Discriminating Non-Native English with 350 Words

John Henderson, Guido Zarrella, Craig Pfeifer and **John D. Burger**
The MITRE Corporation
202 Burlington Road
Bedford, MA 01730-1420, USA
{jhndrsn,jzarrella,cpfeifer,john}@mitre.org

Abstract

This paper describes MITRE's participation in the native language identification (NLI) task at BEA-8. Our best effort performed at an accuracy of 82.6% in the eleven-way NLI task, placing it in a statistical tie with the best performing systems. We describe the variety of machine learning approaches that we explored, including Winnow, language modeling, logistic regression and maximum-entropy models. Our primary features were word and character n-grams. We also describe several ensemble methods that we employed for combining these base systems.

1 Introduction

Investigations into the effect of authors' latent attributes on language use have a long history in linguistics (Labov, 1972; Biber and Finegan, 1993). The rapid growth of social media has sparked increased interest in automatically identifying author attributes such as gender and age (Schler et al., 2006; Burger and Henderson, 2006; Argamon et al., 2007; Mukherjee and Liu, 2010; Rao et al., 2010). There is also a long history of computational aids for language pedagogy, both for first- and second-language acquisition. In particular, automated native language identification (NLI) is a useful aid to second language learning. This is our first foray into NLI, although we have recently described experiments aimed at identifying the gender of unknown Twitter authors (Burger et al., 2011). We performed well using only character and word n-grams as evidence. In the present work, we apply that same approach to NLI, and combine it with several other baseline classifiers.

In the remainder of this paper, we describe our high-performing system for identifying the native language of English writers. We explore a varied set of learning algorithms and present two ensemble methods used to produce a better system than any of the individuals. In Section 2 we describe the data and task in detail as well as the evaluation metric. In Section 3 we discuss details of the particular system configuration that scored best for us. We describe our experiments in Section 4, including our exploration of several different classifier types and parametrizations. In Section 5 we present and analyze performance results, and inspect some of the features that were useful in discrimination. Finally in Section 6 we summarize our findings, and describe possible extensions to the work.

2 Task, data and evaluation

Native Language Identification was a shared task organized as part of the *Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, 2013. The task was to identify an author's native language based on an English essay.

The data provided consisted of a set of 12,100 Test of English as a Foreign Language (TOEFL) examinations contributed by the Educational Testing Service (Blanchard et al., to appear). These were English essays written by native speakers of Arabic, Chinese, French, German, Hindi, Italian, Japanese, Korean, Spanish, Telugu, and Turkish. A set of 1000 essays for each language was identified as training data, along with 100 per language for development,

Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications, pages 101–110,
Atlanta, Georgia, June 13 2013.

and another 100 per language for a final test set. The mean length of an essay is 348 words.

The primary evaluation metric for shared task submissions was simple accuracy: the fraction of the test essays for which the correct native language was identified. A baseline accuracy would thus be about 9% (one out of eleven). Results were also reported in terms of F-measure on a per-language basis. F-measure is a harmonic mean of precision and recall: $F = \frac{2PR}{P+R}$. For the evaluation, the precision denominator was the number of items labeled with a particular language by the system and the recall denominator was the number of items marked with a particular language in the reference set.

The training, development, and test sets all had balanced distributions across the native languages, so error rates and accuracy did not favor any particular language in any set.

3 System overview

The systems we used to generate results for the NLI competition were all machine-learning-based, with no handwritten rules or features. The final submitted systems were ensembles built from the outputs and confidence scores of independent eleven-way multinomial classifiers.

3.1 Features

The features used to build these systems were language-independent and were generated using the same infrastructure designed for the experiments described in Burger et al. (2011).

We incorporated a variety of binary features into our systems, each of which was hashed into a 64-bit numeric representation using MurmurHash3 (Appleby, 2011). The bulk of our features were case-sensitive word- and character-based n-grams, in which a feature was turned "on" if its sequence of words or characters appeared at least once in the text of an essay. We also added binary features describing surface characteristics of the text such as average word length and word count. Features were separated into tracks such that the word unigram "i" and the character unigram "i" would each generate a distinct feature.

