Cognate and Misspelling Features for Natural Language Identification

Garrett Nicolai, Bradley Hauer, Mohammad Salameh, Lei Yao, Grzegorz Kondrak
Department of Computing Science
University of Alberta
Edmonton, AB, Canada
{nicolai,bmhauer,msalameh,lyao1,gkondrak}@ualberta.ca

Abstract

We apply Support Vector Machines to differentiate between 11 native languages in the 2013 Native Language Identification Shared Task. We expand a set of common language identification features to include cognate interference and spelling mistakes. Our best results are obtained with a classifier which includes both the cognate and the misspelling features, as well as word unigrams, word bigrams, character bigrams, and syntax production rules.

1 Introduction

As the world becomes more inter-connected, an increasing number of people devote effort to learning one of the languages that are dominant in the global community. English, in particular, is studied in many countries across the globe. The goal is often related to increasing one's chances to obtain employment and succeed professionally. The language of work-place communication is often not a speaker's native language (L1) but their second language (L2). Speakers and writers of the same L1 can sometimes be identified by similar L2 errors. The weak Contrastive Analysis Hypothesis (Jarvis and Crossley, 2012) suggests that these errors may be a result of L1 causing linguistic interference; that is, common tendencies of a speaker's L1 are superimposed onto their L2. Native Language Identification, or NLI, is an attempt to exploit these errors in order to identify the L1 of the speaker from texts written in L2.

Our group at the University of Alberta was unfamiliar with the NLI research prior to the announcement of a shared task (Tetreault et al., 2013). However, we saw it as an opportunity to apply our expertise in character-level NLP to a new task. Our goal was to propose novel features, and to combine them with other features that have been previously shown to work well for language identification.

In the end, we managed to define two feature sets that are based on spelling errors made by L2 writers. Cognate features relate a spelling mistake to cognate interference with the writer's L1. Misspelling features identify common mistakes that may be indicative of the writer's L1. Both feature sets are meant to exploit the Contrastive Analysis Hypothesis, and benefit from the writer's L1 influence on their L2 writing.

2 Related Work

Koppel et al. (2005b) approach the NLI task using Support Vector Machines (SVMs). They experiment with features such as function-word unigrams, rare part-of-speech bigrams, character bigrams, and spelling and syntax errors. They report 80% accuracy across 5 languages. We further investigate the role of word unigrams and spelling errors in native language identification. We consider not only function words, but also content words, as well as word bigrams. We also process spell-checking errors with a text aligner to find common spelling errors among writers with the same L1.

Tsur and Rappoport (2007) also use SVMs on the NLI task, but limit their feature set to character bigrams. They report 65% accuracy on 5 languages, and hypothesize that the choice of words when writing in L2 is strongly affected by the phonology of

their L1. We also consider character bigrams in our feature set, but combine them with a number of other features.

Wong and Dras (2011) opt for a maximum entropy classifier, and focus more on syntax errors than lexical errors. They find that syntax tree production rules help their classifier in a seven language classification task. They only consider non-lexicalized rules, and rules with function words. In contrast, we consider both lexicalized and non-lexicalized production rules, and we include content words.

Bergsma et al. (2012) consider the NLI task as a sub-task of the authorship attribution task. They focus on the following three questions: (1) whether the native language of the writer of a paper is English, (2) what is the gender of the writer, and (3) whether a paper is a conference or workshop paper. The authors conclude that syntax aids the native language classification task, further motivating our decision to use part-of-speech n-grams and production rules as features for our classifier. Furthermore, the authors suggest normalizing text to reduce sparsity, and implement several meta-features that they claim aid the classification.

3 Classifier

Following Koppel et al. (2005b) and others, we perform classification with SVMs. We chose the SVM-Multiclass package, a version of the SVM-light package(Joachims, 1999) specifically modified for multi-class classification problems. We use a linear kernel, and two hyperparameters that were tuned on the development set: the c soft-margin regularization parameter, which measures the tradeoff between training error and the size of the margin, and ϵ, which is used as a stopping criterion for the SVM. C was tuned to a value of 5000, and epsilon to a value of 0.1.

4 Features

As features for our SVM, we used a combination of features common in the literature and new features developed specifically for this task. The features are listed in the following section.

4.1 Word n-grams

Following previous work, we use word n-grams as the primary feature set. We normalize the text before selecting n-grams using the method of Bergsma et al. (2012). In particular, all digits are replaced with a representative '0' character; for example, '22' and '97' are both represented as '00'. However, unlike Koppel et al. (2005b), we incorporate word bigrams in addition to word unigrams, and utilize both function words and content words.

