LFTK: Handcrafted Features in Computational Linguistics

Bruce W. Lee[1,2,3], **Jason Hyung-Jong Lee**[2]
[1]University of Pennsylvania
[2]LXPER AI Research
brucelws@seas.upenn.edu
jasonlee@lxper.com

Abstract

Past research has identified a rich set of handcrafted linguistic features that can potentially assist various tasks. However, their extensive number makes it difficult to effectively select and utilize existing handcrafted features. Coupled with the problem of inconsistent implementation across research works, there has been no categorization scheme or generally-accepted feature names. This creates unwanted confusion. Also, most existing handcrafted feature extraction libraries are not open-source or not actively maintained. As a result, a researcher often has to build such an extraction system from the ground up.

We collect and categorize more than 220 popular handcrafted features grounded on past literature. Then, we conduct a correlation analysis study on several task-specific datasets and report the potential use cases of each feature. Lastly, we devise a multilingual handcrafted linguistic feature extraction system in a systematically expandable manner. We open-source our system for public access to a rich set of pre-implemented handcrafted features. Our system is coined `LFTK` and is the largest of its kind. Find at `github.com/brucewlee/lftk`.

1 Introduction

Handcrafted linguistic features have long been inseparable from natural language processing (NLP) research. Even though automatically-generated features (e.g., Word2Vec, BERT embeddings) have recently been mainstream focus due to fewer manual efforts required, handcrafted features (e.g., type-token ratio) are still actively found in currently literature trend (Weiss and Meurers, 2022; Campillo-Ageitos et al., 2021; Chatzipanagiotidis et al., 2021; Kamyab et al., 2021; Qin et al., 2021; Esmaeilzadeh and Taghva, 2021). Therefore, it is evident that there is a constant demand for both the identification of new handcrafted features and utilization of existing handcrafted features.

[3]Core contributor

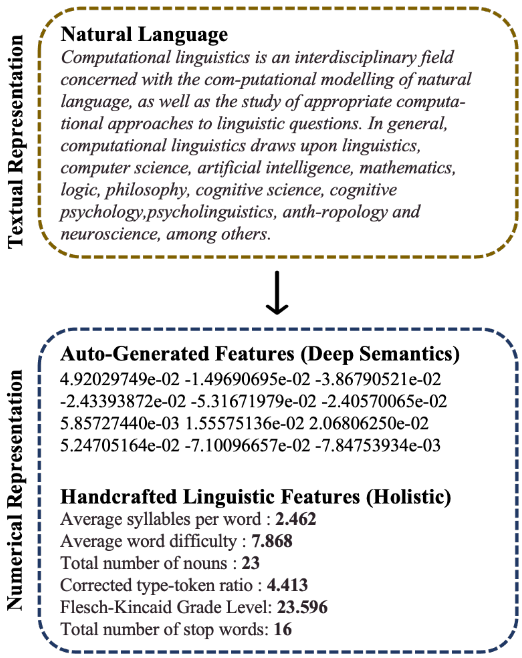

Figure 1: Difference between auto-generated (deep semantic embeddings) and handcrafted features.

After reviewing the recent research, we observed that most research on automatically-generated features tends to focus on creating **deeper** semantic representations of natural language. On the other hand, researchers use handcrafted features to create **wider** numerical representations, encompassing syntax, discourse, and others. An interesting new trend is that these handcrafted features are often used to assist auto-generated features in creating **wide** and **deep** representations for applications like English readability assessment (Lee et al., 2021) and automatic essay scoring (Uto et al., 2020).

The trend was observed across various tasks and languages. For example, there are Arabic speech synthesis (Amrouche et al., 2022), Burmese translation (Hlaing et al., 2022), English-French term alignment (Repar et al., 2022), German readability assessment (Blaneck et al., 2022), Italian pre-

Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023), pages 1–19

trained language model analysis (Miaschi et al., 2020), Korean news quality prediction (Choi et al., 2021), and Spanish hate-speech detection (García-Díaz et al., 2022) systems.

Though using handcrafted features seems to benefit multiple research fields, current feature extraction practices suffer from critical weaknesses. One is the inconsistent implementations of the same handcrafted feature across research works. For example, the exact implementation of the *average words per sentence* feature can be different in Lee et al. (2021) and Pitler and Nenkova (2008) even though both works deal with text readability. Also, there have been no standards for categorizing these handcrafted features, which furthers the confusion.

In addition, no open-source feature extraction system works multilingual, though handcrafted features are increasingly used in non-English applications. The handcrafted linguistic features can be critical resources for understudied or low-resource languages because they often lack high-performance textual encoding models like BERT. In such cases, handcrafted features can be useful in creating text embeddings for machine learning studies (Zhang et al., 2022; Kruse et al., 2021; Maamuujav et al., 2021). In this paper, we make two contributions to address the shortcomings in the current handcrafted feature extraction practices.

1. We systematically categorize an extensive set of reported handcrafted features and create a feature extraction toolkit. The main contribution of this paper is that we collect more than 200 handcrafted features from diverse NLP research, like text readability assessment, and categorize them. We take a systematic approach for easiness in future expansion. Notably, we designed the system so that a fixed set of *foundation features* can build up to various *derivation features*. We then categorize the implemented features into four linguistic branches and 12 linguistic families, considering the original author's intention. The linguistic features are also labeled with available language, depending on whether our system can extract the feature in a language-agnostic manner. `LFTK` (**Linguistic Feature ToolKit**) is built on top of another open-source library, `spaCy`[1], to ensure high-performance parsing, multilingualism, and future reproducibility by citing a specific version. Our feature extraction software aims to cover most of the generally found handcrafted linguistic features in recent research.

[1]github.com/explosion/spaCy

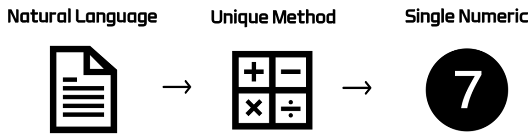

Figure 2: The three constituents of a handcrafted linguistic feature.

2. We report basic correlation analysis on various task-specific datasets. Due to the nature of the tasks, most handcrafted features are from text readability assessment or linguistic analysis studies with educational applications in mind. The broader applications of these handcrafted features to other fields, like text simplification or machine translation corpus generation, have been only reported fairly recently (Brunato et al., 2022; Yuksel et al., 2022). Along with the feature extraction software, we report the predictive abilities of these handcrafted features on four NLP tasks by performing a baseline correlation analysis. As we do so, we identify some interesting correlations that have not been previously reported. We believe our preliminary study can serve as a basis for future in-depth studies.

In a way, we aim to address the recent concern about the lack of ready-to-use code artifacts for handcrafted features (Vajjala, 2022). Through this work, we hope to improve the general efficiency of identifying and implementing handcrafted features for researchers in related fields.

2 Related Work

2.1 What are Handcrafted Features?

The type of linguistic feature we are interested in is often referred to as *handcrafted linguistic feature*, a term found throughout NLP research (Choudhary and Arora, 2021; Chen et al., 2021; Albadi et al., 2019; Bogdanova et al., 2017). Though the term "handcrafted linguistic features" is loosely defined, there seems to be some unspoken agreement among existing works. In this work, we define a handcrafted linguistic feature as ***a single numerical value*** *produced by* ***a uniquely identifiable method*** *on any* ***natural language*** (refer to Figure 2).

Unlike automatic or computer-generated linguistic features, these handcrafted features are often manually defined by combining the text's features with simple mathematical operations like root or division (Lee et al., 2021). For example, the *average difficulty of words* (calculated with an external word difficulty-labeled database) can be considered

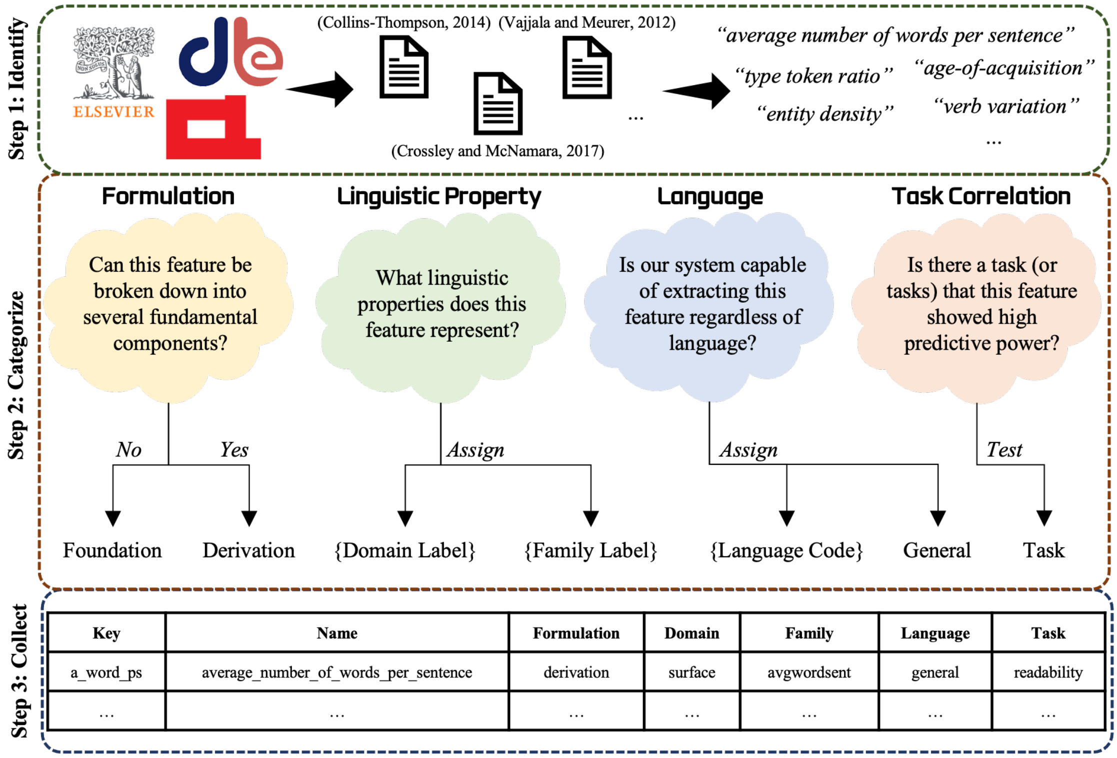

Figure 3: This diagram shows how we collected all handcrafted linguistic features implemented in our extraction software. This is also our general framework for categorizing features for future expansion too.

a handcrafted feature (Lee and Lee, 2020). Though the scope of what can be considered a single handcrafted feature is very broad, each feature always produces a single float or integer as the result of the calculation. More examples of such handcrafted features will appear as we proceed.

2.2 Hybridization of Handcrafted Features

It takes a great deal of effort to make automatic or computer-generated linguistic features capture the full linguistic properties of a text, other than its semantic meaning (Gong et al., 2022; Hewitt and Manning, 2019). For example, making BERT encodings capture **both** semantics and syntax with high quality can be difficult (Liu et al., 2020). On the other hand, combining handcrafted features to capture wide linguistic properties, such as syntax or discourse, can be methodically simpler. Hence, handcrafted features are often infused with neural networks in the last classification layer or directly with a sentence's semantic embedding to enhance the model's ability in holistic understanding (Hou et al., 2022; Lee et al., 2021). Such *feature hybridization* techniques are found in multiple NLP tasks like readability assessment (Vajjala, 2022) and essay scoring (Ramesh and Sanampudi, 2022).

