Combining Shallow and Linguistically Motivated Features in Native Language Identification

Serhiy Bykh Sowmya Vajjala Julia Krivanek Detmar Meurers
Seminar für Sprachwissenschaft, Universität Tübingen
{sbykh, sowmya, krivanek, dm}@sfs.uni-tuebingen.de

Abstract

We explore a range of features and ensembles for the task of *Native Language Identification* as part of the *NLI Shared Task* (Tetreault et al., 2013). Starting with recurring word-based n-grams (Bykh and Meurers, 2012), we tested different linguistic abstractions such as part-of-speech, dependencies, and syntactic trees as features for NLI. We also experimented with features encoding morphological properties, the nature of the realizations of particular lemmas, and several measures of complexity developed for proficiency and readability classification (Vajjala and Meurers, 2012). Employing an ensemble classifier incorporating all of our features we achieved an accuracy of 82.2% (rank 5) in the *closed* task and 83.5% (rank 1) in the *open-2* task. In the *open-1* task, the word-based recurring n-grams outperformed the ensemble, yielding 38.5% (rank 2). Overall, across all three tasks, our best accuracy of 83.5% for the standard TOEFL11 test set came in second place.

1 Introduction

Native Language Identification (NLI) tackles the problem of determining the native language of an author based on a text the author has written in a second language. With Tomokiyo and Jones (2001), Jarvis et al. (2004), and Koppel et al. (2005) as first publications on NLI, the research focus in computational linguistics is relatively young. But with over a dozen new publications in the last two years, it is gaining significant momentum.

In Bykh and Meurers (2012), we explored a data-driven approach using recurring n-grams with three levels of abstraction using parts-of-speech (POS). In the present work, we continue exploring the contribution and usefulness of more linguistically motivated features in the context of the NLI Shared Task (Tetreault et al., 2013), where our approach is included under the team name "Tübingen".

2 Corpora used

T11: TOEFL11 (Blanchard et al., 2013) This is the main corpus of the NLI Shared Task 2013. It consists of essays written by English learners with 11 native language (L1) backgrounds (Arabic, Chinese, French, German, Hindi, Italian, Japanese, Korean, Spanish, Telugu, Turkish), and from three different proficiency levels (low, medium, high). Each L1 is represented by a set of 1100 essays (*train*: 900, *dev*: 100, *test*: 100). The labels for the *train* and *dev* sets were given from the start, the labels for the *test* set were provided after the results were submitted.

ICLE: International Corpus of Learner English (Granger et al., 2009) The ICLEv2 corpus consists of 6085 essays written by English learners of 16 different L1 backgrounds. They are at a similar level of English proficiency, namely higher intermediate to advanced and of about the same age. For the cross-corpus tasks we used the essays for the seven L1s in the intersection with T11, i.e., Chinese (982 essays), French (311), German (431), Italian (391), Japanese (366), Spanish (248), and Turkish (276).

FCE: First Certificate in English Corpus (Yannakoudakis et al., 2011) The FCE dataset consists of 1238 scripts produced by learners taking the First Certificate in English exam, assessing English at an

Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications, pages 197–206,
Atlanta, Georgia, June 13 2013. ©2013 Association for Computational Linguistics

upper-intermediate level. For the cross-corpus tasks, we used the essays by learners of the eight L1s in the intersection with T11, i.e., Chinese (66 essays), French (145), German (69), Italian (76), Japanese (81), Korean (84), Spanish (198), and Turkish (73).

BALC: BUiD (British University in Dubai) Arab Learner Corpus (Randall and Groom, 2009) The BALC corpus consists of 1865 English learner texts written by students with an Arabic L1 background from the last year of secondary school and the first year of university. The texts were scored and assigned to six proficiency levels. For the cross-corpus NLI tasks, we used the data from the levels 3–5 amounting to overall 846 texts. We excluded the two lowest and the highest, sixth level based on pretests with the full BALC data.