Part of speech tag n-grams were added to the feature set after reviewing performance results in Brooke and Hirst (2012). We used the Stanford log-linear part of speech tagger described in Toutanova et al. (2003), with the english-left3words-distsim.tagger pretrained model and the Penn Treebank tagset. The tagger was run on each essay and outputs were incorporated as sequence features with n-grams up to length 5.

3.2 Classifiers

Carnie[1] is a MITRE-developed linear classifier that implements the Winnow2 algorithm of Carvalho and Cohen (2006), generalized for multinomial classification. Carnie was developed to perform classification of short, noisy texts with many training examples. It maintains one weight per feature per output class, and performs multiplicative updates that reinforce weights corresponding to the correct class while penalizing weights associated with the top-scoring incorrect class. The learner is mistake-driven and performs an update of size ϵ after an error or when the ratio of weight masses of the correct and top incorrect classes is below $1 + \delta$. It iterates over the training data, cooling its updates after each iteration. For the purposes of these experiments, an input to Carnie was the text of a single TOEFL essay, and the output was the highest scoring class and several related scores.

SRI's Language Modeling Toolkit (SRILM) is a toolkit for sequence modeling that continues to be relevant after more than a decade of development (Stolcke, 2002). It can be used to both build models of sequence likelihoods and to evaluate likelihoods of previously unseen sequences. Building a multinomial text classifier with a language model toolkit involves building one model for each target class and choosing the label whose model gives the highest probability.

Many smoothing methods are implemented by SRILM, along with a variety of n-gram filtering techniques. The out-of-the-box default configuration produces trigram models with Good-Turing smoothing. It worked well for this competition. Using open vocabulary models (`-unk`), turning off sentence boundary insertion (`-no-sos -no-eos`) and treating each essay as one sentence

[1] It is named for entertainers who guess personal characteristics of carnival goers.

worked best in our development environment.

LIBLINEAR is a popular open source library for classification of large, sparse data. We experimented with several of their standard Support Vector Machine and logistic regression configurations (Fan et al., 2008). We selected multiclass ℓ_2-regularized logistic regression with the dual-form solver and default parameters. Inputs to the model were binary features generated from a single TOEFL essay. Features for this model were generated by Carnie. The model provided probability estimates for each candidate output class (L1) for each essay, which were then combined with the outputs of Carnie and SRILM in an ensemble to produce a single prediction.

3.3 Ensembles

The classifiers described above were selected for inclusion as components in a larger ensemble on the basis of their performance and the observation that errors committed by these systems were not highly correlated. We used the entirety of our training data for construction of each component system, leaving scant data available for estimating parameters of ensembles. This scenario led us to choose naive Bayes to combine the outputs of the original components.

Given $h_1, \ldots, h_k$ hypothesis labels from k different systems, one approximates the conditional likelihood of the reference label $P(R|H_1 \ldots H_k)$ using the Bayes transform and the development set estimates of $P(H_i|R)$. One investigates all possible labels to decode $r^* = \text{argmax}_r P(r) \prod_i P(h_i|r)$. The class balance in every set we operated on made the prior $P(r)$ irrelevant for maximization and simplified many of the denominators along the way. This is a typical formulation of naive Bayes.

Confidence All of our component systems produce scores as well as a predicted label. Carnie produces (non-probability) scores for all of the candidate labels, SRILM produces log-probabilities and perplexities, and LIBLINEAR produces $P(h|r)$, the likelihood of each of the possible labels. We experimented with several transformations of those scores to best use them to predict correctness of their hypothesis. There were several graphical models we could use for folding these scores into the Bayes ensemble, and we chose a simple, discretized $P(H, S|R)$. We evenly partitioned and relabeled our system outputs according to their scores (S), and used those partition labels in the Bayes ensemble. Thus when a particular reference label was scored in the ensemble during decoding, both its prediction and score contributed to the label in the naive Bayes table lookup.

