University of Antwerp
Faculty of Arts

Natural Language Processing

Shared Task 2021

Native Language Identification

Proficiency Level Identification

Structure

1. Introduction
2. Related research
3. Experimental set up
    1. Data set
    2. Feature extraction
    3. Native Language Identification
        1. Included features
        2. Baselines
        3. Best performing: optimized stacked classifier
        4. Results
    4. Proficiency Level Identification
        1. Included features
        2. Baselines
        3. Best performing: Logistic Regression
        4. Results
4. Discussion
5. Conclusion

Native Language Identification

- Automatically identifying L1 based on writing L2 (Markov et al. 2020)
    - Applications: second language acquisition, forensic linguistics, education, advertising, market research
      (Chan et al. 2017; Lotfi et al. 2020; Blanchard et al. 2013)
    - Supervised multi-class classification

Proficiency Level Identification

- Automatically identifying proficiency level and linguistic competence of L2 based on writing L2
- Supervised multi-class classification

Native Language Identification

- 2013 NLI shared task
    - Winning system: **83.6%** classification accuracy on TOEFL11-TEST (closed task) (Tetreault et al. 2013; Jarvis et al. 2013)
        - Word, POS and lemma *n*-grams in range=(1,3)
        - L2-regularized L2-loss SVM classifier
- 2017 NLI Shared Task
    - Winning system: 88.2% classification accuracy (Malmasi et al. 2017; Cimino and Dell'Orletta 2017)
    - Stacked sentence-document architecture
        - 2 SVM classifiers: output sentence classifier → document classifier
    - Number of specific features: character, lemma, word and CPOSn-grams, type-token ratio, and average sentence length
    - Conclusion: findings Malmasi et al. (2016): traditional supervised machine learning models > newer deep learning approaches in performance & training times

Proficiency Level Identification

- Zarco-Tejada (2019): CEFR-levelled English Corpus (CLEC)
    - Explores linguistic features in corpus statistically
    - A2, B1, B2 ~ *low, medium*
- Markov et al. (2020)
    - Traces of L1 in punctuation and emotion words persist in high proficiency levels
    - Ability to choose appropriate words for emotion expression in L2 increases with proficiency level
    - Influence of L2-ed words decreases with increase of proficiency level
    - Influence of cognates increase with proficiency level
    - #L2-ed words > #cognates (across all proficiency levels)

1. Introduction
2. Related research and current state of the art
3. **Experimental set up**
    1. Data set
    2. Feature extraction
    3. Native Language Identification
        1. Included features
        2. Baselines
        3. Best performing: optimized stacked classifier
        4. Results
    4. Proficiency Level Identification
        1. Included features
        2. Baselines
        3. Best performing: Logistic Regression
        4. Results
4. Discussion
5. Conclusion

TOEFL11 (Blanchard et al. 2013)

- 12,100 essays: 1,100 per language included
- 100 per L1 kept apart for test set
- Remaining subset: 11,000 essays with 1,000 per L1
  - Distributed into train and development set with *train_test_split*

L1	Total	Train set	Dev. set
ARA	1,000	900	100
CHI	1,000	900	100
FRA	1,000	900	100
DEU	1,000	900	100
HIN	1,000	900	100
ITA	1,000	900	100
JPN	1,000	900	100
KOR	1,000	900	100
SPA	1,000	900	100
TEL	1,000	900	100
TUR	1,000	900	100

Proficiency	Total	Train set	Dev. set
Low	1,201	1,081	120
Medium	5,964	5,368	596
High	3,835	3,451	384

