

Gradient Boosted Trees for Identification of Complex Words in Context

Raksha Agarwal, Niladri Chatterjee

Indian Institute of Technology Delhi, Hauz Khas, Delhi-110016, India

Abstract

Determining whether a particular word is complex in a given context is an important task for modern NLP, as the presence of complex words may hinder smooth communication. The present work focuses on developing a binary classifier for predicting the complexity of a target word. A set of 51 features, pertaining to eight different classes, has been identified for the said purpose. Four different classifiers have been used, and their performance is compared. CatBoost registered the best performance when tested on CWI2016 dataset, and for the News and Wikinews categories for CWI2018 dataset. In fact, the CatBoost system supersedes the top performers for the 2016 and 2018 contests for the above-mentioned cases. The optimal feature subsets for the datasets are obtained using recursive feature elimination.

Keywords

Complex Word Identification, Linguistic Features, CatBoost, Domain Adaptation

1. Introduction

Presence of difficult words in a text can lower readability and comprehension for second language learners as well as for native speakers with low literacy levels and reading difficulties [1]. This can lead to miscommunication of ideas and/or misunderstanding of contents. Automatic identification of difficult-to-understand words in a given sentence has been considered as a core part of Lexical Simplification (LS) systems by several works in the past [2, 3]. This task is commonly referred to as Complex Word Identification (CWI). Absence of CWI from LS systems, and adopting a 'Simplify Everything' [4] approach may obscure the meaning of the source sentence due to redundant substitutions of simple words.


CWI systems are categorized into four types, namely *Threshold-based*, *Lexicon-based*, *Implicit CWI* and *Machine learning-assisted* [5]. *Threshold-based* system segregate complex and simple words by setting a threshold value on a simplicity metric, such as word frequency [2, 6]. *Lexicon-based* systems make use of domain-specific lexicons for CWI to replace a complex word with a simple word/phrase with similar meaning [7]. *Implicit CWI* systems, instead of identifying complex words, focus on determining whether or not a word can be replaced by a simpler alternative [8]. CWI systems of the above types ignore the effect of context in determining the complexity of a word. *Machine learning-assisted* CWI systems are enabled to design classifiers on an extensive feature space comprising shallow features of the target word (e.g. length of the

Proceedings of the First Workshop on Current Trends in Text Simplification (CTTS 2021), co-located with SEPLN 2021. September 21st, 2021 (Online). Saggion, H., Štajner, S. and Ferrés, D. (Eds).

✉ raksha.agarwal@maths.iitd.ac.in (R. Agarwal); niladri@maths.iitd.ac.in (N. Chatterjee)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

word, POS tag, frequency in lexicon) as well as sentence level features.

In the present work, Gradient Boosted Tree classifiers are trained on a feature space comprising of hand-crafted linguistic features along with vector-based similarity features for CWI. In particular, we have considered three different boosting techniques, namely, CatBoost, XGBoost, LGBM, and compared their performance with traditional Random Forest Classifier. The experiments were conducted on the CWI2016 and CWI2018 datasets. Our experiments establish the superiority of the CatBoost algorithm over others.

The rest of the paper is organised as follows. Section 2 presents the existing related works on CWI. In Section 3, individual features of the proposed feature space are described. Details about the datasets, classification algorithms and optimal feature subset selection are presented in Section 4. Results and domain adaptation study are presented in Section 5. The paper is concluded in Section 6.

2. Related Works

CWI has attracted the attention of various researchers in the past few years with organization of two shared tasks, namely CWI 2016 and CWI 2018 [9, 10]. While CWI 2016 focused only on the English language, the CWI 2018 extended the scope of CWI 2016 by using German, French and Spanish in addition to English. In the present work, CWI has been performed only for English language.

For CWI 2016, a Performance-Oriented Soft Voting ensemble of Threshold-based, Lexicon-based, and Machine Learning-based classifiers trained on morphological, lexical and semantic features of the target word achieved the highest score [11]. Majority of the other systems used Machine Learning-assisted CWI techniques, such as SVMs [12, 13, 14], Random Forests [15, 16, 17], Decision Trees [18], and ensemble systems [19, 20]. Additionally, threshold based methods trained on word frequencies were also used [21, 22, 23].

For CWI 2018, CAMB system [24] trained on an extensive set of hand-crafted features consisting of lexical, psycholinguistic and lexicon based features achieved the highest macro F1 score. It used Adaboost and Random forest Classifiers. Some systems [25, 26, 27] used Word2vec and GloVe word embeddings along with hand-crafted features. Post CWI 2018 task SEQ [28], a BiLSTM-based sequence labelling method with GloVe word embeddings, outperformed CAMB system.

Finnimore et al.[29] trained a Logistic Regression model with a suitably chosen feature set for CWI. Their feature set contained 25 features based on the target word/MWE, sub-word level features, and sentence-level features. However, their system did not outperform CAMB. Sheang [30] presented an approach to CWI based on Convolutional Neural Networks (CNN) trained on pre-trained word embeddings with morphological and linguistic features. Ehara et al.[31] developed a graph-based method for CWI based on similarities among corpus word frequencies. Detailed analysis of the similarity and distance between the word-frequency distributions of five corpora was conducted using four different measures. Zaharia et al. [32] performed CWI using multilingual and language-specific Transformer models, multilingual word embeddings (non-Transformer), and different fine-tuning techniques. Crosslingual Zero-shot, One-shot and Few-shot transfer evaluations were also performed. Aprosio et al. [33] presented a pipeline for

personalised complex word detection adapting to the mother tongue of non-native speakers, and based on false friend identification. Their system utilized manually curated datasets of cognates and false friends for four language pairs.

