An Exploration of Language Identification Techniques for the Dutch Folktale Database

Dolf Trieschnigg[1], Djoerd Hiemstra[1], Mariët Theune[1], Franciska de Jong[1], Theo Meder[2]

[1]University of Twente, Enschede, the Netherlands
[2]Meertens Institute, Amsterdam, the Netherlands
{d.trieschnigg,d.hiemstra,m.theune,f.m.g.dejong}@utwente.nl,theo.meder@meertens.knaw.nl

Abstract

The Dutch Folktale Database contains fairy tales, traditional legends, urban legends, and jokes written in a large variety and combination of languages including (Middle and 17th century) Dutch, Frisian and a number of Dutch dialects. In this work we compare a number of approaches to automatic language identification for this collection. We show that in comparison to typical language identification tasks, classification performance for highly similar languages with little training data is low. The studied dataset consisting of over 39,000 documents in 16 languages and dialects is available on request for followup research.

1. Introduction

Since 1994 the Meertens Institute[1] in Amsterdam has been developing the Dutch folktale database, a large collection of folktales in primarily Dutch, Frisian, 17th century and Middle Dutch and a large variety of Dutch dialects (Meder, 2010). It does not only include fairy tales and traditional legends, but also riddles, jokes, contemporary legends and personal narratives. The material has been collected in the 19th, 20th and 21th centuries, and consists of stories from various periods, including the Middle Ages and the Renaissance. The database has an archival and a research function. It preserves an important part of the oral cultural heritage of the Netherlands and can be used for historical and contemporary comparative folk narrative studies. An online version has been available since 2004[2] and currently contains over 41,000 entries.

A rich amount of metadata has been assigned manually to the documents, including language, keywords, proper names and a summary (in standard Dutch). This metadata is very useful for retrieval and analysis, but its manual assignment is a slow and expensive process. As a result, the folktale database grows at a slow rate. The goal of the FACT (Folktales as Classifiable Text)[3] research project is to study methods to automatically annotate and classify folktales. Ideally, these techniques should aid editors of the folktale database and speed-up the annotation process. Language identification is the first challenge being addressed in the FACT project.

In this paper, we compare a number of automatic approaches to language identification for this collection. Based on the performance of these approaches we suggest directions for future work. The Dutch folktale database poses three challenges for automatic language identification. First, the folktales are written in a large number of similar languages. A total of 196 unique language combinations is present in the metadata; 92 unique (unmixed) language names are used[4]. For most of these languages no official spelling is available; the way words are spelled depends on the annotator who transcribed the oral narrative. As a result, documents in the same language may use a different spelling. For our experiments we have used a selection of 16 languages. Second, the language distribution in the collection is skewed: most of the documents are in Frisian and Standard (or modern) Dutch, but there is a long tail of smaller sets of documents in other languages. Consequently, for many languages only little training data is available to train a classifier. Third, documents in the collection can be multilingual. Most of the documents are monolingual, but some contain fragments in a different language. The length of these fragments ranges from a single passage or sentence to multiple paragraphs.

The contributions of this work are twofold. First, we present an analysis of multiple language identification methods on a challenging collection. Second, we make this collection available to the research community.

The overview of this paper is as follows. In Section 2 we briefly discuss related work. In Section 3 we describe the collection in more detail and outline the experimental setup. In Section 4 the results of the different classification methods are discussed followed by a discussion and conclusion in Section 5.

2. Related work

Early work on language learnability dates back to the 1960s (Gold, 1967). Since the 1990s language detection or language identification has become a well-studied natural language processing problem (Baldwin and Lui, 2010). For clean datasets, with only few and clearly separable languages, language identification is considered a solved problem (McNamee, 2005).

Recent research indicates, however, that language identification still poses challenging problems (Hughes et al., 2006), including: supporting minority languages, such as the dialects encountered in our collection; open class language identification, in such a way that a classifier is capable of indicating that no language could be accurately determined; support for multilingual documents; and classification at a finer level than the document level. Xia et al. (2009) and Baldwin and Lui (2010) also argue that language identification has not been solved for collections con-

[1] `http://www.meertens.knaw.nl`

[2] `http://www.verhalenbank.nl` (in Dutch only)

[3] `http://www.elab-oralculture.nl/FACT`

[4] Sometimes caused by an inconsistent naming convention

taining large numbers of languages. In this work we will focus on the capability of existing classifiers to deal with minority and very similar languages.