3.4 Best configuration

We submitted five systems with a variety of configurations. One of our systems was our individual Carnie system on its own for calibration. The other four were ensembles.

The best system we submitted was a Bayes ensemble of the Carnie, SRILM, and LIBLINEAR components each trained on the train+development sets. Carnie was trained for twelve iterations with $\epsilon = 0.03$, $\delta = 0.05$, and a cooling rate of 0.1. SRILM models were trained for open vocabulary and the default trigram, Good-Turing setting. Logistic regression from LIBLINEAR was run with ℓ_2 regularization and using the dual form solver.

Parameters for the Bayes model were collected from the development set when the components were trained only on the training set. A grid search was performed over likely candidates for λ, the Dirichlet parameter, and ρ, the number of score-based partitions, resulting in $\lambda = 0.03125$ and $\rho = 2$. The grid search was performed with the component models trained only on the training set and using 10-fold cross validation on the development set.

4 Experiments

In all experiments described below, systems were trained initially on the 9900 training examples alone, with the 1100 item development set held back to allow for hyperparameter estimation. When preparing our final test set submissions, the development set was folded into the training data, and all models were re-trained on this new dataset containing 11000 items.

4.1 Baselines

How hard is the NLI task? Simple baselines often give us a quick glimpse into what matters in a NLP task. In Figure 1, we give accuracy results on ten different baselines we trained on the training

Baseline	Accuracy(%)
random	9.1
char length	9.6
SRILM(letter unigram)	10.8
word length	12.0
proficiency	14.9
SRILM(letter bigram)	15.1
JS(vowels)	20.6
JS(consonants)	33.8
JS(vowels+consonants)	34.1
JS(bag-of-words)	52.5

Figure 1: Simple baseline development set scores.

set and evaluated on the development set. Predictions based on simple character and word lengths show only slight gains over random. Using the high/medium/low proficiency score that accompanied the data similarly gives a tiny amount of information over baseline (14.9%). We ignored those ratings elsewhere in our work, to focus on the core task of prediction based on essay content.

We collected some simple distributions of vowel and consonant clusters and used them for prediction, scoring with Jensen-Shannon divergence. JS divergence is a symmetrized form of KL divergence to alleviate the mathematical problem involved with missing observations. It has behaved well in the context of language processing applications (Lee, 1999). The score progression from consonant clusters, to vowel clusters, to words suggests that there is NLI information scattered at various levels of surface features.

4.2 Varied Carnie configurations

Carnie's out-of-the-box configuration is one that has been optimized for application to micro-blogs and other ungrammatical short texts. While our hypothesis was that this configuration would be well suited to analysis of English TOEFL essays, we investigated a number of possible techniques to help Carnie adapt to the new domain.

We began by performing a grid search to select model hyperparameters that enabled our standard configuration to generalize well from the training dataset to the development dataset. These values of ϵ, δ, and cooling rate were then applied to various new feature configurations.

The standard configuration included binary features for word unigrams and bigrams, character n-grams of sizes 1 to 5, and surface features. We experimented here with word trigrams, character 6-grams, and lowercased character n-grams of sizes 1 to 6. We also added skip bigrams, which were ordered word pairs in which 1 to 6 intervening words were omitted. We incorporated part of speech tags in a number of ways, including POS n-grams of lengths 1 to 5, POS k-skip bigrams with k ranging from 1 to 6, and POS n-grams in which closed-class POS tags were replaced with the actual content word used. We also measured the impact of using frequency-weighted features.

Our standard approach with Carnie is to perform multinomial classification using one model trained on all the data simultaneously. We experimented with other ways of framing the NLI problem, such as building eleven binary classifiers, each of which was trained on all of the data but with the sole task of accepting or rejecting a single candidate L1. We also partitioned the training data to build 55 binary classifiers for all possible pairs of L1s. These binary classifiers were then combined via a voting mechanism to select a single winner. This allowed us to apply focused efforts to improve discrimination in language pairs which Carnie found challenging, such as Hindi-Telugu or Japanese-Korean. To this end, we collected a substantial amount of additional out-of-domain training data from the websites lang8.com (70,000 entries) and gohackers.com (40,000 entries). Although we did not use this data in our final submission, we performed experiments to measure the value of this new data in the TOEFL11 domain with no adaptation, with feature filtering to limit training features to items observed in the test sets, and with "frustratingly easy" domain adaptation, EasyAdapt, described in Daumé and Marcu (2007).

