Identifying Real or Fake Articles: Towards better Language Modeling

Sameer Badaskar
School of Computer Science
Carnegie Mellon University
Pittsburgh PA, United States
sbadaska@cs.cmu.edu

Sachin Agarwal
School of Computer Science
Carnegie Mellon University
Pittsburgh PA, United States
sachina@cs.cmu.edu

Shilpa Arora
School of Computer Science
Carnegie Mellon University
Pittsburgh PA, United States
shilpaa@cs.cmu.edu

Abstract

The problem of identifying good features for improving conventional language models like trigrams is presented as a classification task in this paper. The idea is to use various syntactic and semantic features extracted from a language for classifying between real-world articles and articles generated by sampling a trigram language model. In doing so, a good accuracy obtained on the classification task implies that the extracted features capture those aspects of the language that a trigram model may not. Such features can be used to improve the existing trigram language models. We describe the results of our experiments on the classification task performed on a Broadcast News Corpus and discuss their effects on language modeling in general.

1 Introduction

Statistical Language Modeling techniques attempt to model language as a probability distribution of its components like words, phrases and topics. Language models find applications in classification tasks like Speech Recognition, Handwriting Recognition and Text Categorization among others. Conventional language models based on n-grams approximate the probability distribution of a language by computing probabilities of words conditioned on previous n words as follows

$$P(s) \approx \prod_{i=1}^{m} p(w_i|w_{i-n+1}, \ldots, w_{i-1}) \quad (1)$$

In most applications, lower order n-grams (such as bigram or trigram) are used but they are an unrealistic approximation of the underlying language. Higher order n-grams are desirable but they present problems concerning data sparsity. On the other hand, low order n-grams are incapable of representing other aspects of the language like the underlying topics, topical redundancy etc. In order to build a better language model, additional features have to be augmented to the existing language model (e.g. a trigram model) which capture those aspects of the language that the trigram model does not. Now, one way to test the goodness of a feature under consideration is to use it in a framework like an exponential model (Rosenfeld, 1997; Cai et al., 2000) and note the improvement in perplexity. An alternative way (Eneva et al., 2001) is as follows: Let L be the language and $\tilde{L}$ be an approximation of the language obtained by sampling the trigram language model. Also, let X be a piece of text obtained from either L or $\tilde{L}$. Let $y = h(f(X))$ such that $y = 1$ if $X \in L$ and $y = 0$ if $X \in \tilde{L}$ where $f(.)$ is the computed feature and $h(.)$ is the hypothesis function (a classifier like AdaBoost, SVM etc). If $Pr[y = h(f(x))]$ is found to be sufficiently high, it means that the feature $f(x)$ is able to distinguish effectively between the actual language L and the approximate language $\tilde{L}$. In other words, $f(x)$ captures those features of the language that are complementary to the ones captured by the trigram model and therefore $f(x)$ is a *good* feature to augment the trigram language model with.

The formalism explained previously can be interpreted as a classification task in-order to distinguish

between *Real* articles and *Fake* articles. Articles of different lengths drawn at random from the Broadcast News Corpus (BNC)[1] are termed as *Real* articles (from language L). Articles generated by sampling the trigram model trained on the same corpus are termed as *Fake* articles (language $\tilde{L}$). These articles together form the training data for the classifier to associate the features with the classification labels (*real* or *fake*) where the features are computed from the text. The features that give high classification accuracy on the test set of articles are considered good candidates for adding to the trigram model. Furthermore, the confidence that the classifier attaches to a classification decision can be used to compute the perplexity.

In this paper, a classification-task based formalism is used to investigate the goodness of some new features for language modeling. At the same time features proposed in the previous literature on language modeling are also revisited (Cai et al., 2000) Section 2 discusses various syntactic and semantic features used for the classification task, Section 3 gives details about the experiments conducted and the classification results obtained and finally, Section 4 concludes the paper by discussing the implications of the classification results on language modeling with pointers to improvements and future work.

2 Feature Engineering

To differentiate a *real* article from a *fake* one, the empirical, syntactic and semantic characteristics of a given article are used to compute the features for the classification task. The various types of features that were experimented are as follows:

2.1 Empirical Features

Empirical features are based on the statistical analysis of both the *real* and *fake* articles. They include the count of uncommon pairs of words within an article, the ratio of perplexity of trigram and quadgram models for a given article and the nature of the POS tags that occur at the start and end of sentences in an article.

