VTEX System Description for the NLI 2013 Shared Task

Vidas Daudaravičius
VTEX
Akademijos 2a
Vilnius, Lithuania
vidas.daudaravicius@vtex.lt

Abstract

This paper describes the system developed for the NLI 2013 Shared Task, requiring to identify a writer's native language by some text written in English. I explore the given manually annotated data using word features such as the length, endings and character trigrams. Furthermore, I employ k-NN classification. Modified TFIDF is used to generate a stop-word list automatically. The distance between two documents is calculated combining n-grams of word lengths and endings, and character trigrams.

1 Introduction

Native Language Identification (NLI) is the task of identifying the first spoken language (L1) of a person based on the person's written text in another language. As a natural language processing (NLP) task, it is properly categorized as text classification, and standard approaches like support vector machines (SVM) are successufully applied to it. Koppel et al. (2005) trained SVM models with a set of stylistic features, including Part of Speech (POS) and character n-grams (sequences), function words, and spelling error types, achieving 80% accuracy in a 5-language task. Tsur and Rappoport (2007) focused on character n-grams. Wong and Dras (2011) showed that syntactic patterns, derived by a parser, are more effective than other stylistic features. The Cambridge Learner Corpus has been used recently by Kochmar (2011), who concluded that character n-grams are the most promising features. Brooke and Hirst (2012) investigated function words, character n-grams, POS n-grams, POS/function n-grams, CFG productions, dependencies, word n-grams.

A notable problem in the recent NLI research is a clear interaction between native languages and topics in the corpora. The solution in the mentioned work was to avoid lexical features that might carry topical information.

2 Data

The NLI 2013 Shared Task uses the TOEFL11 corpus (Blanchard et al., 2013) which was designed specifically for the task of native language identification. The corpus contains 12 100 English essays from the TOEFL (**T**est of **E**nglish as a **F**oreign **L**anguage) that were collected through ETS (Educational Testing Service) operational test delivery system. TOEFL11 contains eleven native languages: Arabic, Chinese, French, German, Hindi, Italian, Japanese, Korean, Spanish, Telugu, and Turkish. The sampling of essays ensures approximately equal representation of native languages across eight topics, labeled as prompts. The corpus contains more than 1000 essays for each L1 language. Each essay is labelled with an English language proficiency level – high, medium, or low – given by human assessment specialists. The essays are usually 300 to 400 words long. The corpus is split into training, development and test data (9900, 1100 and 1100, respectively). The corpus contains plain text files and the index for these

Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications, pages 89–95,
Atlanta, Georgia, June 13 2013.

File name	Prompt	Native language	Language proficiency
1000025.txt	P2	CHI	high
100021.txt	P1	ARA	low
1000235.txt	P8	TEL	medium
1000276.txt	P4	TEL	high
1000392.txt	P3	JPN	medium
1000599.txt	P6	CHI	medium
1000617.txt	P4	GER	high
1000719.txt	P1	HIN	high
100082.txt	P2	TUR	medium

Table 1: The sample of the training data index.

files. Sample of this index is shown in Table 1.

3 *N*end transformation

The training and the development corpora contain a lot of spelling errors and no POS tagging is provided. For instance, a sentence from the training corpus *"Acachely I write abawet may communitie and who the people support youg people"*. Therefore I needed to find features which encode the information about native language of a writer in a more generalized way. Also, my primary interest was to build a system which does not utilize any language processing tool, such as part of speech or syntactic trees, and topic-related information, such as full words. The reason for that is to have the possibility to apply the same techniques for the texts written in other languages than English in the future. Thus, I choose to use the word length as the number of characters together with the last n characters of that word. Words in the essays were transformed into tokens using five kinds of transformations:

0end – takes the pure length of a word (for example, *make* $\mapsto$ *4*);

1end – adds to the length of a word the last character (*make* $\mapsto$ *4e*);

2end – adds to the length of a word the last two characters (*make* $\mapsto$ *4ke*);

3end – adds to the length of a word the last three characters (*make* $\mapsto$ *4ake*);

4end – adds to the length of a word the last four characters (*make* $\mapsto$ *4make*).

