Native Language Identification: Improving The Quality Of Classification By Focusing On Keywords

Aishy Awawdi, Huda Abulel

Dept. of Computer Science, University of Haifa

aishyawawdi@outlook.com , Huda.Abulel@gmail.com

1 Abstract

We deal with the task of defining the native language of European authors who are fluent in English by researching their social media posts.

we try to improve the quality of the basic classifier by focusing on a small number of keywords. These are words that appear disproportionately in the texts of particular native language speakers, relative to their commonities throughout the corpus.

2 Introduction

The task of improving the quality of classification by focusing on Keywords is based on the research of Shuly Wintner, Gili Goldin and Ella Rabinovich, 2018 - (Native Language Identification with User Generated Content).

We discussed three tasks: -

1- Distinguishing between native and nonnative authors.
2- Determining the language family of the native language of nonnative authors.
3- Identifying the native language of nonnative authors.

The work is distinguished from the original article by three differences: -

-we worked in a larger data set which contains speaker's texts of 30 native languages (compared to 23 in the basic research).

- The text chunks for the classification is smaller. we researched two different sizes:

-each chunk consists of 10 sentences

-each chunk consists of 50 sentences

-after getting the results of each task from the three, we tried to improve the performance of classification in the out-domain.

The research was based on a large data taken from people writings on the social-media site 'Reddit'.

Each chunk of the data is classified to one of these languages: -

English, German, Albanian, Bosnian, Bulgarian, Croatian, Czech, Denish, Estonian, Finnish, French, Greek, Hungarian, Icelandic, Italian, Latvian, Lithuanian, Dutch, Norwegian, Polish, Portuguese, Romanian, Russian, Serbian, Slovakian, Slovenian, Spanish, Swedish, Turkish, Ukrainian

Corresponding to the 34 countries: -

UK, US, Australia, Ireland, Austria, Germany, Albania, Bosnia, Bulgaria, Croatia, Czech, Denmark, Estonia, Finland, France, Greece, Hungary, Iceland, Italy, Latvia, Lithuania, Netherlands, Norway, Poland, Portugal, Romania, Russia, Serbia, Slovakia, Slovenia, Spain, Sweden, Turkey, Ukraine.

Natural languages are divided into families, and in our corpus three distinct families are represented:

Latin, Germanic, Balto-Slavic

We will also include all native speaker text in a separate department, and native speakers of all other languages (Finnish, Turkish, Albanian, etc.) in their own department. In all, we will be assigned a five-class classification task.

We used linguistic and content features for the classification task and got an excellent result up to 96% accuracy for distinguishing between natives and nonnatives, and up to 70% in language family task and up to 45% for the NLI classification task.

3 Experimental setups

3.1 Dataset

As mentioned, the classification data gathered from 'reddit.com'. It is a site of forums on many varied topics, mainly in English. In some forums, writers can tag themselves with flair, which is a kind of metadata field. For example, in the news groups that dealing with Europe, the writers can indicate their country in the flair.

We used the final Reddit dataset released by Wintner, Goldin and Rabinovich,2018.

The training data is from such groups and the country of each writers is what the writer indicated in the flair. Thus, we got a considerable corpus of texts that we know their author's native language. We call this corpus "Europe," or "in-domain," and it will serve us both for training and testing.

And the testing data gathered from other groups, we call this corpus "Non-Europe," or "out-domain," and it will serve us for testing the classifier.

We filtered out data from multilingual countries (Belgium, Canada, and Switzerland) and countries with number of writers that are less than 40.

The final data that we got is made up of chunks, each writer of each country has several chunks consisting of 100 sentences.

3.2 Preprocessing

For each domain we took 20,000 chunks per state, and each chunk consists of 50 random sentences from random authors of the same country. We searched two different methods: -

-each chunk was consisted of 50 random sentences of a single random author.

-each chunk was consisted of 50 random sentences from different country's random authors (shuffling texts of various authors).

Thus, we got a data consisting of 680,000 chunks (34 * 20,000), And up to 34 million sentences.

Each chunk we tagged it according to the task itself (the author language, the family language, native or not).

Different countries with the same official language were tagged with the same language label.

3.3 Task

The main task is to improve the

Performance in the out-domain of the three tasks that we have listed earlier.

(Distinguishing between native and nonnative authors, determining the language family of the native language of nonnative authors, identifying the native language of nonnative authors).

Before the improving we got about 33% accuracy in the out-domains of identifying the native language task, and 50% in family language and 87% in the binary task.

