A Bayesian Approach to Query Language Identification

Jiří Materna[1,2] and Juraj Hreško[2]

[1] Centre for Natural Language Processing
Faculty of Informatics, Masaryk University
Botanická 68a, 602 00, Brno, Czech Republic
xmaterna@fi.muni.cz
[2] Seznam.cz, a.s.
Radlická 608/2, 150 00, Praha 5, Czech Republic
{jiri.materna,juraj.hresko}@firma.seznam.cz

Abstract. In this paper we present a Bayesian approach to language identification of queries sent to an information retrieval system. The aim of the work is to identify both the language of a query as a whole and the language of particular words in the query. The method is evaluated on a test set of manually labelled queries. The evaluation shows that our method performs better than the Google Language Detect API and an implementation of the *n*-gram method on our testing set of queries.

Key words: language identification, query language, information retrieval

1 Introduction

Query language identification is one of the crucial issues that need to be solved in information retrieval systems such as Seznam.cz or Google. Based on the user's location and the query language, the system has to decide how to select resulting web pages, often written in various languages. Apart from identification of the language of a query as a whole, it may also be important to identify the language of particular terms of the query. This information is important, for instance, for correct morphological analysis of a given term.

Language identification of full documents is a well-explored area, however, queries are usually very short and rarely in the form of grammatically correct sentences. This makes the problem of query language identification more complicated. Moreover, since the query language identification is not so common problem as the language identification of a document, it is also less-explored. It turns out that some of the algorithms successfully applied to the language detection of text documents are not convenient for query language detection [5].

The task of language identification, sometimes called language detection or language recognition, is an instance of classification problems. The existing solutions can be divided into three main categories: approaches based on analysis of character *n*-grams [3,2], approaches that use dictionaries [9], and

Aleš Horák, Pavel Rychlý (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2011, pp. 111–116, 2011.

non-lexical approaches that use, for example, phoneme transcriptions [1] or an information about compression rate [6].

In the n-gram based approach, the idea is to compute relative frequencies of character n-grams for each language in the training dataset, and then use these statistics in order to detect language of previously unseen documents. The n-grams represent features in a vector space and similarity metrics such as the cosine measure can be used to find the most similar n-gram statistics to a processed document among statistics of training corpora of known languages.

Other algorithms, also based on the n-gram model, use Markov processes to determine the language of a text [8]. The idea behind the method is to detect language via the probabilities of observing certain character sequences. The probability of seeing particular character is dependent on a limited number of previous characters. The sequence of characters forms states of a Markov model.

The second widely-used approach to language detection is based on dictionaries of words rather than sequences of characters. In the dictionary method, for each language is created a set of language-specific words, where each word is associated with a relevance score. During the classification process, the processed document is compared against the trained dictionaries, and the language with highest scores wins. In comparison to the n-gram model, the dictionary based model requires tokenization and much more training examples, but for the language detection of short texts or queries is more appropriate.

2 Proposed Method

In our work we used the dictionary approach in a Bayesian framework. The training dataset consists of all documents indexed by the Seznam search engine[1] enriched with the information about language of particular documents. Permitted languages are $L = \{cz, en, sk, de, pl, fr, und\}$, where und represents an undefined language[2]. The language detection of web documents is performed using an n-gram classifier whose description is out of the scope of this paper. In order to model the probability $P(w|L)$ of word w being generated by language L we use the relative frequency of w in the corpus of language L smoothed by the Good-Turing Frequency Estimator [4].

For a given query $Q = \{w_1, w_2, \ldots, w_N\}$, the goal of the classifier is to identify probabilities $P(L_Q|w_1, w_2, \ldots, w_N)$ and $P(L_{w_i}|w_1, w_2, \ldots, w_N)$ for each $i \in \{1, 2, \ldots, N\}$, where L_Q stands for the language of the query as a whole and L_{w_i} stands for the language of word w_i. The probabilities are modelled using the Bayesian network shown in figure 1. The only observed variables are words $w_1, w_2, \ldots, w_N$. To be able to infer the required probabilities we need to define prior probabilities $P(L_Q)$ for all languages and conditional probabilities $P(L_{w_i}|L_Q)$ for all combination of L_{w_i} and L_Q.

