UPPSALA
UNIVERSITET

Language Identification of Person Names using Cascaded SVMs

Erik Sterneberg

Uppsala University
Department of Linguistics and Philology
Språkteknologiprogrammet
(Language Technology Programme)
Bachelor's Thesis in Language Technology

April 30, 2012

Supervisor:
Mats Dahllöf

Abstract

A number of state-of-the-art approaches to Language Identification of person names use Support Vector Machines (SVM). Using this approach, a large number of languages in the data set can make the feature space outgrow software restrictions, not allowing the use of all useful features. This paper presents a novel technique for using binary classifiers in cascade as a post-processing step to a multi-class classifier to improve performance.

A data set of 10 languages was extracted from various online sources, most prominently e-petition sites. Using this data, a multi-class SVM classifier and ten binary classifiers were trained. Running binary classifiers in the prediction order given by the multi-class classifier may help correcting mistakes, the theory being that the binary classifiers can be trained on more data and therefore have superior performance on individual languages.

The hypothesis could not be fully tested, but experimental results are encouraging.

This paper also explores new kinds of features to be used in training. One meaningful new feature is combinations of n-gram counts. Finally, a good trade off between training set size and features used is derived through experimentation.

Address for correspondence: Beijing City, Dongcheng District, Juer Hutong no. 85, Apartment 512, Li Jianglong, 100009 CHINA. E-mail: erik.sterneberg@gmail.com

Contents

Acknowledgements

In no special order of importance, the author would like to thank the following for their contributions to this paper:

Jim Breen for the use of the Japanese dictionary ENAMDICT (`http://www.edrdg.org/edrdg/newlic.html`); Marcus Edvinsson, technical consultant; Mats Dahllöf, supervisor; the proofreaders Oliver Stanyer and Anna Gustafsson; and finally Christina Tånnander from Talboks- och Punktskriftsbiblioteket who got me interested in Language Identification and whom without this paper would not have been written.

1 Introduction

1.1 Purpose of thesis

The purpose of this thesis is to investigate if and to what extent a Language Identification System for person names using Support Vector Machines can benefit from using a multi-language classifier in concert with a set of binary language classifiers. The binary classifiers will be used in sequence on every test example as a post-processing step on the output of the multi-language classifier, terminating whenever a classifier accepts the test example as a member of the language it is trained to detect. The order in which to run the binary classifiers on each test example is given by the multi-language classifier, which scores the similarity between the test examples and the training examples for each language.

This paper also experiments with corpora sizes to find the optimal trade off between training set size and number of features for the SVM, and explores the effects of some features not used in previous works.

One potential application of a language classifier for person names is as a preprocessing step to machine transliteration in TTS (Text-To-Speech). A high-performing language classifier might help a TTS system to increase intelligibility for person names in foreign languages. The Language Identification (LID) system in this paper has been developed in cooperation with Talboks- och Punktskriftsbiblioteket, a Swedish government body that, in collaboration with local libraries, provides access to printed materials for people with reading disabilities, some of the materials being produced by a TTS-engine. TPB has contributed to this thesis with Swedish news paper articles to be used as test data.

1.2 Outline

The rest of this paper is organized as follows: the chapter Background presents a selection of previous work on the topic of LID using SVMs; the chapter Method and Data describes the resources and corpora used in this paper, along with a list of feature categories used during testing, and finally the eponymous technique cascaded SVMs is presented; the results of evaluations are presented in the chapter Evaluation; the results of the evaluation are discussed in the chapter Discussion; the paper is concluded in the chapter Conclusions.

2 Background

2.1 LID in the general case

The problem of Language Identification is defined as the task of identifying the language a given document is written in. Language identification is not a "solved task" outside controlled isolated experiments with small numbers of languages. As evaluated in realistic conditions, perfect language identification is still a long way off (Baldwin and Lui, 2010).

LID has been applied in a number of contexts, one of which is Cross Language Information Retrieval (CLIR). Gottron and Lipka write: "The difficulty of a Cross Language Information Retrieval (CLIR) system is to find relevant documents across language boundaries. This induces the need for a CLIR system to be capable of doing translations between documents and queries. If the system has to handle more than one language for queries or documents, it additionally needs to be able to detect the language of a text." It can also be used to spot foreign words in a text in order to increase parsing performance (Baldwin and Lui, 2010).

