Stacked Sentence-Document Classifier Approach for Improving Native Language Identification

Andrea Cimino, Felice Dell'Orletta
Istituto di Linguistica Computazionale "Antonio Zampolli" (ILC–CNR)
via G. Moruzzi, 1 – Pisa (Italy)
{name.surname}@ilc.cnr.it

Abstract

In this paper, we describe the approach of the *ItaliaNLP Lab* team to native language identification and discuss the results we submitted as participants to the essay track of NLI Shared Task 2017. We introduce for the first time a 2-stacked sentence-document architecture for native language identification that is able to exploit both local sentence information and a wide set of general–purpose features qualifying the lexical and grammatical structure of the whole document. When evaluated on the official test set, our sentence-document stacked architecture obtained the best result among all the participants of the essay track with an F1 score of 0.8818.

1 Introduction

Native Language Identification (NLI) is the task of identifying the native language (L1) of a writer based on their writing in another language. Since the seminal work by Koppel et al. (2005), within the Computational Linguistics community there has been a growing interest in the NLP–based Native Language Identification (henceforth, NLI) task. However, so far, due to the unavailability of balanced and wide–coverage benchmark corpora and the lack of evaluation standards it has been difficult to compare the results achieved for this task with different methods and techniques (Tetreault et al., 2012). The First Shared Task on Native Language Identification (Tetreault et al., 2013) was the answer to these mentioned problems.

In this paper, we describe our approach to the essay track of the 2017 Native Language Identification Shared Task (Malmasi et al., 2017). Participating teams of this task were asked to classify the native language of writers of 1,100 English essays solely using the sample of their writings. 11,100 English essays from non-native English writing samples from a standardized, meaningful, and authentic assessment context of English proficiency for academic purposes (the Test Of English as a Foreign Language, TOEFL) (Blanchard et al., 2013) were provided as training data and the 11 native languages covered by the corpus are: Arabic, Chinese, French, German, Hindi, Italian, Japanese, Korean, Spanish, Telugu, and Turkish. Each essay in the TOEFL11 is labeled with an English language proficiency level.

Following the most common approaches and starting from the work of (Cimino et al., 2013), we tackled the Native Language Identification task as a text classification problem. The main novelty of our approach is the proposed classification architecture that combines a sentence and a document classifier in a 2-stacked sentence-document architecture. This system is able to exploit both local sentence information and a wide set of features extracted from the whole document. The features range across different levels of linguistic description, from lexical to morpho–syntactic and syntactic information.

The proposed method was prompted by our studies on sentence and document readability classification (Dell'Orletta et al., 2014), where we shown differences between document and sentence classification problems by focusing on the role of the features and their importance. For example, the classification of the readability of a sentence requires a higher number of features, mainly syntactic ones, and they have different weights with respect to the weights used in the document classification problem. In this work, we show how sentence local information can be exploited also in NLI task providing to the document classifier fruitful local information, thus making some features more effective.

Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications, pages 430–437
Copenhagen, Denmark, September 8, 2017.

2 Related Work

Native Language Identification is most commonly tackled as a multi-class supervised classification task combining NLP–enabled feature extraction and machine learning: see e.g. (Tetreault et al., 2012), and (Malmasi and Dras, 2017). Among the different machine learning algorithms used, systems based on Support Vector Machines obtain the best accuracies. However, the most successful approaches made use of classifier ensemble methods to further improve performance. All recent state-of-the-art systems have relied on some form of multiple classifier system. Among the most recent works, (Ionescu et al., 2014) used multiple string kernels learning using only character n-gram features, reporting an accuracy of 85.3 on the TOEFL11 test set, 1.7 higher than the 2013 state of the art obtained by (Jarvis et al., 2013) in the first shared task on NLI (Tetreault et al., 2013). More recently, (Malmasi and Dras, 2017) made a systematic examination of ensemble methods. By exploiting a classifier stacking architecture, the authors obtained the current state-of-the-art results on three datasets from different languages. As in these previous works, the system presented in this paper uses a stcked architecture, but differently from the previous ones combines a sentence and a document classifier and it is able to exploit in a profitable way both local sentence information and global document information.