[1] http://search.seznam.cz

[2] A language that is not included in our set of supported languages.

The prior probabilities $P(L_Q)$ have been set according to the query language distribution in the search log of the Seznam search engine to the values given by table 1.

Table 1. Prior probabilities of query languages.

Language	cz	en	sk	de	pl	fr	und
Prior probability	61.9 %	19.1 %	3.0 %	1.7 %	0.7 %	0.6 %	13 %

Conditional probabilities $P(L_{w_i}|L_Q)$ of word language L_{w_i} being present in a query of language L_Q is hard to obtain. In order to get correct values we would need a great training corpus of queries with annotated both the language of query and languages of particular words in the query. We have avoided such demanding manual work by approximating the values using the following formula:

$$P(L_{w_i}|L_Q) = \begin{cases} \frac{9}{10} & \text{if } L_{w_i} = L_Q \\ \frac{1}{10} \times \frac{1}{|L|-1} & \text{else,} \end{cases} \tag{1}$$

where $|L|$ stands for the number of languages. In our case $|L| = 7$.

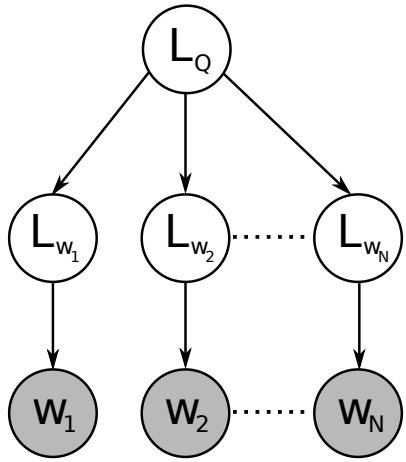

Fig. 1. Graphical model for query language identification.

From the Bayesian graphical model, we can express the probability of L_Q given query $Q = \{w_1, w_2, \ldots, w_N\}$ and the probability of L_{w_i} given query $Q = \{w_1, w_2, \ldots, w_N\}$ by

$$P(L_Q|w_1, w_2, \ldots, w_N) = \frac{P(L_Q)\prod_{i\in<1\ldots N>} P(w_i|L_Q)}{\sum_{L'_Q} P(L'_Q)\prod_{i\in<1\ldots N>} P(w_i|L'_Q)} \quad (2)$$

and

$$P(L_{w_i}|w_1, w_2, \ldots, w_N) = \sum_{L_Q} P(L_{w_i}|L_Q, w_i)P(L_Q|w_1, w_2, \ldots, w_N) \quad (3)$$

respectively, where

$$P(w_i|L_Q) = \sum_{L_w} P(w_i|L_w, L_Q)P(L_w|L_Q) = \sum_{L_w} P(w_i|L_w)P(L_w|L_Q) \quad (4)$$

and

$$P(L_{w_i}|L_Q, w_i) = \frac{P(w_i|L_{w_i})P(L_{w_i}|L_Q)P(L_Q)}{\sum_{L'_{w_i}} P(w_i|L'_{w_i})P(L'_{w_i}|L_Q)P(L_Q)} \quad (5)$$

3 Evaluation

To prove usability of our approach, we compared its results with two other implementations of language detection algorithms. The first one implements an *n*-gram based method. It takes into consideration 1 – 5 letter grams and is the result of the bachelor thesis by [7]. Web interface for the algorithm allows to detect more languages than our one, so we had to reduce the set of detectable languages to be fair. The other tool used to evaluate our method is the Google Language Detect API[3].

Both of these methods are intended to detect languages of documents, so they are not expected to perform so good on shorter examples like queries. We used small testing set of manually classified examples for query language detection. Number of examples was 300. The language distribution was chosen to represent distribution in real data and is listed in table 2. The Undefined language is completely missing because our reference classifiers do not support it. All queries came from Seznam.cz query log and were chosen randomly.

Table 2. Language distribution in the query test set.

Language	cz	en	sk	de	pl	fr
Examples %	65.7 %	18.0 %	6.0 %	5.3 %	2.7 %	2.3 %

In addition to the full test set, we also use its subsets for the evaluation purposes. Two of them were language based – Czech queries (197 samples) and English queries (54 samples). The rest of subsets were based on count of tokens of query[4]. The results are shown in table 3.

