Discriminating similar languages with token-based backoff

Tommi Jauhiainen
University of Helsinki
@helsinki.fi

Heidi Jauhiainen
University of Helsinki
@helsinki.fi

Krister Lindén
University of Helsinki
@helsinki.fi

Abstract

In this paper we describe the language identification system built within the Finno-Ugric Languages and the Internet project for the Discriminating between Similar Languages (DSL) shared task in LT4VarDial workshop at RANLP-2015. The system reached fourth place in normal closed submissions (94.7% accuracy) and second place in closed submissions with the named entities blinded (93.0% accuracy).

1 Introduction

In the Finno-Ugric Languages and the Internet project[1], our aim is to harvest texts written in small Uralic languages from the internet. The project is funded by the Kone Foundation from its language program, which is especially targeted to support the research of Uralic languages (Kone Foundation, 2012). We are particularly interested in gathering material written in the smaller languages, instead of the three largest Uralic languages: Hungarian, Finnish and Estonian. As part of the project, we are developing methods for language identification which are needed to find the relevant texts among the billions of files we are downloading. At the moment, we have a list of 38 relevant languages based on the ISO 639-3 division of the Uralic languages (SIL, 2013). Some of the relevant languages, such as Livvi-Karelian and Ludic, two Finnic languages used in the north-western Russia, are very close to each other. However, the closeness between relevant languages is not as great a problem as the closeness between relevant and irrelevant languages. For example there are many dialectal variations of Finnish which are written differently from the standard Finnish and are actually closer in orthography to some of the very close languages, such as Tornedalen Finnish, than the standard written Finnish. This has led us to introduce separate language models for some of the Finnish dialects. An even greater problem for us is the large number of pages we have found which are written in a language not known to our language identifier (which at the moment has models for 395 languages and variants) or which consist mostly of lists of model abbreviations. Some of the character combinations used in the abbreviations tend to be quite common in some of the relevant languages and are therefore identified as such when the language identifier is forced to choose between languages it knows. Therefore, the opportunity given by the second version of the DSL shared task (Zampieri et al., 2015) to research unknown language detection has been very welcome.

[1] http://suki.ling.helsinki.fi

2 Language identification

The problem of discriminating similar languages given in the DSL shared task is an instance of monolingual language identification. The aim in monolingual language identification is to give one language label to a mystery text. This is different from multilingual language identification, where the mystery text can be labeled with several language labels. An extensive review of the work done in the area of discriminating between similar languages can be found in the report of the first edition of the DSL shared task (Zampieri et al., 2014).

The shared task also includes a group containing texts written in a set of unknown languages to which no training material is provided. Most existing language identifying methods can only categorize between languages they are trained for and do not have the ability to label the text as an unknown language. In order to detect the unknown language the methods usually need to have some

notion of how well they are performing.

2.1 Token-based backoff

The basic language identifier used in this work was developed by Jauhiainen (2010) for his master's thesis. We call the method it uses the *token-based backoff*. In token-based backoff, the text is tokenized and the tokens t are numbered from 1 to the total number of tokens $|m^t|$ in the mystery text m, so that identical tokens can occur several times. The probability of each token $t_1 ... t_{|m^t|}$ for each language is calculated using the longest possible units and backing off to shorter units if needed. For example, if the token itself is not found in any of the language models it is divided into longest character *n*-grams used and the token gets the average of the scores of the *n*-grams in question. For each language l, each token t gets a score $S_{t,l}$ and the whole mystery text gets a score $S_{m,l}$ equal to the average of it's tokens as in (1).

$$S_{m,l} = \frac{S_{t_1,l} + S_{t_2,l} + ... + S_{t_{|m^t|},l}}{|m^t|} \quad (1)$$

In this way, each token in the mystery text is given an equal weight when deciding the language for the whole text. For example, the word "*the*" is given equal weight to the word "*village*". The token-based backoff was recently used successfully in determining the language set in multilingual documents by Jauhiainen et al. (2015).

