# Feature Engineering in the NLI Shared Task 2013: Charles University Submission Report

**Barbora Hladká, Martin Holub** and **Vincent Kríž**
Charles University in Prague
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Prague, Czech Republic
{hladka, holub,kriz}@ufal.mff.cuni.cz

## Abstract

Our goal is to predict the first language (L1) of English essays's authors with the help of the TOEFL11 corpus where L1, prompts (topics) and proficiency levels are provided. Thus we approach this task as a classification task employing machine learning methods. Out of key concepts of machine learning, we focus on feature engineering. We design features across all the L1 languages not making use of knowledge of prompt and proficiency level. During system development, we experimented with various techniques for feature filtering and combination optimized with respect to the notion of mutual information and information gain. We trained four different SVM models and combined them through majority voting achieving accuracy 72.5%.

## 1 Introduction

Learner corpora are collections of texts written by second language (L2) learners, e.g. English as L2 – ICLE (Granger et al., 2009), Lang-8 (Tajiri et al., 2012), Cambridge Learner Corpus,[1] German as L2 – FALKO (Reznicek et al., 2012), Czech as L2 – CzeSL (Hana et al., 2010). They are a valuable resource for second language acquisition research, identifying typical difficulties of learners of a certain proficiency level (e.g. low/medium/high) or learners of a certain native language (L1 learners of L2). Research on the learner corpora does not concentrate on text collections only. Studying the errors in learner language is undertaken in the form of error annotation like in the projects (Hana et al., 2012), (Boyd et al., 2012), (Rozovskaya and Roth, 2010), (Tetreault and Chodorow, 2008). Once the errors and other relevant data are recognized in the learner corpora, automatic procedures for e.g. error correction, author profiling, native language identification etc. can be designed.

Our attention is focused on the task of automatic Native Language Identification (NLI), namely with English as L2.

In this report, we summarize the involment of the Charles University team in the first shared task in NLI co-located with the 8th Workshop on Innovative Use of NLP for Building Educational Applications in June 2013 in Atlanta, USA. The report is organized as follows: we briefly review related works in Section 2. The data sets to experiment with are characterized in Section 3. Section 4 lists the main concepts we pursue during the system development. Our approach is entirely focused on feature engineering and thus Section 5 is the most important one. We present there our main motivation for making such a decision, describe patterns according to which the features are generated and techniques that manipulate the features. We revise our ideas experimentally as documented in Section 6. In total, we submitted five systems to the sub-task of closed-training. In Sections 7 and 8, we describe these systems and discuss their results in detail. We summarize our two month effort in the shared task in Section 9.

[1] http://www.cambridge.org/gb/elt

## 2 Related work

We understand the task of native language identification as a subtask of natural language processing and we consider it as still a young task since the very first attempt to address it occurred eight years ago in 2005, as evident from the literature, namely (Koppel et al., 2005b), (Koppel et al., 2005a).

We appreciate all the previous work concerned with the given topic but we focus on the latest three papers only, all of them published at the 24th International Conference on Computational Linguistics held in December 2012 in Bombay, India, namely (Brooke and Hirst, 2012), (Bykh and Meurers, 2012), and (Tetreault et al., 2012). They provide a comprehensive review of everything done since the very first attempts. We do not want to replicate their chapters. Rather, we summarize them from the aspects we consider the most important ones in any machine learning system, namely the data, the feature design, the feature manipulation, and the machine learning methods - see Table 1.

## 3 Data sets

A new publicly available corpus of non-native English writing called TOEFL11[2] consists of essays on eight different topics written by non-native speakers of three proficiency levels (low/medium/high); the essays' authors have 11 different native languages. The corpus contains 1,100 essays per language with an average of 348 word tokens per essay. A corpus description and motivation to build such corpus can be found in (Blanchard et al., 2013).

The texts from TOEFL11 were released for the purpose of the shared task as three subsets, namely $Train$ for training, $DevTest$ for testing while system development, and $EvalTest$ for final testing. The texts were already tokenized and we processed them with the Standford POS tagger (Toutanova et al., 2003).

## 4 System settings

1. **Task:** Having a collection of English essays written by non-native speakers, the goal is to predict a native language of the essays' authors. Languages L1 are known in advance. Since we have a collection of English essays for which L1 is known (TOEFL11) at our disposal, we formulate this task as a classification task addressed by using supervised machine learning methods.

2. **Feature set**: A set $A = \{A_1, A_2, ..., A_m\}$ of $m$ features where $m$ changes as we perform various feature combinations and filtering steps. We prefer to work with binary features. We do not include two extra features, proficiency level and prompt, provided with the data. In addition, we design features across all 11 languages, i.e. we do not design features separately for a particular L1. Doing so, we address the task of predicting L1 from the text only, without any additional knowledge.