ICNALE: International Corpus Network of Asian Learners of English (Ishikawa, 2011) The version of the ICNALE corpus we used consists of 5600 essays written by college students in ten countries and areas in Asia as well as by English native speakers. The learner essays are assigned to four proficiency levels following the CEFR guidelines (A2, B1, B2, B2+). For the cross-corpus tasks, we used the essays written by learners from Korea (600 essays) and from Pakistan (400).[1] Without access to a corpus with Hindi as L1, we decided to label the essays written by Pakistani students as Hindi. Most of the languages spoken in Pakistan, including the official language Urdu, belong to the same Indo-Aryan/-Iranian language family as Hindi. Our main focus here was on avoiding overlap with Telugu, the other Indian language in this shared task, which belongs to the Dravidian language family.

TÜTEL-NLI: Tübingen Telugu NLI Corpus We collected 200 English texts written by Telugu native speakers from bilingual (English-Telugu) blogs, literary articles, news and movie review websites.

NT11: NON-TOEFL11 We combined the ICLE, FCE, ICNALE, BALC and TÜTEL-NLI sources discussed above in the NT11 corpus consisting of overall 5843 essays for 11 L1s, as shown in Table 1.

L1	**Corpora**					
	ICLE	FCE	BALC	ICNALE	TÜTEL	#
ARA	-	-	846	-	-	846
CHI	982	66	-	-	-	1048
FRE	311	145	-	-	-	456
GER	431	69	-	-	-	500
HIN	-	-	-	400	-	400
ITA	391	76	-	-	-	467
JPN	366	81	-	-	-	447
KOR	-	84	-	600	-	684
SPA	248	198	-	-	-	446
TEL	-	-	-	-	200	200
TUR	276	73	-	-	-	349
#	3005	792	846	1000	200	**5843**

Table 1: Distribution of essays for the 11 L1s in NT11

3 Features

Recurring word-based n-grams (rc. word ng.) Following, Bykh and Meurers (2012), we used all word-based n-grams occurring in at least two texts of the training set. We focused on recurring unigrams and bigrams, which in our previous work and in T11 testing with the *dev* set worked best. For the larger T11 *train* ∪ NT11 set, recurring n-grams up to length five were best, but for uniformity we only used word-based unigrams and bigrams for all tasks. As in our previous work, we used a binary feature representation encoding the presence or absence of the n-gram in a given essay.

Recurring OCPOS-based n-grams (rc. OCPOS ng.) All OCPOS n-grams occurring in at least two texts of the training set were obtained as described in Bykh and Meurers (2012). OCPOS means that the open class words (nouns, verbs, adjectives and cardinal numbers) are replaced by the corresponding POS tags. For POS tagging we used the OpenNLP toolkit (`http://opennlp.apache.org`).

In Bykh and Meurers (2012), recurring OCPOS n-grams up to length three performed best. However, for T11 we found that including four- and five-grams was beneficial. This confirms our assumption that longer n-grams can be sufficiently common to be useful (Bykh and Meurers, 2012, p. 433). Thus we used the recurring OCPOS n-grams up to length five for the experiments in this paper. We again used a binary feature representation.

[1] We did not include ICNALE data for more L1s to avoid overrepresentation of already well-represented Asian L1s.

Recurring word-based dependencies (rc. word dep.) Extending the perspective on recurring pieces of data to other data types, we explored a new feature: recurring word-based dependencies. A feature of this type consists of a head and all its immediate dependents. The dependencies were obtained using the MATE parser (Bohnet, 2010). The words in each n-tuple are recorded in lowercase and listed in the order in which they occur in the text; heads thus are not singled out in this encoding. For example, the sentence *John gave Mary an interesting book* yields the following two potential features *(john, gave, mary, book)* and *(an, interesting, book)*. As with recurring n-grams we utilized only features occurring in at least two texts of the training set, and we used a binary feature representation.

Recurring function-based dependencies (rc. func. dep.) The recurring function-based dependencies are a variant of the recurring word-based dependencies described above, where each dependent is represented by its grammatical function. The above example sentence thus yields the two features *(sbj, gave, obj, obj)* and *(nmod, nmod, book)*.

Complexity Given that the proficiency level of a learner was shown to play a role in NLI (Tetreault et al., 2012), we implemented all the text complexity features from Vajjala and Meurers (2012), who used measures of learner language complexity from SLA research for readability classification. These features consist of lexical richness and syntactic complexity measures from SLA research (Lu, 2010; 2012) as well as other syntactic parse tree properties and traditionally used readability formulae. The parse trees were built using the Berkeley parser (Petrov and Klein, 2007) and the syntactic complexity measures were estimated using the Tregex package (Levy and Andrew, 2006).