2.3 Handcrafted Features in Recent Studies

Until recently, NLP tasks that require a holistic understanding of a given text have utilized machine learning models based only on handcrafted linguistic features. Such tasks include L2 learner's text readability assessment (Lee and Lee, 2020), fake news detection (Choudhary and Arora, 2021), bias detection (Spinde et al., 2021), learner-based reading passage selection (Lee and Lee, 2022). Naturally, these fields have handcrafted and identified a rich set of linguistic features we aim to collect in this study. We highlight text readability assessment research as an important source of our implemented features. Such studies often involve 80$\sim$255 features from diverse linguistic branches of advanced semantics (Lee et al., 2021), discourse (Feng et al., 2010), and syntax (Xia et al., 2016).

3 Assembling a Large-Scale Handcrafted Linguistic Feature Extractor

3.1 Overview

By exploring past works that deal with handcrafted linguistic features, we aim to implement a comprehensive set of features. These features are commonly found across NLP tasks, but ready-to-use

Type	Name	Description	Example
Branch	Lexico-Semantics	attributes associated with words	Total Word Difficulty Score
Branch	Discourse	high-level dependencies between words and sentences	Total # of Named Entities
Branch	Syntax	arrangement of words and phrases	Total # of Nouns
Branch	Surface	no specifiable linguistic property	Total # of Words

Table 1: All available linguistic branches at the current version of our extraction software. The feature names in the example column are given in abbreviated formats due to space limits. We use # to indicate "number of".

Type	Name	Description	Example
Family (F.)	WordSent	basic counts of characters, syllables, words, and sentences	Total # of Sentences
Family (F.)	WordDiff	word difficulty, frequency, and familiarity statistics	Total Word Difficulty Score
Family (F.)	PartOfSpeech	features that deal with POS (UPOS*)	Total # of Verbs
Family (F.)	Entity	named entities or entities, such as location or person	Total # of Named Entities
Family (D.)	AvgWordSent	averages of WordSent features per word, sentence, etc.	Avg. # of Words per Sentence
Family (D.)	AvgWordDiff	averages of WordDiff features per word, sentence, etc.	Avg. Word Difficulty per Word
Family (D.)	AvgPartOfSpeech	averages of PartOfSpeech features per word, sentence, etc.	Avg. # of Verbs per Sentence
Family (D.)	AvgEntity	averages of Entities features per word, sentence, etc.	Avg. # of Entities per Word
Family (D.)	LexicalVariation	features that measure lexical variation (that are not TTR)	Squared Verb Variation
Family (D.)	TypeTokenRatio	type-token ratio statistics to capture lexical richness	Corrected Type Token Ratio
Family (D.)	ReadFormula	traditional readability formulas	Flesch-Kincaid Grade Level
Family (D.)	ReadTimeFormula	basic reading time formulas	Reading Time of Fast Readers

Table 2: All available linguistic families at the current version of our extraction software. As explained in section 3.2.2, family is either *F.*: Foundation or *D.*: Derivation. *UPOS refers to Universal POS <universaldependencies.org/u/pos/>.

public codes rarely exist. We collected and categorized over 200 handcrafted features from past research works, mostly on text readability assessment, automated essay scoring, fake news detection, and paraphrase detection. These choices of works are due to their natural intimate relationships with handcrafted features and also, admittedly, due to the authors' limited scope of expertise. Figure 3 depicts our general process of implementing a single feature. Tables 1 and 2 show more details on categorization.

3.2 Categorization

3.2.1 Formulation

The main idea behind our system is that most handcrafted linguistic features can be broken down into multiple fundamental blocks. Depending on whether a feature can be split into smaller building blocks, we categorized all collected features into either foundation or derivation. Then, we designed the extraction system to build all derivation features on top of the corresponding foundation features. This enables us to exploit all available combinations efficiently and ensure a unified extraction algorithm across features of similar properties.

The derivation features are simple mathematical combinations of one or more foundation features. For example, the *average number of words per sentence* is a derivation feature, defined by dividing *total number of words* by *total number of sentences*. A foundation feature can be the fundamental building block of several derivation features. But again, a foundation feature cannot be split into smaller building blocks. We build 155 derivation features out of 65 foundation features in the current version.

3.2.2 Linguistic Property

Each handcrafted linguistic feature represents a certain linguistic property. But it is often difficult to pinpoint the exact property because features tend to correlate with one another. Such colinear inter-dependencies have been reported by multiple pieces of literature (Imperial et al., 2022; Lee and Lee, 2020). Hence, we only categorize all features into the broad linguistic branches of lexico-semantics, syntax, discourse, and surface. The surface branch can also hold features that do not belong to any specific linguistic branch. The linguistic branches are categorized in reference to Collins-Thompson (2014). We mainly considered the original author's intention when assigning a linguistic branch in unclear cases.

Apart from linguistic branches, handcrafted features are also categorized into linguistic families. The linguistic families are meant to group features into smaller subcategories. The main function of linguistic family is to enable efficient feature search.

		Foundation A	
		General	Specific
Foundation B	General	*General*	*Specific*
	Specific	*Specific*	*Specific*

Table 3: A theoretical example of determining the applicable language of a derivation feature that builds on top of two foundation features.

All family names are unique, and each family belongs to a specific formulation type. This means that the features in a family are either all foundation or all derivation. A linguistic family also serves as a building block of our feature extraction system. Our extraction program is a linked collection of several feature extraction modules, each representing a linguistic family (refer to Figure 4).

3.2.3 Applicable Language

Since handcrafted features are increasingly used for non-English languages, it is important to deduce whether a feature is generally extractable across languages. Though our extraction system is also designed with English applications in mind, we devised a systematic approach to deduce if an implemented feature is language agnostic. Like the example in Table 3, we only classify a derivation feature as generally applicable if all its components (foundation features) are generally applicable.

We can take the example of the *average number of nouns per sentence*, defined by dividing *total number of nouns* by *total number of sentences*. Since both component foundation features are generally applicable (we use UPOS tagging scheme), we can deduce that the derivation is generally applicable too. On the other hand, *Flesch-Kincaid Grade Level* (FKGL) is not generally applicable because our syllables counter is English-specific.

$$\text{FKGL} = 0.39 \cdot \frac{\text{\# word}}{\text{\# sent}} + 11.8 \cdot \frac{\text{\# syllable}}{\text{\# word}} - 15.59$$

There is no guarantee that a feature works similarly in multiple languages. The usability of a feature in a new language is subject to individual exploration.

3.3 Feature Details by Linguistic Family

Due to space restrictions, we only report the number of implemented features in Tables 4 and 5. A full list of these features is available in the Appendices. The following sections are used to elaborate on the motivations and implementations behind features.

Name	**Feature Count**
Lexico-Semantics	70
Discourse	57
Syntax	69
Surface	24
Total	220

Table 4: Feature count by branch

Name	**Feature Count**
WordSent	9
WordDiff	3
PartOfSpeech	34
Entity	19
AvgWordSent	7
AvgWordDiff	6
AvgPartOfSpeech	34
AvgEntity	38
LexicalVariation	51
TypeTokenRatio	10
ReadFormula	6
ReadTimeFormula	3
Total	220

Table 5: Feature count by family

3.3.1 `WordSent` & `AvgWordSent`

`WordSent` is a family of foundation features for character, syllable, word, and sentence count statistics. With the exception of syllables, this family heavily depends on spaCy for tokenization. SpaCy is a high-accuracy parser module that has been used as a base tokenizer in several multilingual projects like the Berkeley Neural Parser (Kitaev et al., 2019). We use a custom syllables count algorithm.

`AvgWordSent` is a family of derivation features for averaged character, syllable, word, and sentence count statistics. An example is the *average number of syllables per word*, a derivation of the *total number of words* and the *total number of syllables* foundation features.

3.3.2 `WordDiff` & `AvgWordDiff`

`WordDiff` is a family of foundation features for word difficulty analysis. This is a major topic in educational applications and second language acquisition studies, represented by age-of-acquisition (AoA, the age at which a word is learned) and corpus-based word frequency studies. Notably, there is the Kuperman AoA rating of over 30,000 words (Kuperman et al., 2012), an implemented feature in our extraction system. Another implemented feature is the word frequency statistics based on SUBLTEXus research, an improved word frequency measure based on American English subtitles (Brysbaert et al., 2012). `AvgWordDiff` averages the `WordDiff` features by word or sentence counts. This enables features like the *average Kuperman's age-of-acquisition per word.*

3.3.3 `PartOfSpeech` & `AvgPartOfSpeech`

`PartOfSpeech` is a family of foundation features that count part-of-speech (POS) properties on the token level based on dependency parsing. Here, we use spaCy's dependency parser, which is available in multiple languages. All POS counts are based on the UPOS tagging scheme to ensure multilingualism. These POS count-based features are found multiple times across second language acquisition research (Xia et al., 2016; Vajjala and Meurers, 2012). The features in `AvgPartOfSpeech` family are the averages of `PartOfSpeech` features by word or sentence counts. One example is the *average number of verbs per sentence.*

3.3.4 `Entity` & `AvgEntity`

Central to discourse analysis, `Entity` is a family of foundation features that count entities. Often used to represent the discourse characteristics of a text, these features have been famously utilized by a series of research works in readability assessment to measure the cognitive reading difficulty of texts for adults with intellectual disabilities (Feng et al., 2010, 2009). `AvgEntity` family are the averages of `Entity` features by word or sentence counts. One example is the *average number of "organization" entities per sentence.*

3.3.5 `LexicalVariation`

Second language acquisition research has identified that the variation of words in the same POS category can correlate with the lexical richness of a text (Vajjala and Meurers, 2012; Housen and Kuiken, 2009). One example of a derivative feature in this module is derived by dividing the *number of unique verbs* by the *number of verbs*, often referred to as "verb variation" in other literature. There are more derivations ("verb variation - 1, 2") using squares or roots, which are also implemented in our system.

3.3.6 `TypeTokenRatio`

Type-token ratio, often called TTR, is another set of features found across second/child language acquisition research (Kettunen, 2014). This is perhaps one of the oldest lexical richness measures in a written/oral text (Hess et al., 1989; Richards, 1987). Though `TypeTokenRatio` features aim to measure similar textual characteristics as `LexicalVariation` features, we separated TTR into a separate family due to its unique prevalence.

Pipeline	Time (sec)
en_core_web_sm + LFTK	12.12
en_core_web_md + LFTK	13.61
en_core_web_lg + LFTK	14.32
en_core_web_trf + LFTK	16.16

Table 6: Average time taken for extracting 220 handcrafted features from a dummy text of 1000 words. spaCy module is quite inconsistent in processing time, varying by at most 2∼3 seconds.

3.3.7 `ReadFormula`

Before machine learning techniques were applied to text readability assessment, linear formulas were used to represent the readability of a text quantitatively (Solnyshkina et al., 2017). Recently, these formulas have been utilized for diverse NLP tasks like fake news classification (Choudhary and Arora, 2021) and authorship attribution (Uchendu et al., 2020). We have implemented the traditional readability formulas that are popularly used across recent works (Lee and Lee, 2023; Horbach et al., 2022; Gooding et al., 2021; Nahatame, 2021).

3.4 LFTK in Context

As we have explored, we tag each handcrafted linguistic feature with three attributes: domain, family, and language. These attributes assist researchers in efficiently searching for the feature they need, one of two research goals we mentioned in section 1. Instead of individually searching for handcrafted features, they can sort and extract features in terms of attributes.