3. **Input data**: A set $X$ of instances being texts from TOEFL11 corpus represented as feature vectors, $\mathbf{x} = \langle x_1, x_2, ..., x_m \rangle \in X, x_i \in A_i$.

4. **Output classes**: A set $C$ of L1 languages, C = {ARA, CHIN, FRE, GER, HIN, ITA, JPN, KOR, SPA, TEL, TUR}, $|C| = 11$.

5. **True prediction**: A set $D = \{< \mathbf{x}, y >: \mathbf{x} \in X, y \in C\}$, $|D| = 12,100$ and its pairwise disjoint subsets $Train$, $DevTest$, $EvalTest$ where $Train \cup DevTest \cup EvalTest = D$, $|Train| = 9,900$, $|DevTest| = 1,100$, $|EvalTest| = 1,100$.

6. **Training data**: $Train \cup DevTest$. No other type of training data is used.

7. **Learning mechanism**: Since we focus on feature engineering, we do not study appropriateness of particular machine learning methods to our task in details. Instead, reviewing the related works, we selected the Support Vector Machine algorithm to experiment with.

8. **Evaluation**: 10-fold cross-validation with the sample $Train \cup DevTest$. Accuracy, Precision, Recall. Proficiency-based evaluation. Topic-based evaluation.

[2] **Source:** Derived from data provided by ETS. Copyright © 2013 ETS. www.ets.org.

| PAPER | DATA | FEATURE DESIGN | FEATURE MANIPULATION | ML METHOD |
|---|---|---|---|---|
| [1] | Lang-8, ICLE, Cambridge Learner Corpus | function words, character n-grams, POS n-grams, POS/function n-grams, context-free-grammar productions, dependencies, word n-grams | frequency-based feature selection | SVM, MaxEnt |
| [2] | ICLE | binary features spanning word-based recurring n-grams, function words, recurring POS based n-grams and combination of them | no special feature treatment | logistic regression |
| [3] | ICLE, TOEFL11 | character n-grams, function words, POS, spelling errors, writing quality | no special feature treatment | logistic regression |

Table 1: A summary of latest related works [1](Brooke and Hirst, 2012), [2](Bykh and Meurers, 2012), [3](Tetreault at al., 2012)

## 5 Feature engineering

We split the process of feature engineering into two mutually interlinked steps. The first step aims at an understanding of the task projected into features describing properties of entities we experiment with. These experiments represent the second step where we find out how the features interact with each other and how they interact with a chosen machine learning algorithm.

We compose a *feature family* as a group of patterns that are relevant for a particular task. The features are then extracted from the data according to them. Since we experiment with English texts written by non-native speakers, we have to search for specific and identifiable text properties, i.e. tendencies of certain first language writers, based on the errors caused by the difference between L1 and L2. In addition, we look for phenomena that are not necessarily incorrect in written English but they provide clear evidence of characteristics typical for L1. Our feature family is built from chunks of various length in the texts, formally lexically and part-of-speech based *n*-grams. In total, the feature family contains eight patterns described in Table 2 - six for binary features *l,n,p,s1,s2,sp* and two for continuous features *a,r*. Outside the feature family, its patterns can be combined into joint patterns, like $l{+}sp$, $n{+}sp{+}r$.

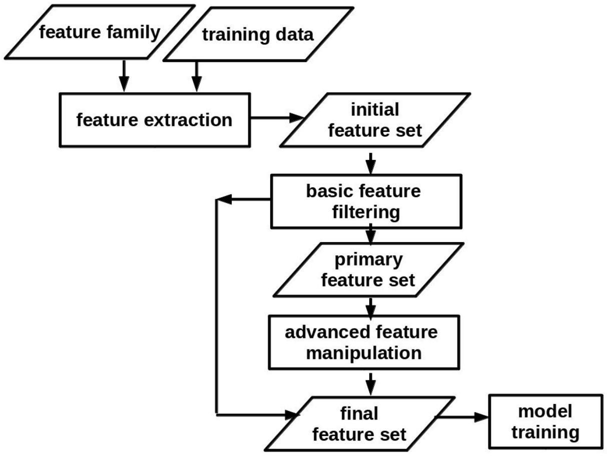

Figure 1: Feature engineering

Considering the key issues of machine learning, we mainly pay attention to overfitting. We are aware of many aspects that may cause overfitting, like complexity of the model trained, noise in training data, a small amount of training data. Features can lead to overfitting as well, thus we address it using elaborated feature engineering visualised in Figure 1. We can see there the data components and the process components having the features in common. The scheme can be traced either with individual patterns from the feature family or with joint patterns.