In addition, we included morphological and POS features from the CELEX Lexical Database (Baayen et al., 1995). The morphological properties of words in CELEX include information about the derivational, inflectional and compositional features of the words along with information about their morphological origins and complexity. POS properties of the words in CELEX describe the various attributes of a word depending on its parts of speech. We included all the non-frequency based and non-word-string attributes from the English Morphology Lemma (EML) and English Syntax Lemma (ESL) files of the CELEX database. We also defined Age of Acquisition features based on the psycholinguistic database compiled by Kuperman et al. (2012). Finally, we included the ratios of various POS tags to the total number of words as POS density features, using the POS tags from the Berkeley parser output.

Suffix features The use of different derivational and inflectional suffixes may contain information regarding the L1 – either through L1 transfer, or in terms of what suffixes are taught, e.g., for nominalization. In a very basic approximation of morphological analysis, we used the porter stemmer implementation of MorphAdorner (`http://morphadorner.northwestern.edu`). For each word in a learner text, we removed the stem it identified from the word, and if a suffix remained, we matched it against the Wiktionary list of English suffixes (`http://en.wiktionary.org/wiki/Appendix:Suffixes:English`). For each valid suffix thus identified, we defined a binary feature (suffix, bin.) recording the presence/absence and a feature counting the number of occurrences (suffix, cnt.) in a given learner text.

Stem-suffix features We also wondered whether the subset of morphologically complex unigrams may be more indicative than considering all unigrams as features. As a simple approximation of this idea, we used the stemmer plus suffix-list approach mentioned above and used all words for which a suffix was identified as features, both binary (stemsuffix, bin.) and count-based (stemsuffix, cnt.).

Local trees Based on the syntactic trees assigned by the Berkeley Parser (Petrov and Klein, 2007), we extracted all local trees, i.e., trees of depth one. For example, for the sentence *I have a tree*, the parser output is: *(ROOT (S (NP (PRP I)) (VP (VBP have) (NP (DT a) (NN tree))) (. .)))* for which the local trees are *(S NP VP .), (NP PRP), (NP DT NN), (VP VBP NP), (ROOT S)*. Count-based features are used.

Stanford dependencies Tetreault et al. (2012) explored the utility of basic dependencies as features for NLI. In our approach, we extracted all Stanford dependencies (de Marneffe et al., 2006) using the trees assigned by the Berkeley Parser. We considered lemmatized typed dependencies (type dep. lm.) such as *nsubj(work,human)* and POS tagged ones (type dep. POS) such as *nsubj(VB,NN)* for our features. We used count-based features for those typed dependencies.

Dependency number (dep. num.) We encoded the number of dependents realized by a verb lemma, normalized by this lemma's count. For example, if the lemma *take* occurred ten times in a document, three times with two dependents and seven times with three dependents, we get the features *take:2-dependents = 3/10* and *take:3-dependents = 7/10*.

Dependency variability (dep. var.) These features count possible dependent-POS combinations for a verb lemma, normalized by this verb lemma's count. If in the example above, the lemma *take* occurred three times with two dependents JJ-NN, two times with three dependents JJ-NN-VB, and five times with three dependents NN-NN-VB, we obtain *take:JJ-NN = 3/10*, *take:JJ-NN-VB = 2/10*, and *take:NN-NN-VB = 5/10*.

Dependency POS (dep. POS) These features are derived from the dep. var. features and encode how frequent which kind of category was a dependent for a given verb lemma. Continuing the example above, *take* takes dependents of three different categories: JJ, NN and VB. For each category, we create a feature, the value of which is the category count divided by the number of dependents of the given lemma, normalized by the lemma's count in the document. In the example, we obtain *take:JJ = (1/2 + 1/3)/10*, *take:NN = (1/2 + 1/3 + 2/3)/10*, and *take:VB = (1/3 + 1/3)/10*.

