Toward Improved Artificial Intelligence in Requirements Engineering: Metadata for Tracing Datasets

Jane Huffman Hayes
Computer Science Department
University of Kentucky
Lexington, Kentucky, USA
hayes@cs.uky.edu

Jared Payne, Mallory Leppelmeier
Computer Science Department
University of Kentucky
Lexington, Kentucky, USA
jared.payne@uky.edu, mallory.leppelmeier@uky.edu

***Abstract*—Data is the driver of artificial intelligence in requirements engineering. While some applications may lend themselves to training sets that are easily accessible (such as sentiment detection, feature request classification, requirements prioritization), other tasks face data challenges. Tracing and domain model building are examples of applications where data is not easily found or in the proper format or with the necessary metadata to support deep learning, machine learning, or other artificial intelligence techniques. This paper surveys datasets available from sources such as the Center of Excellence for Software and Systems Traceability and provides valuable metadata that can be used by researchers or practitioners when deciding what datasets to use, what aspects of datasets to use, what features to use in deep learning, and more.**

***Index Terms*—artificial intelligence, requirement engineering, deep learning, machine learning, datasets, metadata, training sets**

I. Introduction

One need only look at the program for the IEEE International Conference on Requirements Engineering 2018 (RE 2018) to see the impact of artificial intelligence (AI) on academic research for requirements engineering. Topics ranging from classification of requirements, prioritization of requirements, sentiment analysis of reviews of software features, elicitation of security requirements, validation of requirement reviews, and process mining all require the application of artificial intelligence techniques. These topics are not merely ivory tower fodder. Perusal of the industry track papers for RE 2018 indicates that artificial intelligence supports many important undertakings to, for example, detect requirements ambiguity, diagnose requirements violations, identify uncertainty in contextual requirements, and perform requirements classification for redundancy and inconsistency checking.

Those who seek to apply artificial intelligence techniques for requirements engineering such as machine learning, neural networks, and genetic algorithms may quickly encounter their first challenge: lack of data. Once data is obtained, researchers may encounter another challenge: understanding the data.

It may be easier to find appropriate datasets for certain endeavors in requirements engineering than for others. Take sentiment analysis, for example. Feature reviews can be found for software applications, most including a score or number of stars assigned by the author. Researchers can use these scores as a de-facto training set where a higher score indicates a happier user.

Tracing research faces data challenges. It can be difficult to find multiple artifacts for a given application, be it open or closed source. Once software applications with multiple artifacts are located, it is rare for actual trace links (answer sets) to exist. Rath et al. [15] have begun work to use commits to assist in recovering answer sets and Kalim et al. [16] have developed a tool for crowd sourcing of answer sets (called MIDAS), but some projects do not have the required commits and the MIDAS tool requires an initial answer set of some sort.

This paper supports application of artificial intelligence for requirements engineering by analyzing a collection of traceability datasets. The information can be used to assist researchers in selecting datasets and in understanding selected datasets and their attendant results. The metadata can also assist in feature selection for machine learning.

The paper is organized as follows. Section II presents the approach we used to gather metadata on the data sets. Section III presents the metadata. Section IV describes related work, and Section V concludes.

II. Background and Approach

This section provides background information on traceability and its terminology. It presents the tools used to obtain metadata on each dataset: readability-metrics, Stanford Parser, and TraceLab. The metadata is also described.

A. Traceability

Requirements traceability is defined as "the ability to describe and follow the life of a requirement, in both a forwards and backwards direction (i.e., from its origins, through its development and specification, to its subsequent deployment and use, and through periods of on-going refinement and iteration in any of these phases) [1]. " In most tracing research, various techniques are applied to find potential or candidate links between elements of source and target artifacts. For example, one may trace from a requirements specification to a design document.

The candidate links that are retrieved by the technique are then examined or vetted by human analysts in order to determine the final links. The final links are then compared against a gold standard or answer set in order to determine the accuracy of the tracing technique.