[1]http://www.cs.cmu.edu/ roni/11761-s07/project/LM-train-100MW.txt.gz

Ratio of Perplexity of trigram and quad-gram models

Given an article, the ratio of its perplexity for a tri-gram model to a quad-gram model is computed. The trigram and quad-gram models are both trained on the same BNC corpus. Both *real* and *fake* articles would give a low perplexity score for the tri-gram model but for the quad-gram model, *real* articles would have significantly lower perplexity than the *fake* articles. This implies that the ratio of trigram to quad-gram perplexities would be lower for a *fake* article than for a *real* article. In other words, this ratio is similar to computing the likelihood ratio of an article w.r.t the trigram and quad-gram models. The histogram in Figure 1 shows a good separation in the distribution of values of this feature for the *real* and *fake* articles which indicates the effectiveness of this feature. A quadgram language model is a better approximation of *real* text than a trigram model and by using this as a feature, we are able to demonstrate the usefulness of the classification task as a method for identifying good features for language modeling. In the subsequent sections, we investigate other features using this classification framework.

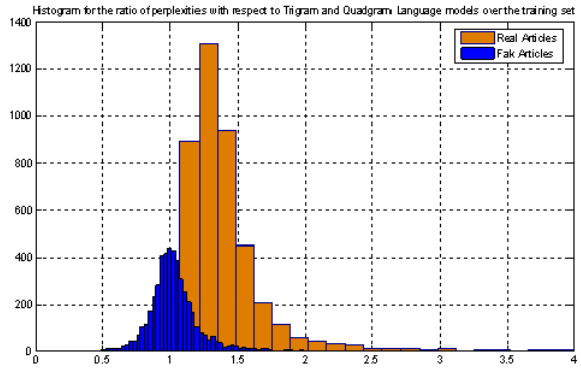


Figure 1: Histogram for the ratio of perplexities with respect to Trigram and Quadgram Language models over the training set

Count of uncommon pairs of words

Content words are the frequently occurring words in the corpus excluding the stop-words. All the words in corpus are ranked according to frequency of their occurrence and content words are defined to be the words with rank between 150 and 6500. A list of common content word pairs (pairs of content words

atleast 5 words apart) is prepared from the *real* corpus by sorting the list of content word pairs by their frequency of occurrence and retaining those above a certain threshold. For a given article, a list of content word pairs is compared against this list and word pairs not in this list form the set of uncommon word pairs.

A *real* article is expected to have lesser number of uncommon content-word pairs than *fake* articles. When normalized by the total number of word pairs, we get the probability of finding an uncommon content-word pair in an article. This probability is greater for *fake* articles than the *real* articles and we use this probability as a feature for the classification task.

Start and End POS Tags

Certain POS tags are more probable than others to appear at the beginning or end of a *real* sentence. This characteristic of *real* text could be used as a feature to distinguish *real* articles from *fake*. The distribution of POS tags of the first and last words of the sentences in an article is used as a feature. Our experiments show that this feature had very little effect in the overall contribution to the classification accuracy over the development set.

2.2 Syntactic Features

These features are derived from the parse structure of the sentence. It is hypothesized that *real* sentences tend to be grammatical while the same may not be the case for *fake* sentences. An objective measure of the grammaticality of a sentence can be obtained by running it through a statistical parser. The log-likelihood score returned by the parser can be used to judge the grammaticality of a sentence and thus determine whether it is *fake* or *real*. The Charniak Parser (Charniak, 2001; Charniak, 2005) was used for assessing the grammaticality of the articles under test. Given an article containing sentences $S_1, S_2, \ldots, S_N$ with lengths $L_1, L_2, \ldots, L_N$, we compute the parser log-likelihood scores $P(S_1), P(S_2), \ldots, P(S_N)$. The overall grammaticality score for an article is given by

$$P_{Gram} = \frac{\sum_{i=1}^{N} L_i P(S_i)}{\sum_{i=1}^{N} L_i} \quad (2)$$

The grammaticality score was normalized using the average and standard deviation over the entire training set. This feature gave small improvement in terms of classification accuracy. There may be several reasons for this: (1) Our training data consisted of spoken transcripts from a broadcast news corpus whereas the Charniak Parser was trained on a different domain (Wall Street Journal) and (2) The parser was trained on mixed case text where as the data we used was all upper case.