3. Feature Space

For the present work we have used a feature space consisting of 51 features, classified into eight categories as described in the following subsections.

3.1. Lexical Features

Shallow features, such as Number of characters (*Nchar*), vowels (*Nvow*), phonemes (*Nphon*), syllables (*Nsyl*), morphemes (*Nmorph*), and percentage of upper case characters (*UpCase*) in the target token are used to model lexical characteristics of the token. A feature (*IsNE*) is used to indicate whether the input token is a Named Entity. The language of etymological¹ origin (e.g., French, Latin) of the target word is also considered as a feature, named *EtymOrig*. A Boolean feature (*IsStopword*) is used to indicate whether the token is a stopword. This feature has been extracted using NLTK's list of English stopwords. Both simple Universal POS tag (*UnivTag*) and detailed Penn POS tag (*PennTag*)² of the input token are also considered as features.

Number of synsets (*Nsyn*), hyponyms (*Nhyppo*) and hypernyms (*Nhyper*) of the target word as extracted from NLTK WordNet are considered as features as well. The number of characters in the words immediately preceding and succeeding the target tokens, named *NcharPrev* and *NcharNext*, respectively, are also included in the feature space. Additionally, we have also considered two sentence level features, namely the total number of tokens in the sentence (*LenSent*), and the relative position (*Relpos*) of the input token in the sentence.

3.2. Lexicon based features

Two lexicon based Boolean features, namely *InGoogle* and *InOgden* were also considered.

- a. *InGoogle* indicates whether the input token belongs to the list of 10,000 most common English words, determined by n-gram frequency in the Google's Trillion Word Corpus³.
- b. *InOgden* indicates the presence of input tokens in the list of 1000 words included in Ogden's Basic English⁴.

3.3. Frequency based features

Several frequency based features have been used to model the familiarity of the target word:

¹<https://pypi.org/project/ety/>

²*IsNE*, *UnivTag* and *PennTag* have been extracted using spaCy

³<https://github.com/first20hours/google-10000-english>

⁴<http://ogden.basic-english.org>

- a. Frequency of the word in Ogden’s Basic English (*OgdenFreq*), Exquisite Corpus (*ECFreq*) and SUBTLEX (*SUBTFreq*). Exquisite Corpus⁵ compiles texts from several domains. SUBTLEX⁶ contains frequency of 51M words calculated on a corpus of Movie Subtitles.
- b. Contextual Diversity (*ContDiverse*) reported in SUBTLEX is also used as a feature. Contextual Diversity is computed as the percentage of movies in which the target word appears.
- c. Furthermore, frequency of the input tokens given in the L count of Thorndike and Lorge [34], and London-Lund Corpus of English Conversation by Brown [35] are also used as features. These are named *TLFreq* and *BrownFreq*, respectively.

3.4. Character Language Model features

Words with unusual letter sequence may add to the complexity of the word. In order to incorporate this, the probability of the input token, calculated using bigram and trigram character language models, have been considered as features (*BiCharProb*, *TriCharProb*). These probabilities are expected to be lower for words with unusual sequence of letters. Letter counts from Google’s Trillion Word Corpus⁷ are used to calculate the letter bigram and trigram probabilities [36].

3.5. Psycholinguistic Features

The cognitive processes in the human brain is influenced by the psycholinguistic properties of a word when presented with either written or spoken forms [37]. These properties include Age of acquisition (*AOA*), Concreteness (*Conc*), Imageability (*Imag*) and Meaningfulness ratings, namely *MeanC* and *MeanP* of the target word. In the present work these features are extracted using MRC psycholinguistic database [38]. Additionally, target token’s written frequency of occurrence (*KFFreq*), and the number of categories (*KFNcats*) and number of samples (*KFNsamp*) of text in which the target word was found are also used as features [39].

3.6. Language Model Features

Statistical n-gram language models are used to study the collocation of words in sentences, and determine the probability of a sequence of words. A trigram language model⁸ trained on the Gigaword corpus [40] has been used to extract two features (*FragScore3*, *FragScore5*) which measure the language model score of a word sequence containing the target token, and the context words in the source sentence in a window of size 3 and 5, respectively [36, 41]. The above-mentioned scores of the given target word help to determine whether or not the word is used in an unusual context in the given source sentence.

⁵<https://pypi.org/project/wordfreq/>

⁶<https://github.com/Wonderlic-AI/wonderlicnlp>

⁷http://norvig.com/ngrams/count_2l.txt, http://norvig.com/ngrams/count_3l.txt

⁸http://www.keithv.com/software/giga/lm_giga_64k_nvp_3gram.zip

3.7. Dependency features

The Dependency tree of a sentence helps to understand the relationship between different words of a given sentence. In this respect, the dependency tag of the input token with its syntactical head (*DepTag*) and, POS tag of the head (*HeadPOS*) are considered as features. Additionally, two features are extracted from the dependency tree, namely depth of the input token in the tree (*TokDepth*), and the number of children of the input token (*NChild*).