4.3 Varied SRILM configurations

SRILM offers a number of parameters for experimentation. We hill-climbed on the training/development split to select a good configuration. We experimented with n-gram lengths from 1-5 (bag of words through word 5-grams), using the tokenization given by the NLI organizers. We tried the lighter weight smoothing techniques offered by

System	Confidence	MRD
Carnie	$s(h_1)/s(h_2)$	343
	$s(h_1)/\sum_i s(h_i)$	268
	$s(h_1) - s(h_2)$	72
SRILM	$\log p(h_1)/\log p(h_2)$	315.7
	$\log p(h_1) - \log p(h_2)$	315.3
	$ppl1(h_1)/ppl1(h_2)$	315.12
	$ppl1(h_1) - ppl1(h_2)$	260
	$ppl1$	77
	$\log p(h_1)$	40
MaxEnt (JCarafe)	$\sum_i p(h_i) \log p(h_i)$	385.7
	$p(h_1)$	383.15
	$\log p(h_1)$	383.15
	$p(h_1)/p(h_2)$	373.75
	$\log p(h_1)/\log p(h_2)$	379.8
LIBLINEAR	$\sum_i p(h_i) \log p(h_i)$	379.8

Figure 2: Confidence candidates measured in Mean Rank Difference between correct and incorrect labels.

SRILM including Good-Turing, Witten-Bell, Ristad's natural discounting, both modified and original Kneser-Ney. We built both closed vocabulary and open vocabulary language models and with special symbols added for sentence boundaries.

4.4 Component confidence experiments

Our components generate scores, but those scores were not always scaled in the same way. Winnow (in Carnie) is a margin-based, mistake-driven learner generating scores which are interpretable only as sums of weights. SRILM produces $\log p(d_j|h_i)$, but renormalizing those (with priors) into estimates of $p(h_i|d_j)$ is unreliable because the different submodels are not connected with smoothing. Logistic regression produces a distribution for $p(h_i|d_j)$. We aimed to express these notions of confidence in a way that was common to all systems. We did this by relabeling system hypotheses after sorting by confidence, but not all metrics were equally good at this sorting.

We performed an ad hoc assessment of several candidate scoring functions. Our goal was to find functions that best separated correct answers from incorrect answers in a sorted ranking. We ran several candidates on our development set and measured the difference between the mean rank of correct answers and the mean rank of incorrect answers. Figure 2 displays the results. In each case h_1 was the best hypothesis generated by the system and h_2 is second best. $p(\cdot)$ indicates probabilities, $s(\cdot)$ indicates non-probability scores. We chose those functions with the highest values.

4.5 Simple models for combination

In this work, we focused our ensembles only on the output of our individual components, ignoring the features from the original data that they attempt to model. The base systems are all trained to minimize errors, and did not appear to have any particular preferential capabilities. Thus we rely on them entirely for the primary processing and focus on their outputs.

In our naive Bayes formulation, the random variables produced by the component systems (H) need not take on values directly comparable with the reference labels to be predicted (R). We experimented with folding in several one-shot systems that produced labels in $\{L, \bar{L}\}$, for particular native language groups, but none of these proved to be good complements for the components described above.

To cope with decode-time configurations of H that hadn't been seen during estimation, we used a Dirichlet prior on R in this ensemble. A single parameter, λ, was introduced. Thus our estimates for $P(h_i|r)$ were based on smoothed counts: $\frac{c(h_i,r)+\lambda}{c(r)+\lambda|R|}$. The search for λ was performed using cross-validation on the development set.