4.1.1 Function Words

Using a list of 295 common function words, we reduce each document to a vector of values representing their presence or absence in a document. All other tokens in the document are ignored. When constructing vectors of bigrams, any word that is not on the list of function words is converted to a placeholder token. Thus, most of our function-word bigrams consist of a single function word preceded or followed by a placeholder token.

4.1.2 Content Words

Other than the normalization mentioned in Section 4.1, all tokens in the documents are allowed as possible word unigrams. No spelling correction is used for reducing the number of word n-grams. Furthermore, we consider all token unigrams that occur in the training data, regardless of their frequency.

An early concern with token bigrams was that they were both large in number, and sparse. In an attempt to reduce the number of bigrams, we conducted experiments on the development set with different numbers of bigrams that exhibited the highest information gain. It was found that using all combinations of word bigrams improved predictive accuracy the most, and did not lead to a significant cost to the SVM. Thus, for experiments on the test set, all token bigrams that were encountered in the training set were used as features.

4.2 Character n-grams

Following Tetreault et al. (2012), we utilize all character bigrams that occur in the training data, rather than only the most frequent ones. However, where the literature uses either binary indicators or relative frequency of bigrams as features, we use a modified form of the relative frequency in our classifier.

In a pre-processing step, we calculate the average frequency of each character bigram across all training documents. Then, during feature extraction, we again determine the relative frequency of each character bigram across documents. We then use binary features to indicate if the frequency of a bigram is higher than the average frequency. Experiments conducted on the development set showed that although this modified frequency was out-performed by the original relative frequency on its own, our method performed better when further features were incorporated into the classifier.

4.3 Part-of-speech n-grams

All documents are tagged with POS tags using the Stanford parser (Klein and Manning, 2003), From the documents in the training data, a list of all POS bigrams was generated, and documents were represented by binary indicators of the presence or absence of a bigram in the document. As with character bigrams, we did not simply use the most common bigrams, but rather considered all bigrams that appeared in the training data.

4.4 Syntax Production Rules

After generating syntactic parse trees with the Stanford Parser. we extract all possible production rules from each document, including lexicalized rules. The features are binary; if a production rule occurs in an essay, its value is set to 1, and 0 otherwise. For each language, we use information gain for feature selection to select the most informative production rules as suggested by Wong and Dras (2011). Experiments on the development set indicated that the information gain is superior to raw frequency for the purpose of syntax feature selection. Since the accuracy increased as we added more production rules, the feature set for final testing includes all production rules encountered in the training set. The majority of the rules are of the form *POS* $\Rightarrow$ *terminal*. We hypothesized that most of the information contained in these rules may be already captured by the word unigram features. However, experiments on the development set suggested that the lexicalized rules contain information that is not captured by the unigrams, as they led to an increase in predictive accuracy.

4.5 Spelling Errors

Koppel et al. (2005a) suggested spelling errors could be helpful as writers might be affected by the spelling convention in their native languages. Moreover, spelling errors also reflect the pronunciation characteristics of the writers' native languages. They identified 8 types of spelling errors and collected the statistics of each error type as their features. Unlike their approach, we focus on the specific spelling errors made by the writers because 8 types may be insufficient to distinguish the spelling characteristics of writers from 11 different languages. We extract the spelling error features from character-level alignments between the misspelled word and the intended word. For example, if the word *abstract* is identified as the intended spelling of a misspelling *abustruct*, the character alignments are as follows:

a	bu	s	t	ru	ct
|	|	|	|	|	|
a	b	s	t	ra	ct

Only the alignments of the misspelled parts, i.e. *(bu,b)* and *(ru,ra)* in this case, are used as features. The spell-checker we use is *aspe*ll[1], and the character-level alignments are generated by *m2m-aligner* (Jiampojamarn et al., 2007).

4.6 Cognate Interference

Cognates are words that share their linguistic origin. For example, English *become* and German *bekommen* have evolved from the same word in a common ancestor language. Other cognates are words that have been transfered between languages; for example, English *system* comes from the Greek word $\sigma\upsilon\sigma\tau\eta\mu\alpha$ via Latin and French. On average, pairs of cognates exhibit higher orthographic similarity than unrelated translation pairs (Kondrak, 2013).

Cognate interference may cause an L1-speaker to use a cognate word instead of a correct English translation (for example, *become* instead of *get*). Another instance of cognate interference is misspelling of an English word under the influence of the L1 spelling (Table 1).

We aim to detect cognate interference by identifying the cases where the cognate word is closer to

[1] http://aspell.net

Misspelling	Intended	Cognate
developped	*developed*	*developpé (Fre)*
exemple	*example*	*exemple (Fre)*
organisation	*organization*	*organisation (Ger)*
conzentrated	*concentrated*	*konzentrierte (Ger)*
comercial	*commercial*	*comercial (Spa)*
sistem	*system*	*sistema (Spa)*

Table 1: Examples of cognate interference in the data.

