The Story of the Characters, the DNA and the Native Language

Marius Popescu and Radu Tudor Ionescu

Department of Computer Science, University of Bucharest, Bucharest, Romania

Our approach to the NLI Shared Task 2013

Aim: investigate if identifying native language is possible with machine learning methods that work at the character level

Advantages:

- completely language independent: the texts will be treated as sequences of symbols (strings)
- theory neutral: rather than restricting the feature space according to theoretical or empirical principles let the learning algorithm select the important features

Tools:

- kernel-based learning methods
- different string kernels

String Kernels

p-spectrum kernel:

$$k_p(s,t) = \sum_{v \in \Sigma^p} \text{num}_v(s) \cdot \text{num}_v(t)$$

p-grams presence bits kernel:

$$k_p^{0/1}(s,t) = \sum_{v \in \Sigma^p} \text{in}_v(s) \cdot \text{in}_v(t)$$

Normalized:

$$\hat{k}_p(s,t) = \frac{k_p(s,t)}{\sqrt{k_p(s,s) \cdot k_p(t,t)}}$$

$$\hat{k}_p^{0/1}(s,t) = \frac{k_p^{0/1}(s,t)}{\sqrt{k_p^{0/1}(s,s) \cdot k_p^{0/1}(t,t)}}$$

Kernel based on Local Rank Distance

Local Rank Distance (LRD):

$$\Delta_{LRD}(S_1,S_2) = \Delta_{left} + \Delta_{right}$$
$$= \sum_{x_s \in S_1} \min_{x_s \in S_2} \{|pos_{S_1}(x_s) - pos_{S_2}(x_s)|, m\} +$$
$$+ \sum_{y_s \in S_2} \min_{y_s \in S_1} \{|pos_{S_1}(y_s) - pos_{S_2}(y_s)|, m\}$$

Kernel:

$$k(s_1,s_2) = e^{-\frac{\Delta_{LRD}(s_1,s_2)}{2\sigma^2}}$$

Combining Kernels

By summing kernels and by kernel alignment

Choosing the Learning Method

- Support Vector Machines / Kernel Ridge Regression (KRR)
- one-versus-one (OVO) / one-versus-all (OVA)

Method	Accuracy
OVO SVM	**72.72%**
OVA SVM	**74.94%**
OVO KRR	**73.99%**
OVA KRR	**77.74%**
KPLS	**74.99%**

Table : Accuracy rates using **10**-fold cross-validation on the train set for different kernel methods with $\hat{k}_5$ kernel.

Parameter Tuning for String Kernel

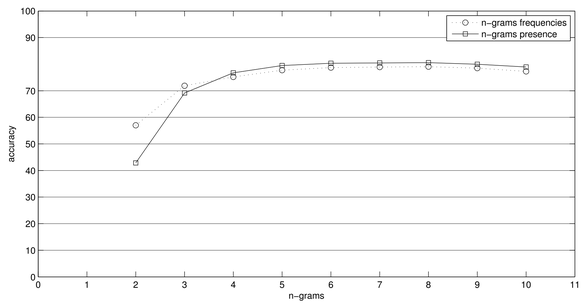

Figure : **10**-fold cross-validation accuracy on the train set for different ***n***-grams.

- The best result was obtained for $\hat{k}_{5-8}^{0/1}$ (with ***n***-grams from **5** to **8**)
- The **10**-fold cross-validation accuracy: **80.94%** (KRR $\lambda = 10^{-5}$)

Parameter Tuning for LRD Kernel

Method	Accuracy
KRR + K_{LRD_6}	**42.1%**
KRR + K_{nLRD_4}	**70.8%**
KRR + K_{nLRD_6}	**74.4%**
KRR + K_{nLRD_8}	**74.8%**

Table : Accuracy rates, using **10**-fold cross-validation on the training set, of LRD with different ***n***-grams, with and without normalization. Normalized LRD is much better.

Parameter Tuning for Kernel Combination

Method	Accuracy
KRR + $K_{nLRD_{6+8}}$	**75.4%**
KRR + $\hat{k}_{5-8}^{0/1} + K_{nLRD_{6+8}}$	**81.6%**
KRR + $(\hat{k}^{0/1} + K_{nLRD})_{6+8}$	**80.9%**

Table : Accuracy rates of different kernel combinations using **10**-fold cross-validation on the training set.

Results - Ranked 3rd in the closed NLI Shared Task

Method	Submission	CV Tr	Dev	CV T+D	Test
KRR + $\hat{k}_{5-8}^{0/1}$	Unibuc-1	**80.9%**	**85.4%**	**82.5%**	**82.0%**
KRR + $K_{nLRD_{6+8}}$	Unibuc-2	**75.4%**	**76.3%**	**75.7%**	**75.8%**
KRR + $\hat{k}_{5-8}^{0/1} + K_{nLRD_{6+8}}$	Unibuc-3	**81.6%**	**85.7%**	**82.6%**	**82.5%**
KRR + $(\hat{k}^{0/1} + K_{nLRD})_{6+8}$	Unibuc-4	**80.9%**	**85.6%**	**82.0%**	**81.4%**
KRR + $\hat{k}_{5-8}^{0/1} + K_{nLRD_{6+8}}$ + heuristic	Unibuc-5	-	-	-	**82.7%**

Table : Accuracy rates of submitted systems on different evaluation sets. The Unibuc team ranked third in the closed NLI Shared Task with the kernel combination improved by the heuristic to level the predicted class distribution.

Future Work - Already started

Method	Test
KLDA + $\hat{k}_{5-8}^{0/1}$	**84.0%**
KLDA + $K_{nLRD_{6+8}}$	**76.4%**
KLDA + $\hat{k}_{5-8}^{0/1} + K_{nLRD_{6+8}}$	**84.1%**

Table : Accuracy rates of systems based on KLDA (not submitted). KLDA improves accuracy because of unmasking.

An explanation for these results is needed

It will be addressed in future work...

popescunmarius@gmail.com raducu.ionescu@gmail.com