2.2 Language models

The language models consist of units x and their scores $S_{x,l}$ for each language l. The scores S are negative logarithms of the relative frequencies of the units as in (2).

$$S_x = -\log_{10}(\text{relative frequency of } x) \quad (2)$$

The relative frequencies are calculated from the training data by dividing the number of units by the total number of units of the same type. If a unit is not found in the training data for some language a penalty value is used instead. The penalty value corresponds to giving every unseen unit a small relative frequency and thus it functions as a form of additive smoothing. The penalty values are optimized separately for each language using the development data. The optimization of the penalty values is done for one language after another and there is generally a more or less clear peak in the accuracy. In case several penalty values produce the highest accuracy, the smallest penalty value is chosen. In earlier experiments we have experimented with Lidstone smoothing, where the small relative frequency is also added to the relative frequencies of the seen units, but it proved out to produce slightly poorer results.

Character *n*-grams are formed from within the tokens so that the beginning and the end of the token are represented by a white-space. White space was omitted from the beginning of the first token where a special character marking the beginning of a text was used. The last token was treated similarly and the same special character was used to mark the end of the mystery text. The beginning and the end of the text were treated in similar way by Goutte et al. (2014).

No information spanning token boundaries were used this time. The types of units used in the system for the shared task in order of backing off are:

- Space-delimited tokens consisting of any characters (*A*)
- Tokens delimited by non-alphabetical characters with capital letters (*C*)
- Tokens delimited by non-alphabetical characters with the letters lowercased (*l*)
- Character *n*-grams of any character varying from the length of 8 to 1.

Examples of the token units can be seen in the Table 1 and character *n*-grams in the Table 2.

A	**C**	**l**
[_¡Que_]	[_Que_]	[_que_]
[_"La_]	[_La_]	[_la_]
[_Además,_]	[_Además_]	[_además_]
[_PP,_]	[_PP_]	[_pp_]

Table 1: Examples of token units from the Spanish language models. Underscore is used to represent a space character.

2.3 Unknown language detection

Unknown language detection is used by the system to decide whether the mystery text is written in one of the languages it knows or not. We are using the unknown language *xx* to denote any language not known by the language identifier. We

Length	*N*-grams from [_Además,_]
8	[_Además,], [Además,_]
7	[_Además], [Además,], [demás,_]
6	[_Adem á], [Además], [demás,] [emás,_]
5	[_Adem], [Adem á], [demás] [emás,], [más,_]
4	[_Ade], [Adem], [dem á] [emás], [más,], [ás,_]
3	[_Ad], [Ade], [dem], [em á] [más], [ás,], [s,_]
2	[_A], [Ad], [de], [em] [má], [ás], [s,], [,_]
1	[_], [A], [d], [e], [m], [á], [s], [,], [_]

Table 2: Examples of character *n*-grams generated from the token [_Además,_]. Underscore is used to represent a space character.

used two methods to determine whether the language identified actually belonged to the unknown language *xx*. In both methods, the system first maps the mystery text into one of the languages it knows. After the first mapping the results are analyzed to detect the presence of an unknown language.

The first method is simply to look at the score given by the token-based backoff and reject identifications with too high scores. The unknown language *xx* is identified as the mystery language L_m, if the best score $S_{m,l}$ for the mystery text is higher than cut-off score C_l for the language l as in (3).

$$L_m = xx, \text{ if } S_{m,l} > C_l \quad (3)$$

The second one is to count how many of the lowercased words consisting of alphabetical characters in the mystery text are found in any of the language models of the language identifier. If the ratio of the words R_m is lower than the cut-off ratio R_l for the language with the best score $S_{m,l}$, the unknown language *xx* is chosen as in (4).

$$L_m = xx, \text{ if } R_m > R_l \quad (4)$$

The exact values for the cut-off ratios R_l and the cut-off scores C_l are determined individually for each language l. The development set is used to find out the values which produce the best combined recall for the language l and the unknown language *xx*.

3 Shared task

In the dataset of the shared task, there were 6 language groups with a total of 13 languages and the additional unknown language marked by *xx*. The unknown language *xx* is used to denote any language not belonging to the group of 13 languages. The goal was to build a system that could identify the language of the excerpts in the test set using only the information provided in the training and the development sets.

