Garuda & Bhasha at SemEval-2016 Task 11: Complex Word Identification Using Aggregated Learning Models

Prafulla Kumar Choubey and **Shubham Pateria**
System Software
Samsung R & D Institute India Bangalore Pvt Ltd
Bangalore, Karanataka, India -560037
prafulla.ch, s.pateria@samsung.com

Abstract

This paper describes aggregated learning models for Complex Word Identification (CWI) task in SemEval 2016. The work focused on selecting the features that determine complexity of words and used different combinations of support vector machine (SVM) and decision tree (DT) techniques for classification. These classifiers were pipelined with pre-processing and post-processing blocks which helped improving accuracy of systems, though had little impact on recall. Four systems were evaluated on the test set; SVM and DT systems by team Bhasha achieved G score of 0.529 and 0.508 respectively and SVM&DT and SVMPP systems by team Garuda achieved G scores of 0.360 and 0.546 respectively.

1 Introduction

CWI constitutes the first stage in lexical simplification (LS) pipeline (Specia et al., 2012). Performance of any LS system highly relies on accurate identification of complex words. (Shardlow, 2014) categorized errors in LS into six types and two of them (type 2A and 2B) are reflections of recall and precision of CWI system. The author concluded that type 2B errors occur with highest frequency, nearly 0.54, followed by type 2A (0.11). Also only 0.1 of simplifications were free from error. Thus, designing a robust CWI system impacts LS task the most.

CWI has long been researched and used in LS systems. (Paetzold, 2015) named some of the approaches used inclusive to LS task, like lexicon based approaches, frequency thresholding based approaches, word length thresholding and user driven approaches. Trained classifiers have also been proposed in literature like, (Shardlow, 2013) used SVM as classifier on feature sets comprising of frequency, length, synonym counts etc.

This work is extension to the method proposed by (Shardlow, 2013). The proposed systems are built on broader set of features and multiple classifiers have been used individually or in aggregation to know trade-off between precision and recall. These systems are trained solely on the training data released by task organizers and no external resource has been used, except the WordNet (Miller, 1995; Fellbaum, 1998).

The rest of paper is organized as follows, section 2 describes the feature sets used. Section 3 describes the approaches used and submitted systems performance w.r.t other participants. Finally, section 4 presents the conclusions and outlines some directions of future work.

2 Feature Sets

Selection of relevant feature is key to better performance of a machine learning model. While in some tasks, feature extraction is an obvious process, in tasks like CWI its hard to determine features aptness. So, in this work all the feature sets whose inclusion improves any of recall, precision or accuracy were used. These features have been described below.

Proceedings of SemEval-2016, pages 931–935,
San Diego, California, June 16-17, 2016.

2.1 Characters n-grams Frequency

These features are based on an assumption that character sequences that are frequently used should be easier to be recognized compared to rarely used sequences. For instance, if it is required to choose between Quixotic and Idealistic, idealistic would be easier to remember. Probably, because ideal, deal, idea are very commonly observed sequences, while quix is a rarely used character sequence in English dictionary. Given infinite possible character sequences, this work limited the sequences to 300 most frequently used uni, bi and tri-grams of characters.

2.2 Word's Features

In most of the LS tasks, word frequency thresholding has been used to determine its complexity (Brysbaert and New, 2009). In our systems, we have also included frequencies of words at positions [-2,-1,1,2] relative to the word, which is accounted only to observed higher accuracy.

We have also used Parts-of-Speech tag of target word and the words at position [-2,-1,1,2] relative to it as features. Brown corpus (Francis and Kucera, 1979) tagged with pos-tags used in the Penn treebank (Marcus et al., 1993) project was used to calculate trigram and emission probabilities for words and their parts of speech. Using Viterbi algorithm (Jurafsky and Martin, 2000) and probabilities calculated above, training and test sentences were tagged. Inclusion of pos-tags as features are again accounted to higher accuracy observed on validation set. The pos-tags are quantized as their occurrence frequency in combined training and testing datasets, table 1 contains the values.

Other word level features used are- word length (Keskisärkkä, 2012), words position, ratio of vowels and consonants, words stem length and frequency of stemmed words.

2.3 WordNet

The objective of LS is to find an alternative to complex word which is supposedly more common to target user. The words which have simpler synonyms available are more likely to be recognized as complex. For instance, succumb is likely to be complex provided yield is used more frequently. To introduce this into our system, we used WordNet to find synonyms of word and used word frequency relative to its SynSets as an input feature. Like in case of succumb, yield, succumb and yield have frequency counts of 4 and 0 respectively. So the input feature will be 0 [0 / (0+4)] for succumb and 1 [4 / (0+4)] for yield.