B. *Readability-metrics*

Earlier work by Hayes et al. [2] demonstrated that certain characteristics of trace datasets could be used to predict trace links between source and target artifacts. Particularly, measures pertaining to the readability of the text contribute most to the accuracy of the predictions. This paper follows that example and mines the same measures for each dataset using the readability-metrics tool and TraceLab component for Stanford Parser [3] developed by the second and third co-authors. The tool provides two types of measures.

1) Readability measures: In general, readability measures attempt to estimate the ease or difficulty with which one can read and comprehend a text. Most measures provided by readable.com are ease of reading measures, meaning the higher the value, the harder the text is to read. These measures tend to follow the grade levels of the United States, so that a measure of 9 means that the text could generally be understood by a student at the 9th grade level. The measures falling under this category include (equations are in Fig. 2 at the end of the paper):

a) Flesch reading ease: Ease with which a text can be read (higher = easier to read), based on sentence length and word length.

b) Flesch-Kincaid grade level: Grade level of a text (higher = harder to read), it uses average sentence length (in words) and average number of syllables per word.

c) SMOG index[1]*:* Estimate of years of education needed to read text (higher = harder to read), it uses a count of polysyllable words and adds a constant.

d) Automated readability index: Alternative means of measuring grade level of a text (higher = harder to read), uses average number of characters per word and average number of words per sentence.

2) Basic text measures: These measures characterize the structure of the text. The measures falling under this category include:

a) Number of words: Complex text elements will have many sentences (higher = harder to read).

b) Number of sentences: Complex text elements will have many sentences (higher = harder to read).

c) Mean words per sentence: Complex sentences have many words (higher = harder to read)

d) Number of polysyllable words: Complex words have many syllables (higher = harder to read).

e) Percentage of polysyllable words: Complex words have many syllables (higher = harder to read). A polysyllable is defined here as a word that contains three or more syllables.

C. *Stanford Parser*

The Stanford Part-Of-Speech Tagger [13] reads in text and assigns parts of speech tags, such as NN for proper noun, to each word or symbol. The english-left3words-distsim.tagger training model was used to tag each data set. Once the data sets were tagged, each word was split from its tag, and tags that started with JJ, NN, PRP, or VB were counted. The tags indicated if the word was an adjective, noun, pronoun or verb, respectively. The total number of tags in each data set were also counted.

D. *TraceLab*

Traceability researchers have long developed their own siloed traceability tools, such as Poirot [4], RETRO.NET [5], and ADAMS [6]. Besides requiring a large investment of time and energy to develop, these tools do not readily permit replication of experiments, comparison of results, and do not support reuse or easy modification. To address this, a group of researchers led by DePaul University and William and Mary University developed TraceLab [6].

TraceLab is an experiment environment that provides a graphical interface in which researchers can create and run tracing, or more generally any, experiments consisting of reusable components that are executed sequentially. These components contain pre-written code that can be placed in the experiment interface to manipulate data stored in the workspace. Having components written in advance with their code isolated from one another allows experiments to be shared more easily among researchers, improves replicability, and decreases development time.

To obtain the parts of speech metadata, a TraceLab component was developed to utilize the Stanford Parser's part-of-speech tagger. The experiment using this component is shown in Figure 1.

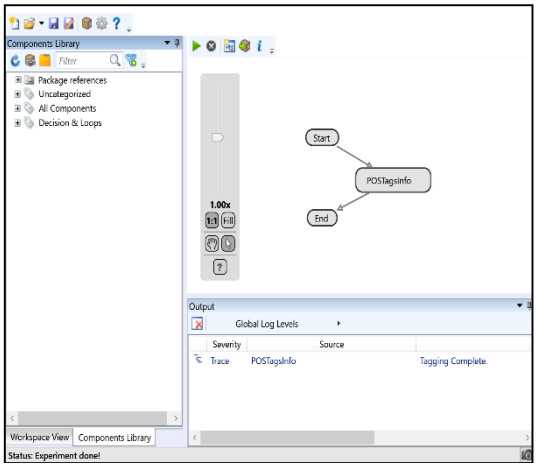

Fig. 1. Stanford Parser component in TraceLab.