Typically, the range of features used is wide and includes characteristics of the linguistic structure underlying the L2 text, encoded in terms of sequences of characters, words, grammatical categories or of syntactic constructions, as well as of the document structure: note however that, in most part of the cases, the exploited features are task–specific. Differently, as in our first system (Cimino et al., 2013), we resort to a wide set of features ranging across different levels of linguistic description (i.e. lexical, morpho–syntactic and syntactic) without any a priori selection: the same set of features was successfully exploited in different tasks focusing on the linguistic form rather than the content of texts, such as readability assessment (Dell'Orletta et al., 2014) or the classification of textual genres (Dell'Orletta et al., 2012).

3 Description of the system

Our approach to the Native Language Identification Task was implemented in a software prototype. The main novelty of our approach is the use of a stack of two SVM classifiers, each one operating on morpho–syntactically tagged and dependency parsed texts. The first classifier is a L1 *sentence classifier* that is aimed at classifying the native language of each sentence of a document. The predictions of the L1 sentence classifier are used as features by the L1 *document classifier*. In addition to the sentence classifier predictions, the second classifier exploits widely used features in native language identification that are used to build the final statistical model. This statistical model is finally used to predict the L1 language of unseen documents. The highest score of the document classifier represents the most probable L1 class. For this work we used LIBLINEAR (Fan et al., 2008) as machine learning library both for the sentence and the document classifiers. The documents were automatically POS tagged by the Part–Of–Speech tagger described in (Cimino and Dell'Orletta, 2016) and dependency–parsed by the DeSR parser (Attardi et al., 2009).

3.1 Training workflow

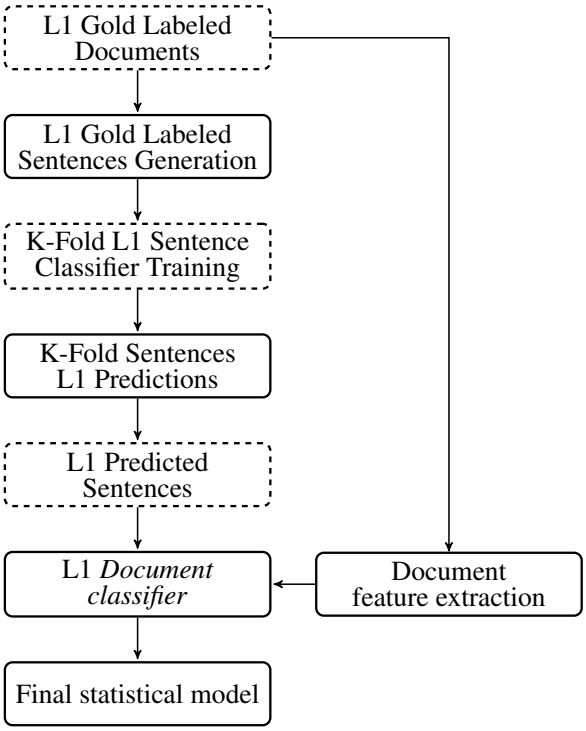

Figure 1: The training workflow of the 2-stacked sentence-document architecture.

Since the document classifier exploits the *predictions* of the sentence classifier in classification

of unseen documents, we devised a specific training workflow that is shown in Figure 1. In the first step of the workflow, the L1 gold labels of the training documents are exploited to build an annotated corpus of L1 sentences where each sentence is labeled according to the label of its belonging document. Once the L1 gold labeled sentence corpus is generated, this is used to train the sentence classifier and used to create the training set of the document classifier. More precisely, the L1 sentence corpus is divided in k different folds [1] where each fold is used to provide the training examples for the sentence classifier. By exploiting widely used NLI features, the sentence classifier produces a specific statistical model for each of the k folds.