Notably, our extraction system is fully implemented in the programming language Python, unlike other systems like Coh-Metrix (Graesser et al., 2004) and L2 Syntactic Complexity Analyzer (Lu, 2017). Considering the modern NLP research approaches (Mishra and Mishra, 2022; Sengupta, 2021; JUGRAN et al., 2021; Sarkar, 2019), the combination of open-source development and Python makes our extraction system more expandable and customizable in the community.

Time with spaCy model's processing time is reported in Table 6. Excluding the spaCy model's processing time (which is not a part of our extraction system), our system can extract 220 handcrafted features from a dummy text of 1000 words on an average of 10 seconds. This translates to about 0.01 seconds per word, and this result is ob-

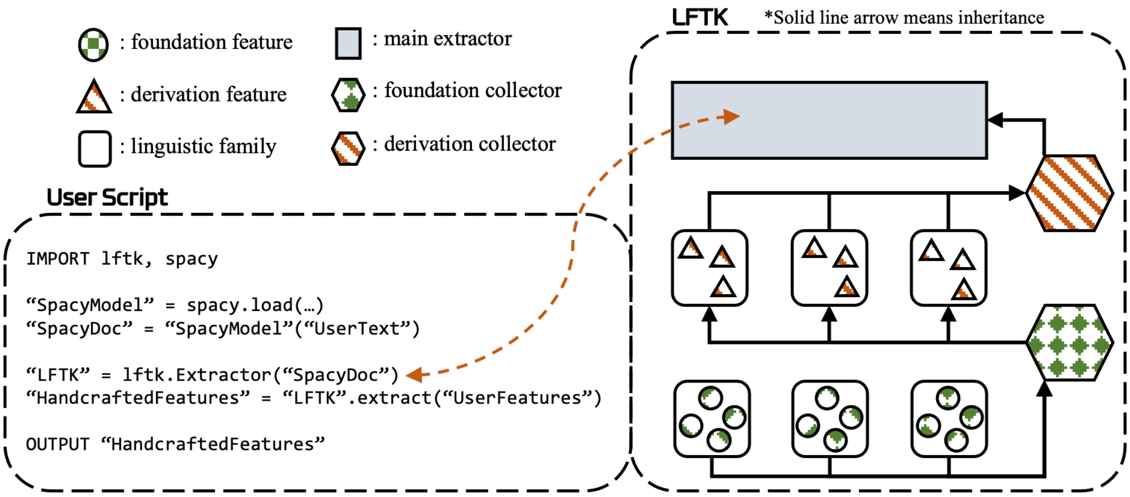

Figure 4: Schematic representation of how a user might use LFTK to extract handcrafted features. Black line arrows represent inheritance relationships. Our extraction system is a collection of multiple linguistic family modules. To interweave this program and resolve multiple dependencies, we designed a foundation collector object to inherit all foundation linguistic families first. Then all derivation linguistic families inherit the same foundation collector object. A derivation collector then inherits all derivation linguistic families, and the main extractor object inherits the derivation collector object. Considering the recent research trend, our program is solely based on the programming language Python.

tained by averaging over 20 trials of randomized dummy texts of exactly 1000 words. This time was taken with a 2.3 GHz Intel Core i9 CPU under a single-core setup. The fast extraction speed makes our extraction system suitable for large-scale corpus studies. Since our extraction system works with a wide variety of tokenizers (different accuracies and processing times) available through spaCy, one might choose an appropriate model according to the size of the studied text. Since spaCy and our extraction system are open sources registered through the Python Package Index (PyPI), reproducibility can easily be maintained by versions.

In addition, our extraction system achieves such a speed improvement due to our systematic breakdown of handcrafted features into foundation and derivation (see section 3.1.1). As depicted in Figure 4, designing the system so that derivation features are built on top of foundation features reduced duplicate program calculation to a minimum. Once a foundation feature is calculated, it is saved and used by multiple derivation features. Indeed, the *total number of words* does not have to be calculated twice for *average word difficulty per word* and *Flesch-Kincaid Grade Level*.

4 Which applies to which? Task-Feature Correlation Analysis

For handcrafted features to be generally useful to the larger NLP community, it can be important to provide researchers with a sense of which features can be potentially good in their problem setup. This section reports simple correlation analysis results of our implemented features and four NLP tasks.

To the best of our knowledge, we chose the representative dataset for each task. Table 7 reports the Pearson correlation between the feature and the dataset labels. We only report the top 10 features and bottom ten features. The full result is available in the Appendices. We used the CLEAR corpus's *crowdsourced algorithm of reading comprehension score controlled for text length* (CAREC_M) for readability labels on 4724 instances (Crossley et al., 2022). We used the ASAP dataset's[2] *domain1_score* on prompt 1 essays for student essay scoring labels on 1783 instances. We used the LIAR dataset for fake news labels on 10420 instances (Wang, 2017). We used SemEval 2019 Task 5 dataset's *PS* for binary hate speech labels on 9000 instances (Basile et al., 2019).

Though limited, our preliminary correlation analysis reveals some interesting correlations that have rarely been reported. For example, n_verb negatively correlates with the difficulty of a text. But there is much room to be explored. One utility behind a large-scale feature extraction system like ours is the ease of revealing novel correlations that might not have been obvious.

[2]www.kaggle.com/c/asap-aes/data

Readability Assessment CLEAR		**Essay Scoring** ASAP		**Fake News Detection** LIAR		**Hate Speech Detection** SemEval-2019 Task 5	
Feature	r	Feature	r	Feature	r	Feature	r
cole	0.716	t_uword	0.832	root_num_var	0.0996	n_sym	0.134
a_char_pw	0.716	t_char	0.820	corr_num_var	0.0996	a_sym_pw	0.109
a_syll_pw	0.709	t_syll	0.819	simp_num_var	0.0992	simp_det_var	0.107
t_syll2	0.700	rt_slow	0.807	a_num_pw	0.0962	root_det_var	0.102
smog	0.685	t_word	0.807	a_num_ps	0.0855	corr_det_var	0.102
a_kup_pw	0.643	rt_fast	0.807	t_n_ent_date	0.0811	t_punct	0.097
t_syll3	0.625	rt_average	0.807	n_unum	0.0810	n_usym	0.096
fogi	0.573	t_kup	0.806	a_n_ent_date_pw	0.0772	t_sent	0.094
a_noun_pw	0.545	t_bry	0.792	a_n_ent_date_ps	0.0763	a_sym_ps	0.091
fkgl	0.544	n_noun	0.779	t_n_ent_money	0.0738	root_pron_var	0.090
...							
n_adv	-0.376	a_subtlex_us_zipf_pw	-0.295	n_upropn	-0.0637	t_n_ent_date	-0.085
t_stopword	-0.378	simp_pron_var	-0.307	a_syll_pw	-0.0712	a_n_ent_pw	-0.086
n_uverb	-0.381	simp_part_var	-0.366	root_propn_var	-0.0719	a_n_ent_date_pw	-0.088
simp_adp_var	-0.462	simp_aux_var	-0.399	corr_propn_var	-0.0720	a_n_ent_gpe_pw	-0.090
a_verb_pw	-0.481	simp_cconj_var	-0.438	a_propn_ps	-0.0745	a_adp_pw	-0.096
n_verb	-0.508	simp_ttr	-0.448	a_verb_pw	-0.0775	simp_ttr_no_lem	-0.122
n_upron	-0.531	simp_ttr_no_lem	-0.448	t_n_ent_person	-0.0790	simp_ttr	-0.122
a_pron_pw	-0.649	simp_punct_var	-0.519	a_n_ent_person_ps	-0.0822	auto	-0.156
n_pron	-0.653	simp_det_var	-0.530	a_n_ent_person_pw	-0.0850	a_char_pw	-0.167
fkre	-0.687	simp_adp_var	-0.533	a_propn_pw	-0.0979	cole	-0.174

Table 7: Task, dataset, and top 10 correlated features (reported both in the positive and negative direction). Under our experimental setup, positive is more difficult in readability assessment. Positive is well-written in essay scoring. Positive is more truthful in fake news detection. Positive is hateful in hate speech detection. We only report feature keys due to space restrictions. The full correlation analysis and key-description pairs are available in the Appendices.

5 Conclusion

In this paper, we have reported our open-source, large-scale handcrafted feature extraction system. Though our extraction system covers a large set of pre-implemented features, newer, task-specific features are constantly developed. For example, *URLs count* is used for Twitter bot detection (Gilani et al., 2017) and *grammatical error count* is used for automated essay scoring (Attali and Burstein, 2006). These features, too, fall under our definition (Figure 2) of handcrafted linguistic features. Our open-source script is easily expandable, making creating a modified, research-specific version of our extraction program more convenient. With various foundation features to build from, our extraction program will be a good starting point.

Another potential user group of our extraction library is those looking to improve a neural or non-neural model's performance by incorporating more features. Performance-wise, the breadth of linguistic coverage is often as important as selection (Lee et al., 2021; Yaneva et al., 2021; Klebanov and Madnani, 2020; Horbach et al., 2013). Our current work has various implemented features, and we believe the extraction system can be a good starting point for many research works.

Compared to other historically important code artifacts like the Coh-Metrix (Graesser et al., 2004) and L2 Syntactic Complexity Analyzer (Lu, 2017), our extraction system is comparable or larger in size. To the best of our knowledge, this research is the first attempt to create a "general-purpose" handcrafted feature extraction system. That is, we wanted to build a system that can be widely used across NLP tasks. To do so, we have considered expandability and multilingualism from architecture design. And such consideration is grounded in the systematic categorization of popular handcrafted linguistic features into the attributes like domain and family. With the open-source release of our system, we hope that the current problems in feature extraction practices (section 1) can be alleviated.

References

Nuha Albadi, Maram Kurdi, and Shivakant Mishra. 2019. Investigating the effect of combining gru neural networks with handcrafted features for religious hatred detection on arabic twitter space. *Social Network Analysis and Mining*, 9(1):41.

Aissa Amrouche, Youssouf Bentrcia, Khadidja Nesrine Boubakeur, and Ahcène Abed. 2022. Dnn-based arabic speech synthesis. In *2022 9th International Conference on Electrical and Electronics Engineering (ICEEE)*, pages 378–382. IEEE.

Yigal Attali and Jill Burstein. 2006. Automated essay scoring with e-rater® v. 2. *The Journal of Technology, Learning and Assessment*, 4(3).

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of the 13th international workshop on semantic evaluation*, pages 54–63.

Patrick Gustav Blaneck, Tobias Bornheim, Niklas Grieger, and Stephan Bialonski. 2022. Automatic readability assessment of German sentences with transformer ensembles. In *Proceedings of the GermEval 2022 Workshop on Text Complexity Assessment of German Text*, pages 57–62, Potsdam, Germany. Association for Computational Linguistics.

Dasha Bogdanova, Jennifer Foster, Daria Dzendzik, and Qun Liu. 2017. If you can't beat them join them: handcrafted features complement neural nets for non-factoid answer reranking. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 121–131.

Dominique Brunato, Felice Dell'Orletta, and Giulia Venturi. 2022. Linguistically-based comparison of different approaches to building corpora for text simplification: A case study on italian. *Frontiers in Psychology*, 13.

Marc Brysbaert, Boris New, and Emmanuel Keuleers. 2012. Adding part-of-speech information to the subtlex-us word frequencies. *Behavior research methods*, 44:991–997.

Elena Campillo-Ageitos, Hermenegildo Fabregat, Lourdes Araujo, and Juan Martinez-Romo. 2021. Nlp-uned at erisk 2021: self-harm early risk detection with tf-idf and linguistic features. *Working Notes of CLEF*, pages 21–24.