Both basic feature filtering and advanced feature manipulation apply selected concepts from informa-

| FEATURE FAMILY PATTERN | DESCRIPTION<br>*n=1,2,3* | EXAMPLES |
|---|---|---|
| l | *n*-grams of lemmas | *picture; to see; you, be, not* |
| n | *n*-grams of words | *picture; to see; you, are, not* |
| p | *n*-grams of function words and POS tags of content words, i.e. nouns, verbs, adjectives, cardinal numbers | *not*; PRP; *you*, VBP; JJ, *to*, VB |
| s1 | skipgrams of words: bigram $w_{i-2}, w_i$ and trigrams $w_{i-3}, w_{i-1}, w_i$, $w_{i-3}, w_{i-2}, w_i$ extracted from a sequence of words $w_{i-3}\ w_{i-2}\ w_{i-1}\ w_i$ | *you,not; able, see; to, see,in; to things, in* |
| s2 | skipgrams of words: bigrams $w_{i-3}, w_i$, $w_{i-4}, w_i$ and trigrams $w_{i-4}, w_{i-3}, w_i$, $w_{i-4}, w_{i-2}, w_i$, $w_{i-4}, w_{i-1}, w_i$ extracted from a sequence of words $w_{i-4}\ w_{i-3}\ w_{i-2}\ w_{i-1}\ w_i$ | *are,see; you,see; you,are,see; you,able,see; you,to,see;* |
| sp | *n*-grams of function words and shrunken POS tags of content words: POS tags N* are shrunken into a tag N, V* into V, J* into J | *not*; PRP; *you* V; J *to* V |
| a | relative frequency of POS tags and function words | |
| r | relative frequency of POS tags | |

Table 2: A feature family. Examples are taken from the file 498.txt, namely from the sentence *You are not able to see things in a big picture.* tagged as follows: (You/you/PRP are/be/VBP not/not/RB able/able/JJ to/to/TO see/see/VB things/thing/NNS in/in/IN a/a/DT big/big/JJ picture/picture/NN ./././)

tion theory.

### 5.1 Concepts from information theory

Consider a random variable $A$ having two possible values 0 and 1 where the probability of 1 is $p$ and 0 is $1 - p$. A degree of uncertainty we deal with when predicting the value of the variable depends on $p$. If $p$ is close to zero or one, then we are almost confident about the value and our uncertainty is low. If the values are equally likely (i.e. $p = 0.5$), our uncertainty is maximal.

The **entropy** $H(A)$ measures the uncertainty. In other words, it quantifies the amount of information needed to predict the value of the variable. The formula 1 for the entropy treats variables with $N \geq 1$ possible values.

$$H(A) = -\sum_{i=1}^{N} p(A = a_i) \log_2 p(A = a_i) \qquad (1)$$

The **conditional entropy** $H(A|B)$ quantifies the amount of information needed to predict the value of the random variable $A$ given that the value of another random variable $B$ is known, see Formula 2. Then $H(A|B) \leq H(A)$ holds.

$$H(A|B) = \sum_{b \in B} p(B = b) H(A|B = b) \qquad (2)$$

The amount $H(A) - H(A|B)$ by which $H(A)$ decreases reflects additional information about $A$ provided by $B$ and is called **mutual information** $I(A;B)$ - see Formula 3. In other words, $I(A;B)$ quantifies the mutual dependence of two random variables $A$ and $B$.

$$I(A;B) = H(A) - H(A|B) \qquad (3)$$

Proceeding from statistics to machine learning, independent random variables correspond to features. Thus we can directly speak about the entropy of a feature, the conditional entropy of a feature given another feature and the mutual information of two features.

**Information gain** of feature $A_k$ - $IG(A_k)$ - measures the expected reduction in entropy caused by partitioning the data set $Data$ according to the values of the feature $A_k$ (Quinlan, 1987):

$$IG(A_k) = H(Data) - \sum_{i=j}^{c} \frac{|D_{v_j}|}{|Data|} H(D_{v_j}), \quad (4)$$

where $A_k^v = \{v_1, v_2, ..., v_c\}$ is a set of possible values of feature $A_k$ and $D_{v_i}$ is a subset of $Data$ containig instances with the feature value $x_k = v_j$.