Lemma realization matrix (lm. realiz.) We specified a set of features that is calculated for each distinct lemma and three feature sets generalizing over all lemmas of the same category:

1. Distinct lemma counts of a specific category normalized by the total count of this category in a document. For example, if the lemma *can* is found in a document two times as a verb and five times as a noun, and the document contains 30 verbs and 50 nouns, we obtain the two features *can:VB = 2/30* and *can:NN = 5/50*.
2. Type-Lemma ratio: lemmas of same category normalized by total lemma count
3. Type-Token ratio: tokens of same category normalized by total token count
4. Lemma-Token Ratio: lemmas of same category normalized by tokens of same category

Proficiency and prompt features Finally, for some settings in the *closed* task we also included two nominal features to encode the *proficiency* (low, medium, high) and the *prompt* (P1–P8) features provided as meta-data along with the T11 corpus.

4 Results

4.1 Evaluation Setup

We developed our approach with a focus on the *closed* task, training the models on the T11 *train* set and testing them on the T11 *dev* set. For the *closed* task, we report the accuracies on the *dev* set for all models (single feature type models and ensembles as introduced in sections 4.2 and 4.3), before presenting the accuracies on the submitted *test* set models, which were trained on the T11 *train* ∪ *dev* set. In addition, for the submitted models we report the accuracies obtained via 10-fold cross-validation on the T11 *train* ∪ *dev* set using the folds specification provided by the organizers of the NLI Shared Task 2013.

The results for the *open-1* task are obtained by training the models on the NT11 set, and the results for the *open-2* task are obtained by training the models on the T11 *train* ∪ *dev* set ∪ NT11 set. For the *open-1* and *open-2* tasks, we report the basic single feature type results on the T11 *dev* set and two sets of results on the T11 *test* set: the results for the actual *submitted* systems and the results for the *complete* systems, i.e., including the features used in the *closed* task submissions that for the open tasks were only computed after the submission deadline (given our focus on the *closed* task and finite computational infrastructure). We include the figures for the complete systems to allow a proper comparison of the performance of our models across the tasks.

Below we provide a description of the various accuracies (%) we report for the different tasks:

- Acc_{test}: Accuracy on the T11 *test* set after training the model on:
  - *closed*: T11 *train* ∪ *dev* set
  - *open-1*: NT11 set
  - *open-2*: T11 *train* ∪ *dev* set ∪ NT11 set
- Acc_{dev}: Accuracy on the T11 *dev* set after training the model on:
  - *closed*: T11 *train* set
  - *open-1*: NT11 set
  - *open-2*: T11 *train* set ∪ NT11 set
- $Acc^{10}_{train \cup dev}$: Accuracy on the T11 *train* ∪ *dev* set obtained via 10-fold cross-validation using the data split information provided by the organizers, applicable only for the *closed* task.

In terms of the tools used for classification, we employed LIBLINEAR (Fan et al., 2008) using L2-regularized logistic regression, LIBSVM (Chang and Lin, 2011) using C-SVC with the RBF kernel and WEKA SMO (Platt, 1998; Hall et al., 2009) fitting logistic models to SVM outputs (the -M option). Which classifier was used where is discussed below.

4.2 Single Feature Type Classifier Results

First we evaluated the performance of each feature separately for the *closed* task by computing the Acc_{dev} values. These results constituted the basis for the ensembles discussed in section 4.3. We also report the corresponding results for the *open-1* and *open-2* tasks, which were partly obtained after the system submission and thus were not used for developing the approach. As classifier, we generally used LIBLINEAR, except for complexity and lm. realiz., where SMO performed consistently better. The summary of the single feature type performance is shown in Table 2.

The results reveal some first interesting insights into the employed feature sets. The figures show that the recurring word-based n-grams (rc. word ng.) taken from Bykh and Meurers (2012) are the best performing single feature type in our set yielding an Acc_{dev} value of 81.3%. This finding is in line with the previous research on different data sets showing that lexical information seems to be highly relevant for the task of NLI (Brooke and Hirst, 2011; Bykh and Meurers, 2012; Jarvis et al., 2012; Jarvis and Paquot, 2012; Tetreault et al., 2012). But also the more abstract linguistic features, such as complexity measures, local trees, or dependency variation measures seem to contribute relevant information, considering the random baseline of 9% for this task.