A large array of methods has been developed for tackling the problem of language identification: categorisation based on n-grams (Cavnar and Trenkle, 1994), words or stopwords (Damashek, 1995; Johnson, 1993), part-of-speech tags (Grefenstette, 1995), syntactic structure (Lins and Gonçalves, 2004), systems based on markov models (Dunning, 1994), SVMs and string kernel methods (Kruengkrai et al., 2005), and information theoretic similarity measures (Martins and Silva, 2005). An extensive overview of techniques is outside the scope of this paper. A more comprehensive overview can be found in Hughes et al. (2006) and Baldwin and Lui (2010). We limit our experiments to the method by Cavnar and Trenkle (1994) and a number of variations based on n-grams and words motivated by positive experimental results of (Baldwin and Lui, 2010).

3. Experimental setup

In the following subsections we describe the collection, investigated classification methods, and evaluation metrics in detail.

3.1. The collection

The complete folktale database[5] consists of over 41,000 documents. After filtering out documents with offensive content (sexual, racist, lese-majesty, etcetera) and copyrighted materials, 39,510 documents remain. From this collection we put all documents with a mixed language where at least one of the languages is Standard Dutch into a single language group labeled "Standard Dutch mixed". Documents in a language with fewer than 50 documents in that language in the collection are removed. This results in a collection of 39,003 documents in 16 different languages. Table 1 lists the 16 languages and their document frequencies. Note that the number of documents per language is strongly skewed: 79% of the collection is written in Frisian or Standard Dutch. The remaining 21% of the documents is distributed over the remaining fourteen languages. Also note that in comparison to previous work by Baldwin and Lui (2010), which uses collections between 1500 and 5000 documents, the collection is relatively large.

3.2. Classification methods

As a baseline classification method, we used the TextCat[6] implementation of the algorithm described by Cavnar and Trenkle (1994). The algorithm creates an n-gram profile for each language and performs classification by comparing each of the n-gram profiles to the n-gram profile of the text to classify. An out-of-place distance measure is used to compare the order of n-grams in the profile and the text. Following the methods investigated by Baldwin and Lui (2010) we used a number of classification methods based on nearest neighbour (NN) and nearest prototype (NP) in combination with the cosine similarity metric.

All tested classification methods use a supervised learning approach: classifications are based on a training set of manually labeled examples. The difference between NN and NP methods is the way the examples are stored. In the NP case, the examples of the same class are aggregated into a *prototype*, a single model representing the class. The prototype is constructed by summing the vectors of the examples. In the NN case, the examples are stored separately. During classification the class(es) of the nearest example(s) is/are returned. In our case we use the class of the first nearest neighbour (or prototype).

The documents are represented by vectors of the unit of analysis, containing the count of that unit. In the case of words, each unique word encountered in the collection forms one dimension of the vector. We use six different units of analysis: overlapping character n-grams of size 1 to 4, a combined representation of n-grams of length 1 to 4, and words (uninterrupted sequences of letters). The text is lowercased and punctuation is removed before features are extracted. The overlapping character n-grams are extracted by sliding a window of n characters over the text one character at a time. In case of the combined n-gram representation, this process is repeated four times (for n=1 to n=4). To reduce the complexity, we experiment with reducing the vector to a selection of 100, 500 and 1000 features. The selection of features is based on the most frequently used features per language appearing the training set. To be more precise: from each language the most frequent feature is taken until the desired number of features is reached. In our experiments we follow the approach described by Baldwin and Lui (2010). Alternatively, we could have used information gain to select the most informative features. We will consider this in future work.

3.3. Evaluation method

We evaluated the different approaches by means of stratified 10-fold cross-validation: the collection was split into

Language	Doc. count
Frisian	17,347
Standard Dutch	13,632
17th century Dutch	2,361
Standard Dutch mixed	1,538
Flemish	882
Gronings[1]	854
Noord-Brabants[1]	677
Middle Dutch	656
Liemers[1]	328
Waterlands[1]	153
Drents[1]	150
Gendts[1]	116
English	97
Overijssels[1]	80
Zeeuws[1]	68
Dordts[1]	64
Total (16 languages)	39,003

[1] Dutch dialects

Table 1: Collection statistics

[5] As of January 2012

[6] http://www.let.rug.nl/vannoord/TextCat

Language	Precision	Recall	F
Frisian	0.999	0.976	0.987
17th century Dutch	0.983	0.978	0.980
Middle Dutch	0.952	0.974	0.963
Liemers	0.861	0.909	0.884
Gronings	0.882	0.785	0.830
Standard Dutch	0.879	0.633	0.736
Gendts	0.942	0.560	0.703
Noord-Brabants	0.331	0.558	0.415
Zeeuws	0.692	0.265	0.383
Flemish	0.229	0.810	0.357
Dordts	0.207	0.609	0.310
Drents	0.196	0.707	0.307
English	0.112	0.887	0.199
Waterlands	0.091	0.824	0.163
Standard Dutch mixed	0.259	0.088	0.131
Overijssels	0.055	0.250	0.090
Macro average	0.542	0.676	0.527
Micro average	0.799	0.799	0.799

Table 2: Per-language classification performance for TextCat, sorted by descending F-score

10 stratified folds (preserving the proportion of languages in the whole collection). Each fold was used to test the method trained on the other nine folds.