- **Part-of-speech tags**
    - spaCy
    - One of core features for NLI
    - Local syntactic patterns
    - Especially POS *n*-grams of (1,3): POS trigrams best single feature type according to Malmasi and Dras (2017a)
- **Lemmatized text**
    - spaCy
    - Lemma *n*-grams are a commonly used surface feature in NLI (Markov et al. 2020; Malmasi and Dras 2017b)
- **Function words**
    - NLTK's stopwords corpus
    - Closed class word category (heavily grammaticalized)
    - Kestemont (2014): not strongly affected by topic/genre, less under control, authorship attribution
    - One of core features in NLI (Malmasi and Dras 2017a)
- **Text length**
    - Number of tokens
    - Number of sentences
    - Number of characters

- Token *n*-grams of (1,4)
- **Lemma *n*-grams of (1,4) ***
- Text length in number of tokens
- Character *n*-grams of (1,3)
- Function word uni- and bigrams

- Dummy Classifier
    - CountVectorizer
    - TfidfVectorizer
- Support Vector Machine
    - CountVectorizer
    - TfidfVectorizer
- BERT

Best performing: stacked classifier

- **83.4%** acc. on development set
- Three consecutive optimized pipelines (RandomizedSearchCV):
  - Pipeline 1.
    - TfidfV. (NLTK word tokenizer, L2 norm)
      - Lemma unigrams with document frequency higher than 0.01% and lower than 80%, lowercase=False
    - Logistic Regression
      - 'one vs. rest' strategy
      - C = 1.0 (inverse regularization strength)
      - Max_iter = 2500
  - Pipeline 2.
    - TfidfV. (NLTK word tokenizer, L2 norm)
      - Lemma *n*-grams (1,4) with document frequency higher than 10%, lowercase=True
    - SGDClassifier
      - 'l1' penalty, 'hinge' loss function
      - Max_iter=1000
  - Pipeline 3.
    - TfidfV. (NLTK word tokenizer, L2 norm)
      - Lemma *n*-grams (1,3) with document frequency higher than 0.01% and lower than 80%, lowercase = False
    - LinearSVC with default parameters

Best performing: stacked classifier

- **83.4%** acc. on development set
- Three consecutive optimized pipelines (RandomizedSearchCV):
  - Pipeline 1.
  - Pipeline 2.
  - Pipeline 3.
- Final estimator: multinomial Logistic Regression (max_iter = 1000)

	Method	(%)*
Baselines	Dummy Classifier (CountVectorizer) *Token unigrams*	8.4
	Dummy Classifier (TfidfVectorizer) *Token unigrams*	8.4
	Support Vector Machine (CountVectorizer) *Token unigrams*	60.4
	Support Vector Machine (TfidfVectorizer) *Token unigrams*	72.3
	BERT *Text column*	48.2
Single Classifiers**	Logistic Regression *Token unigrams*	71.9
	Stochastic Gradient Descent Classifier *Token unigrams*	75.5
	LinearSVC *Token unigrams*	76.3
	Logistic Regression *Token n-grams (1-2), number of tokens, character n-grams (1-3)*	73.3
	Logistic Regression *Lemma n-grams (1-2), function word n-grams (1-2), character n-grams (1-3)*	70.5
Stacked Classifiers**	Logistic Regression, SGDClassifier, LinearSVC *Token n-grams (1-4) (depending on vectorizer)*	82.9
	Logistic Regression, SGDClassifier, LinearSVC *Lemma n-grams (1-4) (depending on vectorizer)*	**83.4**
	Logistic Regression, SGDClassifier, LinearSVC *Token n-grams (1-2), lemma n-grams (1-2), character n-grams (1-3), number of tokens*	81.5

(*) Classification accuracy on development set
(**) All with TfidfVectorizer

Table 3
An overview of classifiers and results for the task of Native Language Identification.