	Acc_{dev}		
Feature type	**closed**	**open-1**	**open-2**
1. rc. word ng.	***81.3***	***42.0***	***80.3***
2. rc. OCPOS ng.	67.6	26.6	64.8
3. rc. word dep.	67.7	30.9	69.4
4. rc. func. dep.	62.4	28.2	61.3
5. complexity	37.6	19.7	36.5
6. stemsuffix, bin.	50.3	21.4	48.8
7. stemsuffix, cnt.	48.2	19.3	47.1
8. suffix, bin.	20.4	9.1	17.5
9. suffix, cnt.	19.0	13.0	17.7
10. type dep. lm.	67.3	25.7	67.5
11. type dep. POS	46.6	27.8	27.6
12. local trees	49.1	26.2	25.7
13. dep. num.	39.7	19.6	41.8
14. dep. var.	41.5	18.6	40.1
15. dep. POS	47.8	21.5	47.4
16. lm. realiz.	70.3	30.3	66.9

Table 2: Single feature type results on T11 *dev* set

Having explored the performance of the single feature type models, the interesting question was, whether it is possible to obtain a higher accuracy than yielded by the recurring word-based n-grams by combining multiple feature types into a single model. We thus investigated different combinations, with a primary focus on the *closed* task.

4.3 Combining Feature Types

We followed Tetreault et al. (2012) in exploring two options: On the one hand, we combined the different feature types directly in a *single vector*. On the other hand, we used an *ensemble* classifier. The ensemble setup used combines the probability distributions provided by the individual classifier for each of the incorporated feature type models. The individual classifiers were trained as discussed above, and ensembles were trained and tested using LIBSVM, which in our tests performed better for this purpose than LIBLINEAR. To obtain the ensemble *training files*, we performed 10-fold cross-validation for each feature model on the T11 *train* set (for internal evaluation) and on the T11 *train* ∪ *dev* set (for submission) and took the corresponding probability estimate distributions. For the ensemble *test files*, we took the probability estimate distribution yielded by each feature model trained on the T11 *train* set and tested on the T11 *dev* set (for internal evaluation), as well as by each feature model trained on the T11 *train* ∪ *dev* set and tested on the T11 *test* set (for submission).

In our tests, the ensemble classifier always outperformed the single vector combination, which is in line with the findings of Tetreault et al. (2012). We thus focused on ensemble classification for combining the different feature types.

4.4 *Closed* Task (Main) Results

We submitted the predictions for the systems listed in Table 3, which we chose in order to test all feature types together, the best performing single feature type, everything except for the best single feature type, and two subsets, with the latter primarily including more abstract linguistic features.

id	system description	system type
1	overall system	ensemble
2	rc. word ng.	single model
3	#1 minus rc. word ng.	ensemble
4	well performing subset	ensemble
5	"linguistic subset"	ensemble

Table 3: Submitted systems for all three tasks

The results for the submitted systems are shown in Table 4. Here and in the following result tables, the system ids in the table headers correspond to the ids in Table 3, the best result on the *test* set is shown in bold, and the symbols have the following meaning:

- x = feature type used
- - = feature type not used
- -* = feature type ready after submission

We report the Acc_{test}, Acc_{dev} and $Acc^{10}_{train \cup dev}$ accuracies introduced in section 4.1. The Acc_{dev} results are consistently better than the Acc_{test} results, highlighting that relying on a single development set can be problematic. The cross-validation results are more closely aligned with the ultimate test set performance.

	systems				
Feature type	**1**	**2**	**3**	**4**	**5**
1. rc. word ng.	x	x	-	x	-
2. rc. OCPOS ng.	x	-	x	x	-
3. rc. word dep.	x	-	x	x	-
4. rc. func. dep.	x	-	x	x	-
5. complexity	x	-	x	x	x
6. stemsuffix, bin.	x	-	x	x	x
7. stemsuffix, cnt.	x	-	x	-	x
8. suffix, bin.	x	-	x	x	x
9. suffix, cnt.	x	-	x	-	x
10. type dep. lm.	x	-	x	-	x
11. type dep. POS	x	-	x	-	x
12. local trees	x	-	x	-	x
13. dep. num.	x	-	x	x	-
14. dep. var.	x	-	x	x	-
15. dep. POS	x	-	x	x	-
16. lm. realiz.	x	-	x	x	-
proficiency	x	-	x	x	-
prompt	x	-	x	x	-
Acc_{test}	**82.2**	79.6	81.0	81.5	74.7
Acc_{dev}	85.4	81.3	83.5	84.9	76.3
$Acc^{10}_{train \cup dev}$	82.4	78.9	80.7	81.7	74.1