2.3 Semantic Features

Real articles contain sentences with correlated pairs of content-words and sentences that are correlated with each other. An article with such sentence/word correlations is said to be semantically coherent. Owing to the use of only the short term word history for computing the probability distribution of a language, a trigram model fails to model semantic coherence and we exploit this fact for the classification task. Specifically, we intend to model both intra-sentence and inter-sentence semantic coherence and use them as features for classification.

Intra-sentence Coherence

To model the intra-sentence word correlations, we use Yule's Q-statistic (Eneva et al., 2001). The word correlations are learned from the BNC corpus as well as the *fake* corpus. The coherence score for an article is defined as the sum of the correlations between pairs of content words present in the article. The coherence score for an article is normalized by the total number of content-word pairs found in the article. Since the trigram and quad-gram language model can capture short distance coherences well, coherences between distant words can be used to differentiate between *real* and *fake* articles. The Yule Q-statistic is calculated for every pair of content words, which are atleast 5 words apart within a sentence, both in the *real* and *fake* corpus.

The articles are scored according to content word-pair correlations learned from the *real* as well as *fake* corpus. Each article is given two scores, one for the word-pair correlations from *real* articles and other for the word-pair correlations from *fake* articles. For a *real* article, the *real* word-pair correlation score would be relatively higher compared to the *fake* word-pair correlation score (and vice-versa

for a *fake* article).

Modeling Topical Redundancy (Inter-sentence Coherence)

A characteristic of *real* articles is that they tend to be cohesive in terms of the topic under discussion. For example, a news-article about a particular event (topic) would have several direct or indirect references to the event. We interpret this as some sort of a *redundancy* in terms of the information content which we term as Topical Redundancy. The *fake* articles would not exhibit such a redundancy. If a *real* article is transformed to another *representation space* where some form of *truncation* is applied, on transformation back to the original space, the amount of information-loss may not be significant due to information redundancy. However, if the same process is applied on a *fake* article, the information-loss would be significant when transformed back to the original space. We intend to exploit this fact for our classification task.

Let $D_{W \times N}$ be an article represented in the form of a matrix, where W is the article vocabulary and N is the number of sentences in that article. Every term of this matrix represents the frequency of occurrence of a vocabulary word in a particular sentence. We construct a sentence-sentence matrix as follows:

$$A = D^T D \tag{3}$$

We now *transform* A into the Eigen-space using Singular Value Decomposition (SVD) which gives

$$A = USU^T \tag{4}$$

Here, $U_{N \times N}$ is the eigen-vector matrix and $S_{N \times N}$ is the diagonal eigen-value matrix. If we retain only the top K eigen-values from S , we get the truncated (lossy) form $S'_{K \times K}$. Thus the *truncated* form of A i.e. A' is

$$A' = US'U^T \tag{5}$$

We believe that the information loss $\| A - A' \|^2$ will not be significant in the case of *real* articles since the topical redundancy is captured in a very compact manner by the eigen-representation. However, in the case of a *fake* article, the loss is considerable. For a *real* article, the matrix would be less sparse than a *fake* article and so is the case for the reconstructed matrix. Therefore, the statistics - mean, median, minimum and maximum computed from the reconstructed matrix have higher values for *real* articles than a *fake* articles. We use these statistics as features for classifying the article. Figure 2 show the histograms of the statistics computed from the reconstructed matrix for the training set. As can be seen, there is a good separation between the two classes *fake* and *real* in all the cases. Using these features increased the classification accuracy by a significant amount as shown later. From another perspective, these features model the inter-sentence semantic coherence (Deerwester et al., 1990) within an article and this is consistent with our notion of topical redundancy as explained previously. The matrix package developed by NIST (Hicklin et al., 2005) was used for SVD.

