INSTITUT FÜR INFORMATIK
Datenbanken und Informationssysteme

Universitätsstr. 1 D–40225 Düsseldorf

Comparison of Language Identification Techniques

Leonid Panich

Bachelorarbeit

Beginn der Arbeit: 06.März 2015
Abgabe der Arbeit: 05.Juni 2015
Gutachter: Prof. Dr. Stefan Conrad
Prof. Dr. Martin Mauve

Erklärung

Hiermit versichere ich, dass ich diese Bachelorarbeit selbstständig verfasst habe. Ich habe dazu keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Düsseldorf, den 05.Juni 2015

Leonid Panich

Abstract

Many researches that analyse huge amounts of text data, eliminate only texts in English or use the datasets of texts, which language is identified. Accordingly, the language identification task for this text data is assumed to be accomplished.

The purpose of the present work is to compare the language identification approaches using the datasets of tweets for the evaluation. The parameters and classifiers with the best performance for the word- and N-gram-based approaches are determined for four different datasets of tweets, that contain 19 different languages. Moreover, the approaches with the highest results are found for each of these datasets. The lists of sentences and words from the Leipzig Corpora Collection are used as the training data.

The results of the present work show that for all used datasets the frequent words approach outperforms the short words approach and works with cumulative frequency addition classifier better than with other classifiers. The frequent words approach achieved the best results using 3100-3800 most frequent words. For most of the used datasets the improved graph-based N-gram approach, that utilises the natural logarithm of the counts of the N-grams, obtains the best performance. This approach shows the best results with the N-grams of the length from 3 to 5 and is used in all comparisons with the cumulative frequency addition classifier. However, for the Non-Latin dataset it is surpassed by the frequent words approach with the cumulative addition classifier and 3100 words.

Contents

1 Introduction **1**

2 Language Modelling Methods **2**

2.1 Word-based Modelling Methods . . . 3

2.2 N-gram-based Modelling Methods . . . 4

3 Classification Methods **7**

3.1 Cumulative Frequency Addition Classifier . . . 7

3.2 Naive Bayesian Classifier . . . 8

3.3 Rank-order Statistics Classifier . . . 9

4 Related work **10**

5 Datasets **13**

5.1 TweetLID Dataset . . . 13

5.2 LIGA Dataset . . . 14

5.3 Annotated Twitter Sentiment Dataset . . . 14

5.4 Non-Latin Dataset . . . 15

5.5 Dataset from the Work of Carter et al. . . . 15

6 Evaluation **16**

6.1 Comparison of the Word-based Methods . . . 17

6.2 Comparison of the N-gram-based Methods . . . 20

7 Conclusion **24**

References **25**

List of Figures **28**

List of Tables **28**

1 Introduction

Language identification is a task of identifying the language of a given document. It is an important preprocessing step for many Natural Language Processing (NLP) tasks. For example, sentiment analysis, question answering, part-of-speech tagging and information retrieval generally assume that the language of the text is identified.

The research history of this area is long and McNamee stated in [McN05] that language identification is a solved problem, because the most methods, that he used, achieved accuracies approaching 100% on a test suite comprised of European languages. This statement is true for long texts, that have a standard orthography. However, in recent years extraction and analyse of the information from the social networks is in demand. The texts there are very short and commonly contain unusual spelling and abbreviations. This makes the language identification task more challenging.

Twitter[1] is one of the social networks, that has a growing popularity as a data source among different researchers. It is valuable due to its huge volume of messages (tweets), that are limited to 140 characters. They are sent immediately and refer to all kinds of topics from news and politics to popular singers and sport events. In 2007 Twitter had 5000 tweets per day and this amount grew up to 500,000,000 tweets per day in 2013 [Kri13]. This service has users worldwide and the tweets are written in many languages. Researchers have used Twitter for different purposes, for example, for prediction of National Football League (NFL) games outcomes [SDGS13], detection [SOM10] and reaction [MRPS11] to the disasters, prediction [AGL$^+$11] and detection [AMM11] of the flu trends. Many researches, that were written until the close of 2011 and use Twitter as a data source, are listed and classified in [WTW13].

The purpose of the present work is to compare the language identification approaches and to find the most effective one for the tweets. The performance of different word- and N-gram-based methods is examined and the best parameters of them are determined. To measure the performance of the approaches, they are implemented in the Java programming language and the F_1 measure of them is calculated.