The statistical models are then used to predict the L1 language of the sentences that do not belong to the training examples of the generated folds. For this work we used the LIBLINEAR L2-regularized logistic regression as learning algorithm since the LIBLINEAR implementation provides the confidence of belonging to a specific class for unseen examples. In addition, features with frequency lower than 2 in the corpus where discarded. By merging the k folds of the L1 predicted sentences, a corpus of L1 predicted sentences is obtained and it is used by the document classifier during its training phase. The document classifier by exploiting widely used NLI features and the predictions of the sentence classifier produces its own statistical model that is finally used to predict the L1 language of unseen documents. The document classifier was trained using the LIBLINEAR L2-regularized L2-loss support vector classification that (Jarvis et al., 2013) have shown to have very good performances in NLI document classification. Features with frequency lower than 3 in the corpus where discarded.

Once the document classifier is trained, for the final settings the sentence classifier is trained using all the sentences of the L1 Gold sentence corpus, this in order to achieve the best possible accuracy in classification of unseen sentences.

The prediction workflow of unseen documents, shown in Figure 2, is similar to the training workflow with the exception that the k fold training procedure is not needed.

All the real valued features were scaled in the range $[0, 1]$ in order to reduce the training times and to maximize the classification performances.

[1] for our runs we have chosen $k = 5$

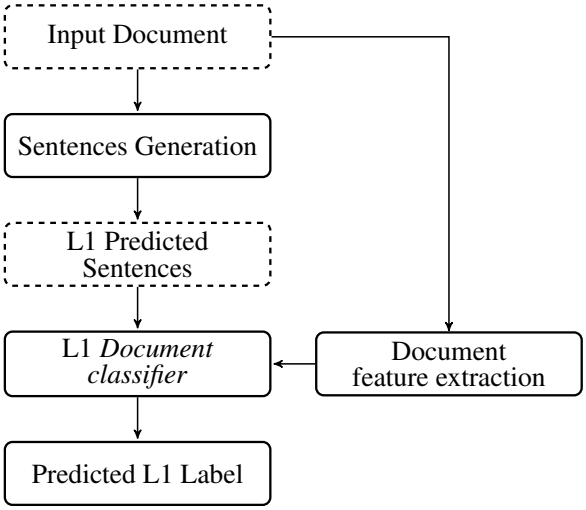

Figure 2: The test workflow of the 2-stacked sentence-document architecture.

3.2 Sentence and Document Features

Here are described the features used both by the sentence and the document classifiers. Hereafter we regard documents and sentences as *texts* in order to avoid ambiguities in the description of the features. In the description below some features are calculated as the normalized frequency and other as the normalized logarithm of the frequency. The choice was made according to empirical evaluation on the development set.

Raw and Lexical Text Features

Text Length, calculated as the number of tokens.

Word Length, calculated as the average number of characters per word.

Character n-grams, calculated as the logarithm of the frequency of each character n-gram in the text and normalized with respect to the text length. A smoothing term is added to the frequency of each n-gram in order to avoid 0 values for n-grams with 1 frequency.

Function word n-grams, calculated as the frequency of each function word n-gram in the text and normalized with respect to the number of tokens in the text. In this work we considered the words belonging to one of the following fine part-of-speech categories: determiners, coordinating conjunctions, preposition or subordinating conjunctions, interjections.

Word n-grams, calculated as presence or absence of a word n-gram in the text.

Lemma n-grams, calculated as the frequency of

each lemma n-gram in the text and normalized with respect to the number of tokens in the text.

Morpho–syntactic Features

Coarse grained Part-Of-Speech n-grams, calculated as the logarithm of the frequency of each coarse grained PoS n-gram in the text and normalized with respect to the number of tokens of the text.

Coarse grained Part-Of-Speech - Lemma n-grams: calculated as the frequency of the n-grams of the Coarse grained Part-of-Speech of the current token and its following token lemma. The frequencies are normalized with respect to the number of tokens of the text.

Syntactic Features

Linear dependency types n-grams, calculated as the frequency of each dependency n-gram in the text with respect to the surface linear ordering of words and normalized with respect to the number of tokens in the text.

Hierarchical dependency types n-grams calculated as the logarithm of the frequency of each hierarchy dependency n-gram in the text calculated with respect to the hierarchical parse tree structure and normalized with respect to the number of tokens in the text. In addition to the dependency relationship, the feature takes into account whether a node is a left or a right child with respect to its parent.

Head-dependents of the syntax tree: the distribution of head and its dependents in the syntax trees.