Table 4: Results for the ***closed*** task

Overall, comparing the results for the different systems shows the following main points (with the system ids in the discussion shown in parentheses):

- The overall system performed better than any single feature type alone (cf. Tables 2 and 4). The ensemble thus is successful in combining the strengths of the different feature types.
- The rc. word ng. feature type alone (2) performed very well, but the overall system without that feature type (3) still outperformed it. Thus apparently the different properties accessed by more elaborate linguistic modelling contribute some information not provided by the surface-based n-gram feature.
- A system incorporating a subset of the different feature types (4) performed still reasonably well. Hence, it is conceivable that a subsystem consisting of some selected feature types would perform equally well (eliminating only information present in multiple feature types) or even outperform the overall system (by removing some noise). This point will be investigated in detail in our future work.

- System 5, combining a subset of feature types, where each one incorporates some degree of linguistic abstraction (in contrast to pure surface-based feature types such as word-based n-grams), performed at a reasonably high level, supporting the assumption that incorporating more linguistic knowledge into the system design has something to contribute.

Putting our results into the context of the NLI Shared Task 2013, with our best Acc_{test} value of 82.2% for *closed* as the main task, we ranked fifth out of 29 participating teams. The best result in the competition, obtained by the team "Jarvis", is 83.6%. According to the significance test results provided by the shared task organizers, the difference of 1.4% is not statistically significant (0.124 for pairwise comparison using McNemar's test).

4.5 *Open-1* Task Results

The Acc_{dev} values for the single feature type models for the *open-1* task were included in Table 2. The results for the *test* set are presented in Table 5. We report two different Acc_{test} values: the accuracy for the actual *submitted* systems (Acc_{test}) and for the corresponding *complete* systems (Acc_{test} with $*$) as discussed in section 4.1.

	systems				
Feature type	**1**	**2**	**3**	**4**	**5**
1. rc. word ng.	x	x	-	x	-
2. rc. OCPOS ng.	x	-	x	x	-
3. rc. word dep.	x	-	x	x	-
4. rc. func. dep.	x	-	x	x	-
5. complexity	x	-	x	x	x
6. stemsuffix, bin.	x	-	x	x	x
7. stemsuffix, cnt.	x	-	x	-	x
8. suffix, bin.	x	-	x	x	x
9. suffix, cnt.	x	-	x	-	x
10. type dep. lm.	-*	-	-*	-	-*
11. type dep. POS	-*	-	-*	-	-*
12. local trees	-*	-	-*	-	-*
13. dep. num.	x	-	x	x	-
14. dep. var.	x	-	x	x	-
15. dep. POS	x	-	x	x	-
16. lm. realiz.	x	-	x	x	-
Acc_{test}	36.4	**38.5**	33.2	37.8	21.2
Acc_{test} with $*$	37.0	n/a	35.4	n/a	29.9

Table 5: Results for the ***open-1*** task

Conceptually, the *open-1* task is a cross-corpus task, where we used the NT11 data for training and T11 data for testing. It is more challenging for several reasons. First, the models are trained on data that is likely to be different from the one of the *test* set in a number of respects, including possible differences in genre, task and topic, or proficiency level. Second, the amount of data we were able to obtain to train our model is far below what was provided for the *closed* task. Thus a drop in accuracy is to be expected.

Particularly interesting is the fact that our best result for the *open-1* task (38.5%) was obtained using the rc. word ng. feature type alone. Thus adding the more abstract features did not improve the accuracy. The reason for that may be the smaller training corpus size, the uneven distribution of the texts among the different L1s in the NT11 corpus, or the mentioned potential differences between NT11 and T11 in genre, task and topic, and learner proficiency. Also interesting is the fact that the system combining a subset of feature types outperformed the overall system. This finding supports the assumption mentioned in section 4.4 that the ensemble classifier can be optimized by informed, selective model combination instead of combining all available information.

To put our results into the context of the NLI Shared Task 2013, our best Acc_{test} value of 38.5% for the *open-1* task achieved rank two out of three participating teams. The best accuracy of 56.5% was obtained by the team "Toronto". While the *open-1* task results in general are much lower than the *closed* task results, highlighting an important challenge for future NLI work, they nevertheless are meaningful steps forward considering the random baseline of 9%.