Cf. slide 8: **unbalanced** data set

Proficiency	Total	Train set	Dev. set
Low	1,201	1,081	120
Medium	5,964	5,368	596
High	3,835	3,451	384

- Set all classes equal to class with least amount of instances?
  → discarding 7,397 training instances 🤔
- Solution: adapt *class_weight* parameter in classifiers
  - Home-made weight dictionary
  - "Balanced" argument: adjusts weights inversely proportional to class frequencies in input data

- Token unigrams
- **Token *n*-grams (1,3) ***
- **Number of tokens ***
- **POS-tag *n*-grams (1,3) ***
- Function word *n*-grams (1,3)

- Dummy Classifier
    - CountVectorizer
    - TfidfVectorizer
- Support Vector Machine
    - CountVectorizer
    - TfidfVectorizer
- BERT

Two best classifiers:

- Logistic Regression: **73.2%** (f1 macro)
    - 'One vs. rest' strategy
    - Balanced class weight
    - *Max_iter* = 5000
    - **(1)** trained on:
        - Token *n*-grams (1,3)
        - POS-tag *n*-grams (1,3)
        - Text length in number of tokens
    - **(2)** trained on:
        - Token *n*-grams (1,3)
        - POS-tag *n*-grams (1,3)
        - Text length in number of tokens
        - Function word *n*-grams (1,3)
- Principle of parsimony (Occam's Razor): choose simplest best performing model

Two best classifiers:

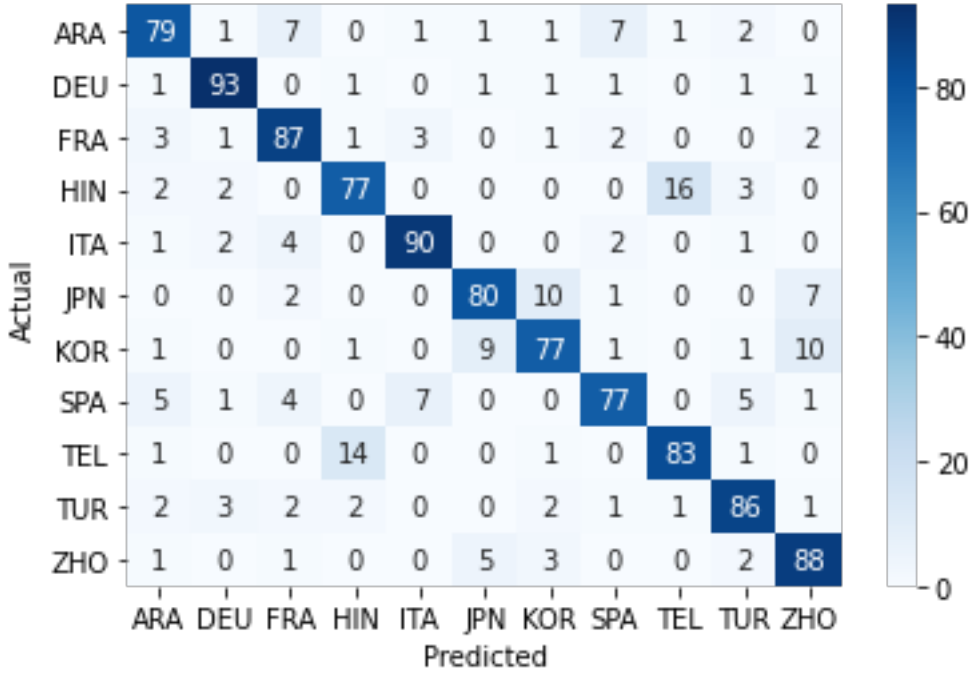
- Logistic Regression: **73.2%** (f1 macro)
    - 'One vs. rest' strategy
    - Balanced class weight
    - *Max_iter* = 5000
    - **(1)** trained on:
        - Token *n*-grams (1,3)
        - POS-tag *n*-grams (1,3)
        - Text length in number of tokens
    - **(2)** trained on:
        - Token *n*-grams (1,3)
        - POS-tag *n*-grams (1,3)
        - Text length in number of tokens
        - Function word *n*-grams (1,3)
- Principle of parsimony (Occam's Razor): choose simplest best performing model