$C$ being a target feature, $H(Data) = H(C)$. Thus the mutual information between $C$ and $A_k$ - $I(C; A_k)$ - is the information gain of the feature $A_k$, i.e.

$$I(C; A_k) = IG(A_k). \quad (5)$$

All mentioned concepts are visualized in Figure 2 for our settings:

- Our target feature $C$ has eleven possible values (i.e. L1 languages). These values are uniformly distributed in the data $D$, thus $H(C) = -\sum_{i=1}^{11} \frac{1}{11} \log_2 \frac{1}{11} = \log_2 11 \doteq 3.46$. Sample features (only for illustration) $A_1, A_2, A_3, A_4 \in A$ are binary features so $H(A_i) \leq 1 < H(C) = 3.46$, $i = 1, ..., 4$. The circle areas correspond to the entropy of features.

- The black areas correspond to mutual information $I(A_i; A_k)$.

- The striped areas correspond to the mutual information $I(C; A_k)$ between $C$ and $A_k$.

- Features $A_1$ and $A_3$ are independent, so $I(A_1; A_3) = 0$.

- $A_2$ has the highest mutual dependence with $C$,

- $H(A_2) = H(A_3)$ and $IG(A_2) > IG(A_3)$

In addition to the concepts from information theory, we introduce another measure to quantify features: the **document frequency** of feature $A_k$ – $df(A_k)$ is the number of texts in which $A_k$ occurs, i.e. $df(A_k) \geq 0$.

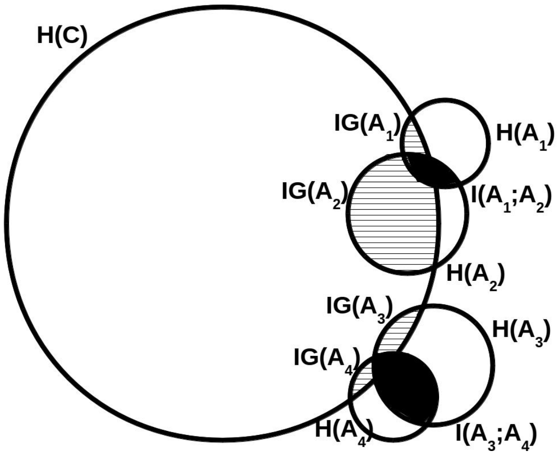

Figure 2: Information gain and mutual information visualization

## 5.2 Discussion on features

We impose a fundamental requirement on features: they should be both **informative** (i.e. useful for the classification task) and **robust** (i.e. not sensitive to training data). We control the criterion of *being informative* by information gain maximization. The criterion of *being robust* is quantified by document frequency. If $df(A_k)$ is high enough, then we can expect that $A_k$ will occur in test data frequently. We propose two techniques to increase $df$: (i) filtering out features with low $df$; (ii) feature combination driven by $IG$.

The fulfillment of both criteria is always dependent on training data, i.e. the final feature set tends to fit training data and our goal is to weaken this tendency in order to get a more robust feature set. Both basic feature filtering and advanced feature combination help us to address this issue.

## 5.3 Basic feature filtering

We obtained the feature set $A^0$ by extracting features according to the feature family patterns) from the training data. Basic feature filtering removes features from $A^0$ in two steps that result in a primary feature set $A^1$:

1. Remove binary feature $A_k$ if $df(A_k) < \delta_{df}$. Remove continous feature $A_k$ if $relative\_frequency(A_k) < \delta_{rf}$ or $df(relative\_frequency(A_k) \geq \delta_{rf}) < \delta_{df}$.

2. Remove binary feature $A_k$ if $IG(A_k) \leq \delta_{IG}$.

### 5.4 Advanced feature manipulation

The process of advanced feature manipulation handles $m$ input features from the primary feature set $A^1$ in two different ways, filter them and combine them, in order to generate a final feature set $A^f$ ready to train the model:

- **Filter them**. We use Fast Correlation-Based Filter (FCBF; (Fleuret, 2004), (Yu and Liu, 2003)) that addresses the correlation between features. It first ranks the features according to their information gain, i.e. $IG(A_1) \geq IG(A_2) \geq ... \geq IG(A_m)$. In the second step, it iteratively removes any feature $A_k$ if there exists a feature $A_j$ such that $IG(A_j) \geq IG(A_k)$ and $I(A_k; A_j) \geq IG(A_k)$, i.e. $A_j$ is better as a predictor of $C$ and $A_k$ is more similar to $A_j$ than to $C$. In the situation visualized in Figure 2, the feature $A_4$ will be filtered out because there is a feature $A_3$ such that $IG(A_3) \geq IG(A_4)$ and $I(A_3; A_4) \geq IG(A_4)$
- **Combine them**. We combine (COMB) binary features using logical operations (AND, OR, XOR, AND NOT, etc.) getting a new binary feature.