Savvas Chatzipanagiotidis, Maria Giagkou, and Detmar Meurers. 2021. Broad linguistic complexity analysis for greek readability classification. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 48–58.

Mingxuan Chen, Xinqiao Chu, and KP Subbalakshmi. 2021. Mmcovar: multimodal covid-19 vaccine focused data repository for fake news detection and a baseline architecture for classification. In *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 31–38.

Sujin Choi, Hyopil Shin, and Seung-Shik Kang. 2021. Predicting audience-rated news quality: Using survey, text mining, and neural network methods. *Digital Journalism*, 9(1):84–105.

Anshika Choudhary and Anuja Arora. 2021. Linguistic feature based learning model for fake news detection and classification. *Expert Systems with Applications*, 169:114171.

Kevyn Collins-Thompson. 2014. Computational assessment of text readability: A survey of current and future research. *ITL-International Journal of Applied Linguistics*, 165(2):97–135.

Scott Crossley, Aron Heintz, Joon Suh Choi, Jordan Batchelor, Mehrnoush Karimi, and Agnes Malatinszky. 2022. A large-scaled corpus for assessing text readability. *Behavior Research Methods*, pages 1–17.

Armin Esmaeilzadeh and Kazem Taghva. 2021. Text classification using neural network language model (nnlm) and bert: An empirical comparison. In *Proceedings of SAI Intelligent Systems Conference*, pages 175–189. Springer.

Lijun Feng, Noémie Elhadad, and Matt Huenerfauth. 2009. Cognitively motivated features for readability assessment. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 229–237.

Lijun Feng, Martin Jansche, Matt Huenerfauth, and Noémie Elhadad. 2010. A comparison of features for automatic readability assessment. In *Coling 2010: Posters*, pages 276–284.

José Antonio García-Díaz, Salud María Jiménez-Zafra, Miguel Angel García-Cumbreras, and Rafael Valencia-García. 2022. Evaluating feature combination strategies for hate-speech detection in spanish using linguistic features and transformers. *Complex & Intelligent Systems*, pages 1–22.

Zafar Gilani, Ekaterina Kochmar, and Jon Crowcroft. 2017. Classification of twitter accounts into automated agents and human users. In *Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining 2017*, pages 489–496.

Zheng Gong, Kun Zhou, Wayne Xin Zhao, Jing Sha, Shijin Wang, and Ji-Rong Wen. 2022. Continual pre-training of language models for math problem understanding with syntax-aware memory network. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5923–5933.

Sian Gooding, Ekaterina Kochmar, Seid Muhie Yimam, and Chris Biemann. 2021. Word complexity is in the eye of the beholder. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4439–4449.

Arthur C Graesser, Danielle S McNamara, Max M Louwerse, and Zhiqiang Cai. 2004. Coh-metrix: Analysis of text on cohesion and language. *Behavior research methods, instruments, & computers*, 36(2):193–202.

Carla W Hess, Holly T Haug, and Richard G Landry. 1989. The reliability of type-token ratios for the oral language of school age children. *Journal of Speech, Language, and Hearing Research*, 32(3):536–540.

John Hewitt and Christopher D Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138.

Zar Zar Hlaing, Ye Kyaw Thu, Thepchai Supnithi, and Ponrudee Netisopakul. 2022. Improving neural machine translation with pos-tag features for low-resource language pairs. *Heliyon*, 8(8):e10375.

Andrea Horbach, Alexis Palmer, and Manfred Pinkal. 2013. Using the text to evaluate short answers for reading comprehension exercises. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 286–295.

Serge PJM Horbach, Jesper W Schneider, and Maxime Sainte-Marie. 2022. Ungendered writing: Writing styles are unlikely to account for gender differences in funding rates in the natural and technical sciences. *Journal of Informetrics*, 16(4):101332.

Shudi Hou, Simin Rao, Yu Xia, and Sujian Li. 2022. Promoting pre-trained lm with linguistic features on automatic readability assessment. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, pages 430–436.

Alex Housen and Folkert Kuiken. 2009. Complexity, Accuracy, and Fluency in Second Language Acquisition. *Applied Linguistics*, 30(4):461–473.

Joseph Marvin Imperial, Lloyd Lois Antonie Reyes, Michael Antonio Ibanez, Ranz Sapinit, and Mohammed Hussien. 2022. A baseline readability model for cebuano. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 27–32.

SWARANJALI JUGRAN, ASHISH KUMAR, BHUPENDRA SINGH TYAGI, and VIVEK ANAND. 2021. Extractive automatic text summarization using spacy in python & nlp. In *2021 International conference on advance computing and innovative technologies in engineering (ICACITE)*, pages 582–585. IEEE.

Marjan Kamyab, Guohua Liu, and Michael Adjeisah. 2021. Attention-based cnn and bi-lstm model based on tf-idf and glove word embedding for sentiment analysis. *Applied Sciences*, 11(23):11255.

Kimmo Kettunen. 2014. Can type-token ratio be used to show morphological complexity of languages? *Journal of Quantitative Linguistics*, 21(3):223–245.

Nikita Kitaev, Steven Cao, and Dan Klein. 2019. Multilingual constituency parsing with self-attention and pre-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505.

Beata Beigman Klebanov and Nitin Madnani. 2020. Automated evaluation of writing–50 years and counting. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 7796–7810.

Jessica Kruse, Paloma Toledo, Tayler B Belton, Erica J Testani, Charlesnika T Evans, William A Grobman, Emily S Miller, and Elizabeth MS Lange. 2021. Readability, content, and quality of covid-19 patient education materials from academic medical centers in the united states. *American Journal of Infection Control*, 49(6):690–693.

Victor Kuperman, Hans Stadthagen-Gonzalez, and Marc Brysbaert. 2012. Age-of-acquisition ratings for 30,000 english words. *Behavior research methods*, 44:978–990.

Bruce W Lee, Yoo Sung Jang, and Jason Lee. 2021. Pushing on text readability assessment: A transformer meets handcrafted linguistic features. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10669–10686.

Bruce W Lee and Jason Lee. 2020. Lxper index 2.0: Improving text readability assessment model for l2 english students in korea. In *Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 20–24.

Bruce W Lee and Jason H Lee. 2022. Auto-select reading passages in english assessment tests? *arXiv preprint arXiv:2205.06961*.

Bruce W Lee and Jason Hyung-Jong Lee. 2023. Traditional readability formulas compared for english. *arXiv preprint arXiv:2301.02975*.

Tao Liu, Xin Wang, Chengguo Lv, Ranran Zhen, and Guohong Fu. 2020. Sentence matching with syntax- and semantics-aware bert. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3302–3312.

Xiaofei Lu. 2017. Automated measurement of syntactic complexity in corpus-based l2 writing research and implications for writing assessment. *Language testing.*, 34(4).

Undarmaa Maamuujav, Carol Booth Olson, and Huy Chung. 2021. Syntactic and lexical features of adolescent l2 students' academic writing. *Journal of Second Language Writing*, 53:100822.

Alessio Miaschi, Gabriele Sarti, Dominique Brunato, Felice Dell'Orletta, and Giulia Venturi. 2020. Italian transformers under the linguistic lens. In *CLiC-it*.

Pradeepta Mishra and Pradeepta Mishra. 2022. Explainability for nlp. *Practical Explainable AI Using Python: Artificial Intelligence Model Explanations Using Python-based Libraries, Extensions, and Frameworks*, pages 193–227.

Shingo Nahatame. 2021. Text readability and processing effort in second language reading: A computational and eye-tracking investigation. *Language learning*, 71(4):1004–1043.

Emily Pitler and Ani Nenkova. 2008. Revisiting readability: A unified framework for predicting text quality. In *Proceedings of the 2008 conference on empirical methods in natural language processing*, pages 186–195.

Han Qin, Yuanhe Tian, and Yan Song. 2021. Relation extraction with word graphs from n-grams. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2860–2868.

Dadi Ramesh and Suresh Kumar Sanampudi. 2022. An automated essay scoring systems: a systematic literature review. *Artificial Intelligence Review*, 55(3):2495–2527.

Andraž Repar, Senja Pollak, Matej Ulčar, and Boshko Koloski. 2022. Fusion of linguistic, neural and sentence-transformer features for improved term alignment. In *Proceedings of the BUCC Workshop within LREC 2022*, pages 61–66.

Brian Richards. 1987. Type/token ratios: What do they really tell us? *Journal of child language*, 14(2):201–209.

Dipanjan Sarkar. 2019. *Text analytics with Python: a practitioner's guide to natural language processing*. Springer.

Sudhriti Sengupta. 2021. Programming languages used in ai. In *Artificial Intelligence*, pages 29–35. Chapman and Hall/CRC.

Marina Solnyshkina, Radif Zamaletdinov, Ludmila Gorodetskaya, and Azat Gabitov. 2017. Evaluating text complexity and flesch-kincaid grade level. *Journal of social studies education research*, 8(3):238–248.

Timo Spinde, Lada Rudnitckaia, Jelena Mitrović, Felix Hamborg, Michael Granitzer, Bela Gipp, and Karsten Donnay. 2021. Automated identification of bias inducing words in news articles using linguistic and context-oriented features. *Information Processing & Management*, 58(3):102505.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship attribution for neural text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8384–8395.

Masaki Uto, Yikuan Xie, and Maomi Ueno. 2020. Neural automated essay scoring incorporating handcrafted features. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6077–6088.

Sowmya Vajjala. 2022. Trends, limitations and open challenges in automatic readability assessment research. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5366–5377.

Sowmya Vajjala and Detmar Meurers. 2012. On improving the accuracy of readability classification using insights from second language acquisition. In *Proceedings of the seventh workshop on building educational applications using NLP*, pages 163–173.

William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426.

Zarah Weiss and Detmar Meurers. 2022. Assessing sentence readability for german language learners with broad linguistic modeling or readability formulas: When do linguistic insights make a difference? In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 141–153.

Menglin Xia, Ekaterina Kochmar, and Ted Briscoe. 2016. Text readability assessment for second language learners. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 12–22.

Victoria Yaneva, Daniel Jurich, Le An Ha, and Peter Baldwin. 2021. Using linguistic features to predict the response process complexity associated with answering clinical MCQs. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 223–232, Online. Association for Computational Linguistics.

Kamer Ali Yuksel, Ahmet Gunduz, Shreyas Sharma, and Hassan Sawaf. 2022. Efficient machine translation corpus generation. In *Proceedings of the 15th biennial conference of the Association for Machine Translation in the Americas (Workshop 2: Corpus Generation and Corpus Augmentation for Machine Translation)*, pages 11–17.

Xiaopeng Zhang, Xiaofei Lu, and Wenwen Li. 2022. Beyond differences: Assessing effects of shared linguistic features on l2 writing quality of two genres. *Applied Linguistics*, 43(1):168–195.