Assignment In many prediction settings, we know that our evaluation data consists of examples drawn from a particular allocation of candidate classes. One can take advantage of this in a probabilistic setting by doing a global search for the maximum likelihood assignment of the test documents to the L1 languages under the constraint that each L1 language must have a particular occupancy by the documents – in this case, an even split. More generally, once we have $p(h_i|d_j)$ for each candidate language h_i and document d_j, we can find an assignment $A = \{(i,j) : \alpha_{i,j} = 1\}$ that maximizes the likelihood $P(H|D) = \prod_{(i,j)\in A} p(h_i|d_j) = \prod_{i,j} p(h_i|d_j)^{\alpha_{i,j}}$ under the constraints that $\sum_i \alpha_{i,j} = |D|/|H|$ and $\sum_j \alpha_{i,j} = 1$. The first constraint says that each language should get an even allocation of documents assigned to it and the second constraint says that

each document should be assigned to only one language. This reduces to a maximum weight matching on $\sum_{i,j} \alpha_{i,j} \log p(h_i|d_j)$. This problem is directly convertible into a max flow problem or a linear program. It can be solved with methods such as the Hungarian algorithm, Ford-Fulkerson, or linear programming. In our case, we used `LPSOLVE`[2] to find this global maximum. This looks at first glance like an integer programming problem, but one can relax the constraints into inequalities and still be guaranteed that the solution will end up with all $\alpha_{i,j}$ landing on either zero or one in the right amounts. We applied this assignment combination as a post-processing step to the probabilities generated in the naive Bayes ensemble and also to the raw LIBLINEAR outputs. The hope in doing this is that the optimizer will move the less likely assignments around appropriately while preserving the assignments where it has more confidence. We observed mixed results on our development set and submitted two systems using this ensemble technique.

4.6 Other components explored

LIBLINEAR provides an implementation of a linear SVM as well as a logistic regression package. We experimented with various combinations of ℓ_1- and ℓ_2 -loss SVMs, with both ℓ_1 and ℓ_2-regularization, but in the end opted to use the ℓ_2-regularized logistic regression due to slightly superior performance and the ease with which we could extract eleven values of $P(H)$ for inclusion in our ensemble.

Another component that was tested in development of our ensemble systems was a maximum entropy classifier. This particular effort used the implementation from JCarafe,[3] which uses L-BFGS for optimization.

We approached the NLI task as document classification, following a typical JCarafe recipe (Gibson et al., 2007). The class of the document is the native language of the author. Each document was treated as a bag of words, and several classes of features were extracted: token n-gram frequency, character n-gram frequency, part of speech n-gram frequency. The feature mix that produced the best score was token bigrams and trigrams, character trigrams and POS trigrams. A feature frequency threshold of 5 was used to curb the number of features.

[2] `http://lpsolve.sourceforge.net`

[3] `https://github.com/wellner/jcarafe`

L1	Mean F		Our Best F	
GER	1	0.776	1	0.921
ITA	2	0.757	2	0.88
CHI	3	0.723	4	0.85
JPN	4	0.708	5	0.837
FRE	5	0.701	7	0.818
TEL	6	0.667	3	0.802
KOR	7	0.665	6	0.827
TUR	8	0.656	8	0.81
ARA	9	0.65	3	0.872
SPA	10	0.631	10	0.768
HIN	11	0.606	11	0.762

Figure 3: L1s by empirical prediction difficulty. Mean F incorporates all submissions by all competition teams.

5 Results

Our best performing ensemble was 82.6% accurate when scored on the competition test set, and was composed of Carnie, SRILM, and logistic regression, using naive Bayes to combine the subsystem outputs and confidence scores into a single prediction. The best performing subsystem during system development scored 79.3% on the test set in isolation, demonstrating once again the value of combining systems that make independent errors.

Certain L1s gave our systems more difficulty than others. Our best submitted F-measure scores ranged from 0.921 for German to 0.762 for Hindi. Figure 3 demonstrates that our systems' scores were highly correlated with average scores from all submissions by all teams ($R^2 = 0.84$). From this we infer that our performance differences between L1s may be explained by inherent difficulties in certain languages or by the selection of similar L1s as a part of the competition task, rather than quirks of our approach. Our submissions do appear to have a particular advantage on Arabic and Korean, relative to the field.