To obtain the readability and basic text metadata, a Python package was developed comprised of functions that can calcu-

[1] SMOG requires a minimum of 30 sentences for a calculation. Some datasets did not meet this and thus had 0 as scores.

late various readability metrics for a body of text. This functionality is facilitated by the Natural Language Toolkit package [11], which provides word and sentence tokenizers as well as an interface to the CMU Pronouncing Dictionary [12]. This dictionary maps English words to their pronunciations as a list of phonemes and was used to determine the number of syllables a word contains.

III. METADATA

This section presents the metadata for the datasets as well as correlation analysis results.

A. Dataset Selection

To identify datasets for use, recent traceability papers, the Community Data Sets page of the Center of Excellence for Software and System Traceability (COEST) website, and the dataset collection of our research group were examined. Once datasets were identified, their artifacts were also examined. Datasets including artifacts that could not be processed by the readability tools were omitted. Reasons for exclusion were varied such as the existence of non-English text or code in the datasets, and lack of complete sentences in the datasets. This resulted in only five datasets being used from the COEST repository (coest.org, Resources, Community Datasets) and only one from the University of Kentucky repository (https://selab.netlab.uky.edu/AIRE-2019-hayes-payne-leppelmeier-meta-data.zip).

Table I gives an overview of each dataset, providing the name, description, source and target artifact types, and number of links in the answer set. As can be seen, there are several domains represented such as health care (CCHIT, Infusion Pump), science (CM-1), business (GANNT), and telephony (Waterloo).

These trace datasets lend themselves well to use of artificial intelligence for requirements engineering. The Waterloo dataset provides 34 student projects all written to the same specification for internet telephony. An answer set is provided for each. A researcher could use a subset of the 34 for model building/training and predict trace links on the test set (the remainder).

Also, per Zogaan et al.'s Traceability-Data Quality Assessment (T-DQA) Framework [17], Waterloo is accessible (available, not licensed, stored in our repository that has been made available), possesses intrinsic characteristics (it is from a useful domain and has 34 development teams represented), it has contextuality (is relevant and trustworthy), and it can be interpreted (thus it possesses representational characteristic).

The CM-1 dataset has been used frequently, possibly more than any other trace dataset (Zogaan et al. show it as the aerospace domain with 22 uses [17]). This dataset is notoriously challenging for tracing techniques when it comes to precision (a measure of how many false positive links a trace method retrieves). A researcher can apply artificial intelligence techniques to strive to improve the precision for this dataset. Per the T-DQA, CM-1 is accessible (available, not licensed, stored in UK repository, which has been made available), intrinsic (aerospace domain), contextual (relevant, trustworthy), and representational (interpretable).

CCHIT and Infusion Pump are in the health care domain. It is easy for researchers to find repositories of health care documents in order to augment queries when tracing, build domain ontologies, generate word embeddings, etc. These datasets are accessible (available and no license and stored at COEST), intrinsic (useful domain), contextual (relevant and trustworthy), and representational (interpretable).

GANTT is a business application that can be used to manage any project including software engineering projects. It has been used in software maintenance tasks that apply trace matrices resulting from tracing techniques. AI researchers could use this dataset with other business applications or could pursue tracing research, perhaps replicating the earlier experiments to see if their improved trace matrices (from AI techniques) yield improved software maintenance results. Looking at the T-DQA, GANTT is accessible (available, not licensed, stored in UK repository, which that has been made available), intrinsic (business domain), contextual (relevant, trustworthy), and representational (interpretable).