2.1.1 Previous work

One technique that works quite well on running text is the Small Word Technique (SWT), described by Ferreira da Silva and Pereira Lopes (2006) as relying on the intuition that function words are good clues for guessing the language. Training on corpus data, tokens meeting certain length and frequency requirements are considered functions words and given a probability higher than that of the non-function words. The probability that a text is written in a certain language is then given by multiplying the token probabilities for each language.

Based on the description given here, SWT would not work well on texts containing only non-functions words. Another method developed by Cavnar and Trenkle (1994), perhaps better suited for performing LID on such texts, calculates and compares n-gram frequency profiles. A profile for a language is created by counting all (overlapping) n-grams for a certain range of n. A similar profile is created for the document to be classified. A distance measure they call "out-of-place" measure is then used to determine of which category the document is a member. They write: "For each N-gram in the document profile, we find its counterpart in the category profile, and then calculate how far out of place it is. [...] The sum of all of the out-of-place values for all N-grams is the distance measure for the document from the category. "(Cavnar and Trenkle, 1994)

The system developed in this paper uses Support Vector Machines (SVMs).

Baldwin and Lui (2010) describes SVMs as one of the most popular methods for text classification, the reason being that they can automatically weight large number of features, capturing feature interactions in the process.

One system developed by Kruengkrai et al. (2005) shows that in experiments on news text, SVM classifiers outperform the n-gram comparison method first presented in (Cavnar and Trenkle, 1994). For 17 European languages the two methods achieved an average accuracy of 99.7% and 90.2% respectively on text samples with the average length of 50 bytes.

Botha et al. (2006) perform LID experiments using SVMs for 11 South African languages. The authors report that "When 50 words are available, SVM is virtually flawless at this task – an error rate of 0.3% is achieved." Using a window size of 15 words the error rate exceeds 5%.

The technique developed by Cavnar and Trenkle (1994) and the SVM-based techniques would work on texts containing only non-function words, but are in turn more sensitive to noise in the data, such as misspelled or foreign words. Baldwin and Lui (2010) writes that "... the models are brittle when the assumption of strict monolingualism is broken, or when the document is dominated by extra-linguistic data."

2.2 LID of person names

LID of person names is a special case of LID with its own challenges. One of its uses is in speech processing. Chen et al. (2006) write: "Letter-to-sound (LTS), which generates pronunciations of words out of the vocabulary (OOV), is very important in both speech synthesis and speech recognition. LTS for person names are the most important and difficult part. [...] To improve the performance of LTS, identifying the origin of language is critical."

LID of person names is made difficult due to the inherent properties of names, such as being international in nature and not language-specific. One name might be common in more than one language. Another problem is that many countries have diverse societies with many inhabitants with names not matching any official language of the region. Names are also typically very short, not long enough for analysis. Also, a full name can contain name-words from different languages (Nobesawa and Tahara).

These characteristics should be kept in mind when interpreting evaluations of a LID system. From a machine learning point of view names can be members of several classes, but instead of saying that this makes them harder to classify one could argue that too fine-grained classification is not always meaningful. For some purposes perhaps it would suffice to assign them to a group of languages.

2.2.1 Previous work

Bhargava and Kondrak (2010) uses SVMs to do LID on person names with n-gram counts as features. In one test with 13 European languages their system gets an accuracy of 79.9% on fullnames. Unfortunately neither the individual accuracies for the languages nor the languages themselves are listed in the paper, making it impossible to compare their system with the one presented in this paper. In this test, the amount of training data used was close to 15,000

full names for 13 languages, meaning on average about 1,150 full names per language.

In another test with only Chinese, Japanese and English in the data set the system gets an accuracy of 97.6%. Explaining this second, much higher score, they write: "One reason that they [the languages in the second test] do better is because of the smaller number of classes. We can further see that the languages in question are very dissimilar, making the problem easier". For these languages, less computationally expensive LID methods have been reported quite accurate (Bhargava and Kondrak, 2010).