#	key	name	branch
1	t_word	total_number_of_words	wordsent
2	t_stopword	total_number_of_stop_words	wordsent
3	t_punct	total_number_of_puntuations	wordsent
4	t_syll	total_number_of_syllables	wordsent
5	t_syll2	total_number_of_words_more_than_two_syllables	wordsent
6	t_syll3	total_number_of_words_more_than_three_syllables	wordsent
7	t_uword	total_number_of_unique_words	wordsent
8	t_sent	total_number_of_sentences	wordsent
9	t_char	total_number_of_characters	wordsent
10	a_word_ps	average_number_of_words_per_sentence	avgwordsent
11	a_char_ps	average_number_of_characters_per_sentence	avgwordsent
12	a_char_pw	average_number_of_characters_per_word	avgwordsent
13	a_syll_ps	average_number_of_syllables_per_sentence	avgwordsent
14	a_syll_pw	average_number_of_syllables_per_word	avgwordsent
15	a_stopword_ps	average_number_of_stop_words_per_sentence	avgwordsent
16	a_stopword_pw	average_number_of_stop_words_per_word	avgwordsent
17	t_kup	total_kuperman_age_of_acquistion_of_words	worddiff
18	t_bry	total_brysbaert_age_of_acquistion_of_words	worddiff
19	t_subtlex_us_zipf	total_subtlex_us_zipf_of_words	worddiff
20	a_kup_pw	average_kuperman_age_of_acquistion_of_words_per_word	avgworddiff
21	a_bry_pw	average_brysbaert_age_of_acquistion_of_words_per_word	avgworddiff
22	a_kup_ps	average_kuperman_age_of_acquistion_of_words_per_sentence	avgworddiff
23	a_bry_ps	average_brysbaert_age_of_acquistion_of_words_per_sentence	avgworddiff
24	a_subtlex_us_zipf_pw	average_subtlex_us_zipf_of_words_per_word	avgworddiff
25	a_subtlex_us_zipf_ps	average_subtlex_us_zipf_of_words_per_sentence	avgworddiff
26	t_n_ent	total_number_of_named_entities	entity
27	t_n_ent_person	total_number_of_named_entities_person	entity
28	t_n_ent_norp	total_number_of_named_entities_norp	entity
29	t_n_ent_fac	total_number_of_named_entities_fac	entity
30	t_n_ent_org	total_number_of_named_entities_org	entity
31	t_n_ent_gpe	total_number_of_named_entities_gpe	entity
32	t_n_ent_loc	total_number_of_named_entities_loc	entity
33	t_n_ent_product	total_number_of_named_entities_product	entity
34	t_n_ent_event	total_number_of_named_entities_event	entity
35	t_n_ent_art	total_number_of_named_entities_art	entity
36	t_n_ent_law	total_number_of_named_entities_law	entity
37	t_n_ent_language	total_number_of_named_entities_language	entity
38	t_n_ent_date	total_number_of_named_entities_date	entity
39	t_n_ent_time	total_number_of_named_entities_time	entity
40	t_n_ent_percent	total_number_of_named_entities_percent	entity

Table 8: Key, Name, and Branch. #1 $\sim$ #40

A All implemented features

Our extraction software is named LFTK, and its current version is **1.0.9**. Tables 8, 9, 10, and 11 reference v.1.0.9. We only report linguistic family here due to space restrictions. Though our feature description will be regularly updated at this address [3] whenever there is a version update, we also put the current version's full feature table in our extraction program. Through PyPI or GitHub, the published version of our program is always retrievable.

B Feature correlations

Tables 12, 13, 14, and 15 report the full feature correlations that are not reported in Table 7. We have used spaCy's en_core_web_sm model, and the library version was **3.0.5**. Pearson correlation was calculated through the Pandas library, and its version was **1.1.4**. All versions reflect the most recent updates in the respective libraries.

[3] https://docs.google.com/spreadsheets/d/1uXtQ1ah0OL9cmHp2Hey0QcHb4bifJcQFLvYlVIAWWwQ/edit?usp=sharing

#	key	name	branch
41	t_n_ent_money	total_number_of_named_entities_money	entity
42	t_n_ent_quantity	total_number_of_named_entities_quantity	entity
43	t_n_ent_ordinal	total_number_of_named_entities_ordinal	entity
44	t_n_ent_cardinal	total_number_of_named_entities_cardinal	entity
45	a_n_ent_pw	average_number_of_named_entities_per_word	avgentity
46	a_n_ent_person_pw	average_number_of_named_entities_person_per_word	avgentity
47	a_n_ent_norp_pw	average_number_of_named_entities_norp_per_word	avgentity
48	a_n_ent_fac_pw	average_number_of_named_entities_fac_per_word	avgentity
49	a_n_ent_org_pw	average_number_of_named_entities_org_per_word	avgentity
50	a_n_ent_gpe_pw	average_number_of_named_entities_gpe_per_word	avgentity
51	a_n_ent_loc_pw	average_number_of_named_entities_loc_per_word	avgentity
52	a_n_ent_product_pw	average_number_of_named_entities_product_per_word	avgentity
53	a_n_ent_event_pw	average_number_of_named_entities_event_per_word	avgentity
54	a_n_ent_art_pw	average_number_of_named_entities_art_per_word	avgentity
55	a_n_ent_law_pw	average_number_of_named_entities_law_per_word	avgentity
56	a_n_ent_language_pw	average_number_of_named_entities_language_per_word	avgentity
57	a_n_ent_date_pw	average_number_of_named_entities_date_per_word	avgentity
58	a_n_ent_time_pw	average_number_of_named_entities_time_per_word	avgentity
59	a_n_ent_percent_pw	average_number_of_named_entities_percent_per_word	avgentity
60	a_n_ent_money_pw	average_number_of_named_entities_money_per_word	avgentity
61	a_n_ent_quantity_pw	average_number_of_named_entities_quantity_per_word	avgentity
62	a_n_ent_ordinal_pw	average_number_of_named_entities_ordinal_per_word	avgentity
63	a_n_ent_cardinal_pw	average_number_of_named_entities_cardinal_per_word	avgentity
64	a_n_ent_ps	average_number_of_named_entities_per_sentence	avgentity
65	a_n_ent_person_ps	average_number_of_named_entities_person_per_sentence	avgentity
66	a_n_ent_norp_ps	average_number_of_named_entities_norp_per_sentence	avgentity
67	a_n_ent_fac_ps	average_number_of_named_entities_fac_per_sentence	avgentity
68	a_n_ent_org_ps	average_number_of_named_entities_org_per_sentence	avgentity
69	a_n_ent_gpe_ps	average_number_of_named_entities_gpe_per_sentence	avgentity
70	a_n_ent_loc_ps	average_number_of_named_entities_loc_per_sentence	avgentity
71	a_n_ent_product_ps	average_number_of_named_entities_product_per_sentence	avgentity
72	a_n_ent_event_ps	average_number_of_named_entities_event_per_sentence	avgentity
73	a_n_ent_art_ps	average_number_of_named_entities_art_per_sentence	avgentity
74	a_n_ent_law_ps	average_number_of_named_entities_law_per_sentence	avgentity
75	a_n_ent_language_ps	average_number_of_named_entities_language_per_sentence	avgentity
76	a_n_ent_date_ps	average_number_of_named_entities_date_per_sentence	avgentity
77	a_n_ent_time_ps	average_number_of_named_entities_time_per_sentence	avgentity
78	a_n_ent_percent_ps	average_number_of_named_entities_percent_per_sentence	avgentity
79	a_n_ent_money_ps	average_number_of_named_entities_money_per_sentence	avgentity
80	a_n_ent_quantity_ps	average_number_of_named_entities_quantity_per_sentence	avgentity
81	a_n_ent_ordinal_ps	average_number_of_named_entities_ordinal_per_sentence	avgentity
82	a_n_ent_cardinal_ps	average_number_of_named_entities_cardinal_per_sentence	avgentity
83	simp_adj_var	simple_adjectives_variation	lexicalvariation
84	simp_adp_var	simple_adpositions_variation	lexicalvariation
85	simp_adv_var	simple_adverbs_variation	lexicalvariation
86	simp_aux_var	simple_auxiliaries_variation	lexicalvariation
87	simp_cconj_var	simple_coordinating_conjunctions_variation	lexicalvariation
88	simp_det_var	simple_determiners_variation	lexicalvariation
89	simp_intj_var	simple_interjections_variation	lexicalvariation
90	simp_noun_var	simple_nouns_variation	lexicalvariation
91	simp_num_var	simple_numerals_variation	lexicalvariation
92	simp_part_var	simple_particles_variation	lexicalvariation
93	simp_pron_var	simple_pronouns_variation	lexicalvariation
94	simp_propn_var	simple_proper_nouns_variation	lexicalvariation
95	simp_punct_var	simple_punctuations_variation	lexicalvariation
96	simp_sconj_var	simple_subordinating_conjunctions_variation	lexicalvariation
97	simp_sym_var	simple_symbols_variation	lexicalvariation
98	simp_verb_var	simple_verbs_variation	lexicalvariation
99	simp_space_var	simple_spaces_variation	lexicalvariation
100	root_adj_var	root_adjectives_variation	lexicalvariation

Table 9: Key, Name, and Branch. #41 $\sim$ #100

#	key	name	branch
101	root_adp_var	root_adpositions_variation	lexicalvariation
102	root_adv_var	root_adverbs_variation	lexicalvariation
103	root_aux_var	root_auxiliaries_variation	lexicalvariation
104	root_cconj_var	root_coordinating_conjunctions_variation	lexicalvariation
105	root_det_var	root_determiners_variation	lexicalvariation
106	root_intj_var	root_interjections_variation	lexicalvariation
107	root_noun_var	root_nouns_variation	lexicalvariation
108	root_num_var	root_numerals_variation	lexicalvariation
109	root_part_var	root_particles_variation	lexicalvariation
110	root_pron_var	root_pronouns_variation	lexicalvariation
111	root_propn_var	root_proper_nouns_variation	lexicalvariation
112	root_punct_var	root_punctuations_variation	lexicalvariation
113	root_sconj_var	root_subordinating_conjunctions_variation	lexicalvariation
114	root_sym_var	root_symbols_variation	lexicalvariation
115	root_verb_var	root_verbs_variation	lexicalvariation
116	root_space_var	root_spaces_variation	lexicalvariation
117	corr_adj_var	corrected_adjectives_variation	lexicalvariation
118	corr_adp_var	corrected_adpositions_variation	lexicalvariation
119	corr_adv_var	corrected_adverbs_variation	lexicalvariation
120	corr_aux_var	corrected_auxiliaries_variation	lexicalvariation
121	corr_cconj_var	corrected_coordinating_conjunctions_variation	lexicalvariation
122	corr_det_var	corrected_determiners_variation	lexicalvariation
123	corr_intj_var	corrected_interjections_variation	lexicalvariation
124	corr_noun_var	corrected_nouns_variation	lexicalvariation
125	corr_num_var	corrected_numerals_variation	lexicalvariation
126	corr_part_var	corrected_particles_variation	lexicalvariation
127	corr_pron_var	corrected_pronouns_variation	lexicalvariation
128	corr_propn_var	corrected_proper_nouns_variation	lexicalvariation
129	corr_punct_var	corrected_punctuations_variation	lexicalvariation
130	corr_sconj_var	corrected_subordinating_conjunctions_variation	lexicalvariation
131	corr_sym_var	corrected_symbols_variation	lexicalvariation
132	corr_verb_var	corrected_verbs_variation	lexicalvariation
133	corr_space_var	corrected_spaces_variation	lexicalvariation
134	simp_ttr	simple_type_token_ratio	typetokenratio
135	root_ttr	root_type_token_ratio	typetokenratio
136	corr_ttr	corrected_type_token_ratio	typetokenratio
137	bilog_ttr	bilogarithmic_type_token_ratio	typetokenratio
138	uber_ttr	uber_type_token_ratio	typetokenratio
139	simp_ttr_no_lem	simple_type_token_ratio_no_lemma	typetokenratio
140	root_ttr_no_lem	root_type_token_ratio_no_lemma	typetokenratio
141	corr_ttr_no_lem	corrected_type_token_ratio_no_lemma	typetokenratio
142	bilog_ttr_no_lem	bilogarithmic_type_token_ratio_no_lemma	typetokenratio
143	uber_ttr_no_lem	uber_type_token_ratio_no_lemma	typetokenratio
144	n_adj	total_number_of_adjectives	partofspeech
145	n_adp	total_number_of_adpositions	partofspeech
146	n_adv	total_number_of_adverbs	partofspeech
147	n_aux	total_number_of_auxiliaries	partofspeech
148	n_cconj	total_number_of_coordinating_conjunctions	partofspeech
149	n_det	total_number_of_determiners	partofspeech
150	n_intj	total_number_of_interjections	partofspeech
151	n_noun	total_number_of_nouns	partofspeech
152	n_num	total_number_of_numerals	partofspeech
153	n_part	total_number_of_particles	partofspeech
154	n_pron	total_number_of_pronouns	partofspeech
155	n_propn	total_number_of_proper_nouns	partofspeech
156	n_punct	total_number_of_punctuations	partofspeech
157	n_sconj	total_number_of_subordinating_conjunctions	partofspeech
158	n_sym	total_number_of_symbols	partofspeech
159	n_verb	total_number_of_verbs	partofspeech
160	n_space	total_number_of_spaces	partofspeech