3.3 Document Classifier Specific Features

In addition to the features described in 3.2, the document classifier uses the following features.

Raw Features

Essay prompt, included in the TOEFL11 corpus.

Average sentence length and standard deviation, calculated in terms of number of tokens for each sentence in the document.

Type/Token Ratio. The Type/Token Ratio (TTR) is a measure of vocabulary variation which has shown to be a helpful measure of lexical variety within a text as well as style marker in an authorship attribution scenario: a text characterized by a low type/token ratio will contain a great deal of repetition whereas a high type/token ratio reflects vocabulary richness and variation. Due to its sensitivity to sample size, the TTR has been computed for different chunk lengths. In this work we considered the first 100, 200, 300 and 400 tokens.

Sentence classifier predictions. Since the sentence classifier provides for each sentence the probability score of each L1 class, the following 55 features were calculated for each document: for each L1 language the *i*) average probability, *ii*) the standard deviation of the probabilities, *iii*) the probability product, *iiii*) the maximum probability and *iiiii*) the minimum probability of all the sentences.

3.4 Models

In order to test the performances of the proposed two-stacked sentence-document classifier, we conducted several experiments exploiting different configurations of our system. Table 1 reports the configurations selected for the official runs in terms of features and values of n-grams used. Stacked1 and Stacked2 use both the 2-stack classifier architecture, but the Stacked2 model does not include the *Functional word n-gram* features and the *head-dependents* features. Not-stacked1 and Not-stacked2 reflect the previous two configurations with the exception that the sentence classifier features were not introduced. The selection of these models was guided by the tuning performed on the official NLI Shared Task 2013 and 2017 test sets. Tables 2 and 3 report the results achieved by the selected models on the official 2013 test set and the 2017 development set.

Model	Prec.	Recall	F1-Score
Jarvis (2013)	-	-	0.836
Cimino (2013)	-	-	0.779
Stacked1	**0.853**	**0.851**	**0.851**
Not-stacked1	0.850	0.848	0.848
Stacked2	0.851	0.849	0.849
Not-stacked2	0.850	0.847	0.847

Table 2: Results obtained by our models on the NLI Shared Task 2013 official test set compared to the overall best run and our best run submitted in the NLI Shared Task 2013 edition.

Feature	Feature-Configuration	Stacked1	Stacked2	Not-stacked1	Not-stacked2
Sentence Classifier and Document Classifier features					
Character n-grams	up to 8	✓	✓	✓	✓
Word n-grams	up to 4	✓	✓	✓	✓
Lemma n-grams	up to 4	✓	✓	✓	✓
CPOS n-grams	up to 4	✓	✓	✓	✓
LEMMA-CPOS n-grams	up to 4	✓	✓	✓	✓
Functional word n-grams	up to 3	✓	✗	✓	✗
Linear dependency n-grams	up to 4	✓	✓	✓	✓
Hierarchical dependency n-grams	up to 4	✓	✓	✓	✓
Text Length	NA	✓	✓	✓	✓
Head-Dependents	NA	✓	✗	✓	✗
Document Classifier specific features					
Type Token Ratio	100,200,300,400	✓	✓	✓	✓
Essay Prompt	NA	✓	✓	✓	✓
Average Sentence Length	NA	✓	✓	✓	✓
Standard Deviation Sentence Length	NA	✓	✓	✓	✓
Average Sentence L1 Confidence	NA	✓	✓	✗	✗
Std. Dev. Sentence L1 Confidence	NA	✓	✓	✗	✗
Product of Sentence L1 Confidences	NA	✓	✓	✗	✗
Maximum Sentence L1 Confidence	NA	✓	✓	✗	✗
Minimum Sentence L1 Confidence	NA	✓	✓	✗	✗

Table 1: Configurations of our system used to train our classifier for the evaluation of the NLI Shared Task 2017 test set.

Model	Prec.	Recall	F1-Score
Stacked1	0.8551	0.8527	0.8525
Stacked2	**0.8567**	**0.8545**	**0.8544**
Not-stacked1	0.8552	0.8527	0.8526
Not-stacked2	0.8524	0.8500	0.8498

Table 3: Results obtained by our models on the NLI Shared Task 2017 official development set.