TABLE I. DATA SET OVERVIEW

Dataset Name	Description	Source artifacts	Target artifacts	Links
Infusion Pump (COEST)	A dataset that extracts requirements and components from a specification for a medical infusion pump.	126 high-level requirements	21 low-level components	131
CCHIT-2-WorldVista (COEST)	An industrial dataset that provides trace links between CCHIT healthcare regulatory codes and requirements for WorldVista.	116 regulatory codes	1064 requirements	587
GANTT (UK)	A dataset for a project management tool.	17 high-level requirements	69 low-level design elements	68
CM-1 (UK)	A partially sanitized dataset for a space science instrument from NASA, written in C with 20 KSLOC	235 high level requirements	220 design elements	361
WARC (UK)	A dataset for a web archive tool.	42 functional requirements, 21 nonfunctional requirements (63 total)	89 software requirement specifications	136
Waterloo (UK)	An internet telephony application developed by 34 student groups as part of a course at Univ. of Waterloo.	1092 functional requirements, 209 nonfunctional requirements (1301 total)	383 use cases	2475

WARC is a multi-artifact dataset that can support, e.g., tracing and requirements classification research (non-functional requirements are a separate artifact). Also, human study data exists [18], indicating how human analysts interacted with trace matrices built using traditional information retrieval methods: AI researchers could repeat that study using improved trace matrices.

B. Metadata

Review of related work on metadata and machine learning shows that there are taxonomies for metadata, that metadata is widely used by services such as YouTube and Google, but that there is little information on what metadata is used or how it is selected: except for an article on doing so for technology-assisted review (automatically classifying documents based on expert input). That work stated "Questions still remain, however, regarding the extent to which metadata fields should be utilized, which fields are likely to be most constructive, and which techniques would prove most efficacious…" and then go on to say that they defer "treatment of the question of exactly which specific metadata fields are best suited for machine learning. [19]" With no prior art to guide the selection, we chose to use the metadata that showed promise in predicting trace links in the AI work of Hayes et al. [2].

It is possible that the readability measures are collinear due to the use of similar measures in each (number of words per sentence, for example). Also, it is clear that the basic text measures are not independent (e.g., number of words, number of sentences, and mean words per sentence). Multi-collinearity analysis and principal-component analysis to reduce correlated data remains as future work.

The tools presented in Section II were used to collect metadata for the datasets. Each measure was described in Section II B. or Section II. C. The tables are provided at the end of the paper[2]. Table IV presents the basic text measures from the readability-metrics Python script (number of words, number of sentences, mean words per sentence, number of polysyllables, % polysyllables). Table V presents the readability measures from the readability-metrics Python script (Automated readability index, Flesch reading ease, Flesch-Kincaid grade level, SMOG index). Table VI presents the parts of speech measures from the Stanford Parser TraceLab component (number of tags, nouns, pronouns, adjectives, and verbs).

This metadata could be used to assist trace researchers using AI techniques. For example, trace researchers could generate trace links for one of the provided datasets using standard techniques such as vector space model (VSM). The researchers could then separate the correct links retrieved by VSM from the incorrect links. Merging in our metadata, a model could be built, which would then be used to predict trace links for another trace dataset (for which our metadata was also generated) using the researchers' particular AI method.

Also, in determining which datasets to use for an AI undertaking, the metadata can assist. Imagine a researcher undertaking sentiment analysis. Datasets with a large number or percentage of adjective tags (JJ) could be good candidates for such research. Classification of requirements as functional or non-functional could lead a researcher to seek datasets with a large number or percentage of verbs. In performing requirements ambiguity detection, a researcher could seek datasets that are at extremes on polysyllabic words (perhaps hypothesizing that requirements that are ambiguous will contain many polysyllabic words (complex words) and that requirements with few polysyllabic words will be less likely to be ambiguous).

C. Analysis of Metadata

Descriptive data for the basic text metadata is provided in Table IV. As can be seen, the number of words varies greatly, ranging from a minimum of 493 words (Gantt dataset) to a maximum of 29409 words (CCHIT dataset). Number of words with polysyllables also varies greatly from 60 (Gantt dataset) to 5416 (CCHIT dataset) though percentage of polysyllables only ranges from 0.1 to 0.19. Table V presents the same data but for the readability measures. Flesch-Kincaid grade level varies from a minimum of 4.173 to a maximum of 13.889 with a mean of 8.46. The automated readability index has a median of 10.226 with a minimum of 5.718 and a maximum of 15.721.