3.4 Strategy

We used the Multinomial NB classifier from sklearn.naive_bayes.

for the in-domain prediction we used the cross_val_score with 10 folds.

The classification work we did with four main features:

POS n-grams and Function words that are content-independent, character n-grams and Bag-of-words that are content-dependent.

The improvement of the out-domain accuracy results is made by reducing the influence of the in-domain content on the work of the classifier in the out-domain.

It can be done since the in-domain discusses issues that related to Europe and related to the country itself.

For example: In the chunks of French writers, we noticed a lot of frequency of words like the names of cities (Paris...), names of landmarks (Evil tower...), names of president of this state and so on. for these words we call 'Key words'.

If we reduce the frequency of these words; their effect in the in-domain and in the out-domain will be reduced, so we expected to get slightly fewer good results in the in- domain but better results in the out-domain.

3.5 features

We used several features in all three tasks. In this section we describe these features.

3.5.1 Content features

Authors are more likely to write about topics that are related to their country and their culture, therefore content related features are expected to yield high classification results in in-domain compared to low classification results in out-domain. This is since the distinguished words of the in-domain are not actually remarkable in the out-domain.

3.5.1.1 Bag of words

This feature is the main content dependent feature. In which we deal with it to try to improve the quality of classification by focusing on keywords. we examine the keywords in in-domain and replace them with a unique symbol, such as "UNK".

3.5.1.2 Character tri-grams

We used the top 1000 most frequent character 3-grams in our dataset as features. For each chunk the value of a certain character 3-gram feature was the number of its occurrences in the chunk.

3.5.2 Content-independent features

In comparison to content features, content independent features are expected to be more effective when they are used out of-domain.

3.5.2.1 Function words

Function words are highly frequent words, which they were been selected without any relation to the text itself. Therefore, they don't reflect content.
we used about 400 function words taken from Volansky et al. (2015).

3.5.2.2 POS tri-grams

3-gram POS are reflecting the grammar of the text. Each author texts in English are affected in a way or another in his Native language grammar structure. Therefore, Using the 3-POS trigrams as features can help in NLI task. We chose the top 300 most frequent POS tri-

grams as features. For each chunk the value of a certain POS 3-gram feature was the number of its occurrences in the chunk.

3.6 Key words

We extracted 500 keywords per country, using the improved log-odds method (Jurafsky et al. (2014))

The method estimates the difference between the frequency of word '*w*' in two corpora *i* and *j* via the log–odds–ratio for *w*, $\delta_w^{(i-j)}$ which is estimated as:

$$\delta_w^{(i-j)} = log\left(\frac{y_w^i + \alpha_w}{n^i + \alpha_0 - (y_w^i + \alpha_w)}\right) - log\left(\frac{y_w^j + \alpha_w}{n^j + \alpha_0 - (y_w^j + \alpha_w)}\right)$$

Where n^i is the size of corpus *i*, n^j is the size of corpus *j*, y_w^i is the count of word *w* in corpus *i*, y_w^j is the count of word *w* in corpus *j*, α_0 is the size of the background corpus, and α_w is the count of word *w* in the background corpus.
Corpus i, represents the country that we interested in, corpus j, represents the all other countries and the Background corpus represents the two corpus i and j together.
We calculate this formula for each word in the corpus i.
In addition, we use of an estimate for the variance of the log–odds–ratio: -

$$\sigma^2\left(\delta_w^{(i-j)}\right) \approx \frac{1}{y_w^i + \alpha_w} + \frac{1}{y_w^j + \alpha_w}$$

The final statistic for a word is then the z–score of its log–odds–ratio: -

$$\frac{\delta_w^{(i-j)}}{\sqrt{\sigma^2\left(\delta_w^{(i-j)}\right)}}$$

We took 500 words with the highest z–score for each country.

the most characteristic words of each country get higher z-score.

The results of some countries are shown below.

France	**Germany**	**Italy**
france 68.5	germany 91.453	italy 49.003
french 66.8	german 78.981	italian 40.653
le 33.550	merkel 45.611	berlusconi 25.610
fn 31.739	spd 42.789	renzi 22.091
delisted 31.4	afd 42.174	italians 19.841
paris 29.897	cdu 41.861	m5s 17.683
sarkozy 29.43	asylum 34.858	grillo 15.701
macron 28.388	refugees 32.254	rome 15.373
pen 27.449	bavaria 32.234	trentino 13.698
les 26.539	berlin 29.565	milan 13.021
fillon 26.467	germans 28.039	veneto 12.771
cf 26.382	csu 25.646	pd 12.702
Hollande 25.7	fdp 25.456	naples 12.021
egregiously 21.5	linke 24.842	lega 11.348
de 20.461	chancellor 23.9	di 11.305
la 19.405	Refugee22.043	monti 11.032
. . .	. . .	. . .