The frequent words approach and the short words approach are used as the word-based methods. At first, it is examined, if the word frequencies are needed for these approaches. Moreover, the amounts of used words, that maximize their performances are found. The best classifier is determined for the word-based methods and the N-gram approach. The rank-order statistics classifier, the cumulative frequency addition classifier and the naive Bayesian classifier are compared in the present work. The N-gram approach, the graph-based N-gram approach and the improved graph-based N-gram approach are used as the N-gram-based methods. The usage of different N-gram lengths is compared for them. The training data for all utilised methods is taken from the Leipzig Corpora Collection.

[1] http://twitter.com/

2 Language Modelling Methods

The process of language identification can be represented as a system shown in Figure 1. This dataflow was used in [CT94] for text categorization . However, instead of languages, the authors used categories and models were called profiles. Accordingly, the first stage is the **modelling stage**, where **language models** are generated. Such models consist of features, representing specific characteristics of language. These features are words or N-grams with their occurrences in the training set. The language models are determined for each language included in the training corpus. On the other hand a **document model** is a similar model that is created from an input document for which the language should be determined.

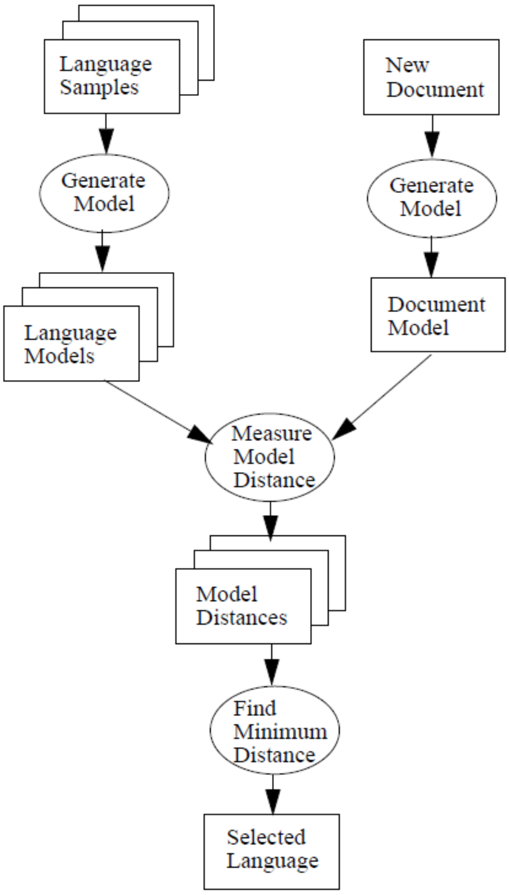

Figure 1: Dataflow for N-gram-based text categorization [CT94]

After all the models have been generated, the document model is compared to all language models in the **classification stage** and the distance between them is measured with the help of classification techniques. The language model that has the minimum distance to the input document represents the language of the document.

All of the methods described below build language models and use the idea which is known as Zipf's Law [Zip49]. It can be formulated as follows: The n-th most common word in a human language text occurs with a frequency inversely proportional to n [CT94]. The consequence of this law is, that in all languages a set of words exists, which are more frequently used than the other words. Therefore, documents from the same language should have similar distributions. From this law also follows, that such classification of the documents is not very sensitive to the amount of words or N-grams taken from the distribution for the comparison.

2.1 Word-based Modelling Methods

2.1.1 Frequent Words Method

One of the direct ways for generating language models is to use words from all languages in the training corpus. Due to the Zipf's Law, words with the highest frequency should be used. Such features are used in the frequent words method, where a language model is generated using a specific amount of the words, having the highest frequency of all words occurring in a text or text corpus. The words are sorted in descending order of their frequencies.

For example, Table 1 shows the most frequent words generated from the datasets of news collected for the Leipzig Corpora Collection [QRB06]. It is quite obvious that many of these words, such as "de", "la", "a", are shared between more than one language making choosing between them more difficult. For the tweet "*los niños de haití agradecen el apoyo de la sociedad española con sus dibujos*" the occurrences of each word in this table are checked. For Spanish it will be 5 of them and for French, Galician, Dutch, Portuguese and Catalan 2. If only the word occurrences are considered in the training set, Spanish, as the language with most occurrences, will be chosen as the language for this tweet.