Figure 4 shows the overall performance of our submissions and subsystems on the development and test evaluation sets.

Our scores dropped 4 to 5% between development and test evaluations, representing significant overfit-

Configuration	dev %	test %
Components		
base Carnie	82.6	
+ trigrams	83.1	
+ POS tags	83.6	79.3
1v1 voted Carnie	79.4	
SRILM	77.1	
MaxEnt	77.7	
Linear SVM	81.9	
Logistic Regression	83.4	
assignment(LR)	82.4	
Ensembles		
bayes(Carnie,SRILM,LR)	87.3	82.6
assign(Carnie,SRILM,LR)	86.5	82.0
assign(Carnie,SRILM,MaxEnt)	86.4	82.3
bayes(Carnie,SRILM)	86.9	81.7

Figure 4: Results.

ting to the development set. The development set was used for model selection, ensemble parameterization, and eventually as additional training data for final submissions. Later tests showed that this final retraining actually reduced the Carnie score by 0.9%.

Figure 4 also shows the effect of various efforts to improve our baseline Carnie system. Adding part-of-speech n-grams and word trigrams as features improved the score on the development set by 1% in total. Meanwhile many of our experiments with new types of features yielded no gains. Lowercased character n-grams, skip bigrams and all non-vanilla formulations of part-of-speech tags provided no improvement and were discarded.

It was observed that all of our systems showed a strong preference for binary features over frequency-weighted inputs. In the case of the JCarafe classifier, switching to binary features yielded a 10% accuracy gain. Although JCarafe didn't provide a gain over the ensemble of Carnie, SRILM, and LIBLINEAR logistic regression, development set results indicated that JCarafe served capably as a replacement for LIBLINEAR in some ensembles.

We also measured the impact of using out-of-domain Japanese and Korean L1 data to train a pairwise JPN/KOR system. Only 78.5% of JPN and KOR texts were correctly identified in our eleven-way baseline system. We restricted train and evaluation data to only those two L1s and found our baseline technique was 86.5% accurate. When we added our out-of-domain data with no domain adaptation technique, that score dropped to 82.0%. Removing features that didn't appear in our test set only raised the score to 82.5%. However, the EasyAdapt technique (Daumé and Marcu, 2007) showed promise. By making an additional source-specific copy of each feature, we were able to raise the score to 88.5%. While this result was of limited applicability in our final submission, and was therefore not submitted to the open data competition task, we believe that this technique may prove useful in enabling cross-domain NLI system transfer.

Rank	L1	Score	Feature
14	GER	21.05	(for,example)
40	GER	15.95	(have,to)
55	HIN	14.80	(as,compared,to)
57	ITA	14.60	(I,think,that)
58	TEL	14.18	(and,also)
60	HIN	13.97	(as,compared)
79	TEL	12.82	(the,people)
96	TEL	12.14	(for,a)
101	ITA	11.83	(that,in)
116	ITA	10.94	(think,that)
119	GER	10.93	(has,to)
120	TEL	10.89	(with,the,statement)

Figure 5: Word n-gram features predicting particular L1.

Figure 5 provides a small sample of word-level features discovered by the Winnow classifier. The table shows the rank of each n-gram relative to all features, and the native language that the feature predicts. The weight assigned by the Winnow2 algorithm is not readily interpretable, although higher weights indicate a stronger association.

Similarly, the top character n-grams can be seen in Figure 7, along with manually selected examples of each. These features can be seen to mainly fall into several broad categories. There are mentions of the authors' home countries as in K[orea]n, I[tal]ian and T[urk]ey. There are also characteristic misspellings and infelicities such as pers[onn]aly, perhaps incorrectly modeled from the French *pers[onn]ellement*.

It is worth noting that the weights (and thus the ranks) for the top character n-gram features are

System	Accuracy (%)	Errors
Carnie	80.4	2153
SRILM	74.5	2800
LIBLINEAR	80.8	2116
ensemble-assign	81.9	1990
ensemble-Bayes	82.2	1961

Figure 6: Training set cross-validation results.

higher than for the top word features, indicating that Winnow found the former to be more informative.