System	F1-Score	Accuracy
Random Baseline	0.0909	0.0909
Organizers baseline	0.7104	0.7109
Stacked1	0.8800	0.8800
Stacked2	**0.8818**	**0.8818**

Table 4: Results of our submitted models for the essay track on the NLI Shared Task 2017 official test set.

4 Results

Table 4 reports the F1-Score and the overall accuracy achieved by our stacked architecture with the feature configuration described in section 3.4 on the NLI Shared Task 2017 official test set. In addition the table reports the results achieved by two different baselines provided by the shared task organizers[2]: a random baseline and a classifier that uses only word unigrams as features. Figure 3 reports the confusion matrix of our best model (Stacked2) on the official NLI essay test set. In addition, Table 5 reports the results obtained by the non stacked version of our architecture. These runs were submitted to the organizers of the task after the official evaluation period.

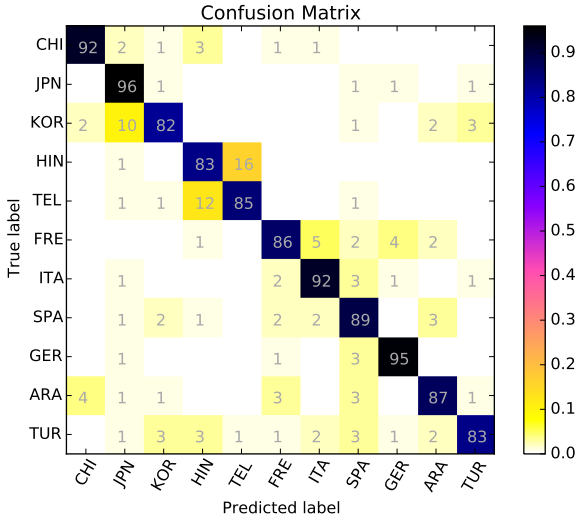

Figure 3: Confusion Matrix of the Stacked2 model on the NLI Shared Task 2017 official test set.

[2] A more detailed description of the baseline system is reported in (Malmasi et al., 2017).

System	F1-Score	Accuracy
Not-stacked1	0.8727	0.8729
Not-stacked2	0.8764	0.8765

Table 5: Results of our not-stacked systems for the essay track on the NLI Shared Task 2017 official test set.

4.1 Discussion

We tested different configurations of our architecture in order to evaluate the contribution on the accuracy of: *i)* the single components of the 2-stacked sentence-document architecture, *ii)* the lexical information and *iii)* the syntactic information. We carried out different experiments on the official NLI Shared Task 2017 development set and on the official NLI Shared Task 2013 test set that reflect the questions we wanted to answer, more specifically the questions are:

- (a) what are the performance obtained by using the 2-stacked sentence-document architecture and by using the sentence and document classifiers separately?
- (b) what is the contribution of the Lexical information on the stacked architecture and on the single components?
- (c) what is the contribution of the Syntactic information on the stacked architecture and on the single components?

In order to answer to these questions, we devised 3 different feature configurations: *All features*, that uses all the features described in section 3.2; *No Lexical*, that does not use the word n-grams features and the character n-grams features; *No Syntax*, that does not use the features extracted from the dependency syntax trees. For each configuration three different classifiers were trained: the Stacked classifier, the Document classifier (Not-Stacked) and the Sentence classifier (Sent). We tested the Sentence classifier in the document classification task by using two different approaches to assign the most probable L1 class of a document according to the predictions of the sentence classifier. The first is a vote approach (VOTE), where we decided to assign to a document the most frequent L1 predicted class among all the sentences of the document. The second is an average approach (AVG): since the sentence classifier assigns for each L1 class its confidence, we took as L1 document class the one that had the highest average among all the probabilities of each sentence. In addition, we tested the accuracy of the Sentence classifier on sentence classification using as test sets the sentences belonging to the documents of the NLI 2017 development set and of the NLI 2013 test set.