3.8. Vector Similarity features

In order to incorporate some additional information about the agreement between the target word and its context the following vector similarity features are included in the feature space. GloVe 300-dimensional word embeddings [42] have been used to calculate the similarity. Other word embeddings, viz. Word2Vec[43] and FastText[44] gave inferior results in our preliminary studies.

- a. The cosine similarity of the target token with the root token and syntactical head in the dependency tree are also taken as features (*RootSim*, *HeadSim*).
- b. The average similarity of the target token with its siblings and children in the dependency tree are also considered (*AvgChildSim*, *AvgSibSim*). Maximum similarity of the target token with its siblings is also considered (*MaxSibSim*).
- c. To further measure the compatibility of the target token with its context, average similarity with k words to its immediate left and right are extracted for $k = 1$ and $k = 5$. These features are named as *LeftSim5*, *RightSim5*, *LeftSim1* and *RightSim1*, respectively.

4. Experimental Details

This section presents the details of datasets, classification algorithms, and the feature selection approach.

4.1. Datasets

In this section the details about the datasets of CWI 2016 and CWI 2018 [9, 10] are presented. CWI 2016 dataset contained 9200 sentences. The target words were manually annotated by 400 non-native English speakers as complex or non-complex. CWI 2018 used the sentences from three different text genres, namely News (professionally written news), WikiNews (news written by amateurs), and articles from Wikipedia [10]. Here, words and phrases of length up to 10 words were annotated by 183 annotators comprising both native and non-native English speakers. Along with the binary complex vs. non-complex label it also contains a probabilistic label representing the proportion of annotators that labelled the item as complex. In this work we have focused on the binary classification only. The data statistics is presented in Table 1. For CWI 2016, the systems were evaluated using a new metric G-Score, which is the harmonic mean of Accuracy and Recall. We have also reported the Accuracy, Precision, Recall and F1-Score. For CWI 2018, macro-average F1-score has been used for evaluation. The above metrics have been chosen as per the instructions of CWI 2016 [9] and CWI 2018 [10].

Table 1
Data Statistics

	CWI2016		CWI2018				
	Words	News		WikiNews		Wikipedia	
		Words	Phrases	Words	Phrases	Words	Phrases
Train	2237	13451	2315	7556	1060	5435	810
Test	88221	1813	282	1138	149	750	120

4.2. Classification Algorithms

In the present work, we have experimented with the following classification algorithms:

- Random Forest (RF): RF Classifiers train a multitude of decision trees on various sub-samples of the dataset, and use averaging to improve accuracy and control over-fitting [45].
- XGBoost : It turns weakly learned decision trees into strong learners by training upon residuals instead of aggregation [46].
- Light Gradient Boosting Machine (LGBM) : This is a histogram-based boosting algorithm. Here, boosting is performed using a specialised gradient-based one-sided sampling of data points of large gradients [47].
- CatBoost: This method makes better utilisation of the categorical features which are otherwise converted to numerical features in traditional gradient boosting [48]. Here, Oblivious trees are used as base predictors. These trees use the same splitting criterion across the entire level of the tree, making it less prone to overfitting.

4.3. Feature Selection

Recursive feature elimination is performed for maximizing the mean 5-fold cross-validation F1-score. We observed that the same set of features did not perform equally well for the two datasets. For CWI 2016, all features except *Nhypo*, *KFNcats*, *NChild* and *AOA*; and for CWI 2018, all features except *SUBTFreq* and *HeadSim* are included in the optimal feature subset.

5. Results and Analysis

The results for CWI 2016 dataset corresponding to the different classification algorithms mentioned in Section 4.2 are mentioned in Table 2. The proposed CWI approach has been compared with state-of-the-art and other top performing systems on the two datasets. It can be observed that CatBoost classifiers outperform other algorithms as well as other existing baselines in terms of Accuracy, Precision, F1-Score, and G-Score. Since many features in the proposed feature space are categorical in nature, the superior performance of CatBoost may be attributed to its effective Ordered Target Encoding [48] for categorical feature preprocessing. For CWI 2018, unlike existing works [24, 27] a unified system is trained for all the three sub-datasets. Initially

Table 2
Results for CWI 2016

System	Classifier	Accuracy	Precision	Recall	F1-Score	G-Score
	CatBoost	0.842	0.189	0.718	0.299	0.775
	XGBoost	0.841	0.185	0.703	0.292	0.766
	RF	0.840	0.185	0.704	0.293	0.766
	LGBM	0.820	0.167	0.713	0.270	0.763
SV000gg [11]	Voting	0.779	0.147	0.769	0.246	0.774
TALN [16]	RF	0.812	0.164	0.736	0.268	0.772
UWB [21]	MaxEntropy	0.803	0.157	0.734	0.258	0.767
PLUJAGH [22]	Threshold	0.795	0.152	0.741	0.252	0.767

Table 3
Results for CWI 2018 for Words

System	Classifier	News		WikiNews		Wikipedia	
		Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
	CatBoost	0.9029	0.8864	0.8822	0.8721	0.8067	0.8025
	XGBoost	0.8758	0.8556	0.8524	0.8377	0.7840	0.7796
	RF	0.8880	0.8703	0.8612	0.8491	0.8053	0.8000
	LGBM	0.8897	0.8712	0.8612	0.8492	0.8093	0.8063
SEQ [28]	BiLSTM	0.8897	0.8763	0.8612	0.8540	0.8147	0.8140
CAMB [24]	RF	0.8902	0.8633	0.8524	0.8317	0.8107	0.7780

all samples corresponding to phrases are discarded from the train and test set; and the classifiers are trained only on single words.