3 Method and Data

The LID-system developed in this paper makes use of a multi-class SVM. A standard SVM is a binary classifer, predicting of which of the two classes the input is a member. It does this by first creating a high-dimensional space and separating training examples as points in the space using a so-called "hyper-plane", then making predictions on the input depending on their position in the space. Some SVMs are constructed in a way so as to allow more classes than two, such as the software used in this paper; SVM^{Multiclass} by Thorsten Joachims (`http://svmlight.joachims.org/svm_multiclass.html`). This software is free for scientific use.

A multi-language classifier was trained by extracting and assigning ID numbers to features from the training data compiled by the author of this paper, feeding the extracted features into SVM^{Multiclass} and letting it create a model, automatically setting weights for the features. Equal amounts of training data from each language in the data set was used.

Ten binary classifiers - one for each language - were similarly trained but use different ratios of training data , see section 3.5 on page 13. These classifiers use the same software as the multi-language counterpart, but with the number of classes reduced to two.

All scripts performing tasks such as downloading and preprocessing data, extracting features and evaluating classifier output have been written solely by the author of this paper, with the exception of the script scoring features by Information Gain (see section *Feature scoring and selection* on page 13). All scripts were written in Python.

3.1 On language selection

The main guideline when choosing languages at the start of construction of the data set was to find languages of sufficiently similar ortography and partly overlapping name sets in order to avoid trivializing the task. The table 3.1 on page 11 lists the selected languages. For many names it might not be meaningful to assign it to a single language within a language family, only to the language family itself, as it might be member of more than one. For this reason the classifier will also be evaluated as a classifier of language families and not only of individual languages.

3.2 Corpus construction

Only full names are used as training, validation and test sets in this paper, and the term "name" will here on throughout the paper be, unless stated otherwise, used to refer to full names, consisting of at least two words. One of the aims of this paper is to test how performance rises with the size of the training set, trying to find the optimal trade off between training set size and number of features. A data set orders of magnitude larger than that used in (Bhargava and Kondrak, 2010) was therefore collected.

The composition of a name corpus is no less important than its size. Frequencies of given names and surnames within the corpus should accurately reflect the demographic cohorts which they are supposed to represent concerning ethnicity, age, social strata etc. In the case of the corpora used in this paper, that group is the general populace, with people of different ages and ethnicities.

With the exception of Japanese, all corpora used in this paper have been extracted from e-petition sites, such as `http://www.gopetition.com`, `http://www.ipetitions.com` and `http://www.namninsamling.com`. These corpora each consist of several petition lists but have had duplicates removed (full names only). Petitions were not considered if the signers were suspected to represent only a small part of the populace, e.g. when the language of the majority of names did not match any offical language of the region. Name (frequency) lists with given names and surnames listed separately such as name statistics in census data have not been used in this paper as they typically yield very poor results [1].

The Japanese corpus was created from the ENAMDICT, a Japanese electronic dictionary file compiled by Jim Breen that is closely associated with the EDICT Japanese dictionary project. The dictionary is available under the Share-Alike license. By the definition given above, the Japanese corpus is indeed not a corpus but a list of names. This seems to have no negative impact on accuracy however, since Japanese ortography is very uniform with a very small set of possible syllables, making the classifier perform with remarkable accuracy concerning this language.

Despite the efforts mentioned above it must be assumed that the corpora used in this paper fails to achieve a good balance in at least one respect — age of the signers, since the very young and the very old are likely to be underrepresented due to differences in Internet habits. It also conceivable that the corpora fails in other respects, such as ethnicity, but any serious attempts at looking in to such matters would be too time-consuming. These assumed short-comings must be accepted due to the lack of time and freely available corpora of sufficient size.

[1] This has been experimentally confirmed by the author of this paper, using English and Swedish census lists. The reason for this is believed to be that when no names (given and surnames) appear more than once in a list, features extracted from them are weighted incorrectly - names that are in reality infrequent are given unwarranted prominence and vice versa - thus affecting performance negatively.