  For example, if we combine two features $A_1$ and $A_2$ using the OR operator, we get a new binary feature $Y = A_1$ OR $A_2$ for which the inequalities $df(Y) > df(A_1)$ and $df(Y) > df(A_2)$ hold. Thus we get a feature that is more robust than the two input features. To know whether it is more informative, we need to know how high $IG(Y)$ is with respect to $IG(A_1)$ and $IG(A_2)$. Without loss of generality, assume that $IG(A_1) > IG(A_2)$. If $IG(Y) > IG(A_1) > IG(A_2)$, then $Y$ is more informative than $A_1$ and $A_2$, but both of these features could be informative enough as well. It depends on the threshold we set up for *being informative*. We can easily iterate this process - let $Y_1 = A_1$ OR $A_2$ and $Y_2 = A_3$ OR $A_4$. Then we can combine $Y_3 = Y_1$ OR $A_5$ or $Y_4 = Y_1$ OR $Y_2$, etc.

Then, advanced feature manupilation runs according to scenarios formed as a series of FCBF and COMB, for example $A^1 \rightarrow$ FCBF $\rightarrow$ COMB $\rightarrow$ FCBF $\rightarrow A^f$ or $A^1 \rightarrow$ COMB $\rightarrow$ FCBF $\rightarrow A^f$.

## 6 System development

During system development, we formulated hypotheses how to avoid overfitting and get features robust and informative enough. In parallel, we run the experiments with parameters using which we controlled this requirement.

**Basic feature filtering** We set the thresholds $\delta_{df}$, $\delta_{IG}$, $\delta_{rf}$ empirically to the values 4, 0 and 0.02, respectively. Table 3 shows the changes in the size of the initial feature set after the basic feature filtering. It is evident that even such trivial filtering reduces the number of features substantially.

| FEATURE FAMILY PATTERN | INITIAL FEATURE SET (i.e. $\lvert A^0 \rvert$) | AFTER *df* FILTERING | AFTER *IG* FILTERING (i.e. $\lvert A^1 \rvert$) |
|---|---|---|---|
| 1 | 2,078,105 | 156,722 | 2,827 |
| n | 2,411,516 | 163,939 | 2,840 |
| p | 1,116,986 | 161,681 | 2,467 |
| s1 | 4,794,702 | 242,969 | 1,877 |
| s2 | 7,632,011 | 382,881 | 4,566 |
| sp | 781,018 | 123,431 | 933 |
| a | 181 | 111 | 111 |
| r | 48 | 48 | 48 |

Table 3: Volumes of initial feature sets extracted from $Train \cup DevTest$ ($1^{st}$ column). Volumes of primary feature sets after basic filtering of $A^0$ ($3^{rd}$ column)

.

**Learning mechanisms** Originally, we started with two learning algorithms, Random Forests (RF) and Support Vector Machines (SVM), running them in the R system.[3]

The **Random forests**[4] algorithm joins randomness with classification decision trees. They iterate the process of two random selections and training a decision tree $k$-times on a subset of $m$ features. Each of them classifies a new input instance $\mathbf{x}$ and the class with the most votes becomes the output class of $\mathbf{x}$.

**Support Vector Machines** (Vapnik, 1995) efficiently perform both linear and non-linear classification employing different *Kernel* functions and

[3] http://www.r-project.org

[4] http://www.stat.berkeley.edu/~breiman/

avoiding the overfitting by two parameters, *cost* and *gamma*.

We run a number of initial experiments with the following settings: the feature family pattern $n$; the basic feature filtering, RF with different values of parameters $k$ and $m$, SVM with different values of parameters *kernel*, *gamma* and *cost*

Cross-validation on the data set $Train$ performed with SVM showed significantly better results than those obtained with RF. We were quite suprised that RF ran with low performance so that we decided to stop experimenting with this algorithm. Step by step, we added patterns into the feature family and carried out experiments with SVM only on the data set $Train \cup DevTest$. We fixed the values of the SVM parameters *kernel*, *degree*, *gamma*, *cost* after several experiments as follows *kernel = polynomial, degree = 1, gamma = 0.0004, cost = 1.* Then we included the advanced feature manipulation into the experiments according to the scenarios $A^1 \rightarrow$ FCBF $\rightarrow$ COMB $\rightarrow$ FCBF $\rightarrow$ $A^f$ and $A^1 \rightarrow$ COMB $\rightarrow$ FCBF $\rightarrow$ $A^f$. COMB was composed using the OR operator only. Unfortunately, none of them outperformed the initial experiments with the basic filtering only.