Table 10: Key, Name, and Branch. #101 $\sim$ #160

#	key	name	branch
161	n_uadj	total_number_of_unique_adjectives	partofspeech
162	n_uadp	total_number_of_unique_adpositions	partofspeech
163	n_uadv	total_number_of_unique_adverbs	partofspeech
164	n_uaux	total_number_of_unique_auxiliaries	partofspeech
165	n_ucconj	total_number_of_unique_coordinating_conjunctions	partofspeech
166	n_udet	total_number_of_unique_determiners	partofspeech
167	n_uintj	total_number_of_unique_interjections	partofspeech
168	n_unoun	total_number_of_unique_nouns	partofspeech
169	n_unum	total_number_of_unique_numerals	partofspeech
170	n_upart	total_number_of_unique_particles	partofspeech
171	n_upron	total_number_of_unique_pronouns	partofspeech
172	n_upropn	total_number_of_unique_proper_nouns	partofspeech
173	n_upunct	total_number_of_unique_punctuations	partofspeech
174	n_usconj	total_number_of_unique_subordinating_conjunctions	partofspeech
175	n_usym	total_number_of_unique_symbols	partofspeech
176	n_uverb	total_number_of_unique_verbs	partofspeech
177	n_uspace	total_number_of_unique_spaces	partofspeech
178	a_adj_pw	average_number_of_adjectives_per_word	avgpartofspeech
179	a_adp_pw	average_number_of_adpositions_per_word	avgpartofspeech
180	a_adv_pw	average_number_of_adverbs_per_word	avgpartofspeech
181	a_aux_pw	average_number_of_auxiliaries_per_word	avgpartofspeech
182	a_cconj_pw	average_number_of_coordinating_conjunctions_per_word	avgpartofspeech
183	a_det_pw	average_number_of_determiners_per_word	avgpartofspeech
184	a_intj_pw	average_number_of_interjections_per_word	avgpartofspeech
185	a_noun_pw	average_number_of_nouns_per_word	avgpartofspeech
186	a_num_pw	average_number_of_numerals_per_word	avgpartofspeech
187	a_part_pw	average_number_of_particles_per_word	avgpartofspeech
188	a_pron_pw	average_number_of_pronouns_per_word	avgpartofspeech
189	a_propn_pw	average_number_of_proper_nouns_per_word	avgpartofspeech
190	a_punct_pw	average_number_of_punctuations_per_word	avgpartofspeech
191	a_sconj_pw	average_number_of_subordinating_conjunctions_per_word	avgpartofspeech
192	a_sym_pw	average_number_of_symbols_per_word	avgpartofspeech
193	a_verb_pw	average_number_of_verbs_per_word	avgpartofspeech
194	a_space_pw	average_number_of_spaces_per_word	avgpartofspeech
195	a_adj_ps	average_number_of_adjectives_per_sentence	avgpartofspeech
196	a_adp_ps	average_number_of_adpositions_per_sentence	avgpartofspeech
197	a_adv_ps	average_number_of_adverbs_per_sentence	avgpartofspeech
198	a_aux_ps	average_number_of_auxiliaries_per_sentence	avgpartofspeech
199	a_cconj_ps	average_number_of_coordinating_conjunctions_per_sentence	avgpartofspeech
200	a_det_ps	average_number_of_determiners_per_sentence	avgpartofspeech
201	a_intj_ps	average_number_of_interjections_per_sentence	avgpartofspeech
202	a_noun_ps	average_number_of_nouns_per_sentence	avgpartofspeech
203	a_num_ps	average_number_of_numerals_per_sentence	avgpartofspeech
204	a_part_ps	average_number_of_particles_per_sentence	avgpartofspeech
205	a_pron_ps	average_number_of_pronouns_per_sentence	avgpartofspeech
206	a_propn_ps	average_number_of_proper_nouns_per_sentence	avgpartofspeech
207	a_punct_ps	average_number_of_punctuations_per_sentence	avgpartofspeech
208	a_sconj_ps	average_number_of_subordinating_conjunctions_per_sentence	avgpartofspeech
209	a_sym_ps	average_number_of_symbols_per_sentence	avgpartofspeech
210	a_verb_ps	average_number_of_verbs_per_sentence	avgpartofspeech
211	a_space_ps	average_number_of_spaces_per_sentence	avgpartofspeech
212	fkre	flesch_kincaid_reading_ease	readformula
213	fkgl	flesch_kincaid_grade_level	readformula
214	fogi	gunning_fog_index	readformula
215	smog	smog_index	readformula
216	cole	coleman_liau_index	readformula
217	auto	automated_readability_index	readformula
218	rt_fast	reading_time_for_fast_readers	readtimeformula
219	rt_average	reading_time_for_average_readers	readtimeformula
220	rt_slow	reading_time_for_slow_readers	readtimeformula

Table 11: Key, Name, and Branch. #161 $\sim$ #220

Readability Assessment CLEAR		Essay Scoring ASAP		Fake News Detection LIAR		Hate Speech Detection SemEval-2019 Task 5	
Feature	r	Feature	r	Feature	r	Feature	r
cole	0.716	t_uword	0.832	root_num_var	0.100	n_sym	0.134
a_char_pw	0.716	t_char	0.820	corr_num_var	0.100	a_sym_pw	0.109
a_syll_pw	0.709	t_syll	0.819	simp_num_var	0.099	simp_det_var	0.107
t_syll2	0.700	rt_slow	0.807	a_num_pw	0.096	root_det_var	0.102
smog	0.685	t_word	0.807	a_num_ps	0.086	corr_det_var	0.102
a_kup_pw	0.643	rt_fast	0.807	t_n_ent_date	0.081	t_punct	0.097
t_syll3	0.625	rt_average	0.807	n_unum	0.081	n_usym	0.096
fogi	0.573	t_kup	0.806	a_n_ent_date_pw	0.077	t_sent	0.094
a_noun_pw	0.545	t_bry	0.792	a_n_ent_date_ps	0.076	a_sym_ps	0.091
fkgl	0.544	n_noun	0.779	t_n_ent_money	0.074	root_pron_var	0.090
t_syll	0.527	t_subtlex_us_zipf	0.770	t_n_ent_percent	0.074	corr_pron_var	0.090
a_noun_ps	0.511	n_unoun	0.752	a_adj_ps	0.073	n_pron	0.083
auto	0.498	n_uverb	0.749	a_n_ent_money_pw	0.073	simp_pron_var	0.080
a_bry_pw	0.495	n_punct	0.740	a_n_ent_percent_pw	0.073	n_upron	0.080
a_syll_ps	0.475	t_syll2	0.739	n_adj	0.071	n_verb	0.078
n_noun	0.454	t_punct	0.738	n_uadj	0.070	rt_fast	0.078
simp_pron_var	0.443	t_stopword	0.731	a_n_ent_money_ps	0.070	t_word	0.078
t_kup	0.442	n_adp	0.727	a_n_ent_percent_ps	0.070	rt_average	0.078
a_char_ps	0.429	n_verb	0.720	n_num	0.069	rt_slow	0.078
a_kup_ps	0.421	n_uadj	0.705	root_adj_var	0.069	n_udet	0.078
a_det_ps	0.420	root_ttr	0.696	corr_adj_var	0.069	corr_aux_var	0.075
a_det_pw	0.419	root_ttr_no_lem	0.696	a_stopword_pw	0.068	root_aux_var	0.075
t_char	0.416	corr_ttr_no_lem	0.696	a_n_ent_cardinal_pw	0.066	n_uaux	0.074
a_adp_pw	0.411	corr_ttr	0.696	simp_sconj_var	0.064	n_uverb	0.073
a_adj_ps	0.403	t_sent	0.693	root_sconj_var	0.064	a_det_pw	0.073
n_unoun	0.392	n_det	0.684	corr_sconj_var	0.064	root_verb_var	0.072
a_adp_ps	0.382	n_adj	0.678	a_n_ent_cardinal_ps	0.062	corr_verb_var	0.072
a_bry_ps	0.374	n_uadv	0.675	a_sconj_pw	0.062	simp_aux_var	0.066
a_adj_pw	0.366	n_uadp	0.667	t_stopword	0.061	corr_sym_var	0.066
n_det	0.340	corr_adj_var	0.651	a_adj_pw	0.061	root_sym_var	0.066
n_adp	0.332	root_adj_var	0.651	n_usconj	0.059	n_aux	0.066
n_adj	0.309	root_adv_var	0.634	t_n_ent_cardinal	0.059	fkre	0.064
n_uadj	0.305	corr_adv_var	0.634	a_stopword_ps	0.058	t_syll3	0.064
a_word_ps	0.289	n_adv	0.634	fkre	0.058	t_subtlex_us_zipf	0.064
t_bry	0.268	root_noun_var	0.625	n_sconj	0.058	t_uword	0.062
corr_adj_var	0.261	corr_noun_var	0.625	a_sconj_ps	0.057	t_stopword	0.061
root_adj_var	0.261	root_verb_var	0.617	simp_adj_var	0.052	t_syll	0.061
root_noun_var	0.243	corr_verb_var	0.617	root_noun_var	0.051	n_adv	0.058
corr_noun_var	0.243	n_aux	0.606	corr_noun_var	0.051	n_det	0.058
a_subtlex_us_zipf_ps	0.236	t_syll3	0.575	n_adp	0.050	n_uadv	0.056
simp_verb_var	0.235	n_upron	0.574	simp_adv_var	0.049	corr_adv_var	0.054
a_n_ent_norp_ps	0.226	n_udet	0.543	corr_adv_var	0.047	root_adv_var	0.054
a_n_ent_ps	0.212	n_cconj	0.530	root_adv_var	0.047	root_noun_var	0.050
a_n_ent_org_ps	0.208	n_pron	0.491	n_noun	0.043	corr_noun_var	0.050
a_aux_ps	0.204	t_n_ent	0.487	a_adp_ps	0.043	n_noun	0.049
a_n_ent_norp_pw	0.201	n_part	0.483	t_subtlex_us_zipf	0.042	corr_ttr	0.048
t_n_ent_norp	0.196	n_upropn	0.469	a_noun_ps	0.042	corr_ttr_no_lem	0.048
simp_adv_var	0.195	root_propn_var	0.466	t_kup	0.042	root_ttr	0.048
a_n_ent_gpe_ps	0.191	corr_propn_var	0.466	t_n_ent	0.042	root_ttr_no_lem	0.048
simp_ttr_no_lem	0.180	n_uaux	0.450	n_det	0.040	a_pron_pw	0.046
simp_ttr	0.180	n_upunct	0.449	n_uadv	0.040	a_pron_ps	0.044
a_stopword_ps	0.180	n_propn	0.430	n_unoun	0.040	simp_sym_var	0.043
simp_punct_var	0.177	n_usconj	0.387	n_adv	0.039	simp_adv_var	0.042
n_udet	0.171	n_sconj	0.353	a_n_ent_ps	0.038	simp_intj_var	0.042
a_propn_ps	0.168	t_n_ent_org	0.334	t_bry	0.038	a_det_ps	0.041
a_n_ent_cardinal_ps	0.165	smog	0.332	root_adp_var	0.038	t_n_ent_loc	0.040
a_num_ps	0.160	n_upart	0.331	corr_adp_var	0.038	root_intj_var	0.040
uber_ttr	0.154	a_punct_ps	0.328	n_uadp	0.037	corr_intj_var	0.040
uber_ttr_no_lem	0.154	t_n_ent_date	0.327	a_subtlex_us_zipf_ps	0.037	n_unoun	0.038
root_propn_var	0.151	a_punct_pw	0.325	a_kup_ps	0.037	n_propn	0.037

Table 12: Task, dataset, and correlated features. Part 1.