Finally, the top part-of-speech n-gram features are shown in Figure 8, again with manually selected examples. These features have similar weights to the character n-gram features and for the most part seem to represent ungrammatical constructions (e.g., the first feature indicates that a personal pronoun followed by an uninflected verb predicts Chinese). However, there are some perfectly grammatical items that are indicative of a particular native language (e.g., *as compared to* for Hindi). One possible explanation might be a dominant L2 pedagogy for that language.

5.1 Cross-validation results

The task organizers requested that the participants run a ten-fold cross validation on a particular split of the union of the training and development sets after the evaluation was over. Results of our leading component systems and ensemble systems are presented in Table 6. These are comparable with the TOEFL-11 column of Figure 3 in Tetreault et al. (2012).

6 Conclusion

In this paper, we have presented MITRE's participation in the native language identification task at BEA-8. Our best system was a naive Bayes ensemble combining component systems that used Winnow, language modeling and logistic regression approaches, all using relatively simple character and word n-gram features. This ensemble performed at an accuracy of 82.6% in the eleven-way NLI task, placing it in a statistical tie with the winning systems submitted by 29 teams. For individual native languages, our submission performed best among the participants on Arabic, as ranked by F-measure.

In addition to the three base systems in our best ensemble, we experimented with a maximum entropy classifier and an assignment-based ensemble method. We described a variety of experiments we performed to determine the best configurations and settings for the various systems. We also covered experiments aimed at using out-of-domain data for several native languages. In future work we will expand upon these, with the goal of applying domain adaptation approaches.

One concern with NLI as framed in this evaluation is the interaction between native language and essay topic. The distribution of topics was very similar in the various subcorpora, but in more natural settings this is unlikely to be the case, and there is a danger of overtraining on topic, to the detriment of language identification performance. This is especially problematic for a highly lexical approach such as ours. In future work, we intend to explore the extent of this effect, using topic-based splits of the corpus. Our initial experiments to remedy this problem are likely to involve domain adaptation approaches, such as Daumé and Marcu (2007).

As described above, we have had success using the Winnow-based system Carnie for other latent author attributes, such as gender. We would like to explore ensembles similar to those described here for these attributes as well.

The techniques described in this paper successfully identified an author's native language 82.6% of the time using a sample of text averaging less than 350 words in length. Future work could study the interaction of text length and NLI performance, including texts shorter than 140 characters in length.

Acknowledgments

This work was funded under the MITRE Innovation Program. Approved for Public Release; Distribution Unlimited: 13-1876.

References

Austin Appleby. 2011. MurmurHash, murmur3. `https://sites.google.com/site/murmurhash/`.

Shlomo Argamon, Moshe Koppel, James W. Pennebaker, and Jonathan Schler. 2007. Mining the blogosphere: Age, gender, and the varieties of self-expression. *First Monday*, 12(9), September.

Rank	L1	Score	Feature	Snippet
1	KOR	57.34	orea	first thing that Korean college students usually buy
2	GER	48.68	,_tha	the fact , that people have less moral values
3	SPA	23.65	omen	consequences related with the environment and the atmosphere
4	ARA	23.23	_alot	becouse you have alot of knowledge
6	TUR	22.84	s_abo	their searchings about the products
11	ITA	21.56	Ital	the Italian scholastic system
19	TEL	20.19	d_als	the whole system and also the concept
20	TUR	19.96	urk	in Turkey all young people go to the parties
21	CHI	19.51	Ta	Take school teachers for example
23	GER	19.34	_-_	constantly - or as mentioned before even exponentially - breaking
27	JPN	17.62	s_,_I	For those reasons , I think
32	FRE	16.90	ndeed	Indeed , facts are just applications of ideas
36	JPN	16.57	apan	been getting weaker these days in Japan .
37	FRE	16.57	onn	I personnaly prefer
38	GER	16.04	,_bec	would be great , because so everyone
41	SPA	15.92	esa	its not necesary to ask
47	HIN	15.23	in_i	the main idea and concept
53	ITA	14.93	act_	due to the fact that too much
74	ITA	13.00	,_in	academic subjects and , in the mean time
81	TEL	12.74	h_ou	cannot do with out a tour guide

Figure 7: Character n-gram features predicting particular L1.