Correlation analysis was performed to support feature selection and analysis. It was hypothesized that datasets from the same domain would be more highly correlated to each other than datasets from different domains. To test this hypothesis, Pearson's Correlation Coefficient was calculated for the basic text metadata, for the readability metadata, and for the parts of speech tag counts. Pearson's coefficient ranges from -1 to +1, where a value of 0 means no correlation. The correlation between the health care datasets (Infusion Pump and CCHIT) and each of the non-health care datasets was also calculated, using the nine measures from the basic text and readability metadata. Finally, correlation analysis between all the datasets was performed. The results were all above 0.99, indicating positive correlation (e.g., as GANTT metadata values increase, so do CM-1 metadata values). Tables II and III are examples of correlation runs. Unfortunately, no interesting results were found that would indicate what set of measures might discriminate between datasets of one domain versus a different domain.

IV. RELATED WORK

Dekhtyar et al. described the need for benchmarks in traceability, emphasizing the need for data and information about the datasets [8]. Sundaram et al. provided datasets and baselines for traceability, with emphasis on the accuracy of various tracing techniques [9]. In May 2019, a panel was held at the Software and System Traceability (SST) workshop [10] held at International Conference on Software Engineering. Panelists explained their thoughts on the application of machine learning and other artificial intelligence techniques to tracing. Data, and lack thereof, was a major topic discussed during the session.

V. CONCLUSION

We have provided metadata for a collection of traceability datasets to support artificial intelligence for requirements engineering. We have provided the scripts and components we used

[2] Table number/order was dictated by page breaks to meet the page limit

to generate the metadata as well as made the datasets and metadata available (see Section III A.). We performed some preliminary analysis on the metadata to support feature selection. The correlation analysis did not immediately reveal features that are domain specific. Future work includes: undertaking a systematic analysis for other possible metadata as well as correlation analysis on that data, collecting additional datasets, and performing analysis for collinearity of the metadata followed by principal components analysis (if warranted).

ACKNOWLEDGMENT

We thank the National Science Foundation for partially funding this work under grant CICI 1642134. We thank Jane Cleland-Huang for helpful feedback on a prior version. We thank Faham Hossain for his assistance with tables.

REFERENCES

[1] Gotel OCZ, Finkelstein A. 'An analysis of the requirements traceability problem.' In Proceedings of the First IEEE International Conference on Requirements Engineering (ICRE), IEEE, 1994; 94–101.

[2] Jane Huffman Hayes, Giuliano Antoniol, Yann-Gaël Guéhéneuc, Adams, 'Inherent Characteristics of Traceability Artifacts: Less is More.' In the Proceedings of IEEE Requirements Engineering (RE) Conference, RENext! Track, 2015.

[3] Readability-metrics tool, https://selab.netlab.uky.edu/AIRE-2019-hayes-payne-leppelmeier-meta-data.zip, Jared Payne; Stanford Parser TraceLab component, Molly Leppelmeier, June 2019.

[4] Jun Lin, Chan Chou Lin, Jane Cleland-Huang, Raffaella Settimi, Joseph Amaya, Grace Bedford, Brian Berenbach, Oussama Ben Khadra, Chuan Duan, Xuchang Zou, "Poirot: A Distributed Tool Supporting Enterprise-Wide Automated Traceability," 14th IEEE International Requirements Engineering Conference (RE'06), Minneapolis/St. Paul, MN, 2006, pp. 363-364.

[5] Jane Huffman Hayes, Alex Dekhtyar, Senthil Sundaram, Ashlee Holbrook, Sravanthi Vadlamudi, Alain April, 'REquirements TRacing On target (RETRO): Improving Software Maintenance through Traceability Recovery,' Innovations in Systems and Software Engineering: A NASA Journal (ISSE) 3(3): 193-202 (2007).

[6] A. De Lucia, F. Fasano, R. Oliveto and G. Tortora, "ADAMS Re-Trace: a traceability recovery tool," Ninth European Conference on Software Maintenance and Reengineering, Manchester, UK, 2005, pp. 32-41.

