A Comparison of Language Identification Approaches on Short, Query-Style Texts

Thomas Gottron[1] and Nedim Lipka[2]

[1] Institut für Informatik, Johannes Gutenberg-Universität Mainz,
55099 Mainz, Germany
gottron@uni-mainz.de
[2] Faculty of Media, Media Systems, Bauhaus University Weimar,
99421 Weimar, Germany
nedim.lipka@uni-weimar.de

Abstract In a multi-language Information Retrieval setting, the knowledge about the language of a user query is important for further processing. Hence, we compare the performance of some typical approaches for language detection on very short, query-style texts. The results show that already for single words an accuracy of more than 80% can be achieved, for slightly longer texts we even observed accuracy values close to 100%.

1 Introduction

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. This is necessary to correctly translate the query or document into the language of the respectively other.

To our knowledge, the focus in research on language detection is usually on analysing full documents, i.e. on reasonably long and well formulated texts. Queries, instead, are rarely formulated as full sentences and are usually very short (typically 2-4 words for web search). Nevertheless, recent systems [1] participating at CLEF detected the language of queries by applying tools intended for long texts. This leads us to the question: how well do these approaches work on texts in the style of queries?

Automatic language detection on written texts, also known as language identification and sometimes as language recognition, is a categorization task. The most distinguished related works are based on statistical learning algorithms and lexical text representations, particularly n-grams, cf. [2,3,4,5,6]. Dictionary-based approaches, concerning words as lexical representation, are discussed in [7,2]. Non-lexical representations, used in the field of language identification, are for example phoneme transcriptions [8] or the rate of compression [9].

The paper in hand studies the potential and reliability of some commonly used language detection approaches on very short, query-style texts. Lacking a large corpus of annotated multi-language queries, we based our experiments on news headlines of the Reuters CV1 and CV2 collection and single words extracted from bilingual dictionaries.

2 n-Gram-Based Language Detection

As short, query-style texts provide too little data for approaches based on words or full sentences, we focus on methods based on character n-grams, for short n-grams. An n-gram consists of n sequential characters; usually its relative occurence in a text is determined.

One such language detection method, that is used quite often is the one of Cavnar and Trankle [3]. Following the observation that each languages has some characteristic n-grams which appear frequently, the idea is to compare the frequency-ranks of n-grams in a previously unseen text with those of reference texts for different languages. The text is then attributed to the language with the most similar frequency-rank according to an out-of-place measure. As this measure is problematic for the few entries in the frequency-rank list of short texts, we "normalised" the ranks in our implementation to values between 0 for the most frequent and 1 for the least frequent n-gram.

As language detection is a classification task, Naive Bayes is a classical approach to the problem. A Naive Bayes classifier uses conditional probabilities of observing features in a text to deduce a probability of a text to be written in a given language. In our case the n-grams serve as features.

Vojtek and Bielikova [4] use Markov processes to determine the language of a text. Here, the idea is to detect the language via the probabilities to observe certain character sequences. The probabilities depend on a limited number k of previously seen characters, which form the states of the Markov process. The conditional probabilities can be estimated via frequencies of k-grams and $k + 1$-grams from a reference text [2].

The last approach we look at in this context is based on the vector space of all possible n-grams. A text can be represented in this space as the vector of the frequencies of its n-grams. Its language can be determined by looking at the cosine-similarity of its vector representation with the vectors of reference texts in different languages.

3 Evaluation

All the algorithms we discussed in the previous section need to be trained on reference documents. We used the English documents from the Reuters collections CV1 [10] and the language annotated Danish, German, Spanish, French, Italian, Dutch, Norwegian, Portuguese and Swedish documents from CV2. Table 1 shows the detailed distribution of the individual languages among these 1,102,410 documents.

The texts of the news articles were used for training the language classifiers. In order to see the influence of the length of n-grams we varied the value of n between 1 and 5 characters. The relatively short and noisy news headlines were retained for classification. They are on average 45.1 characters and 7.2 words long, thus, longer than an average query on the web. However, the titles frequently contain named entities ("*Berlusconi* TV faces legal cliffhanger") or numerical values ("Dollar General Q2 *$0.24* vs *$0.20*"). These entities and a lack of stopwords render the headlines a quite suitable set of short, query-alike texts for language detection. For the evaluation of single word texts, we obtained terms from small, bilingual dictionaries from English to French, German, Spanish, Italian and Portuguese. We extracted the words, which were

Table 1. Distribution of languages in the Reuters corpus and among dictionary terms.

Corpus	da	de	en	es	fr	it	nl	no	pt	sv
Reuters	11.184	116.209	806.788	18.655	85.393	28.406	1.794	9.409	8.841	15.731
Dictionaries	–	3.463	12.391	3.260	1.153	2.432	–	–	501	–

unambiguous from a language point of few (i.e. existed in only one language). This gave us a total of 20.048 words of on average 8.1 characters. Again, see table 1 for details about the indivdual languages.