For instance, the sentence "*Difference makes a lot of opportunities .*" is translated to:

0end:	10	5	1	3	2	13	1
1end:	10e	5s	1a	3t	2f	13s	1.
2end:	10ce	5es	1a	3ot	2of	13es	1.
3end:	10nce	5kes	1a	3lot	2of	13ies	1.
4end:	10ence	5akes	1a	3lot	2of	13ties	1.

4 *N*-gram features

The VTEX NLI 2013 system is based on n-gram features. There are no strict rules for how long n-grams should be. Frequently used n-grams are unigrams, bigrams and trigrams as in Brooke and Hirst (2012; Wong and Dras (2011). The training NLI 2013 corpus is large enough to build higher-order n-grams of $n$end tokens. I use unigrams, bigrams, trigrams, quad-grams and five-grams based on $n$end tokens. Some examples of these n-grams are shown below:

0end					
1-gram:	3				
2-gram:	1	3			
3-gram:	1	10	6		
4-gram:	1	5	3	3	
5-gram:	1	3	3	3	7
3end					
1-gram:	7ess				
2-gram:	2to	7ess			
3-gram:	4est	2to	7ess		
4-gram:	3but	3not	3for	7ess	
5-gram:	3try	5eir	4est	2to	7ess

Beside n-grams of $n$ends, the character n-grams are of interest also. Kochmar (2011) noted that character n-grams provide promissing features for NLI task. Therefore, I tried to use character trigrams also. For instance, from the sentence "*Difference makes a lot of opportunities .*" the following trigrams were generated:

Dif iff ffe fer ere ren enc nce ce_ e_m _ma mak ake kes es_ s_a _a a_l _lo lot ot_ t_o _of of_ f_o _op opp ppo por ort rtu tun uni nit iti tie ies es_ s_.

Whitespace is included in character trigrams and denotes the beginning or the end of a word.

5 CTFIDF for weigthing features

The most widely used technique for weighting items in a list is Term-Frequency–Inverse-Document-Frequency, known as TF–IDF. Daudaravicius (2012) shows that the small change of TF–IDF allows to the generation of stop-word lists automatically. For the NLI 2013 Shared Task I use *Conditional TF–IDF*:

$$\mathrm{CTFIDF}(x) = \mathrm{TF}(x) \cdot \ln \frac{\mathrm{D}_{\max} - \mathrm{d}(x) + 1}{4 \cdot \mathrm{d}(x) + 1},$$

where $\mathrm{TF}(x)$ is the frequency of the item x in the training corpus, $\mathrm{d}(x)$ is the number of documents in the training corpus where the item x appears, known as *document frequency*, $\mathrm{D}_{\max}$ is the maximum of document frequency of any item in the training corpus.

The idea of my Conditional TF–IDF is as follows: if a term occures in less than $\mathrm{D}_{\max}/4$ documents then this term is considered a normal term, and the term is considered as *stop-word* if it occures in more than $\mathrm{D}_{\max}/4$ documents. The range of TF-IDF is between 0 and positive infinity. The range of CTFIDF is from minus infinity to zero for items that are considered stop-words. And the range of CTFIDF is from zero to infinity for the rest of the items.

For instance, the $\mathrm{D}_{\max}$ for the different n-gram length and different Nend transformations is presented in Table 2. The example list of 4end ungrams with positive and negative CTFIDFs are shown in Tables 4 and 3, respectively.

It is important to note that I count $\mathrm{D}_{\max}$ and $\mathrm{d}(x)$ for each training language separately; i.e., when I measure the distance between a document and the document in the training data, I use $\mathrm{D}_{\max}$ and $\mathrm{d}(x)$ of the language which the training document denotes.

	The number of n-grams				
	1	**2**	**3**	**4**	**5**
0end	900	899	834	444	168
1end	900	759	358	320	148
2end	899	581	354	319	148
3end	899	572	320	303	148
4end	899	572	320	303	148

Table 2: The maximum of the document frequency in the training corpus.