As evaluation measures we use macro and micro averaged Precision, Recall and F-measure. The macro (or category) scores indicate the classification performance averaged over the languages, whereas the micro averaged scores indicate the average performance per document. For a particular language, precision is defined as the proportion of predictions in that language which is correct. Recall is the proportion of documents in that language that is correctly predicted. Note that for this classification task the micro average precision, recall and f-measure have the same value (hence the single column P/R/F in Table 4).

4. Results

4.1. TextCat baseline

Table 2 lists the classification performance of TextCat for the 16 languages in the collection. The contingency table in table 3 provides further information about the classification errors made. Its rows list the actual classes where its columns indicate the predicted classes indicated by the system. For example, the second row and first column indicates that 6 documents in Standard Dutch were incorrectly classified by TextCat as Frisian.

We can make the following observations. First, the classification performance of the largest language class (Frisian) is very good. The recall is very high (0.98) at almost perfect precision (0.999). Second, the classification performance of old Dutch languages (17th century Dutch and Middle Dutch) is also good (F-measure larger than 0.96). These languages can be distinguished well from modern Dutch and dialects. Third, the classification performance of the dialects is mixed. Some (Liemers, Gronings) perform relatively well, others (Waterlands, Overijssels) perform poorly. Still the highest F-measure (0.88) does not come close to typical performance scores, which range between 0.91 and 0.99 for the EuroGOV collection (Baldwin and Lui, 2010). Most of the dialects are mistaken for Standard Dutch and vice versa. Gronings shows strong overlap with Drents (both northern dialects); Zeeuws is frequently mistaken for Noord-Brabants, but not the other way around (both southern dialects). Fourth, it is striking that classification of English documents is so poor. Table 3 indicates that Standard Dutch and Standard Dutch mixed is frequently mistaken for English. One possible explanation is that English words or expressions are frequently borrowed in Dutch. It could also indicate that the annotation in the collection is inconsistent: the (Dutch) document contains an English expression but has been classified as Standard Dutch instead of Standard Dutch mixed.

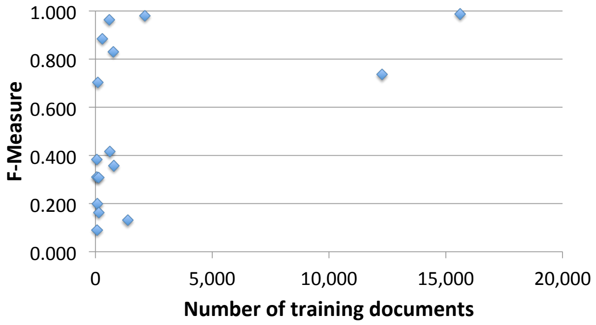

Figure 1: Amount of available training data and classification performance for TextCat

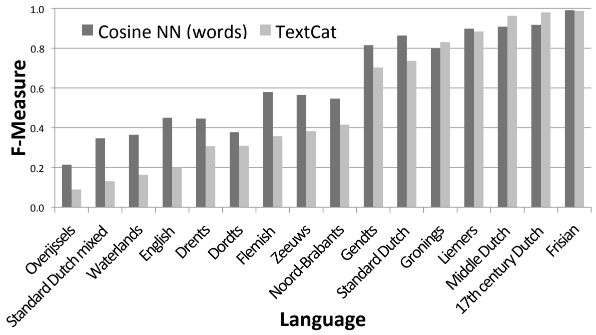

Figure 2: Per language classification performance: TextCat versus cosine (languages sorted according to classification performance of TextCat)

The micro average performance score (see Table 2) indicates a reasonable classification performance of TextCat, but this value has been strongly influenced by the strong performance on the largest language class. The macro averages illustrate that for many smaller languages classification performance is low. Figure 1 shows a scatter plot of the amount of training data available for a language and its classification score.

4.2. Variations of cosine distance

Table 4 summarises the classification performance of a number of variations on language identification systems. TextCat can be viewed as a variation of a nearest prototype system and is therefore in the left part of table.

Again, a number of observations can be made. First, TextCat performs better than all the cosine variants of the nearest prototype method (in terms of F-measure). All the nearest prototype variants based on cosine perform worse.