The results for CWI 2018 dataset for single word targets are presented in Table 3. CatBoost classifiers achieved the highest score in terms of Accuracy and Macro-F1 for the News and WikiNews. For Wikipedia, a low score is achieved which may be due to sub-optimality of the feature space corresponding to this data split.

To analyze the effect of individual feature subsets, the corresponding subset of features is removed from the feature space. CatBoost classifiers are trained on the reduced feature space and results are reported in Table 4. All the feature subsets contribute to increasing the overall performance of the system. While removing Lexical features led to the largest decline in scores, contextual features corresponding to Language Model, Dependency Trees and Vector Similarity also emerged as important features.

5.1. Complex Phrase Identification

CWI 2018 dataset contains phrases along with single words. In order to predict the complexity of target phrases two approaches have been used. In the first approach, all phrases are marked as complex. In the second approach, if the mean predicted complexity of individual component words is above a threshold then the phrase is marked as complex. In subsequent discussions

Table 4
Features Subset Importance

Feature Space	CWI2016		CWI2018	
	Accuracy	G-score	Accuracy	Macro-F1
All	0.8422	0.7751	0.8770	0.8647
w/o Lexical	0.8389	0.7635	0.8592	0.8454
w/o Lexicon	0.8437	0.7710	0.8735	0.8611
w/o Frequency	0.8411	0.7594	0.8692	0.8564
w/o Psycholinguistic	0.8403	0.7705	0.8719	0.8594
w/o Character LM	0.8388	0.7735	0.8725	0.8600
w/o Language Model	0.8374	0.7717	0.8711	0.8585
w/o Dependency	0.8408	0.7726	0.8703	0.8576
w/o Vector Similarity	0.8412	0.7700	0.8701	0.8572

these are referred as Greedy and Threshold, respectively. Individual word complexities are derived using the predictions obtained by CatBoost classifiers. The Accuracy and Macro-F1 for complexity of phrases corresponding to 10 equally spaced threshold values between 0 to 1 is presented in Figure 1. The threshold is chosen to be 0.5 because macro-F1 is maximized at this value. It can be noted that the threshold value 0 corresponding to maximum accuracy is equivalent to the Greedy approach. The results for the entire CWI 2018 dataset including both words and phrases are presented in Table 5. Greedy approach achieves the best scores for News and WikiNews. The scores for Threshold approach is low because a huge proportion (about 80%) of phrases for both training and test set are complex.

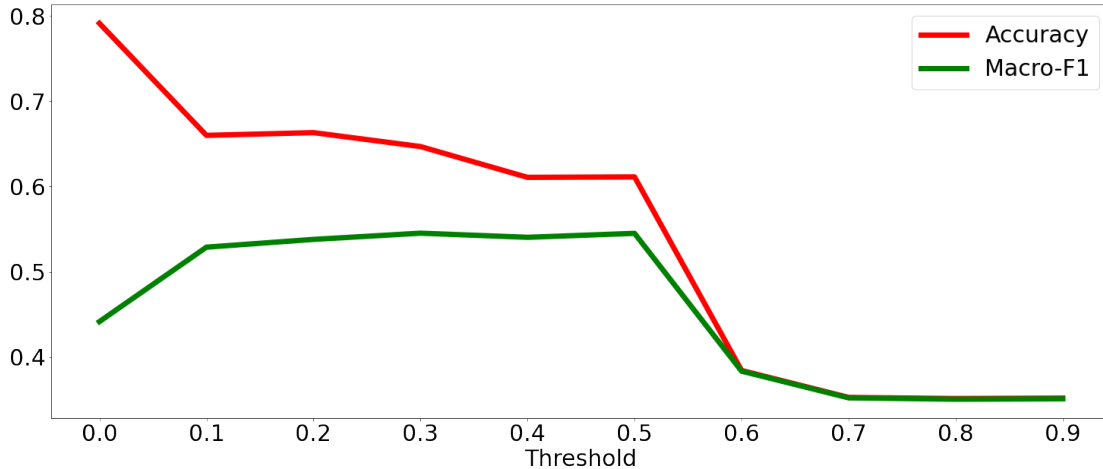


Figure 1: Accuracy and Macro-F1 for Phrase Complexity in the Training Set

McNemar’s test [49, 50] has been used to compare the performance of the CatBoost with well-known baselines, SEQ and CAMB systems, for CWI 2018 dataset. For each system we have constructed 2×2 contingency tables for both Words and Word+Phrases. Figures 2a, 2b show

Table 5
Results for CWI 2018 for Words+Phrases

System		News		WikiNews		Wikipedia	
		Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
	Greedy	0.8902	0.8846	0.8695	0.8670	0.8080	0.8080
	Threshold (0.5)	0.8640	0.8524	0.8547	0.8485	0.7805	0.7798
SEQ [28]	BiLSTM	0.8811	0.8763	0.8524	0.8505	0.8161	0.8158
CAMB [24]	RF	0.8792	0.8736	0.8430	0.8400	0.8115	0.8115
NLP-CIC [27]		0.863	0.8551	0.837	0.8308	0.774	0.7722
Sheang [30]	CNN	-	0.8679	-	0.8386	-	0.8011
Zaharia [32]	XLM-RoBERTa	-	0.808	-	0.811	-	0.808

the values for CatBoost v/s SEQ, while Figures 2c, 2d show the values for CatBoost v/s CAMB. According to the test, the null hypothesis (equal performance of the systems) is rejected with 99% confidence for both Words and Words+Phrases for CAMB system; and it is rejected with 95% and 90% confidence for Words and Words+Phrases, respectively for SEQ system.