Language family	Language	Abbreviation in paper
Germanic	German	de
	Danish	dk
	British English	en
	Swedish	sv
	Dutch	nl
Japonic (or Japanese-Ryukyuan)	Japanese	jp
Slavic	Polish	pl
Romance	Portugese	pt
	French	fr
	Italian	it

Table 3.1: Languages in data set

3.2.1 Normalization

In comparison to names appearing in newspapers, journals and books the data extracted from e-petition sites is "flawed" in the sense that it contains deviations from capitalization standards and numerous non-alpha characters. There are also many non-person names such as names of organizations or even whole sentences that need to be purged. The following measures have been taken to correct these problems:

Truecasing A high percentage of the data is written in all lowercase or all uppercase letters. Working under the assumption that word-initial n-grams have a higher discriminant ability than intra-word n-grams the data has been truecased so as to distinguish these. Truecasing is determining the proper case of words where such information is unavailable. This means capitalizing the first letter of every word except for words like *von* and *la* which are never capitalized. It also entails some word-internal letters, like the *A* in *McAfee*. Word-internal capitalization has not been done one the data set however since it requires too much language-specific knowledge. Tests performed by the author of this paper confirm that truecasing increases performance.

Removal of non-alpha characters Non-alpha characters that do not occur in words, such as " ; * ( ) [ ] @ : = + have been removed from the data set.

Removal of too short names Names that do not consist of two words or more with at least one letter in the first word and two in the second were removed from the data set.

Removal of too long names Strings with more than 7 words were suspected to be sentences and not names and were removed from the data set. It appears as though a number of authentic Portugese names exceeds this limit and were removed. However, if the limit was made higher more non-names would pollute the data. Making the limit lower on the other hand makes a stronger assumption on the length of names that would require further study.

feature category	description
n-gram counts (0 < n < 6)	every n-gram and their frequencies
words	every word in the training example
gapped n-grams (2 < n < 5)	e.g. the 3-gram *M_r_a* from 'Maria'
consonant sequences	for every word the consonants are extracted and separated by a '_', such as *nd_rss_n* from the word *Andersson*
combined bigram counts	combination of every bigram and its count in a name

Table 3.2: Features

Diacritic marks were not removed.

3.3 Features

A great number of different kinds of features have been tested during the development of the classifier. Most do not give a significant increase in performance. The single most useful feature category is the n-gram or counts of n-grams. 3-gram count features for instance would for the name *Henrik Eriksson* be the strings:

```
_He1, Hen1, enr1, nri1, rik2, ik_1,
_Er1, iks1, kss1, sso1, son1, on_1
```

where space is written as underscore for clarity and the trailing digit after the 3-gram represents the count of the 3-gram in the whole string. By themselves n-gram counts enable the SVM to perform almost as well as together with other more advanced features. The features used to train the classifier are listed in table 3.2 on page 12. The second best feature category is the combined n-gram count. The powerset of n-gram count features within every training example grows exponentially with the size of `n`, and due to hardware constraints nothing above `n = 2` has been properly tested during the development of the multiclass classifier. In fact, using compound n-gram count features where `n = 3` will for all ten languages generate so many features that even the feature selection script (see section Feature scoring and selection on page 13) cannot handle to score and reduce them in number on the computers used in this study. Tests show that by using combined 3-gram count features, a dramatic decrease in training data is required, which only serves to lower overall performance. The combined 3-gram counts are used instead of the combined 2-gram counts however, as a feature category when training the binary classifiers (see section 4.2.4 on page 17). Other useful features include the words themselves, although these only have any impact when words are longer than the longest n-gram.

3.4 Feature scoring and selection

Maintaining a reasonable amount of training data when training the 10-language classifier restricts the number of features categories to be used simultaneously. In machine learning there are several algorithms to score and select an optimal subset of features. Using such an algorithm would allow for more kinds of features to be extracted from the training and test set, while keeping only a subset not exceeding the allowed limit of SVM^{Multiclass} [2]. A well known feature scoring metric called Information Gain (IG) has been tested during this study. IG scores a feature according to how indicative its presence or absence is on membership in the various classes. A script using IG has been implemented and tested in this study that allowed for n-gram counts where `n > 8`. Extensive testing shows however, that performance does not increase by the use of IG and n-gram counts where `n > 8`. This might mean that the implementation of the algorithm is incorrect, but it is also reasonable to assume that the model is saturated with information with n-gram count features for `0 < n < 7` and anything beyond this contribute little.

Experiments with other modifications such as not turning the least frequent n-grams into features have had non-significant positive effect on performance. This is not surprising as the SVM works well with large amounts of features. Results from these modifications have been excluded from this paper.

As mentioned in the Features section there is one kind of feature that could prove highly useful - combined 3-gram counts. The feature selection script could unfortunately not handle the vast amount of features it produces and so this kind of feature could not be properly tested.