	Predicted															
Actual ↓	Frisian	Standard Dutch	17th century Dutch	Standard Dutch mixed	Flemish	Gronings	Noord-Brabants	Middle Dutch	Liemers	Waterlands	Drents	Gendts	English	Overijssels	Zeeuws	Dordts
Frisian	16928	106		5	31	27	78	3	5	48	53		7	48		8
Standard Dutch	6	8630	13	366	2065	12	446	6	20	1028	195	1	554	156	4	130
17th century Dutch		21	2308	3	13		2	12		1			1			
Standard Dutch mixed	1	695	9	135	194	15	165	7	4	131	32		92	48	3	7
Flemish		124	1	1	714		4		3	9			24	2		
Gronings		45		3	3	670	19	1		14	84	1		14		
Noord-Brabants		118	1	6	63	8	378	2	5	18	25		3	47	1	2
Middle Dutch		1	15		1			639								
Liemers		14			3		3		298		4	1		5		
Waterlands		15			9		2			126	1					
Drents		4		1		28	1				106			10		
Gendts		1			1		11		10	3	16	65		9		
English	3	4					1	1					86			2
Overijssels		21			7		8			6	18			20		
Zeeuws		6		1	4		23		1	4	5	1		5	18	
Dordts		11			5		2			4	1			2		39

Table 3: Contingency matrix for TextCat

		Nearest prototype				*Nearest neighbour*			
Character	# Features		Macro		Micro		Macro		Micro
n-grams		Precision	Recall	F	P/R/F	Precision	Recall	F	P/R/F
TextCat		**0.542**	0.676	**0.527**	**0.799**	-	-	-	-
Cosine									
n = 1	all (59)	0.234	0.489	0.243	0.498	0.404	0.419	0.407	0.781
n = 2	100	0.356	0.572	0.365	0.577	0.564	0.525	0.531	0.845
	500	0.405	0.598	0.410	0.597	0.629	0.562	0.579	0.864
	1000	0.406	0.599	0.410	0.598	0.631	0.564	0.581	0.865
	all (1,630)	0.406	0.599	0.410	0.598	0.631	0.564	0.581	0.865
n = 3	100	0.340	0.547	0.338	0.569	0.478	0.494	0.475	0.819
	500	0.451	0.628	0.449	0.630	0.606	0.565	0.561	0.855
	1000	0.484	0.635	0.475	0.630	0.628	0.583	0.582	0.863
	all (17,894)	0.503	0.643	0.490	0.631	0.664	0.598	0.606	0.874
n = 4	100	0.309	0.525	0.323	0.583	0.449	0.408	0.418	0.804
	500	0.375	0.632	0.400	0.637	0.588	0.521	0.540	0.852
	1000	0.376	0.654	0.409	0.641	0.621	0.543	0.568	0.864
	all (112,419)	0.403	**0.693**	0.442	0.656	**0.702**	0.584	0.624	0.886
n = 1...4	100	0.289	0.544	0.309	0.562	0.526	0.516	0.514	0.837
	500	0.354	0.607	0.382	0.638	0.585	0.564	0.567	0.866
	1000	0.372	0.624	0.401	0.658	0.611	0.582	0.588	0.874
	all (132,002)	0.400	0.650	0.431	0.687	0.669	0.601	0.624	**0.887**
words	100	0.369	0.650	0.394	0.643	0.474	0.490	0.475	0.828
	500	0.326	0.560	0.338	0.600	0.612	0.581	0.587	0.862
	1000	0.366	0.638	0.389	0.637	0.627	0.591	0.601	0.867
	all (174,180)	0.373	0.659	0.400	0.649	0.675	**0.609**	**0.630**	0.883

Table 4: Classification performance of evaluated systems

The nearest neighbour cosine variants perform similar or better than TextCat in terms of micro and macro F-measure. It should be noted, however, that these nearest neighbour approaches are far more expensive in terms of processing time and required storage than the method implemented by TextCat. Second, the cosine variants perform better with longer representations (longer n-gram windows or words) and with more features. Using all features performs best, but the selection of 1000 features closely approximates the scores based on all features. Figure 2 illustrates the difference between TextCat and the (NN) Cosine distance with word features: Cosine performs better on all languages, except Middle and 17th century Dutch, and Gronings.

5. Conclusions and future work

In this work we have investigated a number of language identification methods on a new and large collection of folktales in a variety and mix of languages. In comparison to other nearest prototype methods, the approach based on mixed n-grams proposed by Cavnar and Trenkle (1994) performs well. The results showed that a nearest neighbour approach using longer and more features performs even better.