French	German	Spanish	Galician	Dutch	Portuguese	Catalan	Italian
de	der	de	de	de	de	de	di
la	die	la	a	van	a	la	e
le	und	que	e	een	que	que	il
à	in	el	que	en	o	i	la
et	den	en	o	het	e	a	che
les	von	y	do	in	do	el	in
des	mit	a	da	is	da	l	a
en	auf	los	en	op	em	en	per
a	das	del	un	te	para	per	un
l	zu	se	unha	met	os	del	del

Table 1: Most frequent words of European languages

2.1.2 Short Words Method

The short word-based approach is similar to the frequent words method, but it only uses words up to a specific length. Common limits are 4 and 5 letters. Words with this length are mostly determiners, conjunctions and prepositions, that are often language specific. Table 1 shows that the top ten words of all represented European languages are actually smaller than 5 characters. In the top hundred words will be also longer words, for example, in English, the word "people" is on the 53th place. Deleting such words is aimed to improve the categorization performance.

2.2 N-gram-based Modelling Methods

2.2.1 N-gram Method

Another successful approach for generating language models is the N-gram approach. Cavnar and Trenkle [CT94] used it for text categorization and found out that it also performed well on the task of language identification. In this approach, a language model is generated from a corpus of documents using N-grams instead of complete words, that are used in the first two approaches.

An **N-gram** is an contiguous N-character slice of a string or a substring of a word and respectively words depending on the size of N [CT94]. The beginning and the end of a word are often marked with an underscore or a space before N-grams are created. This helps to discover start and end N-grams at the beginning and ending of a word and to make the distinction between them and inner-word N-grams.

For instance, the word *data*, surrounded with the underscores, results in the following N-grams:

- unigrams: _, d, a, t
- bigrams: _a, da, at, ta, a_
- trigrams: _da, dat, ata, ta_
- quadgrams: _dat, data, ata_
- 5-grams: _data, data_
- 6-grams: _data_

Cavnar and Trenkle use N-grams of several different lengths simultaneously. The more common approach, however, is to use fixed lengths of N-grams. Grefenstette [Gre95] uses trigrams, Prager [Pra99] uses N-grams with N ranging from 2 to 5. Dunning [Dun94] generates N-grams of sequences of bytes in his work.

To detect the language of a document, at first its N-gram language model is created. Commonly, preprocessing is employed, i.e. punctuation marks are deleted and words are converted to lower case. Moreover, they are tokenized and surrounded with spaces or underscores. From these tokens, N-grams are generated and their occurrences are counted. The list of N-grams is sorted in descending order of their frequencies and the most frequent ones produce the N-gram language model of the document.

The main advantage of the N-gram-based approach is in splitting all strings and words in smaller parts than words. That makes errors, coming from incorrect user input or Optical Character Recognition (OCR) failures, remain only in some of the N-grams, leaving other N-grams of the same word unaffected, which improves correctness of comparing language models.

However, N-grams of small length are not very distinctive and some of them are present in language models of many languages. This does not happen with the first two approaches that are based on words instead of N-grams.

2.2.2 Graph-based N-gram Method

In the work of Tromp and Pechenizkiy [TP11] a Language Identification Graph-based N-gram Approach (**LIGA**) for language identification is described. They not only use N-gram presences and occurrences, but also their ordering, for that they create a graph language model on labelled data. The weights of the nodes represent the frequencies of trigrams and the weights of the edges capture transitions from one character trigram to the next. To create a language model, Tromp and Pechenizkiy use a training corpus of texts in that language. They calculate the frequencies of trigrams and their transitions and divide these counts by the total number of nodes or edges in the language.

In the word *lemon*, the nodes of the graph would be the trigrams *_le*, *lem*, *emo*, *mon* and *on_* and the edges would be (*_le*,*lem*),(*lem*,*emo*),(*emo*,*mon*) and (*mon*,*on_*). In total, there are 5 trigrams and 4 transitions (edges) between them. Each trigram has a frequency of $\frac{1}{5}$ and each transition has a frequency of $\frac{1}{4}$.

If two sentences from different languages are taken, for example, "*een test*" in Dutch and "*a test*" in English, the resulting graph will be as shown in Figure 2. In this figure all nodes and edges for these sentences are shown. It can be seen, that some nodes and transitions are shared between these two languages.