[7] Jane Cleland-Huang, Yonghee Shin, Ed Keenan, Adam Czauderna, Greg Leach, Evan Moritz, Malcom Gethers, Denys Poshyvanyk, Jane Huffman Hayes, Wenbin Li, 'Toward Actionable, Broadly Accessible Contests in Software Engineering.' In the Proceedings of International Conf. on SW Eng. Track on New and Innovative Emerging Results (NIER) 2012.

[8] Alex Dekhtyar, Jane Huffman Hayes, Giulio Antoniol. 'Benchmarks for Traceability?' Published in the Proceedings of Traceability in Emerging Forms of Software Engineering (TEFSE), Slade, KY, March 22/23, 2007.

[9] Senthil Karthikeyan Sundaram, Jane Huffman Hayes, Alex Dekhtyar, 'Baselines in requirements tracing.' ACM SIGSOFT Software Engineering Notes 30(4): 1-6 (2005)

[10] 10th International Workshop at the 41st International Conference on Software Engineering (ICSE), May 27, 2019, https://sst2019.chalmers.se/.

[11] "Natural Language Toolkit—NLTK 3.4.3 documentation", Nltk.org, 2019. [Online]. Available: https://www.nltk.org. [Accessed: 27-Jun-2019].

[12] K. Lenzo, "The CMU Pronouncing Dictionary", Speech.cs.cmu.edu, 2019. [Online]. Available: http://www.speech.cs.cmu.edu/cgi-bin/cmudict. [Accessed: 27-Jun- 2019].

[13] Stanford Part of Speech Tagger, http://nlp.stanford.edu/software/tagger.shtml

[14] Karl Pearson,"Notes on regression and inheritance in the case of two parents," Proceedings of the Royal Society of London, (20 June 1895) 58 : 240–242.

[15] Michael Rath, Jacob Rendall, Jin L. C. Guo, Jane Cleland-Huang, Patrick Mäder, 'Traceability in the Wild: Automatically Augmenting Incomplete Trace Links,' Software Engineering and Software Management 2019: 63, Stuttgart, Germany.

[16] Albert Kalim, Satrio Husodo, Jane Huffman Hayes, Erin Combs, Jared Payne, 'Multi-user Input in Determining Answer Sets (MIDAS),' Proceedings of IEEE Requirements Engineering Conference (RE) 2018, August 2018, Banff, Canada.

[17] W. Zogaan, P. Sharma, M. Mirahkorli and V. Arnaoudova, "Datasets from Fifteen Years of Automated Requirements Traceability Research: Current State, Characteristics, and Quality," 2017 IEEE 25th International Requirements Engineering Conference (RE), Lisbon, 2017, pp. 110-121.

[18] W. K. Kong and J. H. Hayes, "Proximity-based traceability: An empirical validation using ranked retrieval and set-based measures," in Workshop on Empirical Requirements Engineering (EmpiRE), 2011, pp. 45–52.

[19] Jones, A., Bazrafshan, M., Delgado, F.P., Lihatsh, T., Schuyler, T. The Role of Metadata in Machine Learning for Technology Assisted Review. DESI V Workshop, June 14, 2013.

TABLE II. COMPARISON OF CORRELATION VALUES AMONG HEALTH CARE DATASETS

	InfusionPump/ Requirements	**InfusionPump/ Components**	**CCHIT/target**	**CCHIT/source**
InfusionPump/Requirements	1			
InfusionPump/Components	0.99	1		
CCHIT/target	0.99	0.99	1	
CCHIT/source	0.99	0.99	0.99	1

TABLE III. COMPARISON OF CORRELATION VALUES AMONG NON-HEALTH CARE DATASETS

	CCHIT/target	CCHIT/source	GANNT/low	GANNT/high
CCHIT/target	1			
CCHIT/source	0.99	1		
GANNT/low	0.99	0.99	1	
GANNT/high	0.98	0.98	0.99	1