3.1 DSL corpus collection

The dataset for the shared task was the second version of the DSL corpus collection (DSLCC v. 2.0.). The training set consisted of 18000 labeled excerpts for each of the 13 languages. Each of the excerpts contained from 20 to 100 tokens and seemed to comprise mostly of a one complete sentence. Over 99% of the excerpts ended with a punctuation mark, a bracket or a quotation mark. The average number of tokens for each language can be seen in the Table 3. On the average the excerpts in Spanish had clearly more tokens than those of the other languages. The development set had 2000 labeled excerpts for each of the 13 languages as well as for the unknown language *xx*. The length of the excerpts in the development and training sets were comparable as can be seen in the Table 3. The average number of characters in the excerpts of the development set was 219. The number and the identity of the languages used in the excerpts of the unknown language *xx* were not known. Some of the excerpts in the unknown language *xx* were identified as Catalan and Slovenian by Google Translate[2], but also many other languages were present.

The test set A consisted of 14000 unlabeled excerpts from newspaper texts: 1000 excerpts for each of the 13 languages and 1000 excerpts for the unknown language. The test set B had the same number of unlabeled excerpts from newspaper texts, but all of the named entities had been substituted by place holders using a named entity recognizer. The following example excerpt is from the test set B:

- El #NE# #NE# #NE# #NE# asociación civil comprometida con el desarrollo económico y cultural de la ciudad, celebrará el 15º aniversario de su formación con una cena en el

[2]https://translate.google.com

Language l	Train.	Dev.
Croatian (hr)	29.6	29.7
Bosnian (bs)	30.7	30.9
Serbian (sr)	31.7	31.6
Malaysian (my)	30.3	30.2
Indonesian (id)	30.9	30.8
Czech (cz)	30.8	30.9
Slovakian (sk)	30.5	30.4
Portuguese (pt-PT)	33.0	33.3
Braz. Port. (pt-BR)	34.1	34.0
Spanish (es-ES)	55.7	56.4
Arg. Spa. (es-AR)	49.1	48.4
Bulgarian (bg)	29.6	29.7
Macedonian (mk)	30.2	30.0
Unknown (xx)	-	33.5

Table 3: The average number of tokens per excerpt in the training and the development sets for each language.

restaurant #NE# el jueves próximo desde las 20.30.

There was not a separate development set for the test set with the named entities blinded so the settings of our system were exactly the same on the test set A and B. Before running the language identifier on the test set B, we simply removed the place holders from the excerpts.

3.2 Language group identification

We followed the example given by the best performing system from the 2014 shared task (Goutte et al., 2014) and first used the system to discriminate between the six language groups. Development set was used to optimize the units used in the group identification phase and we ended up using character n-grams from 7 to 1 characters in length. The penalty value for unseen units was set at 6.7. With these settings, the system discriminated (at least on the third run, see below) between the groups perfectly on both the development and the test data, if we are not considering the unknown language. The average identification accuracy for individual languages with the development data was already 94.61% (*xx* not included). The Table 4 shows the accuracies with different unit combinations at this point. These combinations were more thoroughly run after the deadline for the shared task to show how much accuracy is gained by backing off to smaller units within the tokens. A small increase in overall accuracy was noticed when the penalty value was raised to 6.8 from 6.7. It would not have affected the end result of the system used in the shared task as the language identifier was only used to identify the language groups at this point and it did so perfectly already with the penalty value of 6.7.

Units	Pen.	Accuracy.
n-grams: 7 to 1	6.8	94.63%
n-grams: 7 to 1	6.7	94.61%
n-grams: 6 to 1	6.8	94.52%
n-grams: 8 to 1	6.7	94.50%
n-grams: 5 to 1	7.0	94.08%
$C + l + n$-grams: 8 to 1	6.4	94.31%
$l + n$-grams: 8 to 1	6.3	94.20%
$A + C + l + n$-grams: 8 to 1	6.4	94.15%
6-grams	6.8	93.98%
7-grams	6.6	93.80%
C	6.2	93.80%
5-grams	7.0	93.75%
l	6.2	93.70%
A	6.2	93.46%
n-grams: 4 to 1	7.2	92.97%
4-grams	7.2	92.88%
8-grams	6.2	92.81%
n-grams: 3 to 1	6.9	90.19%
3-grams	6.9	90.19%
n-grams: 2 to 1	7.6	83.22%
2-grams	7.6	83.22%
1-grams	6.5	73.63%

Table 4: The average accuracies for known languages using different unit combinations on the development set.