3.7 Evaluation

We defined two evaluation scenarios: in-domain, where training and testing is done only on chunks from the European subreddits; and out-of-domain, where we train on chunks from the European subreddits and test on chunks from other subreddits, making sure they were authored by different users.

We report *accuracy*, defined as the percentage of chunks that were classified correctly out of the total number of chunks. we use the same evaluation strategy for the three tasks.

4 Results

We implemented the features that we discussed and evaluated the accuracy of the three classification tasks as we mentioned above.

The trivial baseline for the binary classification task is 50%, for language family classification 20%, and for the language identification task 3.3%.

4.1 Before masking the keywords

We used the mentioned four features with the data set consists of 20,000 chunk per country in two different methods described in Section 3.2.

Table 1 shows the accuracy obtained in-domain While each chunk was consisted of 50 random sentences of a single user. Table 2 shows the accuracy obtained out-domain.

Feature Set	Binary	Families	NLI
Char. 3-grams	71.23%	56.02%	65.3%
POS 3-grams	74.84%	36.74%	16.31%
Function words	81.26%	40.88%	22.27%
Bag-of-words	81.39%	84.06%	91.66%

Table 1: In-domain accuracy, chunk: single user

Feature Set	Binary	Families	NLI
Char. 3-grams	67.84%	39.61%	28.51%
POS 3-grams	76.7%	33.52%	10.71%
Function words	83.62%	37.94%	11.37%
Bag-of-words	87.92%	50.93%	33.63%

Table 2: Out-domain accuracy, chunk: single user

Table 3 shows the accuracy obtained in-domain While each chunk was consisted of 50 random sentences from different country's authors (shuffling texts of various users). Table 4 shows the accuracy obtained out-domain.

Feature Set	Binary	Families	NLI
Char. 3-grams	81.03%	71.84%	94.07%
POS 3-grams	83.32%	46.49%	29.97%
Function words	89.59%	52.02%	38.44%
Bag-of-words	88.43%	97.38%	99.92%

Table 3: In-domain accuracy, chunk: various users

Feature Set	Binary	Families	NLI
Char. 3-grams	83.52%	48.97%	40.02%
POS 3-grams	84.38%	40.4%	16.49%
Function words	89.66%	45.58%	16.45%
Bag-of-words	96.29%	70.84%	45.52%

Table4: Out-domain accuracy, chunk: various users

Evidently, all feature sets outperform the baseline, although some are far better than others. The feature that yields the best accuracy is Bag of words, with 96.29% accuracy on the binary task, 70.84% on the language family task and 45.52% on the NLI task. As expected, the content-based features yield relatively high results when the evaluation is in-domain.

POS 3-grams and function words yield better results when the evaluation was done by shuffling texts of various users. As we evaluate on chunks of single users, the personal style of the user may dominate the subtler signal of his or her native language.

we display a table that shows the classification results of NLI task on 20,000 chunk per country using bag of words feature on different chunk sizes while using chunks from one or various users:

Lines in chunk	shuffling	In-domain	Out-domain
50	V	99.92%	45.36%
50	X	91.66%	33.44%
10	V	80.53%	27.25%
10	X	69.74%	25.11%
100	V	99.98%	50.52%
100	X	95.21%	37.27%

Table 5: Out-domain, In-domain accuracy of the NLI task using bag of words

4.2 After masking the keywords

4.2.1 Expectation results

We extracted the top 500 keywords of each country and masked a different number (K) of these words while using bag of word feature for classification. we tried K in range 10 to 500.

Our expectation was to get better accuracy of NLI task in out-domain with less accuracy in the in-domain, (for some K-s).

4.2.2 Real results

We tried different K-s on the same size of Dataset and the same size of chunks. (20,000 chunks per country, 50 sentences per chunk, without shuffling the sentences)

K	In-domain	Out-domain
10	90.38%	32.61%
50	89.86%	32.5%
100	89.14%	31.58%
250	88.23%	29.86%
500	87.46%	28.54%

Table 6: Out-domain, In-domain accuracy of the NLI task (without shuffling the sentences)

As we can see the larger the K, the lower the accuracy in the two domains.