token	**ctfidf**	**token**	**ctfidf**	**token**	**ctfidf**
5earn	0.00	4Most	1.16	10ents	2.51
7ally	0.04	7lity	1.20	4your	2.59
10sion	0.10	2Of	1.22	7arly	2.59
7ieve	0.10	6ance	1.22	6eple	2.64
5hing	0.12	6mous	1.22	7tory	2.71
10ence	0.12	5hier	1.24	8tics	2.94
9tion	0.15	3Now	1.25	9gers	3.00
2us	0.22	5eing	1.27	4cool	3.07
6rson	0.23	12tion	1.30	3Let	3.13
7hout	0.29	2He	1.30	4rule	3.29
3may	0.30	4ways	1.41	5imes	3.52
3say	0.31	6hers	1.43	3job	3.53
3see	0.34	5reat	1.45	13ties	3.60
3try	0.35	9rent	1.53	8cial	3.68
3did	0.36	3him	1.55	5eals	3.81
2”	0.42	5ower	1.61	6lent	3.81
2“	0.44	12ties	1.65	4lose	3.95
2he	0.46	3You	1.68	8naly	4.13
4hard	0.52	11lity	1.74	6skes	4.34
7pany	0.58	4cost	1.76	7cted	4.34
5akes	0.60	5ince	1.78	7test	4.34
4kind	0.68	6ills	1.82	6alth	4.36
7blem	0.70	5isks	1.82	5eall	4.60
5ever	0.71	5oney	1.89	9dent	4.73
4been	0.74	6rget	2.07	7cess	4.75
4same	0.81	5ired	2.10	7kers	5.36
8king	0.86	9nies	2.11	9ters	5.46
6king	0.93	4ever	2.15	2D.	5.52
5ften	0.96	6ates	2.15	5neof	5.52
6urse	0.97	3his	2.22	8idnt	5.52
7ling	0.97	10ered	2.24	8klin	5.52
4Even	0.98	4love	2.24	9velt	5.52
8ible	0.99	6ited	2.24	10sful	6.62
4used	1.02	9ties	2.27	4four	7.62
10tely	1.07	4earn	2.30	3oil	8.05
4best	1.09	6llow	2.30	9cans	8.26
7ught	1.10	9ated	2.37	4jobs	8.96
4easy	1.12	3got	2.42	3FDR	11.04
4Then	1.12	8ngly	1.13		

Table 3: The list of 4end unigrams with positive CTFIDFs of one document from the training corpus.

token	ctfidf	token	ctfidf	token	ctfidf
1.	-224.19	3but	-3.48	3lot	-0.92
1,	-127.63	5bout	-2.58	2we	-0.88
2to	-69.62	3get	-2.57	5hich	-0.85
2of	-56.92	7mple	-2.54	9ment	-0.84
3the	-45.09	2by	-2.39	3who	-0.84
3and	-27.25	4from	-2.26	3The	-0.81
2is	-24.79	4they	-2.18	4them	-0.79
1a	-23.19	3can	-2.12	3one	-0.77
6ople	-22.78	4will	-2.11	4only	-0.75
3not	-22.31	3all	-1.83	4much	-0.70
3are	-18.11	2If	-1.72	4what	-0.68
3for	-15.82	2at	-1.63	4also	-0.64
4that	-14.39	2In	-1.50	4want	-0.57
2do	-13.16	6ings	-1.38	6cond	-0.56
2it	-12.50	5irst	-1.35	9tant	-0.43
4have	-11.53	3For	-1.33	3how	-0.35
4with	-9.39	5gree	-1.33	3new	-0.31
1I	-8.72	3you	-1.31	6ould	-0.31
7ause	-7.73	2so	-1.30	4need	-0.20
2in	-6.40	4time	-1.15	5oing	-0.15
5heir	-6.23	3was	-1.08	4take	-0.11
2be	-5.44	7ever	-0.98	2So	-0.10
4many	-5.40	5ther	-0.95	6ally	-0.09
2as	-5.06	4make	-0.93	3But	-0.08
5here	-3.92	5hink	-3.64		

Table 4: The list of 4end unigrams with negative CTFIDFs of the same document as in Fig. 3.

6 Distance between documents

Cosine distance is a widely used technique to measure the distance between two feature vectors. It is calculated as follows:

$$\cos(X,Y) = \frac{\sum_i (X_i Y_i)}{\sqrt{\sum_i X_i^2} + \sqrt{\sum_i Y_i^2}}.$$

CTFIDF allows the splitting of feature vectors into the list of "informative" items and the list of functional items. For the NLI 2013 Shared task, I combine two cosine distances of negative and positive CTFIDFs as follows:

$$\cos'(X,Y) = \frac{2\cos(X',Y') + \cos(X'',Y'')}{3},$$

where

$$X' = \text{filter}_{\geq 0}\, X, \qquad Y' = \text{filter}_{\geq 0}\, Y,$$
$$X'' = \text{abs}(\text{filter}_{<0}\, X), \quad Y'' = \text{abs}(\text{filter}_{<0}\, Y),$$

so X' and Y' contain features with positive CTFIDF, while X'' and Y'' contain features with negative CTFIDF.