Readability Assessment CLEAR		**Essay Scoring** ASAP		**Fake News Detection** LIAR		**Hate Speech Detection** SemEval-2019 Task 5	
Feature	r	Feature	r	Feature	r	Feature	r
corr_propn_var	0.151	n_ucconj	0.320	corr_punct_var	0.036	a_aux_ps	0.035
bilog_ttr	0.147	n_unum	0.297	root_punct_var	0.036	n_upropn	0.035
bilog_ttr_no_lem	0.147	n_num	0.290	a_det_ps	0.036	n_uintj	0.035
simp_propn_var	0.147	corr_num_var	0.283	n_upunct	0.036	a_aux_pw	0.034
a_punct_ps	0.145	root_num_var	0.283	a_adv_ps	0.036	a_subtlex_us_zipf_pw	0.032
a_n_ent_gpe_pw	0.142	corr_pron_var	0.258	a_adv_pw	0.034	t_n_ent_product	0.031
a_n_ent_org_pw	0.140	root_pron_var	0.258	a_subtlex_us_zipf_pw	0.033	t_kup	0.030
a_n_ent_loc_ps	0.140	t_n_ent_cardinal	0.250	t_uword	0.032	root_part_var	0.029
n_upropn	0.134	a_char_pw	0.242	a_word_ps	0.031	corr_part_var	0.029
t_n_ent_gpe	0.132	cole	0.228	a_n_ent_ordinal_ps	0.031	n_upart	0.029
a_cconj_ps	0.129	t_n_ent_person	0.228	corr_ttr	0.031	t_bry	0.029
t_n_ent_org	0.127	a_syll_pw	0.223	corr_ttr_no_lem	0.031	n_punct	0.028
a_n_ent_cardinal_pw	0.115	t_n_ent_gpe	0.214	root_ttr	0.031	simp_part_var	0.027
a_n_ent_loc_pw	0.108	a_n_ent_pw	0.207	root_ttr_no_lem	0.031	n_intj	0.027
corr_sym_var	0.105	corr_sconj_var	0.205	rt_average	0.031	a_verb_pw	0.026
root_sym_var	0.105	root_sconj_var	0.205	rt_slow	0.031	n_usconj	0.026
simp_sym_var	0.104	simp_num_var	0.202	a_bry_ps	0.031	n_sconj	0.026
t_n_ent_loc	0.101	t_n_ent_time	0.191	t_word	0.031	corr_sconj_var	0.026
n_unum	0.101	a_propn_pw	0.183	rt_fast	0.031	root_sconj_var	0.026
t_n_ent_cardinal	0.099	a_n_ent_org_pw	0.166	t_n_ent_gpe	0.030	a_verb_ps	0.026
simp_cconj_var	0.099	a_n_ent_ps	0.166	a_noun_pw	0.029	a_stopword_pw	0.025
n_usym	0.098	a_n_ent_person_ps	0.164	t_n_ent_ordinal	0.028	simp_sconj_var	0.025
corr_cconj_var	0.095	a_n_ent_person_pw	0.153	n_udet	0.028	simp_cconj_var	0.024
root_cconj_var	0.095	corr_adp_var	0.146	t_punct	0.027	n_part	0.024
a_num_pw	0.093	root_adp_var	0.146	n_cconj	0.026	t_syll2	0.024
corr_ttr_no_lem	0.090	a_adv_pw	0.145	n_punct	0.026	simp_verb_var	0.024
corr_ttr	0.090	a_n_ent_org_ps	0.143	n_ucconj	0.026	t_char	0.023
root_ttr_no_lem	0.090	simp_propn_var	0.143	a_n_ent_gpe_ps	0.025	simp_adj_var	0.022
root_ttr	0.090	a_n_ent_date_pw	0.142	corr_cconj_var	0.025	t_n_ent_org	0.021
corr_num_var	0.088	a_n_ent_date_ps	0.138	root_cconj_var	0.025	a_n_ent_loc_ps	0.020
root_num_var	0.088	a_propn_ps	0.125	a_adp_pw	0.024	root_cconj_var	0.019
a_n_ent_money_pw	0.084	a_kup_pw	0.111	a_det_pw	0.024	corr_cconj_var	0.019
a_n_ent_percent_pw	0.084	a_n_ent_time_pw	0.101	a_n_ent_ordinal_pw	0.024	a_intj_ps	0.019
simp_part_var	0.083	a_n_ent_gpe_pw	0.094	root_det_var	0.024	t_n_ent_art	0.018
a_n_ent_pw	0.082	t_n_ent_quantity	0.091	corr_det_var	0.024	corr_adj_var	0.018
t_n_ent_percent	0.082	a_n_ent_cardinal_pw	0.090	simp_cconj_var	0.023	root_adj_var	0.018
t_n_ent_money	0.082	a_num_pw	0.088	a_punct_ps	0.023	a_n_ent_loc_pw	0.018
a_n_ent_percent_ps	0.081	n_uintj	0.088	a_kup_pw	0.023	a_adv_ps	0.017
a_n_ent_money_ps	0.081	n_intj	0.088	a_n_ent_pw	0.023	a_n_ent_product_pw	0.017
n_num	0.075	a_n_ent_time_ps	0.084	t_char	0.023	root_propn_var	0.015
a_n_ent_language_ps	0.073	a_adp_pw	0.082	a_cconj_ps	0.021	corr_propn_var	0.015
a_sym_ps	0.072	corr_aux_var	0.081	a_n_ent_gpe_pw	0.020	a_adv_pw	0.014
a_sym_pw	0.071	root_aux_var	0.081	t_sent	0.019	n_space	0.014
a_n_ent_event_ps	0.071	t_n_ent_percent	0.080	simp_adp_var	0.018	simp_noun_var	0.014
a_n_ent_law_pw	0.068	t_n_ent_money	0.080	simp_noun_var	0.016	n_adj	0.013
n_sym	0.068	a_n_ent_cardinal_ps	0.080	a_n_ent_quantity_pw	0.015	a_sconj_ps	0.013
a_n_ent_quantity_ps	0.068	corr_intj_var	0.077	a_char_ps	0.014	smog	0.012
a_n_ent_law_ps	0.067	root_intj_var	0.077	t_syll	0.014	n_ucconj	0.012
t_n_ent_law	0.065	a_n_ent_gpe_ps	0.075	simp_det_var	0.014	a_stopword_ps	0.012
a_n_ent_date_ps	0.064	uber_ttr	0.070	a_cconj_pw	0.014	a_sconj_pw	0.012
a_n_ent_language_pw	0.060	uber_ttr_no_lem	0.070	a_n_ent_quantity_ps	0.012	a_n_ent_product_ps	0.011
t_n_ent_language	0.058	a_det_pw	0.068	a_bry_pw	0.012	n_uadj	0.010
a_sconj_ps	0.057	a_n_ent_quantity_pw	0.068	t_n_ent_norp	0.011	t_n_ent_norp	0.008
a_n_ent_event_pw	0.057	a_n_ent_percent_pw	0.067	n_pron	0.010	a_subtlex_us_zipf_ps	0.008
a_n_ent_quantity_pw	0.056	a_n_ent_money_pw	0.067	t_n_ent_quantity	0.010	a_noun_pw	0.008
t_n_ent_quantity	0.054	a_n_ent_percent_ps	0.067	a_n_ent_loc_ps	0.009	a_n_ent_art_pw	0.007
t_n_ent_event	0.054	a_n_ent_money_ps	0.067	a_pron_ps	0.008	uber_ttr	0.007
a_verb_ps	0.052	a_n_ent_quantity_ps	0.065	a_n_ent_event_ps	0.008	uber_ttr_no_lem	0.007
t_n_ent	0.052	simp_intj_var	0.065	a_n_ent_norp_ps	0.008	t_n_ent_ordinal	0.007
a_n_ent_product_ps	0.046	a_num_ps	0.058	t_n_ent_event	0.008	t_n_ent_money	0.006

Table 13: Task, dataset, and correlated features. Part 2.