The reduction rate is slow, but we noticed that if we continue to increase the K (more than 500) the reduction rate become faster.

We also show results for different sizes of chunks (20,000 chunks per country, shuffling the sentences)

K	lines in chunk	In-domain	Out-domain
5	50	99.88%	38.55%
10	10	70.23%	21.97%
30	50	99.82%	35.21%

Table 7: Out-domain, In-domain accuracy of the NLI task (with shuffling the sentences)

5 Analysis

5.1 Before masking the keywords

Bag-of-words feature

This feature is very robust for classifying in the in-domain because in the in-domain the data is very distinguishable, as we said before, the in-domain discuss topics related to Europe, each writer will usually speak about his own country and then will use words that reflects that.

If we look at the keywords that are the results of the log-comparison of each country in the in-domain with the other

countries in the same domain, we will see how in each country its keywords are words that identify the country itself, for example in Germany the first keywords of the 500 were words like german, Merkel, Berlin, refugees ... words that are very identify Germany itself.

For that the presence of such words in each country will make this feature (Bag-of-words) more powerful and through it the classifier can more easily identifies the origin country of each speaker, therefore we got the highest accuracy results of the native language identification (NLI) in the in-domain.

We got much lower results in the out-domain for the NLI and that was expected.

Since the words that differentiate between the countries in the in-domain like the Key-words will not necessarily appear much in the out-domain as it appears in the in-domain, this will weaken the classification job in the out data set because the bag of words is built from such words.

5.2 After masking the keywords

The reducing in the accuracy of NLI in out-domain after masking the top K keywords in the in-domain is due to the large number of the keywords that also appear in the out-domain, thus masking these words will reduce the accuracy results in the two domains as reflected in section 4.2.2

If we train the classifier only with keywords; we achieve 40% accuracy out domain in compared with 33% while using the whole words. This may lead us to think that maybe the keywords exist in out domain in the same proportion as in the in-domain.

The appearance of a large portion of the keywords in the out-domain is not very surprising, because a lot of these words are still very likely to appear in subjects that are not necessarily related to europe. for example, words like 'Germany' or 'german' that are in the top of the keywords list of the Germany in-domain data are also expected to appear in germen people's other posts even if they are talking about different subjects that are not related to europe.

we extracted results of the log-comparison of each country in the out-domain with the other countries in the out domain and we notice many keywords that appear in the in-domain results of specific country are also shown in the results of its out-domain keywords with close distribution.

After getting the results that shown in section 4.2.2, we tried to cover a specific keyword instead of simply covering every time the top K keywords.

First, we chose to cover the words that doesn't appear completely in the out-domain, we extracted these words and received very little improvement in the out domain: it goes from 45.39% to 45.4% accuracy.

Then we tried to cover the words that are not much frequent in the out-domain, we chose the words that its frequency out domain is less than in the in-domain or it doesn't appear out domain.

another way was to choose a specific threshold; for example '10' and the keywords that appear less than 10 times in the out-domain (or didn't appear at all) will be our new keywords that we will cover later.

Since these new keywords are very frequent in the in-domain and much less frequent in the out-domain; we expected now to see much more improvement in the out-domain accuracy after masking these new keywords, But we got results that are not much different from the results of

covering the origin keywords: It goes from 45.62% to 41.64% in the out-domain.

We concluded that if the keywords appeared in the out-domain no matter if they were much less frequent or not, masking them in the in-domain will eventually reduce the classification accuracy results in the two domains data set.

So, we will really see an improvement on the results outside the domain when these keywords are covered only if a large part of these keywords are completely not exist in the outside domain.

To demonstrate that, we removed all the keywords in the out-domain data set and re-experiment covering the K keywords in the in-domain.

with k=0 i.e. not covering any keyword in the in-domain, the NLI accuracy in the out-domain that we got with bag of words feature was around 5% and that verifies that a lot of the keywords appear in the out-domain.

After that we re-tried the same K ranges as we did in section 4.2.2.

Now we have really seen the effect of covering words in the internal field and we noticed that the more we cover keywords the higher the accuracy in the external field, the accuracy rose from 5% to 24.65% when 500 keywords of each country were covered in the internal field.

6 Conclusion

Covering words really leads to higher accuracy of classification in the external field only if a large part of these words is not exist at all in the external field.