Compared to other language identification tasks carried out by Baldwin and Lui (2010), the classification results stay behind. Baldwin and Lui (2010) report a maximum macro F-measure of 0.729 for a skewed collection containing 67 languages. With similar methods, we achieve only 0.630, for a collection with fewer languages. These results indicate that this collection indeed poses a challenge for language identification. The collection therefore is a valuable resource for future language identification research. The collection is available on request (users are required to sign a license agreement).

An important note has to be made on the consistency of the language annotations in the collections. The folktales in the database have been gathered and annotated (in a free text field) by more than 50 people. It is an open question whether these editors have used the same method for labelling the language of a document; some might have annotated a document with Standard Dutch, where another would have labeled it as a mix of Standard Dutch and another language. This might explain why the automatic methods cannot discriminate between these classes.

Our future work will focus on the following aspects of language identification. First, we intend to focus on multilingual document detection. Almost 10% of the documents in the complete collection contains multiple languages. Therefore, it would be useful to detect languages at the sentence level. Second, it would be useful to assign a level of certainty to the detected language. In the work described in this paper we view the task as a closed classification problem with a fixed number of languages. Especially for the long tail of documents in minority languages it would be useful to indicate if no known language was confidently determined. Third, since the language identification system is intended to be used in a semi-automatic setting, it is useful to have a mechanism to present proof for the detected language. Especially when the annotator has no in-depth knowledge of the different languages this would be useful. This could be achieved, for example, by showing sentences from the suggested language(s) similar to the sentence under classification. Fourth and finally, since classification performance is still relatively low, we intend to investigate how contextual information can be used to improve classification performance. In the line of recent work from Carter et al. (2013), who improved the language identification of Twitter messages by incorporating classification features based on for example language of the blogger and language of the document linked to, we could introduce additional features for this particular domain. One can think of features based on the date, source, and place of narrative of the folktale. Or a feature based on the geographical locations encountered in the text. In addition, it might be possible to incorporate knowledge from dialect lexicons to improve classification.

6. Acknowledgements

This work has been carried out within the Folktales as Classifiable Texts (FACT) project, which is part of the CATCH programme funded by the Netherlands Organisation for Scientific Research (NWO).

7. References

T. Baldwin and M. Lui. 2010. 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 ACL*, pages 229—237, Los Angeles, California, USA.

S. Carter, W. Weerkamp, and E. Tsagkias. 2013. Microblog language identification: Overcoming the limitations of short, unedited and idiomatic text. *Language Resources and Evaluation Journal*. To appear.

W.B. Cavnar and J.M. Trenkle. 1994. N-gram-based text categorization. In *Proceedings of the Third Symposium on Document Analysis and Information Retrieval*, Las Vegas, USA.

M. Damashek. 1995. Gauging similarity with n-grams: Language-independent categorization of text. *Science*, 267(5199):843.

T. Dunning. 1994. Statistical identification of language. *Computing Research Laboratory Technical Memo MCCS*, pages 94–273.

E.M. Gold. 1967. Language identification in the limit. *Information and control*, 10(5):447–474.

G. Grefenstette. 1995. Comparing two language identification schemes. In *JADT 1995, 3rd International Conference on Statistical Analysis of Textual Data*, Rome, Italy.

B. Hughes, T. Baldwin, S. Bird, J. Nicholson, and A. MacKinlay. 2006. Reconsidering language identification for written language resources. In *Proc. International Conference on Language Resources and Evaluation*, pages 485–488.

S. Johnson. 1993. Solving the problem of language recognition. Technical report, Technical report, School of Computer Studies, University of Leeds.

C. Kruengkrai, P. Srichaivattana, V. Sornlertlamvanich, and H. Isahara. 2005. Language identification based on string kernels. In *Communications and Information Technology, 2005. ISCIT 2005. IEEE International Symposium on*, volume 2, pages 926–929. IEEE.

R.D. Lins and P. Gonçalves. 2004. Automatic language identification of written texts. In *Proceedings of the 2004 ACM symposium on Applied computing*, pages 1128–1133. ACM.

B. Martins and M.J. Silva. 2005. Language identification in web pages. In *Proceedings of the 2005 ACM symposium on Applied computing*, pages 764–768. ACM.

P. McNamee. 2005. Language identification: a solved problem suitable for undergraduate instruction. *J. Comput. Small Coll.*, 20:94–101, February.

T. Meder. 2010. From a Dutch folktale database towards an international folktale database. *Fabula*, 51(1-2):6–22.

F. Xia, W.D. Lewis, and H. Poon. 2009. Language id in the context of harvesting language data off the web. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 870–878. Association for Computational Linguistics.