		SEQ	
		correct	incorrect
CatBoost	correct	a=3043	b=203
	incorrect	c=164	d=291

(a) For Words ($\chi^2 = 3.935, p = 0.047$)

		SEQ	
		correct	incorrect
CatBoost	correct	a=3484	b=203
	incorrect	c=166	d=399

(b) For Words+Phrases ($\chi^2 = 3.51, p = 0.061$)

		CAMB	
		correct	incorrect
CatBoost	correct	a=3038	b=208
	incorrect	c=109	d=346

(c) For Words ($\chi^2 = 30.29, p < 0.00001$)

		CAMB	
		correct	incorrect
CatBoost	correct	a=3479	b=208
	incorrect	c=109	d=456

(d) For Words+Phrases ($\chi^2 = 30.3, p < 0.00001$)

Figure 2: Contingency Table for CWI 2018

5.2. Domain Adaptation Study

In this section we study the importance of domain specific data for training CWI systems. Here, the classifiers are trained on a Source dataset to predict the complexity for Target test dataset. A fraction, denoted by Tgt , of training samples from the Target dataset along with Source training data is included in the training set. $Tgt=0$ corresponds to the case when the training data contains no samples from the Target dataset. The features used for training the classifier for each Source dataset is as described in Section 4.3. Figure 3 depicts the Accuracy and Macro-F1 scores corresponding to different classifiers and Tgt values. As expected, the highest scores for each of the classifiers are obtained for $Tgt=100\%$ i.e. when the the entire training data of the Target dataset is used for training. However, it can be observed that for CWI 2018 (See

Figure 3a), including just 10% (about 2000 samples) training data improved the performance significantly. In fact, the performance is only marginally increased for $Tgt > 50\%$ (about 10,000 samples). Similarly, for CWI 2016 (See Figure 3b), across all the classifiers the performance plateaued after $Tgt > 50\%$ (about 1000 samples). This indicates that classifiers trained on the proposed feature space are well applicable to unseen data.