The $\cos'$ combines two cosine distances giving the weight for cosine of positive CTFIDFs equal to 2 and for the negative CTFIDFs equal to 1. I have also tested combinations of 1 to 0, 0 to 1, 1 to 1, and 1 to 2. But these combinations did not achieve better results. Therefore, for all submitted system results I used the same combination of 2 to 1.

I utilize 26 feature vectors and obtain 26 combined cosine distances for each document: one for character trigrams and other 25 for token n-grams of diverse word transformations. Each combined cosine distance has an assigned weight to get the final distance between two documents. The distance between two documents X and Y is calculated as follows:

$$\text{dist}(X,Y) = \frac{\sum_i w_i \cos'(X_i, Y_i)}{\sum_i w_i} \in [0,1],$$

where w_i is the weight of ith feature vector.

The most difficult task was to find the best combination of these 26 weights. For the NLI 2013 Shared Task I have used the combinations shown in Table 5. The n-gram weights in most cases are diagonal with the highest value at the 0end unigram and the lowest at the 4end five-gram. In the beggining I tested the opposite combination, but this led to worse results. Also, the influence of character trigrams on the results was high. The first and second combinations in Table 5 differ in the use of five-grams and 4end transformations, while the leverage of character trigrams were kept the same. The final official results show that richer features improve results. Also, I found that the higher leverage is for character trigrams over n-grams the better the results are. But, the results of character trigrams only resulted in lower performance. It is a long way to find the optimal combination of the weights.

	Token n-gram				
	1	**2**	**3**	**4**	**5**
1-closed					
Character trigrams					64
0end	7	6	5	4	0
1end	6	5	4	3	0
2end	5	4	3	2	0
3end	4	3	2	1	0
4end	0	0	0	0	0
2-closed					
Character trigrams					125
0end	9	8	7	6	5
1end	8	7	6	5	4
2end	7	6	5	4	3
3end	6	5	4	3	2
4end	5	4	3	2	1
3-closed					
Character trigrams					25
0end	1	1	1	1	1
1end	1	1	1	1	1
2end	1	1	1	1	1
3end	1	1	1	1	1
4end	1	1	1	1	1
4-closed					
Character trigrams					225
0end	17	15	13	11	9
1end	15	13	11	9	7
2end	13	11	9	7	5
3end	11	9	7	5	3
4end	9	7	5	3	1
5-closed					
Character trigrams					550
0end	17	15	13	11	9
1end	15	13	11	9	7
2end	13	11	9	7	5
3end	11	9	7	5	3
4end	9	7	5	3	1

Table 5: Weights of the NLI 2013 different submissions.

7 Assigning native language to a text

I used the k-NN technique to assign native language to a text. I counted the distances between the test document and all training documents, and take some amount of closest documents for each language. To reduce the influnce of outliers, I dropped off the n closest documents and only then take some amount from the rest. At first, I remove the 10 top documents from each language, and then kept the 20 closest documents for each language. In total, I obtained 220 documents and ranked them by distance. Then, I employed voting for the closest 20 documents. A winner language is assigned to a document as the native language. This technique was used for VTEX-closed-(1, 2 and 3) system submitions. For the VTEX-closed-(4 and 5) I used another number for outliers and the top closest ones: the 50 closest documents for each language were dropped off, the remianing 25 for each language were kept, and, finally, the closest 25 documents are used for the voting of native language.

8 Results

My primary interest in participating in the NLI 2013 Shared Task was to investigate new features that were not used earlier, and what the value of each feature in the identification of a writer's native language is. The results of five submitted systems are shown in Tables 6 and 7. The best submitted system had 31.9 percent accuracy. This result was the worst of all participating teams. At the time of writing this report, I tested new combinations of outliers and tops, "stop-words" and significant items, $n$end n-grams and character trigram weights. New settings improved my best submitted system accuracy from 31.9 to 63.9 percent. This result was achieved with the following settings. I took the last 50 percent of closest documents for each language. I set to use only stop-words and to exclude significant items, i.e., items with only negative CTFIDF. Finaly, I set n-gram weights accordingly: 84 for character trigrams, and for $n$end 1,1,1,1,1, 1,3,3,3,1, 1,3,5,3,1, 1,3,3,3,1, 1,1,1,1,1. This result shows that 2end and 3end transformation trigrams have the highest impact on the results. Nevertheless, all tested transformations help to improve the results. In conclusion, I investigated the influence of features, such as character trigrams and Nend n-grams, to the identification of writer's native language and found them very informative.