Flesch reading ease[3] = $206.835 - 1.015 \times \langle words \div sentences \rangle - 84.6 \times \langle syllables \div words \rangle$

Flesch-Kincaid grade level[2] = $0.39 \times \langle words \div sentences \rangle + 11.8 \times \langle syllables \div words \rangle - 15.59$

SMOG Index[4] = $1.0430 \sqrt{\langle numbr\ of\ polysyllables \times \frac{30}{number\ of\ sentences} \rangle} + 3.1291$

Automated Readability Index[5] = $4.71 \times \langle \frac{characters}{words} \rangle + 0.5 \times \langle \frac{words}{sentences} \rangle - 21.43$

Fig. 2. Equations for readability metrics.

TABLE IV. READABILITY-METRICS BASIC TEXT METADATA

Name	Number of words	Number of sentences	Mean words per sentence	Number of polysyllables	% polysyllables
InfusionPump/Requirements	3601	256	14.07	537	0.15
InfusionPump/Components	1470	68	21.62	280	0.19
CCHIT/source	2785	111	25.09	407	0.15
CCHIT/target	29409	1018	28.89	5416	0.18
GANNT/low	1554	106	14.66	204	0.13
GANNT/high	493	28	17.61	60	0.12
CM1/source_artifacts	556	30	18.53	61	0.11
CM1/target_artifacts	5884	204	28.84	629	0.11
WARC/SRS	1737	103	16.86	257	0.15
WARC/NFR	534	23	23.22	98	0.18
WARC/FRS	597	44	13.57	93	0.16
waterloo/low	12527	436	28.73	1718	0.14
waterloo/high	19429	1406	13.82	1944	0.1
Mean	6198.15	294.85	20.42	900.31	0.14
Min	493	23	13.57	60	0.10
Max	29409	1406	28.89	5416	0.19
Median	1737	106	18.53	280	0.15

[3] https://www.webfx.com/tools/read-able/flesch-kincaid.html

[4] https://en.wikipedia.org/wiki/SMOG

[5] http://www.readabilityformulas.com/automated-readability-index.php

TABLE V. READABILITY-METRICS READABILITY METADATA

Name	Automated readability index	Flesch reading ease	Flesch-Kincaid grade level	SMOG index
InfusionPump/Requirements	5.86	82.91	5.19	11.4
InfusionPump/Components	12.81	49.76	11.69	14.72
CCHIT/source	12.47	58.62	11.32	14.07
CCHIT/target	15.26	46.94	13.89	16.3
GANNT/low	7.7	74.2	6.55	11.05
GANNT/high	6.62	80.34	6.43	0
CM1/source_artifacts	10.23	85.62	5.92	11.28
CM1/target_artifacts	13.86	72.54	10.3	13.16
WARC/SRS	8.14	82.32	5.97	12.15
WARC/NFR	11.48	61.28	10.48	0
WARC/FRS	5.81	85.65	4.68	11.43
waterloo/low	15.72	49.66	13.47	14.47
waterloo/high	5.72	89.76	4.17	9.85
Mean	10.13	70.74	8.47	10.76
Min	5.72	46.94	4.17	0
Max	15.72	89.76	13.89	16.31
Median	10.23	74.2	6.55	11.43

TABLE VI. STANFORD PARSER METADATA

Name	Number of tags	Number of nouns	Number of pro-nouns	Number of ad-jectives	Number of verbs
InfusionPump/Requirements	3697	1239	23	252	478
InfusionPump/Component	1470	642	10	60	228
CCHIT/target	29334	5714	52	1044	2418
CCHIT/source	2781	888	12	164	413
GANNT/Low	1554	527	18	68	286
GANNT/High	493	148	6	16	80
CM1/source	598	121	1	14	57
CM1/target	6064	765	28	123	362
WARC/SRS	1738	528	16	132	263
WARC/NFR	536	183	3	29	77
WARC/FRS	595	195	7	38	92
waterloo/low	13222	4829	82	448	2014
waterloo/high	21307	5963	106	697	2273