Table 4 contains candidates for the final submission. The highlighted candidates were finally selected for the submission.

| FEATURE PATTERNS | CROSS-VALIDATION on $Train$ | Acc (%) on $DevTest$ |
|---|---|---|
| **l + a** | **72.97 ± 0.76** | **71.09** |
| n + a | 72.45 ± 0.98 | 63.00 |
| **l + sp + a** | **72.00 ± 0.72** | **70.64** |
| **l+sp** | **71.09 ± 0.72** | **71.45** |
| n+sp | 70.38 ± 0.69 | 52.27 |
| l | 71.67 ± 0.57 | 70.18 |
| n | 71.27 ± 0.84 | 68.72 |
| **l+p** | **71.17 ± 2.41** | **71.27** |
| n+s1 | 69.90 ± 1.04 | 66.72 |
| n+s2 | 68.75 ± 1.50 | 67.63 |
| n+s1+s2 | 67.97 ± 0.96 | 66.81 |

Table 4: Candidates for the final submission. Candidates in bold were submitted.

| MODEL | FEATURE FAMILY PATTERN | Acc (%) |
|---|---|---|
| CUNI-closed-1 | majority voting of CUNI-closed-[2-5] | 72.5 |
| CUNI-closed-2 | l+a | 71.6 |
| CUNI-closed-3 | l+p | 71.6 |
| CUNI-closed-5 | l+sp+a | 71.1 |
| CUNI-closed-4 | l+sp | 69.7 |

Table 5: An overview of models submitted.

| MODEL | Acc (%) |
|---|---|
| CUNI-closed-1 | 74.2 |
| CUNI-closed-2 | 73.4 |
| CUNI-closed-3 | 73.9 |
| CUNI-closed-4 | 73.1 |
| CUNI-closed-5 | 72.9 |

Table 6: Cross-validation results for all submitted CUNI-closed systems.

## 7 Submission to the shared task

In total, we submitted five systems to the closed-training sub-task - see their overview in Table 5. The results correspond to our expectations that we made based on the results of cross-validation presented in Table 4. The best system, CUNI-closed-1, was the outcome of majority voting of the remaining four systems. The performance of this system per language is presented in Table 7.

Table 6 reports accuracy results when doing 10-fold cross-validation on $Train \cup DevTest$. The folds for this experiment were provided by the organizers to get more reliable comparison of the NLI systems.

It is interesting to analyse the complementarity of the CUNI-closed-[2-5] systems that affects the performance of CUNI-closed-1. In Table 8, we list the numerical characteristics of five possible situations that can occur when comparing the outputs of two systems $i$ and $j$. Situations **2** and **3** capture how complementary the systems are. The numbers for our systems are presented in Table 9.

We grouped languages according to the thresholds of F-measure. First we did it across the data, no matter what the proficiency level and prompt are - see the first row of Table 10. Second we did grouping

|  | Acc(%) | P(%) | R(%) | F(%) |
|---|---|---|---|---|
| ARA | 72 | 67 | 72 | 69,6 |
| CHI | 78 | 71 | 78 | 74,3 |
| FRE | 73 | 74 | 73 | 73,7 |
| GER | 83 | 83 | 83 | 83,0 |
| HIN | 75 | 68 | 75 | 71,4 |
| ITA | 83 | 85 | 83 | 83,8 |
| JPN | 70 | 65 | 70 | 67,6 |
| KOR | 64 | 70 | 64 | 67,0 |
| SPA | 66 | 70 | 66 | 68,0 |
| TEL | 68 | 72 | 68 | 69,7 |
| TUR | 65 | 72 | 65 | 68,4 |

Table 7: CUNI-closed-1 on $EvalTest$: Acc, P, R, F

**1.** the number of instances both systems predicted correctly;
**2.** the number of instances both systems predicted incorrectly;
**3.** the number of instances the systems predicted differently: $i$ system correctly and $j$ system incorrectly;
**4.** the number of instance the systems predicted differently: $i$ system incorrectly and $j$ system correctly;
**5.** the number of instances the systems predicted differently and both incorrectly.

Table 8: Pair of two systems $i$ and $j$ and their predictions.

|  | pair of CUNI-closed-$i$ and CUNI-closed-$j$ systems |  |  |  |  |  |
|---|---|---|---|---|---|---|
|  | 2-3 | 2-4 | 2-5 | 3-4 | 3-5 | 4-5 |
| 1 | 707 | 717 | 745 | 701 | 710 | 732 |
| 2 | 161 | 215 | 242 | 183 | 181 | 250 |
| 3 | 81 | 71 | 43 | 87 | 78 | 35 |
| 4 | 81 | 50 | 37 | 66 | 72 | 50 |
| 5 | 70 | 47 | 33 | 63 | 59 | 33 |

Table 9: CUNI-closed-[2-5]: complementary rates.