Results for VTEX-closed-1

	ARA	CHI	FRE	GER	HIN	ITA	JPN	KOR	SPA	TEL	TUR	Precision	Recall	F-measure
ARA	**30**	5	2	5	5	11	12	6	10	13	1	26.3%	30.0%	28.0%
CHI	4	**20**	2	5	5	6	21	20	5	9	3	24.1%	20.0%	21.9%
FRE	6	8	**9**	13	3	14	14	9	8	10	6	28.1%	9.0%	13.6%
GER	6	4	5	**30**	7	13	4	1	7	20	3	35.3%	30.0%	32.4%
HIN	15	5	0	7	**17**	5	6	5	3	31	6	23.0%	17.0%	19.5%
ITA	7	2	4	3	4	**47**	9	3	4	15	2	34.8%	47.0%	40.0%
JPN	4	5	1	4	5	7	**44**	12	4	14	0	25.3%	44.0%	32.1%
KOR	2	8	1	3	2	9	35	**27**	3	9	1	26.0%	27.0%	26.5%
SPA	13	10	4	3	5	15	13	8	**12**	13	4	19.0%	12.0%	14.7%
TEL	13	8	0	1	13	4	2	1	4	**52**	2	26.3%	52.0%	34.9%
TUR	14	8	4	11	8	4	14	12	3	12	**10**	26.3%	10.0%	14.5%

Accuracy = 27.1%

Results for VTEX-closed-2

	ARA	CHI	FRE	GER	HIN	ITA	JPN	KOR	SPA	TEL	TUR	Precision	Recall	F-measure
ARA	**31**	5	1	3	5	11	13	6	8	15	2	26.5%	31.0%	28.6%
CHI	6	**23**	1	4	6	5	21	15	6	10	3	27.7%	23.0%	25.1%
FRE	5	8	**7**	12	7	15	12	10	6	10	8	25.9%	7.0%	11.0%
GER	7	4	4	**28**	9	12	6	1	6	20	3	35.0%	28.0%	31.1%
HIN	13	5	2	6	**17**	4	6	5	4	30	8	20.2%	17.0%	18.5%
ITA	7	2	4	3	4	**47**	9	3	4	15	2	35.1%	47.0%	40.2%
JPN	4	7	0	5	6	7	**36**	16	3	15	1	22.0%	36.0%	27.3%
KOR	3	7	1	3	2	9	34	**26**	4	9	2	25.7%	26.0%	25.9%
SPA	15	7	3	5	6	17	10	7	**10**	15	5	16.4%	10.0%	12.4%
TEL	13	6	1	0	15	2	2	1	6	**52**	2	25.5%	52.0%	34.2%
TUR	13	9	3	11	7	5	15	11	4	13	**9**	20.0%	9.0%	12.4%

Accuracy = 26.0%

Results for VTEX-closed-3

	ARA	CHI	FRE	GER	HIN	ITA	JPN	KOR	SPA	TEL	TUR	Precision	Recall	F-measure
ARA	**27**	6	1	5	6	11	11	7	11	13	2	25.2%	27.0%	26.1%
CHI	6	**22**	2	6	8	2	21	14	5	12	2	27.2%	22.0%	24.3%
FRE	6	8	**6**	12	8	14	15	7	5	10	9	17.1%	6.0%	8.9%
GER	7	4	6	**24**	9	13	1	2	7	22	5	27.3%	24.0%	25.5%
HIN	15	4	2	7	**17**	4	6	3	5	30	7	19.5%	17.0%	18.2%
ITA	7	0	6	3	4	**45**	8	5	4	16	2	34.1%	45.0%	38.8%
JPN	4	9	0	5	6	8	**32**	15	4	16	1	21.2%	32.0%	25.5%
KOR	2	6	1	5	2	9	31	**26**	4	12	2	27.7%	26.0%	26.8%
SPA	15	7	4	6	8	16	7	6	**11**	14	6	15.3%	11.0%	12.8%
TEL	10	6	2	0	13	5	2	1	10	**50**	1	23.9%	50.0%	32.4%
TUR	8	9	5	15	6	5	17	8	6	14	**7**	15.9%	7.0%	9.7%

Accuracy = 24.3%

Table 6: The results for closed-task VTEX systems.