4.6 *Open-2* Task Results

For the *open-2* task we provide the same information as for *open-1*. The Acc_{dev} values for the single feature type models are shown in Table 2, and the two Acc_{test} values, i.e., the accuracy for the actual *submitted* systems (Acc_{test}) and for the *complete* systems (Acc_{test} with $*$) can be found in Table 6.

For the *open-2* task, we put the T11 *train* $\cup$ *dev* and NT11 sets together to train our models. The interesting question behind this task is, whether it is possible to improve the accuracy of NLI by adding

	systems				
Feature type	**1**	**2**	**3**	**4**	**5**
1. rc. word ng.	x	x	-	x	-
2. rc. OCPOS ng.	x	-	x	x	-
3. rc. word dep.	$-^*$	-	$-^*$	$-^*$	-
4. rc. func. dep.	x	-	x	x	-
5. complexity	x	-	x	x	x
6. stemsuffix, bin.	x	-	x	x	x
7. stemsuffix, cnt.	x	-	x	-	x
8. suffix, bin.	x	-	x	x	x
9. suffix, cnt.	x	-	x	-	x
10. type dep. lm.	$-^*$	-	$-^*$	-	$-^*$
11. type dep. POS	x	-	x	-	x
12. local trees	x	-	x	-	x
13. dep. num.	x	-	x	x	-
14. dep. var.	x	-	x	x	-
15. dep. POS	x	-	x	x	-
16. lm. realiz.	x	-	x	x	-
Acc_{test}	83.5	81.0	79.3	82.5	64.8
Acc_{test} with $*$	**84.5**	n/a	83.3	82.9	79.8

Table 6: Results for the ***open-2*** task

data from corpora other than the one used for testing. This is far from obvious, especially considering the low results obtained for the *open-1* task pointing to significant differences between the T11 and the NT11 corpora.

Overall, when using all feature types, our results for the *open-2* task (84.5%) are better than those we obtained for the *closed* task (82.2%). So adding data from a different domain improves the results, which is encouraging since it indicates that something general about the language used is being learned, not (just) something specific to the T11 corpus. Essentially, the *open-2* task also is closest to the real-world scenario of using whatever resources are available to obtain the best result possible.

Putting the results into the context of the NLI Shared Task 2013, our best Acc_{test} value of 83.5% (84.5%) is the highest accuracy for the *open-2* task, i.e, first rank out of four participating teams.

5 Conclusions

We explored the task of Native Language Identification using a range of different feature types in the context of the NLI Shared Task 2013. We considered surface features such as recurring word-based n-grams system as our basis. We then explored the contribution and usefulness of some more elaborate, linguistically motivated feature types for the given task. Using an ensemble model combining features based on POS, dependency, parse trees as well as lemma realization, complexity and suffix information features, we were able to outperform the high accuracy achieved by the surface-based recurring n-grams features alone. The exploration of linguistically-informed features thus is not just of analytic interest but can also make a quantitative difference for obtaining state-of-the-art performance.

In terms of future work, we have started exploring the various feature types in depth to better understand the causalities and correlations behind the results obtained. We also intend to explore more complex linguistically motivated features further, such as features based on syntactic alternations as used in Krivanek (2012). Studying such variation of linguistic properties, instead of recording their presence as we mostly did in this exploration, also stands to provide a more directly interpretable perspective on the feature space identified as effective for NLI.

Acknowledgments

We thank Dr. Shin'ichiro Ishikawa and Dr. Mick Randall for providing access to the ICNALE corpus and the BALC corpus respectively. We also thank the shared task organizers for organizing this interesting competition and sharing the TOEFL11 corpus. Our research is partially funded through the European Commission's 7th Framework Program under grant agreement number 238405 (CLARA).

References

R. H. Baayen, R. Piepenbrock, and L. Gulikers. 1995. The CELEX lexical database (cd-rom). CDROM, `http://www.ldc.upenn.edu/Catalog/readme_files/celex.readme.html`.

Daniel Blanchard, Joel Tetreault, Derrick Higgins, Aoife Cahill, and Martin Chodorow. 2013. TOEFL11: A corpus of non-native english. Technical report, Educational Testing Service.

Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING)*, pages 89–97.

Julian Brooke and Graeme Hirst. 2011. Native language detection with 'cheap' learner corpora. In

Learner Corpus Research 2011 (LCR 2011), Louvain-la-Neuve.