3 Experimental Results

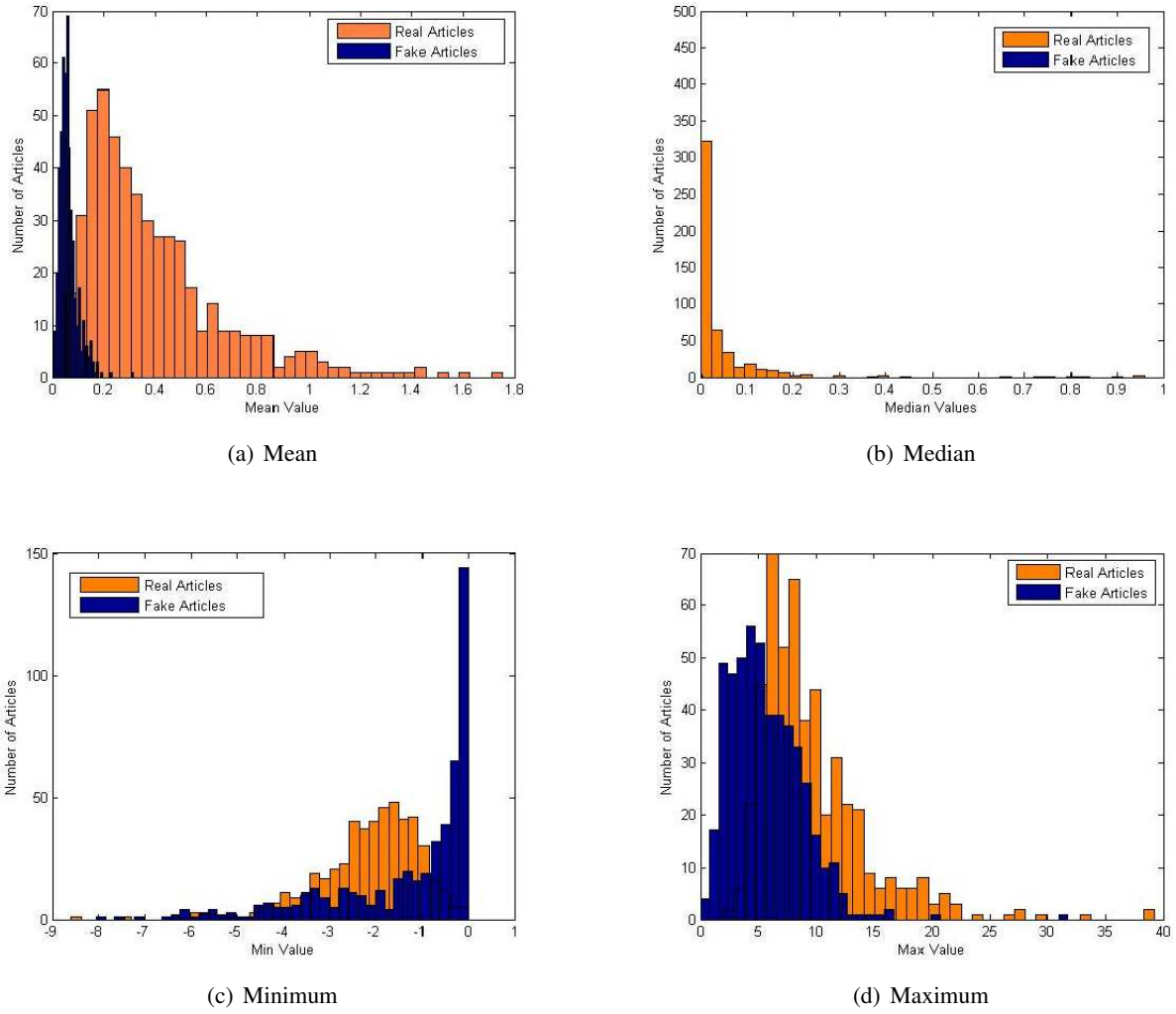
3.1 Data Distribution

The training data consisted of 1000 articles (500 *real* and 500 *fake*) obtained from Broadcast News Corpus (BNC) and the test set consisted of 200 articles (100 *real* and 100 *fake*). Additionally, a development dataset consisting of 200 articles and having the same distribution as that of the test dataset was used for tuning the parameters of the classifiers. To ensure that the training and test data come from the same article length distribution, the training data was resampled to have the same percentage of articles of a given length as in the test set. The article length distribution for both the training(resampled) and test datasets is shown in Tables 1 and 2.