|  | $\geq 90\%$ | $\geq 80\%$ | $\geq 70\%$ | $< 70\%$ |
|---|---|---|---|---|
| overall |  | GER,<br>ITA | CHI,<br>FRE,<br>HIN | TEL,<br>ARA,<br>TUR,<br>SPA,<br>JPN,<br>KOR |
| high |  | GER,<br>ITA | CHI,<br>HIN,<br>FRE | KOR,<br>TUR,<br>SPA,<br>TEL,<br>ARA,<br>JPN |
| medium |  | ITA,<br>GER,<br>FRE,<br>TEL | CHI,<br>ARA,<br>SPA,<br>TUR | JPN,<br>KOR,<br>HIN |
| low | GER | ITA,<br>FRE,<br>JPN | ARA | KOR,<br>TEL,<br>HIN,<br>TUR,<br>SPA,<br>CHI,<br>FRE |

Table 10: CUNI-closed-1 on $EvalTest$: Groups of languages sorted according to F-measure w.r.t. proficiency level.

for a particular proficiency level - see the remaining rows in Table 10. We can see that both GER and ITA are languages with the highest F-measure on all levels. Third we grouped by a particular prompt - see Table 11. We can see there diversed numbers for L1 languages despite the fact that prompts are formulated generally. Even more, we observe a topic similarity between prompts P2, P3, and P8, between P4 and P5, and between P1 and P7.

## 8 Future plans

In our future research, w want to elaborate ideas that concern the feature engineering. We plan to work with the feature family that we designed in our initial experiments. However, we will think about more specific patterns in the essays, like the average count of tokens/punctuation/capitalized nouns/articles per sentence. As Table 12 shows, there is only one candidate, namely the number of tokens in sentence, to be taken into considerations since there is the largest difference between minimum and maximum.

We confronted Ken Lackman,[5] an English teacher, with the task of *manual* native language identification by English teachers. He says: "*I think*

[5] http://kenlackman.com

|  | ≥ 90% | ≥ 80% | ≥ 70% | < 70% |
|---|---|---|---|---|
| P1 | GER,<br>ITA | FRE,<br>HIN,<br>ARA,<br>TEL | CHI,<br>KOR,<br>TUR | SPA,<br>JPN |
| P2 | GER,<br>FRE,<br>ITA,<br>TEL | ARA,<br>HIN,<br>JPN | SPA,<br>KOR,<br>CHI | TUR |
| P3 | GER | CHI,<br>KOR | HIN,<br>ITA | FRE,<br>JPN,<br>TUR,<br>ARA,<br>SPA,<br>TEL |
| P4 |  | ITA | CHI,<br>TUR,<br>HIN,<br>FRE | TEL,<br>SPA,<br>GER,<br>JPN,<br>ARA,<br>KOR |
| P5 | ITA | TUR,<br>JPN,<br>GER | FRE,<br>TEL,<br>KOR | HIN,<br>CHI,<br>SPA,<br>ARA |
| P6 |  | ITA,<br>CHI,<br>SPA | KOR,<br>ARA,<br>JPN | HIN,<br>FRE,<br>TEL,<br>GER,<br>TUR |
| P7 |  | ITA,<br>CHI,<br>TUR | SPA,<br>GER,<br>HIN,<br>FRE | ARA,<br>JPN,<br>KOR,<br>TEL |
| P8 |  | ARA | GER,<br>TEL,<br>SPA,<br>ITA | FRE<br>HIN,<br>KOR,<br>JPN,<br>TUR,<br>CHI |

Table 11: CUNI-closed-1 on $EvalTest$: Groups of languages sorted according to F-measure w.r.t. prompt.

| AVG COUNT PER SENTENCE | $Train$<br>MIN (L1) - MAX (L1) |
|---|---|
| TOKEN | 18 (JPN) -25.8 (SPA) |
| PUNCTUATION | 1.5 (HIN, TEL) - 2.1 (SPA) |
| CAPITALIZED NOUN | 0.1 (CHI) - 0.3 (HIN) |
| *the* | 0.6 (KOR) - 1.2 (ITA, SPA, TEL) |
| *a/an* | 0.3 (JPN, KOR) - 0.7 (ITA, SPA) |

Table 12: Data counts on $Train$.

*it's quite possible to do but you would need a set of guidelines to supply teachers with. The guidelines would list tendancies of certain first language writers, based on errors caused by difference between L1 and L2. For example, Germans tend to capitalize too many nouns, since there are far more nouns capitalized in their language, Asians tend to leave out articles and Arab students tend to use the verb "to be" inappropriately before other verbs.*" Looking into the data, we observe the phenomena Ken is speaking about, but the quantity of them is not statistically significant to distinguish L1s.