Serhiy Bykh and Detmar Meurers. 2012. Native language identification using recurring n-grams – investigating abstraction and domain dependence. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING)*, pages 425–440, Mumbay, India.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC-2006)*, Genoa, Italy, May 24-26.

R.E. Fan, K.W. Chang, C.J. Hsieh, X.R. Wang, and C.J. Lin. 2008. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874. Software available at `http://www.csie.ntu.edu.tw/~cjlin/liblinear`.

Sylviane Granger, Estelle Dagneaux, Fanny Meunier, and Magali Paquot, 2009. *International Corpus of Learner English, Version 2*. Presses Universitaires de Louvain, Louvain-la-Neuve.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The weka data mining software: An update. In *The SIGKDD Explorations*, volume 11, pages 10–18.

Shin'ichiro Ishikawa. 2011. A new horizon in learner corpus studies: The aim of the ICNALE projects. In G. Weir, S. Ishikawa, and K. Poonpon, editors, *Corpora and language technologies in teaching, learning and research*, pages 3–11. University of Strathclyde Publishing, Glasgow, UK. `http://language.sakura.ne.jp/icnale/index.html`.

Scott Jarvis and Magali Paquot. 2012. Exploring the role of n-grams in L1-identification. In Scott Jarvis and Scott A. Crossley, editors, *Approaching Language Transfer through Text Classification: Explorations in the Detection-based Approach*, pages 71–105. Multilingual Matters.

Scott Jarvis, Gabriela Castañeda-Jiménez, and Rasmus Nielsen. 2004. Investigating L1 lexical transfer through learners' wordprints. Presented at the 2004 Second Language Research Forum. State College, Pennsylvania, USA.

Scott Jarvis, Gabriela Castañeda-Jiménez, and Rasmus Nielsen. 2012. Detecting L2 writers' L1s on the basis of their lexical styles. In Scott Jarvis and Scott A. Crossley, editors, *Approaching Language Transfer through Text Classification: Explorations in the Detection-based Approach*, pages 34–70. Multilingual Matters.

Moshe Koppel, Jonathan Schler, and Kfir Zigdon. 2005. 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 (KDD '05)*, pages 624–628, New York.

Julia Krivanek. 2012. Investigating syntactic alternations as characteristic features of learner language. Master's thesis, University of Tübingen, April.

Victor Kuperman, Hans Stadthagen-Gonzalez, and Marc Brysbaert. 2012. Age-of-acquisition ratings for 30,000 english words. *Behavior Research Methods*, 44(4):978–990.

Roger Levy and Galen Andrew. 2006. Tregex and tsurgeon: tools for querying and manipulating tree data structures. In *5th International Conference on Language Resources and Evaluation*, Genoa, Italy.

Xiaofei Lu. 2010. Automatic analysis of syntactic complexity in second language writing. *International Journal of Corpus Linguistics*, 15(4):474–496.

Xiaofei Lu. 2012. The relationship of lexical richness to the quality of ESL learners' oral narratives. *The Modern Languages Journal*.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 404–411, Rochester, New York, April.

John C. Platt. 1998. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report MSR-TR-98-14, Microsoft Research.

Mick Randall and Nicholas Groom. 2009. The BUiD Arab learner corpus: a resource for studying the acquisition of L2 english spelling. In *Proceedings of the Corpus Linguistics Conference (CL)*, Liverpool, UK.

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 the 24th International Conference on Computational Linguistics (COLING)*, 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 Building Educational Applications Using NLP*, Atlanta, GA, USA, June. Association for Computational Linguistics.

Laura Mayfield Tomokiyo and Rosie Jones. 2001. You're not from round here, are you? naive bayes detection of non-native utterance text. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 239–246.

Sowmya Vajjala and Detmar Meurers. 2012. On improving the accuracy of readability classification using insights from second language acquisition. In Joel Tetreault, Jill Burstein, and Claudial Leacock, editors, *Proceedings of the 7th Workshop on Innovative Use of NLP for Building Educational Applications (BEA7) at NAACL-HLT*, pages 163—-173, Montréal, Canada, June. Association for Computational Linguistics.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 180–189, Stroudsburg, PA, USA. Association for Computational Linguistics. Corpus available from `http://ilexir.co.uk/applications/clc-fce-dataset`.