Pos-tags	Values	Pos-tags	Values
{PRP$}	0.000645	{WP}	0.000227
{VBG}	0.039401	{VBZ}	0.001063
{VBD}	0.005699	{DT}	0.001253
{VBN}	0.047874	{NN}	0.537558
{VBP}	0.000189	{FW}	0.000113
{WDT}	0. 000152	{TO}	0.000008
{JJ}	0.075610	{PRP}	0.000873
{RB}	0.024962	{Other}	0.0

Table 1: Pos-tags and the corresponding numerical values used in Classifiers

3 Proposed Systems

We are proposing three stage pipelined systems for CWI task, comprising of pre-processing, classification by one or ensemble of trained models, and post-processing stages. These stages are described below.

3.1 Pre-processing

This stage comprises of four preliminary checks that directly classify words based on rules.

1. All the named entities are classified as non-complex.
2. The words which dont belong to English are classified as complex. We have used words available in various corpora provided by nltk package and English word list (Lawler, 1999) for this.
3. Words with length less than 2 are classified as non-complex.
4. Words with pos-tags CD, DT and TO as per UPenn tag-sets are classified as non-complex.

3.2 Trained Classifiers

This stage consists of a trained classifier based on features described in section 2. We submitted four systems with classifiers trained using different classification algorithms. Systems submitted

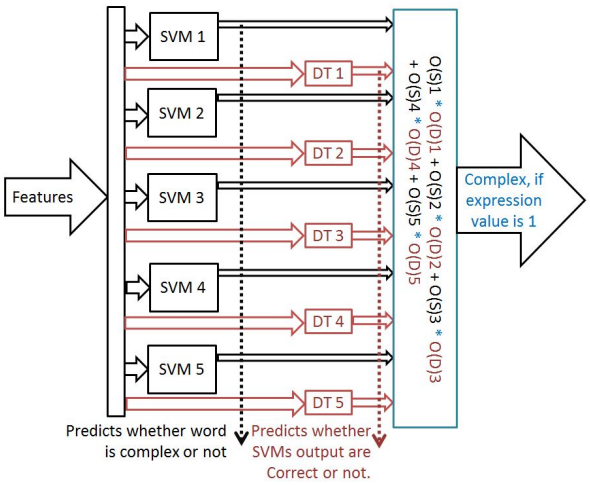

Figure 1: HSVM&DT Architecture for CWI.

by team Bhasha were based on support vector machines (SVM) and decision tree (DT). We used standard classifiers available in scikit-learn package (Pedregosa et al., 2011). Systems submitted by Garuda team were based on hybrid of classifiers. They are described below.

HSVM&DT: This is a hybrid model, inspired by bagging ensembles, comprising of multiple SVMs and DT for classifying words. We have used 5 SVMs with different training parameters (gamma, kernel and C values) and 5 different DT classifiers. SVMs perform classification of words as complex or non-complex, while DTs classify the predictions of SVM as correct or incorrect. Figure 1 explains the architecture.

A pair of SVM and DT model works together to classify the words. For each pair, training data is divided into two sets; first set is used to train the SVM and on second set, trained SVM is used to predict the output. In second set, if predicted output of SVM is same as the actual word label, we define target output as 1 for DT classifier and other wise 0. With this new target output and input features same as SVM, DT model is trained. Five such pairs were trained, each trained on randomly sampled training data. Random sampling allows learning more generalized. This algorithm is described in figure 2. One clear distinction of this aggregated model from conventional bagging is in terms of combining methodology. Output of only those SVM classifiers are used for which corresponding DT predicts 1.

SVMPP: This model is built on CWI_training_allannotations data. Twenty separate SVMs were trained on each annotation available in training data. It was designed with a belief that this system would achieve higher recall, though precision may be substantially low. In training data, a word was defined as complex, if any of the annotation was complex. However in our design, we have calculated coefficients for all 20 classifiers that define the contribution of each SVM classifier in final output. Also, output label for this model was modified as {-1,1} instead of {0, 1}. Detailed algorithm is described in figure 3.

HSVM&DT Algorithm

Given: Training Data S
for i = 1 to 5:
  Divide data in two sets S_λ, S_μ in ratio 3:2
  Train an SVM Classifier on S_λ
  Predict class of $X_\mu \in S_\mu$ by trained SVM: Y'_μ
  Generate training data for DT classifier:
    if Y'_μ == Y_μ:
      $Y_{\mu,DT}$=1
    else:
      $Y_{\mu,DT}$=0
  Train DT Classifier with X_μ,$Y_{\mu,DT} \in S_\mu$
Output Hypothesis: $0 \iff \Sigma O_{S[i]} * O_{D[i]} = 0$
1, otherwise.

Figure 2: HSVM&DT Algorithm.

SVMPP Algorithm

Given: Training Data S_i, i=1..20 #20 annotators
Output Pre-processing: Y'= -1, if Y=0
Y'= 1, if Y=1
for i = 1 to 20:
  Train SVM Classifier SVC_i, on S_i.
  Calculate λ_i & μ_i coefficients for classifier:
  λ_i = T.P./(T.P. + F.P.) μ_i = T.N./(T.N. + F.N.)
  T.P.→ True Positive F.P.→ False Positive
  T.N.→ True Negative F.N.→ False Negative
Output Hypothesis:
  $0 \iff \Sigma$ f * $O_{SVC_i} < 0$;
  1, otherwise.
  where, f = λ_i, when O_{SVC_i} == 1
  f = μ_i, when O_{SVC_i} == -1

Figure 3: SVMPP Algorithm.