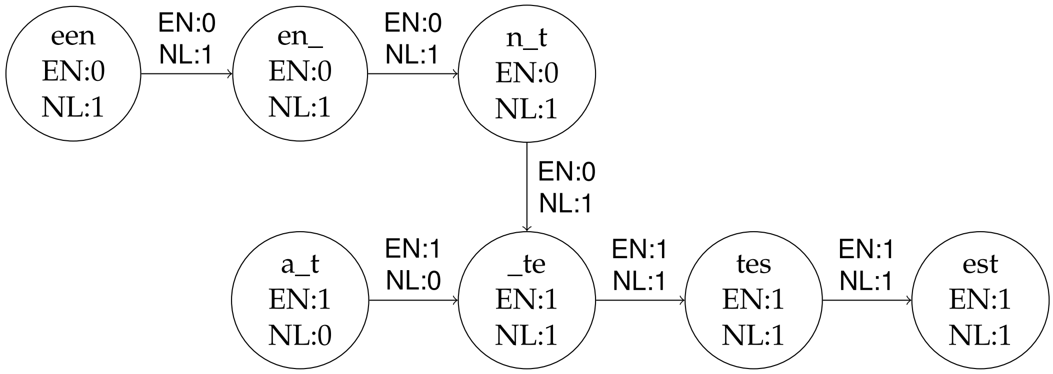

Figure 2: The graph resulting from the example training set

For example, if the training corpus contains the texts above, there would be 4 total trigrams in English and 6 in Dutch, 3 total edges in English and 5 in Dutch. To detect the language of the text "*a tee*", a flat graph is made, as shown in Figure 3. For each language a so-called path-matching score is computed. Only one node and no edges from the Dutch corpus is matched. On the other hand, two nodes and one edge from English corpus are matched. The path-matching score for Dutch is $\frac{1}{6}$ and for English is $\frac{1}{4} + \frac{1}{3} + \frac{1}{4} = \frac{5}{6}$. If this is the highest score out of all the language models, the text "*a tee*" will be classified as English.

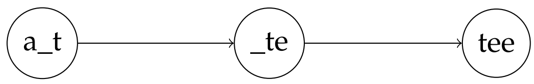

Figure 3: The graph resulting from the evaluation text

Tromp and Pechenizkiy have made experiments of their approach on their dataset, which are described later. Using only 5% of data for the training, they had results of 94.9% correctly identified texts. In contrast, the N-gram approach only identified 87.5%. With 50% of dataset used for training, the LIGA-approach achieved 97.5%, while the N-gram approach led to 93.1% correctly identified texts. Further increasing of the training size has not resulted in better results, but the LIGA-approach has always outperformed the N-gram approach.

2.2.3 Improved Graph-based N-gram Method

John Vogel et al. tried [VTK12] to improve the graph-based N-gram method and evaluated their approach on the same dataset used in [TP11]. They have found four different improvements to the basic LIGA algorithm: using word-length information, reducing the weight of repeated information, using median scoring, using log frequencies and combinations of these methods. The most successful of their improvements is to use log frequencies.

Due to the Zipf's Law the frequencies of N-grams have an exponential distribution. So, if one of the most frequent N-grams of one language is in the text from an another language, this text will most likely be missclassified. To reduce this effect, Vogel et al. take the natural logarithm of counts of N-grams and their transitions, eliminating N-grams that occur only one time. This improvement increased the percentage of correctly identified tweets of the LIGA dataset from 97.5% to 99.8%.

3 Classification Methods

To classify an input document with regard to the language models, the distances between them are calculated. The language with the minimal distance to the input document is chosen as the language of the document. Although, the calculation of distances is different for all classifiers, the first steps are basically the same. The classifiers from this chapter are described and later implemented in the same way as they are defined in [ACT04].

After preprocessing, different N-grams together with their frequencies of occurrence are extracted for each language in the training corpus. These counts are converted by taking the natural logarithm of them, then dividing each value by the highest count of entire dataset and adding one. Subsequently, for each N-gram the internal frequencies are calculated as shown in Equation 1.

$$F_I(i,j) = \frac{C(i,j)}{\sum_i C(i,j)} \tag{1}$$

$F_I(i,j)$ = Internal frequency of a N-gram i in language j
$C(i,j)$ = Count of the i-th N-gram in the j-th language
$\sum_i C(i,j)$ = Sum of the counts of all the N-grams in language

Table 2 shows an example calculation of the internal frequency for the N-gram "abc".