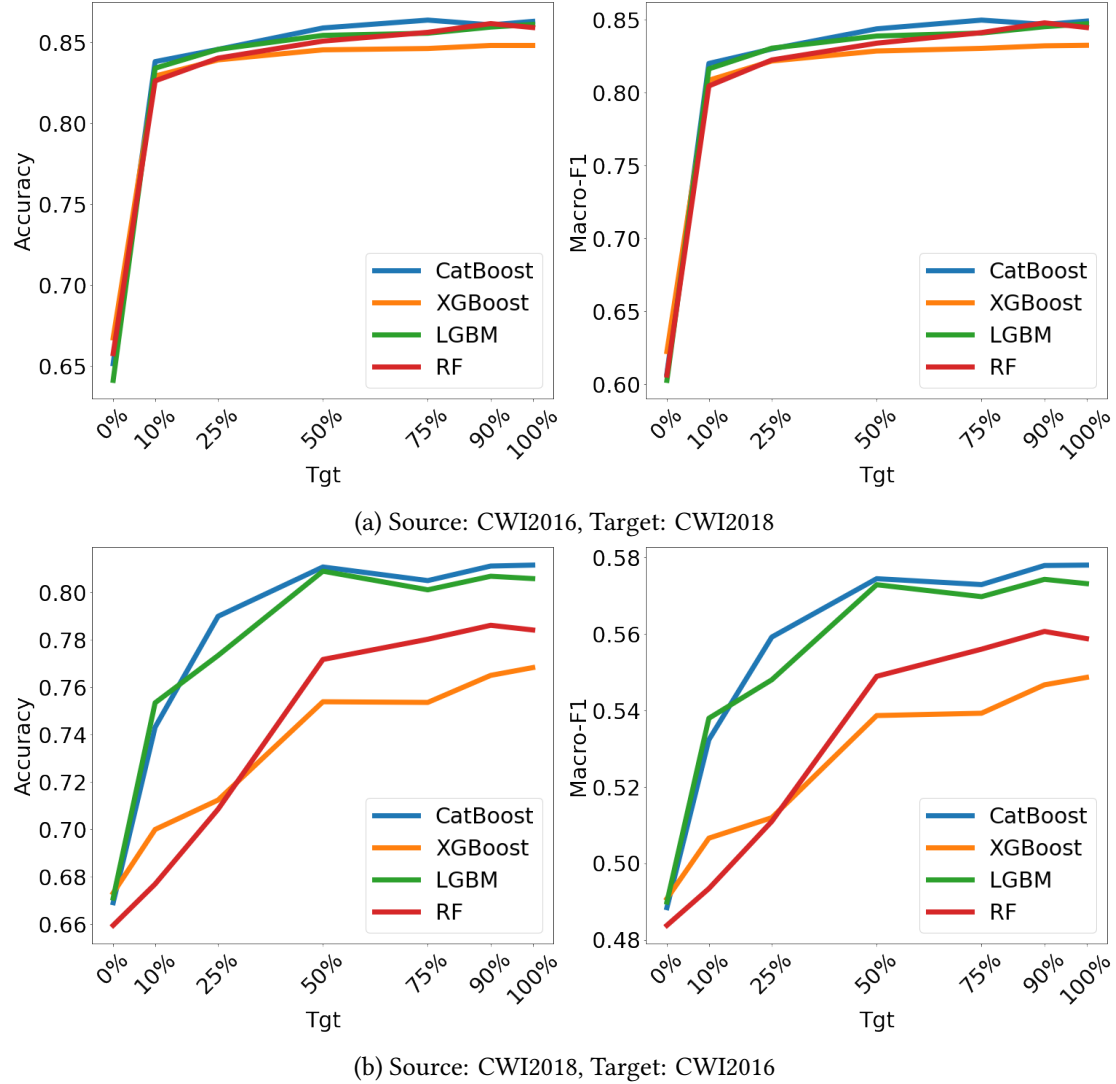


Figure 3: Domain Adaptation Results

6. Conclusion

The present work aims at training and testing with different Gradient Boosted Tree Classifiers for determining whether a given word of a sentence is complex in its context. Four classifiers,

namely CatBoost, XGBoost, LGBM and Random Forest, are trained on a feature space comprising lexical, semantic and frequency based features. The context of the target word is incorporated in the feature space using features derived from Language Model, Dependency Parse Trees and Cosine similarity of the target with its context. Experiments on two datasets, namely CWI2016 and CWI2018 indicate that the classifier trained on the proposed feature space using CatBoost algorithm outperforms known baseline works. Domain adaptation has also been studied between the two datasets to determine the generalizability of the proposed CWI system. It was observed that even with 50% reduction in domain-specific training data, the performance is not degraded significantly. This observation is very encouraging as it extends the applicability of the above techniques to unseen data belonging to a variety of domains.

One major takeaway from the present work is that although it is based on Tree classifiers and hand-crafted feature space, the gradient boosting (CatBoost) system outperforms other techniques based on Deep Neural Networks. In future, we would also like to assess the performance of the proposed feature space with other machine learning schemes, such as LSTM, BiLSTM, CNN.

Acknowledgments

Raksha Agarwal acknowledges Council of Scientific and Industrial Research (CSIR), India for supporting the research under Grant no: SPM-06/086(0267)/2018-EMR-I.

References

- [1] M. Zampieri, S. Malmasi, G. Paetzold, L. Specia, Complex word identification: Challenges in data annotation and system performance, arXiv preprint arXiv:1710.04989 (2017).
- [2] M. Shardlow, A comparison of techniques to automatically identify complex words., in: 51st Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop, 2013, pp. 103–109.
- [3] G. H. Paetzold, L. Specia, Plumberr: An automatic error identification framework for lexical simplification, in: Proceedings of the first international workshop on Quality Assessment for Text Simplification (QATS), 2016, pp. 1–9.
- [4] M. Shardlow, A survey of automated text simplification, International Journal of Advanced Computer Science and Applications 4 (2014) 58–70.
- [5] P. Sikka, V. Mago, A survey on text simplification, ArXiv abs/2008.08612 (2020).
- [6] G. Leroy, J. E. Endicott, D. Kauchak, O. Mouradi, M. Just, User evaluation of the effects of a text simplification algorithm using term familiarity on perception, understanding, learning, and information retention, Journal of medical Internet research 15 (2013) e144.
- [7] N. Elhadad, K. Sutaria, Mining a lexicon of technical terms and lay equivalents, in: Biological, translational, and clinical language processing, Association for Computational Linguistics, Prague, Czech Republic, 2007, pp. 49–56. URL: <https://aclanthology.org/W07-1007>.
- [8] C. Horn, C. Manduca, D. Kauchak, Learning a lexical simplifier using wikipedia, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2014, pp. 458–463.

