tokens to all noun_quant tokens
ratio frequent noun tokens to all noun tokens
ratio frequent noun* tokens to all noun* tokens
ratio frequent part* tokens to all part* tokens
ratio frequent prep tokens to all prep tokens
ratio frequent pron_dem tokens to all pron_dem tokens
ratio frequent pron_rel tokens to all pron_rel tokens
ratio frequent pron* tokens to all pron* tokens
ratio frequent verb tokens to all verb tokens
maximum frequency rank of adj tokens
maximum frequency rank of all adj* tokens
maximum frequency rank of all conj* tokens
maximum frequency rank of all noun* tokens
maximum frequency rank of all part* tokens
maximum frequency rank of all pron* tokens
maximum frequency rank of conj_sub tokens
maximum frequency rank of conj tokens
maximum frequency rank of noun_num tokens
maximum frequency rank of noun_prop tokens
maximum frequency rank of noun_quant tokens
maximum frequency rank of noun tokens
maximum frequency rank of prep tokens
maximum frequency rank of pron_dem tokens
maximum frequency rank of pron_rel tokens

Continued on next page

Table B.1 – *Continued from previous page*

maximum frequency rank of verb tokens
mean frequency rank of adj tokens
mean frequency rank of all adj* tokens
mean frequency rank of all conj* tokens
mean frequency rank of all noun* tokens
mean frequency rank of all part* tokens
mean frequency rank of all pron* tokens
mean frequency rank of conj_sub tokens
mean frequency rank of conj tokens
mean frequency rank of noun_num tokens
mean frequency rank of noun_prop tokens
mean frequency rank of noun_quant tokens
mean frequency rank of noun tokens
mean frequency rank of prep tokens
mean frequency rank of pron_dem tokens
mean frequency rank of pron_rel tokens
mean frequency rank of verb tokens
median frequency rank of adj tokens
median frequency rank of all adj* tokens
median frequency rank of all conj* tokens
median frequency rank of all noun* tokens
median frequency rank of all part* tokens
median frequency rank of all pron* tokens
median frequency rank of conj_sub tokens
median frequency rank of conj tokens
median frequency rank of noun_num tokens
median frequency rank of noun_prop tokens
median frequency rank of noun_quant tokens
median frequency rank of noun tokens
median frequency rank of prep tokens
median frequency rank of pron_dem tokens
median frequency rank of pron_rel tokens
median frequency rank of verb tokens
minimum frequency rank of adj tokens
minimum frequency rank of all adj* tokens
minimum frequency rank of all conj* tokens
minimum frequency rank of all noun* tokens
minimum frequency rank of all part* tokens
minimum frequency rank of all pron* tokens
minimum frequency rank of conj_sub tokens
minimum frequency rank of conj tokens

Continued on next page

Table B.1 – *Continued from previous page*

minimum frequency rank of noun_num tokens
minimum frequency rank of noun_prop tokens
minimum frequency rank of noun_quant tokens
minimum frequency rank of noun tokens
minimum frequency rank of prep tokens
minimum frequency rank of pron_dem tokens
minimum frequency rank of pron_rel tokens
minimum frequency rank of verb tokens
range of frequency ranks of adj tokens
range of frequency ranks of all adj* tokens
range of frequency ranks of all conj* tokens
range of frequency ranks of all noun* tokens
range of frequency ranks of all part* tokens
range of frequency ranks of all pron* tokens
range of frequency ranks of conj_sub tokens
range of frequency ranks of conj tokens
range of frequency ranks of noun_num tokens
range of frequency ranks of noun_prop tokens
range of frequency ranks of noun_quant tokens
range of frequency ranks of noun tokens
range of frequency ranks of prep tokens
range of frequency ranks of pron_dem tokens
range of frequency ranks of pron_rel tokens
range of frequency ranks of verb tokens
Type-To-Token POS Ratio Features
adjective-to-token ratio
comparative adjective-to-token ratio
numeric adjective-to-token ratio
adverb-to-token ratio
relational adverb-to-token ratio
conjunction-to-token ratio
subordinating conjunction-to-token ratio
noun-to-token ratio
numeric noun-to-token ratio
proper noun-to-token ratio
quantitative noun-to-token ratio
particle-to-token ratio
focus particle-to-token ratio
interrogative particle-to-token ratio
negative particle-to-token ratio
verb particle-to-token ratio