Rank	L1	Score	Feature	Snippet
35	CHI	16.58	(PRP,VB)	What if he go and see
43	CHI	15.85	(NNS,POS)	products 's
45	SPA	15.41	(NNS,NNS)	companies universities
59	TEL	14.05	(RB,IN,VBG)	Usually in schooling
64	TEL	13.95	(DT,NNS,WDT)	the topics which
65	TUR	13.71	(IN,DT,IN)	after a while
66	TEL	13.69	(IN,VBG)	in telling
69	TUR	13.42	(VBG,DT,NNS)	learning the ways
70	HIN	13.39	(IN,VBN,TO)	as compared to
80	HIN	12.81	(FW)	[foreign word]

Figure 8: Part of Speech n-gram features predicting particular L1.

Douglas Biber and Edward Finegan, editors. 1993. *Sociolinguistic Perspectives on Register*. Oxford studies in sociolinguistics. Oxford University Press.

Daniel Blanchard, Joel Tetreault, Derrick Higgins, Aoife Cahill, and Martin Chodorow. to appear. TOEFL11: A Corpus of Non-Native English. Technical report, Educational Testing Service.

Julian Brooke and Graeme Hirst. 2012. Robust, Lexicalized Native Language Identification. In *Proceedings of COLING 2012*, pages 391–408, Mumbai, India, December.

John D. Burger and John C. Henderson. 2006. An exploration of observable features related to blogger age. In *Computational Approaches to Analyzing Weblogs: Papers from the 2006 AAAI Spring Symposium*. AAAI Press.

John D. Burger, John Henderson, George Kim, and Guido Zarrella. 2011. Discriminating gender on twitter. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1301–1309, Edinburgh, Scotland, UK, July. Association for Computational Linguistics.

Vitor R. Carvalho and William W. Cohen. 2006. Single-pass online learning: performance, voting schemes and online feature selection. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 548–553, New York, NY, USA. ACM.

Hal Daumé and D Marcu. 2007. Frustratingly easy domain adaptation. In *Proceedings of the Association for Computational Linguistics*, volume 45, page 256.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

John Gibson, Ben Wellner, and Susan Lubar. 2007. Adaptive web-page content identification. In *Proceedings of the 9th Annual ACM International Workshop on Web information and Data Management*, WIDM '07, pages 105–112, New York, NY, USA. ACM.

William Labov. 1972. *Sociolinguistic Patterns*. Conduct & Communication Series. University of Pennsylvania Press.

Lillian Lee. 1999. Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, ACL '99, pages 25–32, Stroudsburg, PA, USA. Association for Computational Linguistics.

Arjun Mukherjee and Bing Liu. 2010. Improving gender classification of blog authors. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, Cambridge, MA, October. Association for Computational Linguistics.

Delip Rao, David Yarowsky, Abhishek Shreevats, and Manaswi Gupta. 2010. Classifying latent user attributes in Twitter. In *2nd International Workshop on Search and Mining User-Generated Content*. ACM.

Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James Pennebaker. 2006. Effects of age and gender on blogging. In *Computational Approaches to Analyzing Weblogs: Papers from the 2006 AAAI Spring Symposium*. AAAI Press, March.

Andreas Stolcke. 2002. SRILM—an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, volume 2, pages 901–904.

Joel Tetreault, Daniel Blanchard, Aoife Cahill, and Martin Chodorow. 2012. Native tongues, lost and found: Resources and empirical evaluations in native language identification. In *Proceedings of COLING 2012*, pages 2585–2602, Mumbai, India, December. The COLING 2012 Organizing Committee.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of HLT-NAACL*, pages 252–259.