	Methods	(%)*
Baselines	Dummy Classifier (CountVectorizer) *Token unigrams*	32.5
	Dummy Classifier (TfidfVectorizer) *Token unigrams*	32.5
	Support Vector Machine (CountVectorizer) *Token unigrams*	71.8
	Support Vector Machine (TfidfVectorizer) *Token Unigrams*	58.9
	BERT *Text column*	NA
Single classifiers**	LinearSVC *Token uni-, bi-, trigrams*	66.6
	Logistic Regression *Token uni-, bi-, trigrams*	72.2
	Stochastic Gradient Descent Classifier *Token uni-, bi-, trigrams*	60.9
	Logistic Regression *Token n-grams (1-3), number of tokens*	72.4
	Logistic Regression *Lemma n-grams (1-3), number of tokens*	70.8
	Logistic Regression *Token n-grams (1-3), number of tokens,* *POS-tag n-grams (1-3)*	**73.2**
	Logistic Regression *Token n-grams (1-3), number of tokens,* *POS-tag n-grams (1-3), function word n-grams (1-3)*	73.2

(*) Macro-average f1-score on development set
(**) All with CountVectorizer

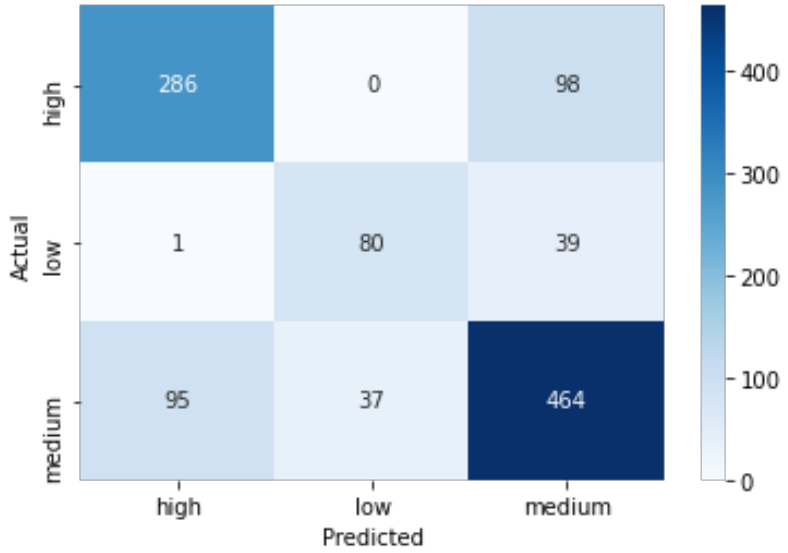
Table 4
An overview of classifiers and results for the task of Proficiency Level Identification.

- NLI: 83.4% acc.
- PLI: 73.2% f1 macro
- Not able to outperform SOTA
- Traditional ML approaches > BERT baseline



- German: 93% recall
- Hindi, Telugu most often confused
    - Both spoken in India → language contact → influence
- Spanish, Italian, French, Arabic, Turkish
- Korean, Japanese, Chinese

- Morphological similarities that persist and remain noticeable in the second language performance



- *Low*: problematic
- *Medium* & *high* more successful
- Most successful features:
    - Text length in number of tokens, token *n*-grams and POS-tag *n*-grams
    - Mean number of tokens differs noticeably per proficiency level:
        - *High*: 363 tokens / essay
        - *Medium*: 306 tokens / essay
        - *Low:* 206 tokens / essay
    - Inclusion of morphosyntactic features of POS-tag *n*-grams → improvement in classification performance

- NLI: stacked classifier trained on lemma *n*-grams (1,4) → **83.4%**
- PLI: Logistic Regression trained on token *n*-grams (1,3), POS-tag *n*-grams (1,3), text length in number of tokens, → **73.2%**

- Performed relatively well without outperforming the current state of the art
- Further research: adding more useful morphosyntactic and grammatical features such as POS-tags, function words, punctuation usage in combination with the proposed stacked classifier