After the group of the mystery text was identified, the text was given to a group optimized version of the token-based language identifier. The units and the penalty value used within each group can be seen in the Table 5. In the token column *A* refers to tokens including all characters, *C* to tokens with only alphabetical characters and *l* to tokens with only lowercased alphabetical characters.

In the Table 5, we can see that the only time we use complete tokens for calculating the score is when we are discriminating between Malaysian and Indonesian. Ranaivo-Malançon (2006) used exclusive lists of words together with the formatting of numbers to decide whether the mystery text was written in Indonesian or Malaysian. The results of our experiments would also suggest that whole words are especially important when discriminating this pair of languages.

Group	Tokens	N-grams	Pen.
A-F	-	1-7	6.7
A (bs, hr, sr)	-	1-7	6.5
B (id, my)	*A*, *C*, *l*	1-8	7.0
C (cz, sk)	-	1-7	6.7
D (pt-PT, pt-BR)	-	1-7	6.7
E (es-ES, es-AR)	-	1-8	6.7
F (mk, bg)	-	1-7	6.7

Table 5: The language models used when discriminating within the language groups.

3.3 First run

The main difference between the first and the second runs is that in the first run, the language identifier was optimized so that it made as few positive errors with the unknown language *xx* as possible. Positive errors with the unknown language are errors where a language known to the language identifier is labeled as the unknown language *xx*. We wanted to continue developing unknown language detection methods (to be used one after another in a serialized manner) and once a positive error was made it was impossible to recover from it. We also wanted to see how high recall we would achieve with the known languages. When considering the overall accuracy, we did not believe that the results of the first run could compete with the results of the second run.

When we were optimizing the parameters, we took a look at the errors the language identifier made on the development set. After the optimization the unknown language *xx* was erroneously identified as one of the 13 languages known by the language identifier 324 times, while a known language was identified as unknown 4 times. With Malaysian we allowed the language identifier to make three 'errors' on the development set, as the sentences were actually in English:

- Daim not attending UMNO assembly, Tengku Adnan confirms © UTUSAN MELAYU (M) BHD, 46M Jalan Lima Off Jalan Chan Sow Lin, 55200 Kuala Lumpur.
- Complete signature forms should be mailed by August 23 to "Save Vui Kong" Campaign, Kuala Lumpur and Selangor Chinese Assembly Hall, 1, Jalan Maharajalela, 50150 Kuala Lumpur, Malaysia.
- Ishak said Jalan Perdana, Jalan Hishamuddin, Jalan Travers (opposite Keretapi Tanah Melayu Berhad), Jalan Mahameru and Jalan Istana Baru would be closed at 9.20 am for the cortege to be taken to Istana Negara.

Furthermore, we allowed it to make one error with Macedonian where the latter half of the sentence was actually written in Latin script instead of the Cyrillic normally used in Macedonian. This kind of errors in the dataset itself were not noticed in the test set.

The parameters used for the unknown language detection on the first run can be seen in the Table 6. R_l is the cut-off ratio and C_l is the cut-off score. The cut-off ratio for Slovak stayed as high as it did because the Slovak development set included some sentences where all the accents were omitted from the characters. We could have coped with this problem by creating separate language models for these languages with de-accented characters, but we did not have time to move further with this idea.

Language l	R_l	C_l
Croatian (hr)	32	5.4
Bosnian (bs)	35	5.0
Serbian (sr)	24	5.1
Malaysian (my)	20	5.3
Indonesian (id)	30	5.4
Czech (cz)	39	5.3
Slovakian (sk)	45	5.3
Portuguese (pt-PT)	25	4.9
Braz. Port. (pt-BR)	25	4.9
Spanish (es-ES)	12	6.5
Arg. Spa. (es-AR)	14	4.9
Bulgarian (bg)	30	5.3
Macedonian (mk)	35	5.1

Table 6: The cut-off ratios used with lowercased tokens and cut-off scores to judge the excerpt to be in the unknown language *xx* on the first run.