- [9] G. Paetzold, L. Specia, Semeval 2016 task 11: Complex word identification, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), 2016, pp. 560–569.
- [10] S. M. Yimam, C. Biemann, S. Malmasi, G. Paetzold, L. Specia, S. Štajner, A. Tack, M. Zampieri, A report on the complex word identification shared task 2018, in: Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications, Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 66–78. URL: <https://www.aclweb.org/anthology/W18-0507>. doi:10.18653/v1/W18-0507.
- [11] G. Paetzold, L. Specia, Sv000gg at semeval-2016 task 11: Heavy gauge complex word identification with system voting, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), 2016, pp. 969–974.
- [12] S. Sanjay, A. Kumar M, K. P. Soman, AmritaCEN at SemEval-2016 task 11: Complex word identification using word embedding, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), Association for Computational Linguistics, San Diego, California, 2016, pp. 1022–1027. URL: <https://aclanthology.org/S16-1159>. doi:10.18653/v1/S16-1159.
- [13] O. Kuru, Ai-ku at semeval-2016 task 11: Word embeddings and substring features for complex word identification, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), 2016, pp. 1042–1046.
- [14] P. K. Choubey, S. Pateria, Garuda & bhasha at semeval-2016 task 11: Complex word identification using aggregated learning models, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), 2016, pp. 1006–1010.
- [15] J. Brooke, A. L. Uitdenbogerd, T. Baldwin, Melbourne at semeval 2016 task 11: Classifying type-level word complexity using random forests with corpus and word list features, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), 2016, pp. 975–981.
- [16] F. Ronzano, A. Abura'ed, L. Espinosa-Anke, H. Saggion, TALN at SemEval-2016 task 11: Modelling complex words by contextual, lexical and semantic features, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), Association for Computational Linguistics, San Diego, California, 2016, pp. 1011–1016. URL: <https://aclanthology.org/S16-1157>. doi:10.18653/v1/S16-1157.
- [17] E. Davoodi, L. Kosseim, Clac at semeval-2016 task 11: Exploring linguistic and psycholinguistic features for complex word identification, arXiv preprint arXiv:1709.02843 (2017).
- [18] M. Quijada, J. Medero, Hmc at semeval-2016 task 11: Identifying complex words using depth-limited decision trees, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), 2016, pp. 1034–1037.
- [19] S. Malmasi, M. Zampieri, Maza at semeval-2016 task 11: Detecting lexical complexity using a decision stump meta-classifier, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), 2016, pp. 991–995.
- [20] G. Nat, Sensible at semeval-2016 task 11: Neural nonsense mangled in ensemble mess, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), 2016, pp. 963–968.
- [21] M. Konkol, Uwb at semeval-2016 task 11: Exploring features for complex word identification, in: Proceedings of the 10th International Workshop on Semantic Evaluation

- (SemEval-2016), 2016, pp. 1038–1041.
- [22] K. Wróbel, Plujagh at semeval-2016 task 11: Simple system for complex word identification, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), 2016, pp. 953–957.
- [23] D. Kauchak, Pomona at semeval-2016 task 11: Predicting word complexity based on corpus frequency, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), 2016, pp. 1047–1051.
- [24] S. Gooding, E. Kochmar, Camb at cwi shared task 2018: Complex word identification with ensemble-based voting, in: Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications, 2018, pp. 184–194.
- [25] A. AbuRa’ed, H. Saggion, LaSTUS/TALN at complex word identification (CWI) 2018 shared task, in: Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications, Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 159–165. URL: <https://www.aclweb.org/anthology/W18-0517>. doi:10.18653/v1/W18-0517.
- [26] D. De Hertog, A. Tack, Deep learning architecture for complexword identification, in: Thirteenth Workshop of Innovative Use of NLP for Building Educational Applications, Association for Computational Linguistics (ACL); New Orleans, Louisiana, 2018, pp. 328–334.
- [27] S. T. Aroyehun, J. Angel, D. A. P. Alvarez, A. Gelbukh, Complex word identification: Convolutional neural network vs. feature engineering, in: Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications, 2018, pp. 322–327.
- [28] S. Gooding, E. Kochmar, Complex word identification as a sequence labelling task, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 1148–1153.
- [29] P. Finnimore, E. Fritsch, D. King, A. Sneyd, A. U. Rehman, F. Alva-Manchego, A. Vlachos, Strong baselines for complex word identification across multiple languages, arXiv preprint arXiv:1904.05953 (2019).
- [30] K. C. Sheang, Multilingual complex word identification: Convolutional neural networks with morphological and linguistic features, in: Proceedings of the Student Research Workshop Associated with RANLP 2019, 2019, pp. 83–89.
- [31] Y. Ehara, Graph-based analysis of similarities between word frequency distributions of various corpora for complex word identification, in: 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), IEEE, 2019, pp. 1982–1986.
- [32] G.-E. Zaharia, D.-C. Cercel, M. Dascalu, Cross-lingual transfer learning for complex word identification, in: 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI), IEEE, 2020, pp. 384–390.
- [33] A. Palmero Aprosio, S. Menini, S. Tonelli, Adaptive complex word identification through false friend detection, in: Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization, 2020, pp. 192–200.
- [34] E. L. Thorndike, I. Lorge, The teacher’s word book of 30,000 words. (1944).
- [35] G. D. Brown, A frequency count of 190,000 words in the london-lund corpus of english conversation, Behavior Research Methods, Instruments, & Computers 16 (1984) 502–532.
- [36] R. Agarwal, N. Chatterjee, LangResearchLab_NC at CMCL2021 shared task: Predicting gaze behaviour using linguistic features and tree regressors, in: Proceedings of the Workshop