Language	N-gram	N-gram count	Total N-grams in this language	Total N-grams in all languages	Internal frequency
English	abc	23	150	1000	23/150
German	abc	47	350	1000	47/350
French	abc	82	500	1000	82/500

Table 2: Example of an internal frequencies calculation

3.1 Cumulative Frequency Addition Classifier

For this classifier, a list of N-grams is generated from the input document, without considering their amounts or any sorting. This list can contain duplicates. Each N-gram in this list is searched in the language model of a considered language. The internal frequencies of the N-grams found are summed up. The bigger the sum is, the smaller is the distance between the language model and the document model. Finally, the language with the maximal sum is chosen as the language of the document.

3.2 Naive Bayesian Classifier

Bayesian classifiers are used to classify a document (tweet) D to one of a set of predefined categories (languages) $C = \{c_1, c_2, ...c_n\}$ [PS03]. These classifiers use Bayes theorem, shown in Equation 2.

$$P(c_j|D) = \frac{P(D|c_j) \cdot P(c_j)}{P(D)} \tag{2}$$

$P(c_j|D)$ = Probability of belonging of a tweet D to the language c_j
$P(D|c_j))$ = Probability of generating a tweet D given language c_j
$P(c_j)$ = Probability of occurrence of language c_j
$P(D)$ = Probability of a tweet D occurring

A tweet is represented by a vector $D = (f_1, f_2, ..f_m)$ of m features, that are words or N-grams with their internal frequencies. The computation of $P(D|c_j))$ can be simplified with the additional assumption that each feature is conditionally independent of other features given the language. This assumption is embodied in the naive Bayesian classifier [Goo65] and the equation 2 is reduced to 3.

$$P(c_j|D) = P(c_j) \cdot \frac{\prod_{i=1}^{m} P(f_i|c_j)}{P(D)} \tag{3}$$

To find the most probable language of the tweet c maximum a posterior classifier (MAP) c_{MAP} is constructed in Equation 4. It maximizes the posterior $P(c_j|D)$. In Equation 6 $P(D)$ is eliminated as it is a constant for all languages . The probability of occurrence of each language $P(c)$ is assumed equal and is also excluded. Therefore, the c_{MAP} becomes equal to maximum likelihood classifier as shown in Equation 7.

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} \{P(c|D)\} \tag{4}$$

$$= \underset{c \in C}{\operatorname{argmax}} \left\{ P(c) \cdot \frac{\prod_{i=1}^{m} P(f_i|c)}{P(D)} \right\} \tag{5}$$

$$= \underset{c \in C}{\operatorname{argmax}} \left\{ P(c) \cdot \prod_{i=1}^{m} P(f_i|c) \right\} \tag{6}$$

$$= \underset{c \in C}{\operatorname{argmax}} \left\{ \prod_{i=1}^{m} P(f_i|c) \right\} \tag{7}$$

As a result, for this classifier, a list of N-grams with possible duplicates is generated as in previous chapter and their internal frequencies, obtained from the training data, are multiplied. The language with the maximal result is chosen as the language of the document.

3.3 Rank-order Statistics Classifier

To determine the language of a document, Cavnar and Trenkle [CT94] use a technique that calculates a so called out-of-place measure for each N-gram of the document model. It determines the distance between an N-gram of the document model and the different language models. This technique is also called rank-order statistics. An example is shown in Figure 4.