The first run achieved 93.87% accuracy on the development set and 93.73% accuracy on the test set.

3.4 Second run

The language models used for the second run were the same as for the first run and can be seen in the Table 5.

The unknown language detection parameters for the second run were optimized to reach the best overall identification accuracy. These ratios

for unknown language detection differ considerably between languages as can be seen in the Table 7 which shows the ratios used for the second run.

Language l	R_l	C_l
Croatian (hr)	16	5.1
Bosnian (bs)	21	5.0
Serbian (sr)	23	5.1
Malaysian (my)	20	5.3
Indonesian (id)	30	5.4
Czech (cz)	39	5.3
Slovakian (sk)	45	5.2
Portuguese (pt-PT)	25	4.9
Braz. Port. (pt-BR)	25	4.9
Spanish (es-ES)	11	4.4
Arg. Spa. (es-AR)	14	4.9
Bulgarian (bg)	30	5.3
Macedonian (mk)	35	5.1

Table 7: The cut-off ratios used with lowercased tokens and cut-off scores to judge the excerpt to be in the unknown language *xx* for the second and the third runs.

In the development set, there was a clear tendency to identify Bosnian sentences as Croatian. We, therefore, experimented with giving a small bonus to Bosnian over Croatian. If the first identified language was Croatian but Bosnian came second within a score margin of 0.01, the text was identified as Bosnian. Twenty-three errors (out of 713 errors between Croatian, Bosnian and Serbian) were corrected by this very ad-hoc weight.

The unknown language was erroneously identified as one of the known languages 82 times. A known language was identified as the unknown language *xx* 58 times.

The second run achieved 94.61% accuracy on the development set and 94.36% accuracy on the test set.

3.5 Third run

The language models used for the third run were the same as for the first and second runs. The parameters for ratio and score cut-offs for determining the unknown language were the same for our third run as our second run and can be seen in the Table 7. The ad-hoc weight given to Bosnian in the second run was still used in the third run.

The third run included a special modifying addition αS_x to the scores S_x of individual character *n*-grams if they were not found in other languages within the group. The new score S'_x was calculated as in (5).

$$S'_x = S_x + \alpha S_x \tag{5}$$

This was done for the groups A (bs, hr, sr) and E (es-ES, es-AR) only. We concentrated our efforts to finding ways to further the identification accuracy of the group A and did not have the time to find the optimal parameters for the other groups. We also did not expect to gain much in overall accuracy had we done so. The multipliers α used in the third run can be seen in the Table 8.

Found in	**Not found**	**Multiplier** α
hr	bs, sr	1.50
bs	sr, hr	2.00
sr	bs, hr	0.15
es-ES	es-AR	0.75
es-AR	es-ES	1.50

Table 8: The multipliers α for groups A and E.

The third run achieved 94.86% accuracy on the development set and 94.67% accuracy on the test set.

The confusion table for the third run on the test data with blinded named entities can be seen in the Table 9.

The within group accuracies for normal test set can be seen in the Table 10. It is clear that our system has a special problem with the group A, where our results are almost 6% lower than the best results of the 2014 shared task.

Group	**Accuracy.**
A-F	94.7%
A (bs, hr, sr)	87.7%
B (id, my)	99.7%
C (cz, sk)	99.8%
D (pt-PT, pt-BR)	92.4%
E (es-ES, es-AR)	90.4%
F (mk, bg)	99.8%
xx	98.2%

Table 10: The accuracies within the language groups for the third run on normal test set.