3.5 Cascaded SVMs

The training data size per language and the number of languages in the data set governs what feature categories will fit into memory when training the classifier.

Useful features such as combined 3-gram counts could not be used when training the 10-language classifier in this study. When adding more languages to the data set, the amount of features needs to be reduced further, resulting in a performance drop. This paper proposes the use of binary classifiers used in sequence, or cascade, after the multi-class classifier to diminish the adverse effect of adding languages to the data set.

A binary classifier trained on 5000 names for the target language and 5000 names from the other languages in the data set combined is assumed to have superior performance over any multi-language classifier when determining whether or not a test example is a member of that class. This is due to the smaller total size of the training data, enabling the SVM to handle more feature categories, such as the combined 3-gram counts.They can not however determine of which language the test example is a part if several classifiers claims ownership, since the scores produced by each classifier are not useful for comparison.

In the system used in this paper, the multi-language classifier is first used to give each language a similarity score for each test example, establishing

[2]The software can make use of about 1 million features.

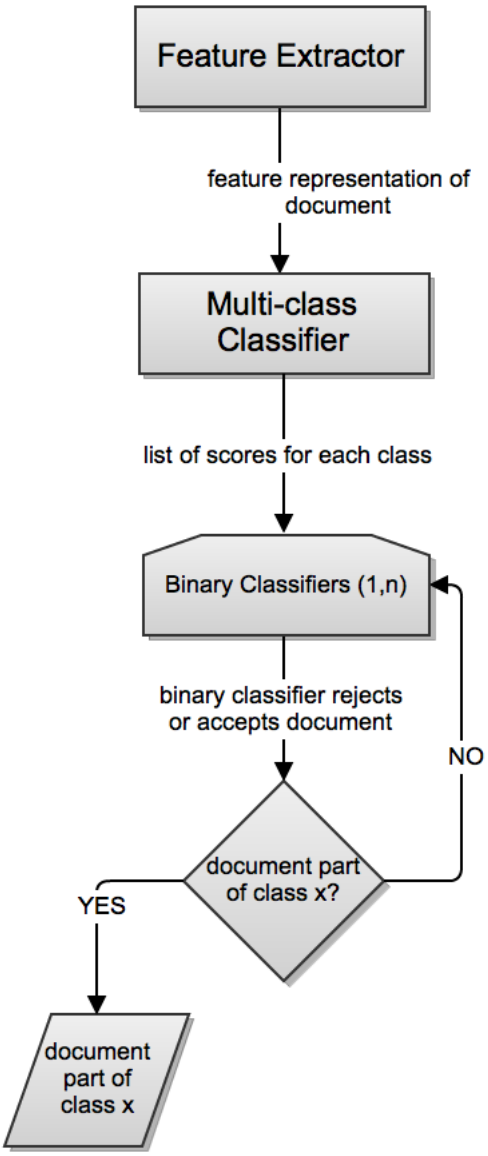

Figure 3.1: Workflow of LID system where n is the number of languages

an order in which to run the binary classifiers on each example. Running the binary classifiers in any other order will produce biased results. The binary classifiers are then used on each test example in the order given by the multi-language classifier, from highest to lowest score, until one of them accepts the test example as a member of its class, looping anywhere between 1 and n times, where n is the number of binary classifiers. Any test case that is rejected by all binary classifiers is automatically assigned the original prediction made by the multiclass classifier. This, however, never happened during testing. See the workflow of the system in 3.1 on page 14.

4 Evaluation

4.1 Corpora sizes used in testing

When training a classifier for all ten languages not all available data could be used. The reason concerns software constraints. The amount of training data SVM^{Multiclass} can handle partly depends on the number of features used. It seems to be more important to have the right kind of information extracted than to have a very large training set. Training on tens of thousands of names per language does not generally give a significant increase in performance compared to a classifier trained on just a few thousand names per language. A good compromise seems to be to use about 5000 names per language.

4.2 Results

All tests have been performed using 5-fold crossvalidation unless explicitly stated otherwise. A subset of 6250 names per language was randomly selected from the much larger constructed data set and split into 5 parts. Five consecutive runs performed with the same features but with rotated test and training sets together constitute a single test. The classifier will be evaluated using the standard measurements precision, recall, F-score and accuracy which is the percentage of the test examples classified correctly.