We formulate an idea of a *bootstrapped feature extraction* that has not been published yet, at least to our knowledge. Let us assume a set of operations that can be performed over a feature set (so far, we have proposed two possible operations with the features, filtering them out and their combinations). Determining whether a condition to perform a given operation holds is done on the *high* number of random samples. If the condition holds on the *majority* of them, then the operation is performed. The only parameter that must be set up is the *majority*. Instead of setting a threshold that is adjusted for all the features, bootstrapped feature extraction deals with fitting the data individually for each feature.

## 9 Conclusion

It was the very first experience for our team to address the task of NLI. We assess it as very stimulating and we understand our participation as setting the baseline for applying other ideas. An overall table of results (Tetreault et al., 2013) for all the teams involved in the NLI 2013 Shared Task shows that there is still space for improvement of our baseline.

We really appreciate all the work done by the organizers. They've made an effort to prepare the high-quality data and set up the framework by which the use of various NLI systems can be reliably compared.

## Acknowledgments

The authors would like to thank Eva Hajičová and Jirka Hana for their valuable comments. We also thank Ken Lackman and Leslie Ryan[6] for sharing their teaching experience. This research was

[6] http://lesliestreet.cz

supported by the Czech Science Foundation, grant no. P103/12/G084, the Technology Agency of the Czech Republic, grant no. TA02010182 and the Charles University in Prague, Faculty of Mathematics and Physics.

## References

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

Adriane Boyd, Marion Zepf, and Detmar Meurers. 2012. Informing Determiner and Preposition Error Correction with Hierarchical Word Clustering. In *Proceedings of the 7th Workshop on Innovative Use of NLP for Building Educational Applications (BEA7)*, pages 208–215, Montreal, Canada. Association for Computational Linguistics.

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

Serhiy Bykh and Detmar Meurers. 2012. Native Language Identification using Recurring $n$-grams – Investigating Abstraction and Domain Dependence. In *Proceedings of COLING 2012*, pages 425–440, Mumbai, India, December.

F. Fleuret. 2004. Fast Binary Feature Selection with Conditional Mutual Information. *Journal of Machine Learning Research (JMLR)*, 5:1531–1555.

Sylviane Granger, Estelle Dagneaux, Fanny Meunier, and Magali Paquot. 2009. *International Corpus of Learner English v2 (Handbook + CD-ROM)*. Presses universitaires de Louvain, Louvain-la-Neuve.

Jirka Hana, Alexandr Rosen, Svatava Škodová, and Barbora Štindlová. 2010. Error-tagged Learner Corpus of Czech. In *Proceedings of the Fourth Linguistic Annotation Workshop (LAW IV)*, pages 11–19, Stroudsburg, USA. Association for Computational Linguistics.

Jirka Hana, Alexandr Rosen, Barbora Štindlová, and Petr Jäger. 2012. Building a learner corpus. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, İstanbul, Turkey. European Language Resources Association.

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 11th ACM SIGKDD*, pages 624–628, Chicago, IL. ACM.

John Ross Quinlan. 1987. Simplifying decision trees. *International Journal of ManMachine Studies, 27, 221-234*.

Marc Reznicek, Anke Ludeling, Cedric Krummes, Franziska Schwantuschke, Maik Walter, Karin Schmidt, Hagen Hirschmann, and Torsten Andreas. 2012. Das Falko-Handbuch. Korpusaufbau und Annotationen Version 2.01. Technical report, Department of German Studies and Linguistics, Humboldt University, Berlin, Germany.

Alla Rozovskaya and Dan Roth. 2010. Annotating ESL Errors: Challenges and Rewards. In *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 28–36, Los Angeles, California, June. Association for Computational Linguistics.

Toshikazu Tajiri, Mamoru Komachi, and Yuji Matsumoto. 2012. Tense and Aspect Error Correction for ESL Learners Using Global Context. In *In Proceedings of the 50th ACL: Short Papers*, pages 192–202.

Joel R. Tetreault and Martin Chodorow. 2008. Native judgments of non-native usage: experiments in preposition error detection. In *Proceedings of the Workshop on Human Judgements in Computational Linguistics*, HumanJudge '08, pages 24–32, Stroudsburg, PA, USA. Association for Computational Linguistics.

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.

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, June. Association for Computational Linguistics.

Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *Proceedings of HLT-NAACL 2003*, pages 252–259.

Vladimir N. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York.

L. Yu and H. Liu. 2003. Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution. In *Proceedings of The Twentieth International Conference on Machine Leaning (ICML-03)*, pages 856–863, Washington, D.C., USA. Association for Computational Linguistics.