As described in Figure 3, each SVM classifier has two associated coefficients $\{\lambda, \mu\}$. λ defines the precision of classifier in predicting non-complex words and μ defines precision for identifying complex words.

3.3 Post-Processing

This stage comprises of final check to remove highly probable miss-classifications.

1. Character bigrams (402 character bigrams were present in training set) which were not observed in training data but is present in test data and has occurrence frequency less than 0.00005 are classified as complex (Ex: zt, kz etc.).

2. Words with occurrence frequency above 0.001 and labelled as non-complex (if label available) in training data but vice versa by trained model, are classified non-complex.

3.4 Performance Comparison

Table 2 compares the performance of our systems with respect to performance statistics of all submitted systems, including baselines. From the analysis, it can be easily concluded that none of our systems are very effective in deciding complexity of words. Three of our systems- SVMPP, DT and SVM could reach around average performances and were ranked 31, 38 and 41 based on G-Score among 52 systems. The performances of SVM and SVMPP are almost comparable to other SVM based submitted systems and SVM seems to be less effective for CWI. But our DT system failed in achieving performance scores comparable to other DT and RFC based systems.

System HSVM&DT performed even worse than individual SVM and DT classifiers and was ranked 48 among all the systems. To determine the reasons, we analyzed the system's performance on only two features- word's frequency and word's length and plotted the decision boundaries. And it was observed that the system failed to achieve its objective of using multiple classifiers to predict the complexity of word and then deciding a classifier that has highest confidence on its prediction. The main reason for its failure was that all SVMs had almost overlapping decision boundaries. And the error in DTs prediction further worsened the prediction.

Systems	Precision	Recall	Fscore	Gscore
maximum	0.299	1	0.353	0.774
sv000gg	0.147	0.769	0.246	0.774
plujagh	0.289	0.453	0.353	0.608
average	0.114	0.606	0.173	0.560
s.d. [σ]	0.065	0.252	0.081	0.194
svmpp	0.099	0.415	0.160	0.546
dt	0.118	0.387	0.181	0.529
svm	0.119	0.363	0.179	0.508
hsvm&dt	0.112	0.226	0.149	0.360
minimum	0.0	0.0	0.0	0.0

Table 2: Systems Performance Measures

HSVM&DT, however, can be a very effective classifier for any task if we can manage to generate non-overlapping decision boundaries for all SVMs or any other constituent classifier. The performance of this system, though, is open for examination and needs to be explored further.

4 Conclusion

We tried several combinations of SVM and DT classifiers, but they could only achieve average of G-scores of all the submitted systems. As concluded by (Paetzold and Specia, 2016) in the CWI task description paper, DT and RFC perform better than most of the submitted systems. Also, the authors concluded that frequency measure possesses the highest confidence in deciding complexity of words. In our later study, we tried reducing the number of features and it was found that by merely including word-frequency, pos-tags, word-length and WordNet features, and training a DT model on same training set, system was able to achieve better G score on the test set. So, average performance of our systems can be mainly accounted to improper selection of features for the task. Inclusion of too many features shadowed the impact of frequency feature. In our future study, we plan to work on finding a suitable feature selection method for CWI and than work on classifier. Also as discussed before, failure of HSVM&DT model in this task is mainly accounted to lack of diversity in classifiers. We will continue our work on an ensemble classifier based on the principle of this model.

References

Marc Brysbaert and Boris New. 2009. Moving beyond kucera and francis: a critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for american english. *BEHAVIOR RESEARCH METHODS*, 41(4):977–990.

Christiane Fellbaum, editor. 1998. *WordNet An Electronic Lexical Database*. The MIT Press, Cambridge, MA ; London, May.

W. N. Francis and H. Kucera. 1979. Brown corpus manual. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island, US.

Daniel Jurafsky and James H. Martin. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition.

Robin Keskisärkkä. 2012. Automatic text simplification via synonym replacement.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, 19(2):313–330, June.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, November.

Gustavo H. Paetzold and Lucia Specia. 2016. Semeval 2016 task 11: Complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*.

Gustavo Paetzold. 2015. Reliable lexical simplification for non-native speakers. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 9–16, Denver, Colorado, June. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Matthew Shardlow. 2013. A comparison of techniques to automatically identify complex words. In *ACL (Student Research Workshop)*, pages 103–109. The Association for Computer Linguistics.

Matthew Shardlow. 2014. Out in the open: Finding and categorising errors in the lexical simplification pipeline. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may. European Language Resources Association (ELRA).

Lucia Specia, Sujay Kumar Jauhar, and Rada Mihalcea. 2012. Semeval-2012 task 1: English lexical simplification. In *Sixth International Workshop on Semantic Evaluation*, *SEM, pages 347–355, Montréal, Canada.