- on Cognitive Modeling and Computational Linguistics, Association for Computational Linguistics, Online, 2021, pp. 79–84. URL: <https://www.aclweb.org/anthology/2021.cmcl-1.8>. doi:10.18653/v1/2021.cmcl-1.8.
- [37] G. Paetzold, L. Specia, Inferring psycholinguistic properties of words, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, San Diego, California, 2016, pp. 435–440. URL: <https://www.aclweb.org/anthology/N16-1050>.
- [38] M. Wilson, Mrc psycholinguistic database: Machine-usable dictionary, version 2.00, Behavior research methods, instruments, & computers 20 (1988) 6–10.
- [39] H. Kučera, W. N. Francis, Computational analysis of present-day American English, University Press of New England, 1967.
- [40] D. Graff, J. Kong, K. Chen, K. Maeda, English gigaword, Linguistic Data Consortium, Philadelphia 4 (2003) 34.
- [41] R. Agarwal, N. Chatterjee, LangResearchLab NC at SemEval-2021 task 1: Linguistic feature based modelling for lexical complexity, in: Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), Association for Computational Linguistics, Online, 2021, pp. 120–125. URL: <https://aclanthology.org/2021.semeval-1.10>. doi:10.18653/v1/2021.semeval-1.10.
- [42] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543. URL: <http://www.aclweb.org/anthology/D14-1162>.
- [43] T. Mikolov, E. Grave, P. Bojanowski, C. Puhresch, A. Joulin, Advances in pre-training distributed word representations, in: Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018), 2018.
- [44] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, Transactions of the Association for Computational Linguistics 5 (2017) 135–146.
- [45] T. K. Ho, The random subspace method for constructing decision forests, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (1998) 832–844. doi:10.1109/34.709601.
- [46] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, Association for Computing Machinery, New York, NY, USA, 2016, p. 785–794. URL: <https://doi.org/10.1145/2939672.2939785>. doi:10.1145/2939672.2939785.
- [47] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, Lightgbm: A highly efficient gradient boosting decision tree, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017. URL: <https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf>.
- [48] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, A. Gulin, Catboost: unbiased boosting with categorical features, in: S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 31, Curran Associates, Inc., 2018. URL: <https://proceedings.neurips.cc/paper/2018/file/14491b756b3a51daac41c24863285549-Paper.pdf>.
- [49] Q. McNemar, Note on the sampling error of the difference between correlated proportions

or percentages, *Psychometrika* 12 (1947) 153–157.

- [50] A. L. Edwards, Note on the “correction for continuity” in testing the significance of the difference between correlated proportions, *Psychometrika* 13 (1948) 185–187.

A. Feature Space

A List of all the features included in the feature space is presented in Table A.1

Table A.1
List of all Features

Subset	Features	Count
	<i>Nchar, Nvow, Nphon, Nsyl, Nmorph, UpCase, NcharPrev, NcharNext</i>	
	<i>IsNE, EtymOrig, IsStopword</i>	
Lexical	<i>UnivTag, PennTag, Nsyn, Nhypho, Nhyper</i>	18
	<i>LenSent, Relpos</i>	
Lexicon	<i>InGoogle, InOgden</i>	2
Frequency	<i>OgdenFreq, ECFreq, SUBTFreq, ContDiverse, TLFreq, BrownFreq</i>	6
Psycholinguistic	<i>AOA, Conc, Imag, Meanc, Meanp, KFreq, KFncats, KFnsamp</i>	8
Character LM	<i>BiCharProb, TriCharProb</i>	2
Language Model	<i>FragScore3, FragScore5</i>	2
Dependency	<i>DepTag, HeadPOS, TokDepth, NChild</i>	4
	<i>RootSim, HeadSim, AvgChildSim, AvgSibSim, MaxSibSim</i>	
Vector Similarity	<i>LeftSim5, RightSim5, LeftSim1, RightSim1</i>	9

B. Feature Importance

For CatBoost the top 5 features corresponding to feature importance based on loss function change are mentioned in Table B.1. For each feature this value represents the difference between the loss value of the model with this feature and without it. For both the datasets, the feature importance is positive for all the features included in the feature space.

Table B.1
Top 5 Features

CW12016		CW12018	
Feature	Importance	Feature	Importance
<i>NChar</i>	4.08	<i>IsNE</i>	7.64
<i>Nphon</i>	3.79	<i>PerUP</i>	4.94
<i>BrownFreq</i>	3.73	<i>ECFreq</i>	4.86
<i>EtymOrig</i>	3.30	<i>Nvow</i>	4.46
<i>ECFreq</i>	3.24	<i>ContDiverse</i>	3.75

C. Domain Adaptation

The Macro-F1 and Accuracy values for different classifiers are mentioned in Table C.1.

Table C.1
Domain Adaptation

Source	Tgt	CatBoost		XGBoost		LGBM		RF	
		Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
Target: CWI2018									
CWI 2016	0%	0.6514	0.6073	0.6674	0.6224	0.6412	0.6026	0.6577	0.6062
	10%	0.8382	0.8200	0.8292	0.8085	0.8341	0.8163	0.8263	0.8046
	25%	0.8457	0.8300	0.8392	0.8217	0.8457	0.8305	0.8403	0.8224
	50%	0.8590	0.8438	0.8454	0.8386	0.8544	0.8389	0.8509	0.8339
	90%	0.8603	0.8468	0.8481	0.8321	0.8595	0.8452	0.8617	0.8479
	100%	0.8630	0.8491	0.8481	0.8325	0.8611	0.8472	0.8592	0.8447
Target: CWI2016									
CWI 2018	0%	0.6690	0.4884	0.6731	0.4908	0.6710	0.4899	0.6594	0.4837
	10%	0.7432	0.5323	0.6999	0.5066	0.7534	0.5379	0.6769	0.4934
	25%	0.7900	0.5592	0.7124	0.5119	0.7734	0.5480	0.7085	0.5110
	50%	0.8108	0.5745	0.7539	0.5386	0.8091	0.5729	0.7718	0.5489
	90%	0.8112	0.5779	0.7650	0.5467	0.8069	0.5743	0.7862	0.5607
	100%	0.8116	0.5780	0.7683	0.5486	0.8059	0.5732	0.7842	0.5588