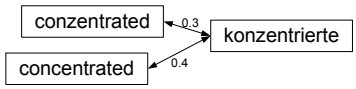

Figure 1: A cognate word influencing the spelling.

the misspelling than to the intended word (Figure 1). We define one feature to represent each language L, for which we could find a downloadable bilingual English-L dictionary. We use the following algorithm:

1. For each misspelled English word m found in a document, identify the most likely intended word e using a spell-checking program.
2. For each language L:
   (a) Look up the translation f of the intended word e in language L.
   (b) Compute the orthographic edit distance D between the words.
   (c) If $D(e, f) < t$ then f is assumed to be a cognate of e.
   (d) If f is a cognate and $D(m, f) < D(e, f)$ then we consider it as a clue that $L = L1$.

We use a simple method of computing orthographic distance with threshold $t = 0.58$ defined as the baseline method by Bergsma and Kondrak (2007). However, more accurate methods of cognate identification discussed in that paper could also be used.

Misspellings can betray cognate interference even if the misspelled word has no direct cognate in language L1. For example, a Spanish speaker might spell the word *quick* as *cuick* because of the existence of numerous cognates such as *question/cuestión*. Our misspelling features can detect such phenomena at the character level; in this case, *qu:cu* corresponds to an individual misspelling feature.

4.7 Meta-features

We included a number of document-specific *meta-features* as suggested by Bergsma et al. (2012): the average number of words per sentence, the average word length, as well as the total number of characters, words, and sentences in a document. We reasoned that writers from certain linguistic backgrounds may prefer many short sentences, while other writers may prefer fewer but longer sentences. Similarly, a particular linguistic background may influence the preference for shorter or longer words.

5 Results

The dataset used for experiments was the TOEFL11 Non-Native English Corpus (Blanchard et al., 2013). The dataset was split into three smaller datasets: the Training set, consisting of 9900 essays evenly distributed across 9 languages, the Development set, which contained a further 1100 essays, and the Test set, which also contained 1100 essays. As the data had a staggered release, we used the data differently. We further split the Training set, with a split of 80% for training, and 10% for development and testing. We then used the Development set as a held-out test set. For held-out testing, the classifier was trained on all data in the Training set, and for final testing, the classifier was trained on all data in both the Training and Development sets.

We used four different combinations of features for our task submissions. The results are shown in Table 2. We include the following accuracy values: (1) the results that we obtained on the Development set before the Test data release, (2) the official Test set results provided by the organizers (Tetreault et al., 2013), (3) the actual Test set results, and (4) the mean cross-validation results (for submissions 1 and 3). The difference between the official and the actual Test set results is attributed to two mistakes in our submissions. In submission 1, the feature lists used for training and testing did not match. In submissions 3 and 4, only non-lexicalized syntax production rules were used, whereas our intention was to use all of them.

No.	Features	Dev	Org	Test	CV
1	Base	82.0	61.2	80.4	58.2
2	– cont. words	67.4	68.7	68.7	–
3	+ char	81.4	80.3	81.7	58.5
4	+ char + meta	81.2	80.0	80.8	–

Table 2: Accuracy of our submissions.

All four submissions used the following base combination of features:

- word unigrams
- word bigrams
- error alignments
- syntax production rules
- word-level cognate interference features

In addition, submission 3 includes character bigrams, while submission 4 includes both character bigrams and meta-features. In submission 2, only function words are used, with the exclusion of content words.

Our best submission, which achieves 81.73% accuracy on the Test set, includes all features discussed in Section 4 except POS bigrams. Early tests indicated that any gains obtained with POS bigrams were absorbed by the production rules, so they were excluded form the final experiments. Character bigrams help on the Test set but not on the Development set. The meta-features decrease accuracy on both sets. Finally, the content words dramatically improve accuracy. The reason we included a submission which did not use content words is that it is a common practice in previous work. In our analysis of the data, we found content words that were highly indicative of the language of the writer. Particularly, words and phrases which contained the speaker's home country were useful in predicting the language. It should be noted that this correspondence may be dependent upon the prompt given to the writer. Furthermore, it may lead to false positives for L1 speakers who live in multi-lingual countries.

5.1 Confusion Matrix

We present the confusion matrix for our best submission in Table 5.1. The highest number of incorrect classifications are between languages that are either linguistically or culturally related (Jarvis and Crossley, 2012). For example, Korean is often misclassified as Japanese or Chinese. The two languages are not linguistically related to Korean, but both have historically had cultural ties with Korean. Likewise, while Hindi and Telugu are not related linguistically, they are both spoken in the same geographic area, and speakers are likely to have contact with each other.