Comparison of the performance of our system to other systems which submitted results to the shared task can be found in the overview of the DSL Shared Task (Zampieri et al., 2015).

	bs	hr	sr	id	my	cz	sk	pt-PT	pt-BR	es-ES	es-AR	mk	bg	*xx*
bs	803	136	54	0	0	0	0	0	0	0	0	0	0	7
hr	76	905	7	0	0	0	0	0	0	0	0	0	0	12
sr	80	37	882	0	0	0	0	0	0	0	0	0	0	1
id	0	0	0	989	11	0	0	0	0	0	0	0	0	0
my	0	0	0	3	997	0	0	0	0	0	0	0	0	0
cz	0	0	0	0	0	1000	0	0	0	0	0	0	0	0
sk	0	0	0	0	0	0	997	0	0	0	0	0	0	3
pt-PT	0	0	0	0	0	0	0	869	131	0	0	0	0	0
pt-BR	0	0	0	0	0	0	0	103	897	0	0	0	0	0
es-ES	0	0	0	0	0	0	0	0	0	879	116	0	0	5
es-AR	0	0	0	0	0	0	0	0	0	158	842	0	0	0
mk	0	0	0	0	0	0	0	0	0	0	0	999	0	1
bg	0	0	0	0	0	0	0	0	0	0	0	0	999	1
xx	3	5	6	0	0	4	13	0	0	1	2	0	0	965

Table 9: The confusion table for the third run on the test set with the named entities blinded.

4 Discussion

The parameters for the language identifier and the language models used were exactly the same for the runs on development set and the corresponding test runs. We did not find the time to use the data in the development set as an additional training material for the actual test runs, even though we suspect it might have slightly improved the results on the test set.

The exact reason for the positive effect caused by the ad-hoc weight used with Bosnian and Croatian is not known. It is possible that the Bosnian training material is not as representative of the language as the Croatian. All data is biased to some extent and if the training data for a language identifier is biased differently from the data it is used on, situations such as this can arise.

The special character used to mark the beginning and the end of the text did not affect the results much. Using it gave a 0.03% increase in average individual language identification accuracy at the group identification phase.

After the shared task submissions, we optimized the multiplier α also for the other languages using the development set. Optimization resulted in a slight improvement with the Portuguese pair achieving 94.88% average accuracy on the development set. The optimized multipliers for the other languages were zero except for the Portuguese, as can be seen in the Table 11.

Acknowledgments

We thank the Kone Foundation for the funding that made possible the research presented in this paper. We also thank the anonymous reviewers for their extremely helpful suggestions.

Found in	Not found	Multiplier α
id	my	0.00
my	id	0.00
cz	sk	0.00
sk	cz	0.00
pt-PT	pt-BR	0.40
pt-BR	pt-PT	0.00
mk	bg	0.00
bg	mk	0.00

Table 11: The multipliers α for groups B, C, D and F.

References

Cyril Goutte, Serge Léger, and Marine Carpuat. 2014. The NRC System for Discriminating Similar Languages. In *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, pages 139–145, Dublin, Ireland, August. Association for Computational Linguistics.

Tommi Jauhiainen, Krister Lindén, and Heidi Jauhiainen. 2015. Language Set Identification in Noisy Synthetic Multilingual Documents. In *Proceedings of the Computational Linguistics and Intelligent Text Processing 16th International Conference, CICLing 2015*, pages 633–643, Cairo.

Tommi Jauhiainen. 2010. Tekstin kielen automaattinen tunnistaminen. Master's thesis, University of Helsinki, Helsinki. http://urn.fi/URN:NBN:fife201012223157.

Kone Foundation. 2012. The Language Programme 2012-2016. http://www.koneensaatio.fi/en.

Bali Ranaivo-Malançon. 2006. Automatic Identification of Close Languages–Case Study: Malay and Indonesian. *ECTI Transaction on Computer and Information Technology*, 2(2):126–133.

SIL. 2013. *ISO 639-3 Codes for the representation of names of languages*. SIL International. http://www.sil.org/iso639-3/.

Marcos Zampieri, Liling Tan, Nikola Ljubešić, and Jörg Tiedemann. 2014. A Report on the DSL Shared Task 2014. In *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, pages 58–67, Dublin, Ireland, August. Association for Computational Linguistics.

Marcos Zampieri, Liling Tan, Nikola Ljubešić, Jörg Tiedemann, and Preslav Nakov. 2015. Overview of the DSL Shared Task 2015. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects (LT4VarDial)*, Hissar, Bulgaria.