4.2.1 Performance of cascaded SVMs

Using the features described above and running a 5-fold crossvalidation test where not only the multi-language classifier but also the binary classifiers are retrained for each part of the test gave a modest increase in performance - only 0.3 percentage points. This is of course due to the fact that almost identical features had been extracted in training. Performing similar tests, but reducing the amount of features available for training the multi-language classifier predictably enabled the cascade binary classifiers to boost performance by several percentage points depending on how low the average accuracy of the multi-language classifier was.

4.2.2 Performance of multi-language classifier using best features

Table 4.1 on page 16 shows a confusion matrix over the distributions of guesses SVM^{Multiclass} trained on all languages makes on the test data. The average

l\p	de	dk	en	fr	it	jp	nl	pl	pt	sv
de	**75.71**	1.92	5.46	3.97	2.42	0.38	4.18	2.96	0.37	2.64
dk	6.86	**77.15**	3.74	2.22	1.12	0.58	2.40	0.66	0.14	5.12
en	3.70	0.75	**80.16**	6.13	2.54	1.06	2.46	1.30	0.46	1.44
fr	3.38	0.45	5.42	**82.85**	3.50	0.40	1.94	0.69	0.50	0.88
it	0.86	0.14	1.14	0.98	**95.17**	0.21	0.37	0.46	0.37	0.30
jp	0.00	0.02	0.11	0.02	0.10	**99.60**	0.06	0.08	0.00	0.02
nl	3.71	1.70	3.78	3.52	1.17	0.48	**83.36**	0.82	0.16	1.31
pl	0.98	0.02	0.54	0.19	0.40	0.14	0.11	**97.38**	0.03	0.21
pt	0.16	0.16	0.38	0.51	2.08	0.06	0.13	0.13	**96.32**	0.06
sv	6.05	3.25	4.26	1.66	2.58	1.06	1.94	1.63	0.38	**77.20**

Table 4.1: Confusion matrix over the distributions of predictions in % for each language (evaluated on corpus data)

Language	Precision %	Recall %	F-score %
jp	95.80	99.60	97.66
pt	97.55	96.32	96.93
pl	91.79	97.38	94.50
it	85.68	95.17	90.17
nl	85.99	83.36	84.65
dk	90.18	77.15	83.16
fr	81.21	82.85	82.01
sv	86.57	77.20	81.61
en	76.37	80.16	78.21
de	74.69	75.71	75.18

Table 4.2: Precision, recall and F-score for each language measured in % (evaluated on corpus data)

accuracy was 87.04%. Precision, recall and F-score for the ten languages are shown in table 4.2 on page 16. Similar scores for the same test but evaluating precision, recall and F-score for language families are shown in 4.3 on page 17. These numbers were calculated by relaxing the evaluation criteria, counting a prediction within the correct language family as being correct.

4.2.3 Performance on different training set sizes

A model could not be trained on more than 5000 names per language using the features in table 3.2, but tests with a reduced feature set and more training data all resulted in drops or insignificant increases in performance. The results of these tests are therefore excluded from this paper.

Test results show that the classifier in this study benefits from amounts of training data greater than what was used in (Bhargava and Kondrak, 2010). The graph 4.1 on page 17 shows how performance increases with available training

Language	Precision %	Recall %	F-score %
Japonic	95.80	99.60	97.66
Slavic	91.79	97.38	94.50
Germanic	96.27	92.05	94.11
Romance	90.52	94.09	92.27

Table 4.3: Precision, recall and F-score for each language family measured in % (evaluated on corpus data)