2017 NLI development set			
Model	**Prec.**	**Recall**	**F1-Score**
All Features			
Stacked	0.8551	0.8527	0.8525
NotStacked	**0.8552**	**0.8527**	**0.8526**
Sent. (AVG)	0.7968	0.7900	0.7886
Sent. (VOTE)	0.7516	0.7436	0.7405
No Lexical			
Stacked	**0.8070**	**0.8036**	**0.8033**
Not-stacked	0.7947	0.7927	0.7923
Sent. (AVG)	0.7345	0.7182	0.7148
Sent. (VOTE)	0.6592	0.6409	0.6343
No Syntax			
Stacked	**0.8545**	**0.8527**	**0.8526**
Not-stacked	0.8519	0.8500	0.8498
Sent. (AVG)	0.8017	0.7936	0.7925
Sent. (VOTE)	0.7472	0.7409	0.7384

Table 6: Results of our experiments on the NLI Shared Task 2017 development set.

2013 NLI test set			
Model	**Prec.**	**Recall**	**F1-Score**
All features			
Stacked	**0.8537**	**0.8518**	**0.8516**
NotStacked	0.8502	0.8482	0.8480
Sent. (AVG)	0.7983	0.7909	0.7896
Sent. (VOTE)	0.7474	0.7418	0.7393
No Lexical			
Stacked	**0.8042**	**0.8018**	**0.8014**
Not-stacked	0.7840	0.7818	0.7814
Sent. (AVG)	0.7325	0.7200	0.7160
Sent. (VOTE)	0.6515	0.6373	0.6311
No Syntax			
Stacked	**0.8571**	**0.8555**	**0.8553**
Not-stacked	0.8513	0.8491	0.8489
Sent. (AVG)	0.7957	0.7891	0.7880
Sent. (VOTE)	0.7429	0.7373	0.7346

Table 7: Results of our experiments on the official NLI Shared Task 2013 test set.

Model	Prec.	Recall	F1-Score
2017 NLI development set			
Baseline	0.3533	0.3531	0.3515
All features	0.3937	0.3948	0.3936
2013 NLI test set			
Baseline	0.3541	0.3536	0.3519
All features	0.3946	0.3956	0.3946

Table 8: Performances of the sentence classifier on sentences belonging to the official NLI Shared Task 2017 development set and on the official NLI Shared Task 2013 test set.

Tables 6, 7 and 8 report the results of all the experiments. With the exception of the results obtained by the All features model on the 2017 development set, the stacked architecture always outperforms the not-stacked architecture in all the feature configurations used, showing that our devised stacked architecture is effectively able to exploit some information hidden in L2 sentences that are not fully captured at document level. For what concerns the sentence classifier when used as a document classifier, the average approach (AVG) always outperforms the results of VOTE version in all the configuration tested. We can see also that for each feature configuration there is a drop of only 5-6 points with respect to the 2-stacked classifier.

Table 8 reports the performances of the standalone sentence classifier on the L1 sentence classification task. For each dataset we report a baseline result calculated by using only word unigrams features. We have chosen this baseline following the approach used by NLI Shared Task organizers for calculating their baseline system. The baseline results are compared with the results achieved by the *All Features* configuration. As expected, the L1 sentence classification task is extremely more difficult than the L1 document classification task: the results achieved by the baseline system on the document classification task are extremely higher than the ones on the sentence classification task (+35% in terms of F1-Score). An interesting result to notice is the contribution in the sentence and document classification tasks of the features we used to develop our system. While we can observe an improvement of almost 14% (F1-Score) with respect to the baseline system on the L1 document classification task, only 4% (F1-Score) of improvement are achieved on the sentence classification task, confirming the complexity of the sentence classification task and the need of a specific process of feature selection for this task.

For what concerns question (b), we can observe that the lexical features (word n-grams and character n-grams) are extremely relevant for NLI. Both the *All Features Stacked* configuration and the *No Syntax Stacked* configuration report an accuracy of approximately 0.85% on the performed experiments, which is almost 5 points more than the results obtained by using the *No Lexical Stacked* configuration. The same drop in classification performance can be also observed when using the not-stacked architecture and the sentence classifier as document classifier with the AVG and the VOTE approaches.