	A	C	F	G	H	I	J	K	S	T	Tu
ARA	83	0	0	0	2	2	2	1	4	5	1
CHI	1	81	2	0	1	0	8	6	1	0	0
FRE	6	0	82	2	1	3	0	0	1	0	5
GER	1	0	0	90	1	1	1	0	2	0	4
HIN	1	2	2	0	76	1	0	0	0	16	2
ITA	1	1	0	1	0	89	1	0	5	1	1
JPN	2	1	1	1	0	1	86	6	0	0	2
KOR	1	8	0	0	0	0	11	78	0	1	1
SPA	2	2	7	0	3	5	0	2	75	0	4
TEL	2	0	0	2	15	0	0	0	1	80	0
TUR	4	3	2	1	0	1	1	5	2	2	79

Table 3: Confusion Matrix for our best classifier.

Features	Test
Full system	81.7
w/o error alignments	81.3
w/o word unigrams	81.1
w/o cognate features	81.0
w/o production rules	80.6
w/o character bigrams	80.4
w/o word bigrams	76.7

Table 4: Accuracy of various feature combinations.

5.2 Ablation Study

Table 4 shows the results of an ablation experiment on our best-performing submission. The word bigrams contribute the most to the classification; their removal increases the relative error rate by 27%. The word unigrams contribute much less., This is unsurprising, as much of the information contained in the word unigrams is also contained in the bigrams. The remaining features are also useful. In particular, our cognate interference features, despite applying to only 4 of 11 languages, reduce errors by about 4%.

6 Conclusions and Future Work

We have described the system that we have developed for the NLI 2013 Shared Task. The system combines features that are prevalent in the literature with our own novel character-level spelling features and word cognate interference features. Most of the features that we experimented with appear to increase the overall accuracy, which contradicts the view that simple bag-of-words usually perform better than more complex feature sets (Sebastiani, 2002).

Our cognate features can be expanded by including languages that do not use the Latin script, such as Russian and Greek, as demonstrated by Bergsma and Kondrak (2007). We utilized bilingual dictionaries representing only four of the eleven languages in this task[2]; yet our cognate interference features still improved classifier accuracy. With more resources and with better methods of cognate identification, the cognate features have the potential to further contribute to native language identification.

Our error-alignment features can likewise be further investigated in the future. Currently, after analyzing texts with a spell-checker, we automatically accept the first suggestion as the correct one. In many cases, this leads to faulty corrections, and misleading alignments. By using context sensitive spell-checking, we can choose better corrections, and obtain information which improves classification.

This shared task was a wonderful introduction to Native Language Identification, and an excellent learning experience for members of our group,

References

Shane Bergsma and Grzegorz Kondrak. 2007. Alignment-based discriminative string similarity. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 656–663.

Shane Bergsma, Matt Post, and David Yarowsky. 2012. Stylometric analysis of scientific articles. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 327–337, Montréal, Canada.

Daniel Blanchard, Joel Tetreault, Derrick Higgins, Aoife Cahill, and Martin Chodorow. 2013. TOEFL11: A Corpus of Non-Native English. Technical report, Educational Testing Service.

Scott Jarvis and Scott Crossley, editors. 2012. *Approaching Language Transfer Through Text Classification: Explorations in the Detection-based Approach*, volume 64. Multilingual Matters Limited, Bristol, UK.

Sittichai Jiampojamarn, Grzegorz Kondrak, and Tarek Sherif. 2007. Applying many-to-many alignments and HMMs to letter-to-phoneme conversion. In *Proceedings of NAACL-HLT*, pages 372–379.

Thorsten Joachims. 1999. Making large-scale support vector machine learning practical. In *Advances in kernel methods*, pages 169–184. MIT Press.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430.

Grzegorz Kondrak. 2013. Word similarity, cognation, and translational equivalence. To appear.

Moshe Koppel, Jonathan Schler, and Kfir Zigdon. 2005a. Automatically determining an anonymous author's native language. *Intelligence and Security Informatics*, pages 41–76.

Moshe Koppel, Jonathan Schler, and Kfir Zigdon. 2005b. Determining an author's native language by mining a text for errors. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 624–628, Chicago, IL. ACM.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.

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.

Joel Tetreault, Daniel Blanchard, and Aoife Cahill. 2013. A report on the first native language identification shared task. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, Atlanta, GA, USA.

Oren Tsur and Ari Rappoport. 2007. Using classifier features for studying the effect of native language on the choice of written second language words. In *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*, pages 9–16, Prague, Czech Republic.

Sze-Meng Jojo Wong and Mark Dras. 2011. Exploiting parse structures for native language identification. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1600–1610, Edinburgh, Scotland, UK.

[2]French, Spanish, German, and Italian.