The algorithms were implemented from scratch and trained on the Reuters articles. For the frequency-rank approach, we additionally used a readily trained implementation of the original algorithm, which we included in the evaluation process as LC4J[3]. We used each of the algorithms to detect the languages of the previously unused Reuters headlines and the words obtained from dictionaries.

Table 2 shows the accuracies[4] for detecting the language of the Reuters headlines and the dictionary entries across all algorithms and all settings for n. But, the values of LC4J need to be treated carefully: in many cases the algorithm could not detect any language at all. This might be, because the language models provided with the implementation are too sparse for short texts. The values given here are solely based on those cases where language detection was successful. When taking into account the unclassified documents, the accuracy drops drastically to 39.24% for the headlines and to 30.33% for the dictionary words.

Table 2. Accuracy of language classifiers (in %)

Data	Method	1-grams	2-grams	3-grams	4-grams	5-grams
Headlines	Naive Bayes	87.90	95.01	98.52	99.40	**99.44**
	Multinomial	65.42	90.08	97.63	99.17	99.22
	Markov	10.28	85.87	73.13	4.50	0.00
	Frequency-rank	6.07	14.90	59.93	25.91	3.47
	Vectorspace	54.68	47.21	61.04	69.67	75.37
	LC4J (where successful)	–	–	67.72	–	–
Dictionaries	Naive Bayes	52.26	64.40	73.49	79.13	**81.61**
	Multinomial	35.65	57.04	68.27	75.74	77.88
	Markov	19.95	57.34	55.14	21.52	2.95
	Frequency-rank	12.32	24.04	42.82	23.25	6.70
	Vectorspace	29.99	33.98	44.28	52.73	59.23
	LC4J (where successful)	–	–	49.93	–	–

The poor performance of the Markov process and our own frequency-rank implementation for higher values of n can be explained with data sparseness, too. The accuracy of Markov drops probably due to a higher number of n-grams not seen during

[3] http://olivo.net/software/lc4j/

[4] Confusion matrices with more detailed results are available in an online Appendix at http://www.informatik.uni-mainz.de/forschung/ir/ecir2010.php

training and an unequal language distribution in the training data. The frequency-rank approach instead suffers from the sparseness of n-grams in the query-like documents, resulting in skewed rankings. Even with the normalised ranking, the performance drops for larger values of n.

The best performing approach for short texts is the Naive Bayes classifier (and its Multinomial variation without the class distribution normalisation). For larger values of n they perform remarkably good and achieve an accuracy close to 100% on the headlines. This observation holds also when looking at individual languages. On a language level, the accuracy varies between 99.71% for Italian and 96.52% for Norwegian. The misclassifications of Norwegian headlines were mostly assigned to Danish. In general, the Scandinavian languages tend to be confused more than other languages. A similar observation was made for dictionary terms of Latin-based languages. Here the most mistakes occured between Spanish, Portuguese and Italian.

4 Conclusions

We looked into language detection for short, query-style texts. Comparing different approaches based on n-grams, it turned out, that Naive Bayes classifiers perform best on very short texts and even on single words. Errors tend to occur within language families, i.e. among Scandinavian or Latin languages.

Future work will comprise a closer look at an adaptation of the frequency-rank approach for short texts, a hierarchical approach to better distinguish between texts from the same language family and the evaluation on real-world multilingual user queries.

References

1. Oakes, M., Xu, Y.: A search engine based on query logs, and search log analysis at the university of Sunderland. In: CLEF'09: Proceedings of the 10th Cross Language Evaluation Forum. (2009)
2. Dunning, T.: Statistical identification of language. Technical Report MCCS-94-273, Computing Research Laboratory, New Mexico State University (1994)
3. Cavnar, W.B., Trenkle, J.M.: N-gram-based text categorization. In: SDAIR'94, Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval. (1994)
4. Vojtek, P., Bieliková, M.: Comparing natural language identification methods based on Markov processes. In: Computer Treatment of Slavic and East European Languages, 4th Int. Seminar. (2007) 271–282
5. Suen, C.Y.: N-gram statistics for natural language understanding and text processing. IEEE Transactions on Pattern Analysis and Machine Ingelligence **PAMI-1**(2) (1979) 164–172
6. Sibun, P., Reynar, J.C.: Language identification: Examining the issues (1996)
7. Řehůřek, R., Kolkus, M.: Language identification on the web: Extending the dictionary method. In: Computational Linguistics and Intelligent Text Processing. Volume 5449/2009 of Lecture Notes in Computer Science. (2009) 357–368
8. Berkling, K., Arai, T., Barnard, E.: Analysis of phoneme-based features for language identification. In: Proc ICASSP. (1994) 289–292
9. Teahan, W.J.: Text classification and segmentation using minimum cross-entropy. In: RIAO '00. Volume 2. (2000) 943–961
10. Lewis, D.D., Yang, Y., Rose, T., Li, F.: RCV1: A new benchmark collection for text categorization research. Journal of Machine Learning Research **5** (2004) 361–397