3.2 Classifier

Classifiers like AdaBoost (Freund et al., 1999) and Max-Entropy (Rosenfeld, 1997) models were used for the classification task.

The number of iterations for AdaBoost was estimated using 5-fold cross-validation. Given a subset of features, Maxent classified 74.5% of the documents correctly compared to 82% for AdaBoost. Therefore, Adaboost was chosen as the classifier for further experiments.



(a) Mean

(b) Median

(c) Minimum

(d) Maximum

Figure 2: Histograms of topical redundancy features computed over the training set. In (b) , the median values for the fake articles are close to zero and hence cannot be seen clearly.

3.3 Results and Discussion

We used two performance measures to evaluate our model. First is the accuracy which measures the number of articles correctly classified as *real* or *fake* and the second measure is the log-probability that the model assigns to the classification decision i.e. it measures the confidence the model has in its classification. Table 3 shows our experimental results on the syntactic, semantic and empirical features.

The combination of syntactic, semantic and empirical features gave an accuracy of 91.5% with an average log-likelihood of -0.22 on development data set. The accuracy on the test dataset was 87% with an average log-likelihood of -0.328.

4 Conclusions and Future Work

In this work, we have used a classification-task based formalism for evaluating various syntactic, semantic and empirical features with the objective of improving conventional language models. Features that perform well in the task of classifying *real* and trigram-generated *fake* articles are useful for augmenting the trigram model. Semantic features, such as topical redundancy, model long-range dependencies which are not captured by a trigram language model. Therefore, the semantic features contribute significantly to the classification task accuracy. Additionally, linguistic resources such as WordNet (WordNet, 1998) can be used to model

# Sentences per article	# Real Art.	# Fake Art.	% Total (Real & Fake)
1	938	940	19.76
2	440	471	9.58
3	502	474	10.26
4	507	533	10.94
5	497	525	10.75
7	431	524	10.05
10	475	479	10.04
15	482	421	9.50
20	421	446	9.12

Table 1: Distribution of article lengths for training dataset.

# Sentences per article	# Real Art.	# Fake Art.	% Total (Real & Fake)
1	20	20	20
2	10	10	10
3	10	10	10
4	10	10	10
5	10	10	10
7	10	10	10
10	10	10	10
15	10	10	10
20	10	10	10

Table 2: Distribution of article lengths for test dataset.

topical redundancy using synonyms and other inter-word dependencies. The semantic features we explored assume a single underlying topic for an article which may not be always true. An article can be a representation of different topics and we aim to explore this direction in future.

Feature Combination	Classification Accuracy	Avg. Log Likelihood
Syntactic	60.5%	-0.663
Semantic	79.5%	-0.510
Empirical	83.0%	-0.446
Semantic + Syntactic	80.0%	-0.553
Semantic + Empirical	86.0%	-0.410
Semantic + Syntactic + Empirical	91.5%	-0.220

Table 3: Performance of different features on the development-set.

References

Can Cai, Larry Wasserman and Roni Rosenfeld. 2000. *Exponential language models, logistic regression, and semantic coherence*. Proceedings of the NIST/DARPA Speech Transcription Workshop.

Eugene Charniak. 2001. *Immediate-Head Parsing for Language Models*. Proceedings of 39th Annual Meeting of the ACL, 124-131.

Eugene Charniak. 2005. ftp://ftp.cs.brown.edu/pub/ nlparser/parser05Aug16.tar.gz

Thomas M. Cover and Joy A. Thomas. 1991. *Elements of Information Theory*. John Wiley & Sons, New York.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer and Richard Harshman. 1990. *Indexing by Latent Semantic Analysis*. Journal of Japanese Society for Artificial Intelligence, 41(6).

Elena Eneva, Rose Hoberman and Lucian Lita. 2001. *Learning within-sentence semantic coherence*. Proceedings of the EMNLP 2001.

Yoav Freund and Robert E. Schapire. 1999. *A short introduction to boosting* Journal of Japanese Society for Artificial Intelligence, 14(5):771-780.

Joe Hicklin, Cleve Moler, Peter Webb. 2005. http://math.nist.gov/javanumerics/jama/

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.

Roni Rosenfeld. 1997. *A whole sentence maximum entropy language model*. In Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding, 1997.

WordNet: An Electronic Lexical Database, ISBN-13: 978-0-262-06197-1.