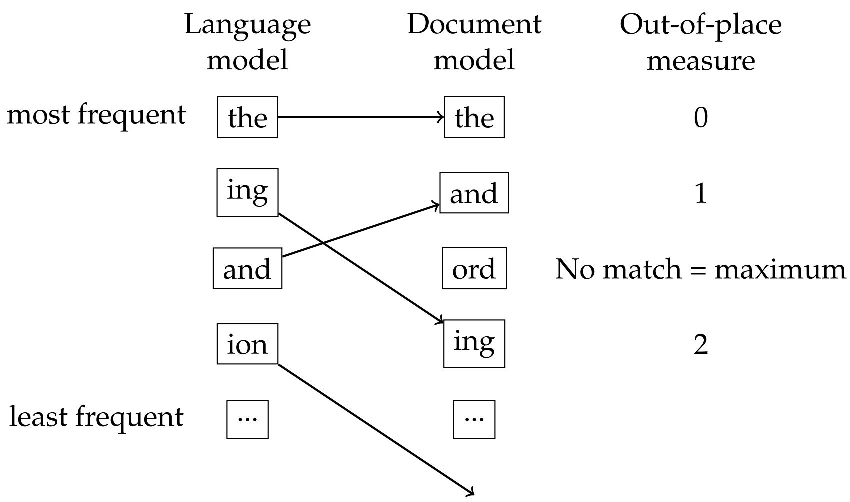

Figure 4: Example of the rank-order statistics classifier

If equal N-grams have the same rank in both models, like the N-gram "*the*", distance between them is zero. If the respective ranks for equal N-grams vary, their distance is the number of ranks between them, so the distance between the N-grams "*ing*" is 2. If an N-gram from the document model, like the N-gram "*ord*", is not found in the language model, their distance is defined as a maximum out-of-place value, which is generally the amount of N-grams in the language model. This is used to distinguish the correct language from the one with no matches. Subsequently, the sum of all out-of-place measures is the distance between the document model and the language model. Such distance is calculated for all languages and the smallest one indicates the language of the document.

4 Related work

In this chapter, the most related researches and approaches in language identification are summarized. Due to the long research history of this area it is increasingly difficult to give a comprehensive overview of the most important ideas. For more information about various approaches and applications of language identification with their accuracies and limitations, please, refer to the [GGJ14].

In the work of Cavnar and Trenkle [CT94] training sets in 8 languages on the order of 20K to 120K bytes in length have been used. Their validation set consisted of 3478 articles from a newsgroup hierarchy of Usenet that were fairly pure samples of a single language. From these articles, punctuation marks were deleted. Words were tokenized and delimited by white space before and after. But in contrast to other researches, described in this chapter, N-grams with length from 1 to 5 were extracted from these tokens with their total occurrences. Uni-grams were at the top of the list, but they were discarded, because they simply reflect the alphabetical distribution of a language. Remaining N-grams formed the language models of the documents.

Cavnar and Trenkle kept track if an article was over or under 300 bytes in length and varied the number of the N-gram frequencies from 100 to 400. The average text size was 1700 bytes. As shown in Table 3, the article length had a minor impact on the overall results of the language identification compared to the number of N-gram frequencies. Overall, their system showed the best performance at a training language model length of 400 N-grams, misclassifying only 7 articles out of 3478 and having an overall classification rate of 99.8%.

Article length (bytes)	< 300	< 300	< 300	< 300	> 300	> 300	> 300	> 300
Training model length (N-grams)	100	200	300	400	100	200	300	400
Overall correct	92.9%	97.6%	98.6%	98.3%	97.2%	99.5%	99.8%	99.8%

Table 3: Comparison of the results depending on the article and training model lengths from the work of Cavnar and Trenkle

They had also found interesting anomalies. An increasing N-gram model length decreased the percentage of correctly detected languages. This was mainly, due to the multiple languages that had a similar distance measures from the tested article.

The frequent words approach in their work is criticized, because for some of the lesser-used languages, building the representative language model is difficult. Another problem that was noted, is that some languages have different forms of words to indicate tense, case or other attributes. Therefore, the language models of them should be larger or include only words stems, what makes the process of building a model much more difficult.

The work of Dunning [Dun94] is similar to the one from Cavnar and Trenkle, but he omitted the tokenization step for building the N-grams. He used training sets of the length between 1000 and 50000 bytes for the English and Spanish language and validation sets of length from 10 to 500 bytes. In contrast to all other researches, to classify texts, he used a Bayesian classifier. He stated, that with small amounts of training data, lower order N-gram models work better, while with more training data, higher orders become practical. His approach worked with an accuracy of 92% with 20 bytes of validation data and 50K of training. With validation strings of 500 bytes and training text of 5 Kbytes he obtained an accuracy of 97%.

Souter et al. used the frequent words approach in their work [SCH+94]. They have identified one hundred high frequent words per language and calculated their probability values using training sets of roughly 100 kilobytes of text with one tenth of this data reserved for testing. Nine languages have been used: Friesian, English, French, Gaelic, German, Italian, Portuguese, Serbo-Croat, and Spanish. Machine-readable samples of each of these languages were obtained from the Oxford Text Archive at Oxford University. 91% of the test samples were correctly identified and the bigram method successfully identified 88% of the samples.