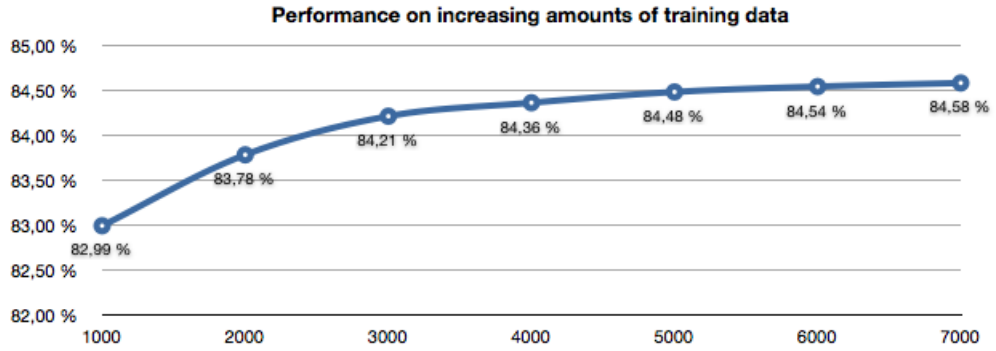

Figure 4.1: Performance measured in average accuracy in % with increasing amounts of training data available per language (multi-class classifier, 10 languages)

data, using first only the first 1000 names in every language corpus then adding more and more. This test was run using a subset of features, as $\text{SVM}^{\text{Multiclass}}$ cannot handle all useful features generated from a training data of more than about 5000 names per language.

4.2.4 Performance of binary classifiers

The tests for the binary classifiers have also been performed using 5-fold cross-validation. Ten separate data sets were constructed, each consisting of two classes with 6250 randomly selected names from a language's corpus in one class and about 700 names per corpus from the remaining 9 languages' corpora in the second class. The performance of the classifiers are shown in table 4.4 on page 18.

4.2.5 Evaluation on newscorpora

As a final test of performance a new test set constructed from a Swedish news corpus was constructed and classified using the multi-language classifier and the Swedish binary classifier separately. The test set contained 188 names manually extracted from 30 randomly selected articles. Each name was assigned a class based on the country of residence of the person, not based on country of birth. Names belonging to languages not in this study were used for the binary classifier only. American English names were labeled as British English. This

Language	Accuracy %	Precision %	Recall %	F-score %
jp	99.85	99.78	99.76	99.77
pt	98.44	99.38	97.33	98.34
pl	98.17	98.50	97.70	98.10
it	96.44	96.12	96.66	96.38
nl	92.47	93.35	91.34	92.33
fr	91.37	91.98	90.56	91.26
dk	91.23	93.87	88.13	90.91
en	89.90	88.60	91.52	90.04
sv	90.25	91.99	88.10	90.00
de	89.44	88.76	90.27	89.50

Table 4.4: Performance of binary classifiers (evaluated on corpus data)

Lang	de	dk	en	fr	it	jp	nl	pl	pt	sv	Prec. %	Rec. %	F-score %
de	**2**	0	0	0	0	0	0	0	0	1	20.00	66.67	30.77
dk	0	**0**	0	0	0	0	0	0	0	0	0.00	0.00	0.00
en	0	0	**11**	1	0	1	0	0	0	1	57.89	78.57	66.67
fr	0	0	0	**0**	0	0	0	0	0	0	0.00	0.00	0.00
it	0	0	0	0	**3**	0	0	0	0	0	60.00	100.00	75.00
jp	0	0	0	0	0	**0**	0	0	0	0	0.00	0.00	0.00
nl	0	0	1	0	0	0	**0**	0	0	0	0.00	0.00	0.00
pl	0	0	1	0	0	0	0	**8**	0	0	88.89	88.89	88.89
pt	0	0	0	0	0	0	0	0	**0**	0	0.00	0.00	0.00
sv	8	2	6	1	2	1	5	1	0	**105**	98.13	80.15	88.24

Table 4.5: Confusion matrix and performance of multi-language classifier on news corpus in % for each language

Language	Precision %	Recall %	F-score %
Rom	42.86	100.00	60.00
Sla	88.89	88.89	88.89
Ger	99.30	95.30	97.26

Table 4.6: Performance of of multi-language classifier in % for each language family (evaluated on news corpus data)

might potentially have adverse effects on performance of the classifier when identifying English names.

The test results shown in 4.5 on page 18 show a much higher F-score for Swedish than in previous tests. The average accuracy for the multi-language classifier was 69.05%. The Swedish binary classifier scored an average of 86.84%.

Language / predictions	sv	non-sv	Accuracy %
sv	**119**	10	92.25
non-sv	13	**44**	77.19

Table 4.7: Performance of binary classifier on news corpus

5 Discussion

The effect of postprocessing the result of a multi-language classifier with cascaded binary classifiers could not be properly tested. When more languages are added to the training set performance will drop since less features can be extracted from each language corpus and still fit into memory. The hypothesis is that binary classifiers trained on a full set of the most useful features can in part compensate for this drop in performance.