Finally, for what concerns question (c) we can observe that surprisingly the syntax features bring almost or no contribution when joined with all the other features we used. When the results of the stacked, non-stacked and sentence rows achieved by the *All Features* configuration are compared with the respective ones achieved by the *No Syntax* configuration, no statistical difference in accuracy can be observed. In our opinion, this result is due to the correlation of lexical information and part-of-speech tag information, but a more in depth analysis would be required to analyze these results.

5 Conclusions

In this paper, we reported the results of our participation to the essay track of the Second Native Language Identification Shared Task. By resorting to a novel 2-stacked sentence-document architecture and to a set of general purpose features qualifying the lexical and grammatical structure of a text, we achieved very promising results and the first position in this shared task.

We have shown that our novel stacked architecture outperforms the results achieved by a single document classifier, showing that sentence local information is useful for NLI.

In future works, we would like to carry out a more in depth study of the sentence level classifier, focusing in particular on the features that most maximize its accuracy on L2 sentences. In addition, we want to investigate the combination of different sentence-document models in order to deepen the study of the interaction between the sentence and document levels in the task of native language identification.

References

Giuseppe Attardi, Felice Dell'Orletta, Maria Simi, and Joseph Turian. 2009. Accurate Dependency Parsing with a Stacked Multilayer Perceptron. In *Proceedings of the 2nd Workshop of Evalita 2009*.

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

Andrea Cimino and Felice Dell'Orletta. 2016. Building the state-of-the-art in POS tagging of Italian Tweets. In *Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016), Napoli, Italy, December 5-7, 2016.*.

Andrea Cimino, Felice Dell'Orletta, Giulia Venturi, and Simonetta Montemagni. 2013. Linguistic Profiling based on General-purpose Features and Native Language Identification. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications, BEA@NAACL-HLT 2013, June 13, 2013, Atlanta, Georgia, USA*. pages 207–215.

Felice Dell'Orletta, Simonetta Montemagni, and Giulia Venturi. 2012. Genre–oriented Readability Assessment: a Case Study. In *Proceedings of the Workshop on Speech and Language Processing Tools in Education (SLP-TED)*.

Felice Dell'Orletta, Martijn Wieling, Giulia Venturi, Andrea Cimino, and Simonetta Montemagni. 2014. Assessing the Readability of Sentences: Which Corpora and Features? In *Proceedings of the Ninth Workshop on Innovative Use of NLP for Building Educational Applications, BEA@ACL 2014, June 26, 2014, Baltimore, Maryland, USA*. pages 163–173.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research* 9:1871–1874.

Radu-Tudor Ionescu, Marius Popescu, and Aoife Cahill. 2014. Can characters reveal your native language? A language-independent approach to native language identification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. pages 1363–1373.

Scott Jarvis, Yves Bestgen, and Steve Pepper. 2013. Maximizing Classification Accuracy in Native Language Identification. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, Atlanta, Georgia, pages 111–118.

Moshe Koppel, Jonathan Schler, and Kfir Zigdon. 2005. Automatically Determining an Anonymous Author's Native Language. In *Intelligence and Security Informatics, IEEE International Conference on Intelligence and Security Informatics, ISI 2005, Atlanta, GA, USA, May 19-20, 2005, Proceedings*. pages 209–217.

Shervin Malmasi and Mark Dras. 2017. Native Language Identification using Stacked Generalization. *arXiv preprint arXiv:1703.06541* .

Shervin Malmasi, Keelan Evanini, Aoife Cahill, Joel Tetreault, Robert Pugh, Christopher Hamill, Diane Napolitano, and Yao Qian. 2017. A Report on the 2017 Native Language Identification Shared Task. In *Proceedings of the 12th Workshop on Building Educational Applications Using NLP*. Association for Computational Linguistics, Copenhagen, Denmark.

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*. Association for Computational Linguistics, Atlanta, GA, USA.

Joel R. Tetreault, Daniel Blanchard, Aoife Cahill, and Martin Chodorow. 2012. Native Tongues, Lost and Found: Resources and Empirical Evaluations in Native Language Identification. In *COLING 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, 8-15 December 2012, Mumbai, India*. pages 2585–2602.