[3] http://code.google.com/apis/ajax/playground/#language_detect

[4] URLs in queries were split by stops, e.g. “www.wetter.de” was split into three tokens.

Table 3. Language identification accuracy on various test sets.

Set/Method	Bayesian	Google API	*n*-gram
All languages	91.67 %	61.33 %	51.67 %
Czech	91.37 %	50.76 %	46.70 %
English	92.59 %	75.93 %	52.26 %
1 token	79.31 %	36.21 %	39.66 %
2 tokens	95.80 %	61.54 %	47.55 %
3 or more tokens	93.00 %	76.00 %	64.00 %

All samples have been manually labelled with correct language, and after it classified using all three classifiers. Performance is expressed using the accuracy, i.e. number of correctly classified samples divided by total number of examples in the test set.

As we can see from the resulting scores, the hardest problem is the language identification for one-token queries. On two or more tokens all methods performed better. Worse performance of our Bayesian approach on "3 or more tokens" than on "2 tokens queries" category can be explained by the type of some of these queries. They more likely contain URLs with common words like *tchibo*, *mobile* or *ebay*, and country specific domain names.

Apart from the detection of query language, we also proposed a method for language detection of particular words in the query. To have at least some notion about the word language identification accuracy, we create a small test set consisting of two parts. The first part (150 queries) has been chosen randomly as in previous experiment and the other one (also 150 queries) has been taken from the suspected set of queries, created as a result of processing full query set when only queries with at least two languages were chosen. Before all, we picked a threshold 0.9 that defines the minimum probability of the word language for a word to be considered as comming from this language instead of being the same as the detected query language. This has to be done because our approach had some problems with combinations of some languages, especially the Slavic ones (Czech, Slovak, Polish).

With this adjustment we reached accuracy of 73.33% on our testing set. Most errors have been caused by the presence of URLs in queries and by the occurrence of very common words from one language in a query of another language in which the word occurs too, but with smaller frequency (e.g. "ou" in French means "or" and in Czech it is an abbreviation for "municipal office").

The performance on this task has not been compared to the other implementations as in the previous task because neither the *n*-gram implementation nor the Google Language Detect API does not support such functionality.

4 Conclusions

We have presented a method for automatic language identification of a query in a fulltext information retrieval system. The method supports detection of

both language of a query as a whole and particular words in the query. In contrast to the most of available language detection tools, our method uses full Bayesian approach and is able to correctly classify even short text like queries. The method has been compared to the Google Language Detect API and the *n*-gram based tool by Josef Toman. Both tools have been outperformed in all test by the implementation of our Bayesian approach.

The method for identification of language of words also performs well, but to use it in a practical application, it needs some modifications. One possible approach is to learn the word language matrix on some relevant data instead of using just the simple function.

Acknowledgements This work has been partly supported by the Ministry of Education of CR within the Center of basic research LC536.

References

1. Kay Berkling, Takayuki Arai, and Etienne Barnard. Analysis of phoneme-based features for language identification. In *Proceedings of ICASSP-94*, pages 289–292, 1994.
2. William B. Cavnar and John M. Trenkle. N-gram based text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, 1994.
3. Ted Dunning. Statistical identification of language. In *Technical Report MCCS-94-273*, volume 94. Computing Research Laboratory, 1994.
4. Irving John Good. The population frequencies of species and the estimation of population parameters. In *Biometrika*, volume 40, pages 237–264, 1953.
5. Thomas Gottron and Nedim Lipka. A comparison of language identification approaches on short, query-style texts. volume 5993, pages 611–614. Springer, 2010.
6. W. J. Teahan. Text classification and segmentation using minimum cross-entropy. In *RIAO'00*, volume 2, pages 943–961.
7. Josef Toman. Statistical language recognition, 2006.
8. Peter Vojtek and Mária Bieliková. Comparing natural language identification methods based on markov processes. In *Slovko, International Seminar on Computer Treatment of Slavic and East European Languages*, pages 271–282, 2007.
9. Radim Řehůřek and Milan Kolkus. Language identification on the web: Extending the dictionary method. In *Proceedings of Computational Linguistics and Intelligent Text Processing 10th International Conference CICLing 2009*, volume 5449, pages 357–368. Springer, 2009.