A binary classifier trained on names from the target language and as many names from other languages in the data set combined has superior performance over any multi-language classifier when determining whether or not a test example is a member of that class, provided that the size of the data set does not allow for the multi-language classifier to be trained using all beneficial features and a substantial training set size.

Features extracted from the collected data set using almost all of the best features did fit into memory however, and due to time constraints, data for more than 10 languages could not be collected. Reducing the amount of features extracted from each of these without adding more languages, and then running the binary classifiers in cascade does however provide a significant boost to performance. Although these results cannot verify the hypothesis, they are indicative of the outcome of future experiments.

Experiments confirm that features other than simple n-gram counts are beneficial to building the SVM model. The most noteworthy of those features are the individual words in a name and combined n-gram counts for `1 < n < 4`. Combined n-gram counts for `n > 3` could not be tested due to the sheer amount of features that would be extracted from even a very small training data.

It is also confirmed that making training corpora larger than what was used in (Bhargava and Kondrak, 2010) increases performance. The model seems to be saturated with data at around 5000 names per language. Decreasing the amount of training data and instead increasing the amount of features extracted only led to a drop in performance.

This paper also evaluates the performance of a person name SVM LID-system on news corpora. The names in this small corpus being manually labeled, the corpus enables a necessary "real world" evaluation. The average accuracy of the multi-language classifier was only 69.05%, but the small size of this test set does not allow one to draw too many conclusions from the results. The average accuracy of the Swedish/non-Swedish classifier performed a few percentage points worse than in the crossvalidation test: 86.84%.

5.1 Conclusions

This paper has presented a novel technique for using binary classifiers in cascade as a post-processing step to diminish the adverse effects of greatly expanding the number of languages in the data set. The hypothesis could not be fully tested but the experimental results are encouraging. Confirmation of the hypothesis is left as a task for the reader.

This paper has also shown that around 5000 full names per language is an optimal trade off between training set size and number of features when training an SVM to perform language identification of person names. Finally, a new helpful kind of feature is introduced: the combined n-gram feature. Using not only n-grams or n-gram counts as features but also combinations thereof seems to be a meaningful feature category.

Bibliography

Timothy Baldwin and Marco Lui. Language identification: the long and the short of the matter. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 229–237, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 1-932432-65-5. URL `http://portal.acm.org/citation.cfm?id=1857999.1858026`.

Aditya Bhargava and Grzegorz Kondrak. Language identification of names with svms. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 693–696, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 1-932432-65-5. URL `http://portal.acm.org/citation.cfm?id=1857999.1858101`.

Gerrit Botha, Victor Zimu, and Etienne Barnard. Text-based language identification for the south african languages. In *17th Annual Symposium of the Pattern Recognition Association of South Africa, Parys, South Africa, 29 Nov - 1 Dec 2006*, page 7, 2006.

William B. Cavnar and John M. Trenkle. N-gram-based text categorization. In *In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, 1994.

Yining Chen, Jiali You, Min Chu, Yong Zhao, and Jinlin Wang. Identifying language origin of person names with n-grams of different units. *Proc ICASSP*, 1: 729–732, 2006. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1660124`.

Joaquim Ferreira da Silva and Gabriel Pereira Lopes. Identification of document language is not yet a completely solved problem. In *Proceedings of the International Conference on Computational Inteligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce*, pages 212–, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2731-0. doi: 10.1109/CIMCA.2006.117. URL `http://portal.acm.org/citation.cfm?id=1191826.1192273`.

Thomas Gottron and Nedim Lipka. A comparison of language identification approaches on short, query-style texts.

Canasai Kruengkrai, Prapass Srichaivattana, Virach Sornlertlamvanich, and Hitoshi Isahara. Language identification based on string kernels. In *In Proceed-*

ings of the 5th International Symposium on Communications and Information Technologies (ISCIT-2005, pages 896–899, 2005.

Shiho Nobesawa and Ikuo Tahara. Language identification for person names based on statistical information. In *Proceedings of PACLIC, te 19th Asia-Pacific Conference on Language, Information and Computation*.