Results for VTEX-closed-4

	ARA	CHI	FRE	GER	HIN	ITA	JPN	KOR	SPA	TEL	TUR	Precision	Recall	F-measure
ARA	**21**	5	1	6	4	14	15	6	14	12	2	30.4%	21.0%	24.9%
CHI	2	**22**	2	5	5	5	24	18	7	7	3	26.2%	22.0%	23.9%
FRE	4	9	**8**	13	3	14	16	9	6	12	6	22.2%	8.0%	11.8%
GER	5	4	8	**25**	8	13	5	2	6	19	5	28.7%	25.0%	26.7%
HIN	7	7	1	7	**15**	5	7	7	4	31	9	22.1%	15.0%	17.9%
ITA	2	3	3	4	2	**48**	12	3	4	16	3	33.8%	48.0%	39.7%
JPN	1	5	1	5	4	8	**42**	17	4	13	0	21.8%	42.0%	28.7%
KOR	1	6	1	2	1	7	36	**33**	2	10	1	30.0%	33.0%	31.4%
SPA	9	11	5	6	4	18	14	5	**10**	14	4	15.9%	10.0%	12.3%
TEL	8	5	3	1	15	5	2	1	4	**53**	3	27.0%	53.0%	35.8%
TUR	9	7	3	13	7	5	20	9	2	9	**16**	30.8%	16.0%	21.1%
Accuracy = 26.6%														

Results for VTEX-closed-5

	ARA	CHI	FRE	GER	HIN	ITA	JPN	KOR	SPA	TEL	TUR	Precision	Recall	F-measure
ARA	**40**	7	0	2	2	14	10	4	7	11	3	33.9%	40.0%	36.7%
CHI	6	**32**	4	0	4	4	21	16	4	8	1	27.8%	32.0%	29.8%
FRE	5	13	**13**	9	2	15	14	8	6	12	3	28.9%	13.0%	17.9%
GER	10	5	8	**22**	2	13	7	3	8	16	6	45.8%	22.0%	29.7%
HIN	12	9	4	5	**11**	5	6	6	4	30	8	28.9%	11.0%	15.9%
ITA	3	5	6	2	1	**54**	7	4	5	11	2	36.5%	54.0%	43.5%
JPN	2	6	0	3	1	8	**48**	16	3	12	1	26.4%	48.0%	34.0%
KOR	1	12	1	0	2	6	29	**39**	2	7	1	35.1%	39.0%	37.0%
SPA	12	9	5	1	3	20	14	5	**16**	12	3	27.1%	16.0%	20.1%
TEL	14	6	0	0	8	5	2	0	3	**59**	3	31.4%	59.0%	41.0%
TUR	13	11	4	4	2	4	24	10	1	10	**17**	35.4%	17.0%	23.0%
Accuracy = 31.9%														

Table 7: The results for closed-task VTEX systems.

References

Blanchard D., Tetreault J. and Cahill A. 2013. Summary Report on the First Shared Task on Native Language Identification. *In Proceedings of the Eighth Workshop on Building Educational Applications Using NLP*, Association for Computational Linguistics, Atlanta, GA, USA

Brooke, J. and Hirst, G. 2012. Robust, Lexicalized Native Language Identification. *In Proceedings of COLING 2012*, Mumbai, India, 391–408.

Daudaravicius, V. 2012. Collocation segmentation for text chunking. *PhD thesis, Vytautas Magnus University.*

Kochmar, E. 2011. Identification of a Writer's Native Language by Error Analysis. *Master's thesis, University of Cambridge.*

Koppel M., Schler J. and Zigdon, K. 2005. Determining an author's native language by mining a text for errors. *In Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining (KDD '05)*, 624-628.

Tsur, O. and Rappoport, A. 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 (CACLA'07)*, 9-16.

Wong, S.J. and Dras, M. 2011. Exploiting parse structures for native language identification. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 1600-1610.