Readability Assessment CLEAR		**Essay Scoring** ASAP		**Fake News Detection** LIAR		**Hate Speech Detection** SemEval-2019 Task 5	
Feature	r	Feature	r	Feature	r	Feature	r
a_propn_pw	0.044	t_n_ent_loc	0.056	n_aux	0.007	t_n_ent_percent	0.006
n_ucconj	0.042	t_n_ent_product	0.049	root_pron_var	0.007	a_punct_pw	0.005
a_n_ent_ordinal_ps	0.041	t_n_ent_fac	0.048	corr_pron_var	0.007	a_noun_ps	0.005
root_punct_var	0.038	root_sym_var	0.034	a_n_ent_time_ps	0.006	n_cconj	0.003
corr_punct_var	0.038	corr_sym_var	0.034	n_upron	0.006	t_n_ent	0.003
simp_num_var	0.032	simp_sym_var	0.034	a_n_ent_loc_pw	0.005	a_n_ent_art_ps	0.001
a_n_ent_product_pw	0.031	n_usym	0.034	simp_pron_var	0.005	a_n_ent_percent_ps	0.001
t_n_ent_product	0.030	a_adj_pw	0.030	t_n_ent_loc	0.005	a_n_ent_money_ps	0.001
a_n_ent_fac_ps	0.024	root_det_var	0.028	a_n_ent_event_pw	0.005	a_word_ps	0.001
a_n_ent_art_ps	0.023	corr_det_var	0.028	t_n_ent_time	0.002	a_n_ent_ordinal_ps	-0.001
a_n_ent_fac_pw	0.019	t_n_ent_art	0.028	n_space	0.002	a_n_ent_percent_pw	-0.002
t_n_ent_fac	0.016	a_n_ent_loc_pw	0.026	a_syll_ps	0.002	a_n_ent_money_pw	-0.002
n_propn	0.015	t_n_ent_norp	0.025	a_punct_pw	0.002	a_intj_pw	-0.002
simp_space_var	0.009	n_sym	0.021	uber_ttr_no_lem	0.001	a_n_ent_law_ps	-0.005
a_n_ent_ordinal_pw	0.005	a_n_ent_product_pw	0.020	uber_ttr	0.001	n_upunct	-0.006
corr_det_var	0.001	simp_space_var	0.019	a_n_ent_time_pw	0.001	t_n_ent_law	-0.006
root_det_var	0.001	corr_space_var	0.019	simp_sym_var	0.001	a_cconj_pw	-0.007
a_n_ent_art_pw	-0.002	root_space_var	0.019	simp_aux_var	0.000	a_n_ent_fac_pw	-0.007
t_n_ent_ordinal	-0.005	t_n_ent_ordinal	0.019	a_n_ent_norp_pw	0.000	a_space_ps	-0.008
t_n_ent_art	-0.009	a_noun_pw	0.019	root_sym_var	0.000	a_n_ent_law_pw	-0.008
t_uword	-0.010	a_n_ent_loc_ps	0.017	corr_sym_var	0.000	simp_propn_var	-0.008
a_n_ent_date_pw	-0.013	a_bry_pw	0.016	a_pron_pw	-0.001	t_n_ent_fac	-0.008
a_part_ps	-0.016	n_uspace	0.015	simp_punct_var	-0.001	simp_punct_var	-0.009
a_aux_pw	-0.022	a_adv_ps	0.011	a_n_ent_language_pw	-0.002	corr_punct_var	-0.009
t_n_ent_date	-0.025	a_n_ent_fac_pw	0.010	n_usym	-0.003	root_punct_var	-0.009
a_adv_ps	-0.033	t_n_ent_event	0.008	root_aux_var	-0.003	a_space_pw	-0.009
simp_adj_var	-0.035	a_n_ent_norp_ps	0.006	corr_aux_var	-0.003	a_n_ent_quantity_ps	-0.009
a_cconj_pw	-0.054	n_space	0.004	n_sym	-0.003	t_n_ent_quantity	-0.010
simp_noun_var	-0.063	a_n_ent_product_ps	0.004	a_aux_ps	-0.003	a_n_ent_event_pw	-0.010
root_space_var	-0.072	a_n_ent_norp_pw	0.004	n_uspace	-0.003	n_uspace	-0.010
corr_space_var	-0.072	a_n_ent_event_ps	0.001	a_sym_pw	-0.003	a_n_ent_quantity_pw	-0.011
a_sconj_pw	-0.073	a_n_ent_event_pw	-0.001	t_n_ent_language	-0.004	a_n_ent_fac_ps	-0.011
n_aux	-0.081	a_space_pw	-0.001	n_uaux	-0.005	a_part_ps	-0.011
simp_sconj_var	-0.088	a_space_ps	-0.007	a_sym_ps	-0.005	a_n_ent_time_ps	-0.012
a_n_ent_time_ps	-0.091	a_n_ent_fac_ps	-0.015	t_n_ent_product	-0.005	a_n_ent_event_ps	-0.012
n_sconj	-0.096	fogi	-0.021	a_n_ent_language_ps	-0.006	simp_adp_var	-0.013
n_cconj	-0.104	a_sym_pw	-0.023	a_n_ent_product_ps	-0.007	a_punct_ps	-0.013
n_upunct	-0.115	a_sym_ps	-0.026	auto	-0.008	t_n_ent_event	-0.013
n_usconj	-0.120	a_n_ent_art_pw	-0.030	a_space_pw	-0.009	a_n_ent_ordinal_pw	-0.014
root_part_var	-0.128	fkgl	-0.032	a_n_ent_fac_pw	-0.009	a_adj_ps	-0.014
corr_part_var	-0.128	simp_adj_var	-0.033	a_n_ent_fac_ps	-0.009	a_kup_ps	-0.015
n_uadp	-0.129	auto	-0.038	simp_verb_var	-0.010	a_cconj_ps	-0.015
root_sconj_var	-0.129	a_adj_ps	-0.040	t_n_ent_fac	-0.010	a_kup_pw	-0.016
corr_sconj_var	-0.129	corr_punct_var	-0.053	root_space_var	-0.011	t_n_ent_cardinal	-0.016
a_n_ent_person_ps	-0.140	root_punct_var	-0.053	corr_space_var	-0.011	corr_space_var	-0.019
a_n_ent_time_pw	-0.145	a_n_ent_art_ps	-0.054	t_syll3	-0.011	root_space_var	-0.019
t_n_ent_time	-0.152	a_intj_pw	-0.057	a_n_ent_law_ps	-0.012	a_part_pw	-0.019
simp_det_var	-0.154	a_det_ps	-0.064	a_n_ent_art_ps	-0.012	a_adj_pw	-0.019
corr_verb_var	-0.195	a_part_pw	-0.065	a_aux_pw	-0.012	a_n_ent_time_pw	-0.021
root_verb_var	-0.195	a_adp_ps	-0.065	a_n_ent_product_pw	-0.013	root_adp_var	-0.021
n_uspace	-0.197	a_syll_ps	-0.071	n_uintj	-0.013	corr_adp_var	-0.021
root_pron_var	-0.201	a_intj_ps	-0.074	a_n_ent_law_pw	-0.013	a_syll_ps	-0.021
corr_pron_var	-0.201	fkre	-0.075	simp_intj_var	-0.013	a_bry_ps	-0.022
a_subtlex_us_zipf_pw	-0.211	a_char_ps	-0.076	corr_intj_var	-0.013	a_n_ent_norp_ps	-0.022
rt_average	-0.214	root_part_var	-0.091	root_intj_var	-0.013	t_n_ent_time	-0.022
rt_slow	-0.214	corr_part_var	-0.091	n_intj	-0.013	simp_space_var	-0.024
t_word	-0.214	a_noun_ps	-0.096	t_n_ent_art	-0.013	n_uadp	-0.025
rt_fast	-0.214	a_kup_ps	-0.096	t_n_ent_law	-0.014	a_n_ent_norp_pw	-0.031
a_intj_ps	-0.214	simp_adv_var	-0.103	t_syll2	-0.015	a_n_ent_org_ps	-0.032
simp_aux_var	-0.214	a_bry_ps	-0.110	a_space_ps	-0.016	a_n_ent_language_pw	-0.033

Table 14: Task, dataset, and correlated features. Part 3.

Readability Assessment CLEAR		**Essay Scoring** ASAP		**Fake News Detection** LIAR		**Hate Speech Detection** SemEval-2019 Task 5	
Feature	r	Feature	r	Feature	r	Feature	r
a_space_ps	-0.236	a_n_ent_ordinal_pw	-0.112	simp_space_var	-0.016	n_adp	-0.034
a_intj_pw	-0.245	a_word_ps	-0.115	smog	-0.017	t_n_ent_language	-0.034
n_intj	-0.247	a_n_ent_ordinal_ps	-0.118	a_n_ent_art_pw	-0.019	a_n_ent_org_pw	-0.035
a_part_pw	-0.250	a_part_ps	-0.118	a_intj_pw	-0.019	a_bry_pw	-0.035
a_n_ent_person_pw	-0.257	a_cconj_pw	-0.133	a_intj_ps	-0.022	a_n_ent_language_ps	-0.035
simp_intj_var	-0.263	bilog_ttr_no_lem	-0.144	fogi	-0.026	a_propn_ps	-0.037
corr_adv_var	-0.266	bilog_ttr	-0.144	fkgl	-0.030	a_n_ent_cardinal_ps	-0.039
root_adv_var	-0.266	simp_sconj_var	-0.149	t_n_ent_org	-0.032	t_n_ent_person	-0.040
n_uintj	-0.267	a_subtlex_us_zipf_ps	-0.157	n_verb	-0.036	t_n_ent_gpe	-0.044
t_n_ent_person	-0.269	root_cconj_var	-0.158	a_n_ent_org_ps	-0.040	a_n_ent_cardinal_pw	-0.045
a_space_pw	-0.275	corr_cconj_var	-0.158	cole	-0.040	n_num	-0.047
root_intj_var	-0.278	simp_noun_var	-0.159	root_verb_var	-0.041	simp_num_var	-0.047
corr_intj_var	-0.278	a_verb_ps	-0.162	corr_verb_var	-0.041	n_unum	-0.048
n_space	-0.283	a_stopword_ps	-0.166	simp_propn_var	-0.043	corr_num_var	-0.050
n_part	-0.284	a_aux_pw	-0.176	n_uverb	-0.044	root_num_var	-0.050
n_upart	-0.286	a_cconj_ps	-0.177	n_upart	-0.046	a_propn_pw	-0.051
a_punct_pw	-0.287	a_sconj_pw	-0.186	n_part	-0.046	fogi	-0.053
a_stopword_pw	-0.288	a_aux_ps	-0.192	a_verb_ps	-0.047	fkgl	-0.055
t_punct	-0.290	a_pron_ps	-0.201	corr_part_var	-0.049	a_n_ent_person_pw	-0.058
n_uaux	-0.292	a_sconj_ps	-0.203	root_part_var	-0.049	a_char_ps	-0.061
n_punct	-0.301	simp_verb_var	-0.204	simp_part_var	-0.050	a_n_ent_ps	-0.062
corr_aux_var	-0.308	a_pron_pw	-0.209	a_n_ent_org_pw	-0.051	a_n_ent_person_ps	-0.062
root_aux_var	-0.308	a_verb_pw	-0.220	a_part_ps	-0.052	a_syll_pw	-0.066
a_pron_ps	-0.319	a_stopword_pw	-0.236	a_char_pw	-0.055	a_num_ps	-0.070
n_uadv	-0.333	a_subtlex_us_zipf_pw	-0.295	n_propn	-0.057	a_adp_ps	-0.073
t_subtlex_us_zipf	-0.334	simp_pron_var	-0.307	bilog_ttr_no_lem	-0.059	a_n_ent_date_ps	-0.074
a_adv_pw	-0.338	simp_part_var	-0.366	bilog_ttr	-0.059	a_n_ent_gpe_ps	-0.074
t_sent	-0.339	simp_aux_var	-0.399	simp_ttr	-0.059	a_num_pw	-0.080
corr_adp_var	-0.359	simp_cconj_var	-0.438	simp_ttr_no_lem	-0.059	bilog_ttr_no_lem	-0.083
root_adp_var	-0.359	simp_ttr	-0.448	a_part_pw	-0.060	bilog_ttr	-0.083
n_adv	-0.376	simp_ttr_no_lem	-0.448	n_upropn	-0.064	t_n_ent_date	-0.085
t_stopword	-0.378	simp_punct_var	-0.519	a_syll_pw	-0.071	a_n_ent_pw	-0.086
n_uverb	-0.381	simp_det_var	-0.530	root_propn_var	-0.072	a_n_ent_date_pw	-0.088
simp_adp_var	-0.462	simp_adp_var	-0.533	corr_propn_var	-0.072	a_n_ent_gpe_pw	-0.090
a_verb_pw	-0.481			a_propn_ps	-0.074	a_adp_pw	-0.096
n_verb	-0.508			a_verb_pw	-0.077	simp_ttr_no_lem	-0.122
n_upron	-0.531			t_n_ent_person	-0.079	simp_ttr	-0.122
a_pron_pw	-0.649			a_n_ent_person_ps	-0.082	auto	-0.156
n_pron	-0.653			a_n_ent_person_pw	-0.085	a_char_pw	-0.167
fkre	-0.687			a_propn_pw	-0.098	cole	-0.174

Table 15: Task, dataset, and correlated features. Part 4.