Continued on next page

Table B.1 – *Continued from previous page*

preposition-to-token ratio
pronoun-to-token ratio
demonstrative pronoun-to-token ratio
relative pronoun-to-token ratio
punctuation-to-token ratio
verb-to-token ratio
pseudo verb-to-token ratio
Token & Type Frequency Features
frequent type-to-token ratio
maximum dispersion of frequent types
maximum frequency rank of frequent types
mean dispersion of frequent tokens
mean dispersion of frequent types
mean frequency log of frequent tokens' dictionary frequency count
mean frequency log of frequent types' dictionary frequency count
mean frequency rank of frequent types
mean frequency rank of frequent tokens
median dispersion of frequent tokens
median dispersion of frequent types
median frequency rank of frequent types
median frequency rank of frequent tokens
minimum dispersion of frequent types
minimum frequency rank of frequent types
range of dispersion of frequent types
range of frequency ranks of frequent types
ratio frequent tokens to morpheme tokens
ratio of frequent types to morpheme types
Discourse Connective Features
all connectives count
average connectives per sentence
multi-word connective count
ratio of single word connectives to all connectives
ratio of multi-word connectives to all connectives
ratio of multi-word connectives to single word connectives
single word connective count
Homographic Features
average of: each frequent homograph token count x its dictionary entry count
frequent homograph type-to-token ratio
ratio of frequent homograph tokens to morpheme tokens
ratio of frequent homograph tokens to morpheme types
sum of: each frequent homograph token count x its dictionary entry count

Continued on next page

Table B.1 – *Continued from previous page*

Frequency-based Discourse Connective Features
frequent single word connective count
non-frequent single word connective count
ratio of non-frequent single word connectives to all connectives
ratio of frequent single word connectives to all connectives
Type-To-Token Features
lexeme type-to-token ratio
morpheme type-to-token ratio
square root of morpheme type-to-token ratio
square root of lexeme type-to-token ratio
Word Length Features
average character length of surface forms
average length of diacritized words
average morpheme length of words
Sentence Length Features
average sentence morpheme length
average sentence token length
Token Count Feature
token count
Foreign Word Feature
ratio of foreign words to tokens

Table B.1: *Full Feature List Grouped By Sets (All POS Features Are Based on MADA's POS Output.)*

Appendix C

Automated MSA Readability Prediction System

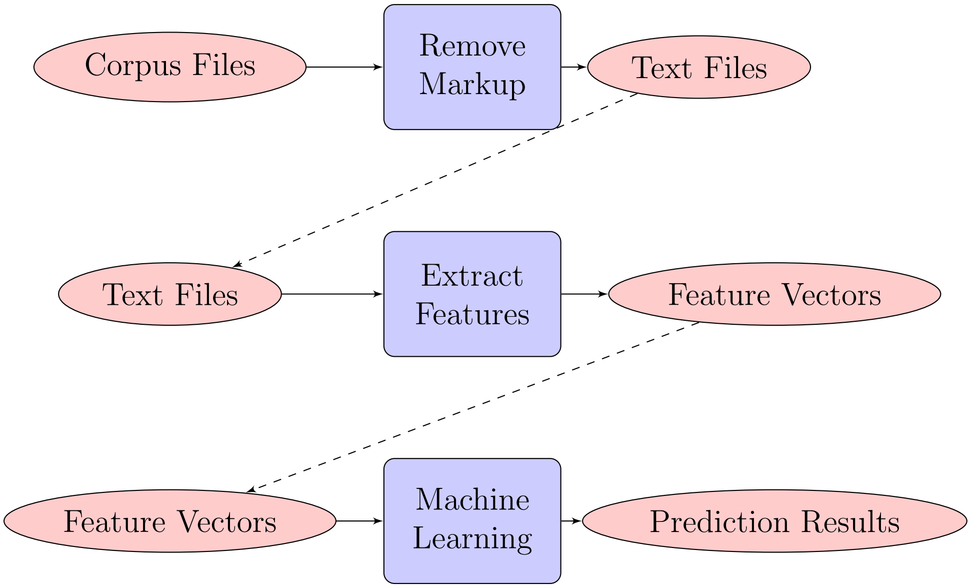

Figure C.1: *Overview of the MSA Automated Readability Prediction System*