Grefenstette [Gre95] compared the short words approach with trigrams. He used one million characters of text from the European Corpus Initiative (ECI) collection [2] and considered the following ten languages: Danish, Dutch, English, French, German, Italian, Norwegian, Portuguese, Spanish and Swedish. He tokenized the sentences and counted all words and trigrams occurrences. He took words that have a length of five or less characters. Moreover, punctuation marks were not removed before generating the N-gram-based language models, which resulted in N-grams that contain only commas or dots. Each language was characterized with trigrams appearing at least 100 times of amount resulting from 2550 to 3560 N-grams and with words that occur at least three times resulting from 980 to 2750 words, depending on the language.

On test strings with 1 to 5 words, the results of the short words approach were worse, because of the high probability that no word is found in the language model. But with at least 15 words in test string round 99.9% of strings were correctly recognized. The trigram approach has shown better results than the short words approach almost in all of his tests. The samples with more words performed better, but starting with 15 words all methods performed equally well with round 99.9%.

Prager [Pra99] used to generate a training set from 100 Kbytes of text of 13 Western European languages. They were chosen, because they were of particular interest to IBM and these languages share etymological roots and have largely overlapping character sets, what made the task more difficult. As a validation set, he took chunks of text with sizes from 20 to 1000 bytes. He compared the results for all sizes of chunks, tried to find the N-gram length and performed additional experiments using both N-grams and words together as features. When they were used together, character sequences recognized both as a word and an N-gram were treated solely as a word in both indexing and matching processes.

Unlike Grefenstette, Prager used words up to four letters for the short words approach and calls them “stop-words”. He noted, that words of unrestricted length did better than short words, as he had expected, but both of them had good performance. He also

[2] http://www.elsnet.org/eci.html

thought that a set of only function words, such as pronouns, prepositions, articles and auxiliaries tend to be quite distinctive, and it should perform as good as a set of short words, but actual lists of such words were not available to Prager.

The combination of short words and N-grams showed better results than either of these methods alone. Quadgrams and words of unrestricted length had the best performance. The best N-gram length was 4, followed by 5, 3 and 2, which performed poor on the small sizes of chunks. As it was predicted by Prager, the longer input texts was better recognized than the small ones.

5 Datasets

To test the language identification methods described above datasets of tweets were used, that have only one language. Most of the datasets are collections of identification numbers of tweets, because in the Twitter API Terms of Service the redistribution of texts from tweets or information about users is prohibited. With the help of the identification numbers of tweets and Twitter REST API, the actual tweets were retrieved. Some of them have become unavailable, because the account or tweet was deleted or made private. To avoid having tweets from one user near each other, the tweet were shuffled. In these datasets all numbers, special characters and punctuation marks, except for apostrophes and dashes, were deleted, because they can possibly have a meaning for the task, and others are assumed as language independent. All words were converted to lower case and series of whitespace characters between the words were converted to only one whitespace character. References to users of Twitter, to locations of users, term references preceded by a hashtag sign and links were also deleted. The tweets, that turned up to be the same after preprocessing, were eliminated.

5.1 TweetLID Dataset

TweetLID [ZSVG+14] is a workshop with a shared task for the automatic identification of the language in which tweets are written. For the purpose of this workshop has been generated a corpus of tweets, which contains 34984 tweets. All tweets are annotated with their languages. Tweets in the dataset are in 6 languages: English, Portuguese, Basque, Catalan, Galician and round 62% of them are in Spanish, as shown in Table 4. Some tweets are multilingual or the language of them is not determined. The main interesting feature of this dataset is the presence of some not widely used languages and four of them (Spanish, Portuguese, Catalan, and Galician) belong to the same language family, which makes the distinction of them harder. Some of them had more than one language in one tweet, but they were not used. The corpus is released under the Creative Commons License.

Language tweets	Original size	After preprocessing
Spanish (es)	21417	20407
Portuguese (pt)	4320	4295
Catalan (ca)	2959	2916
English (en)	1970	1834
Galician (gl)	963	954
Basque (eu)	754	735
Total	34984	31141

Table 4: TweetLID dataset

5.2 LIGA Dataset

A Graph-based N-gram algorithm for language identification (**LIGA**), described before, was introduced in the work of Tromp and Pechenizkiy [TP11]. This algorithm was tested on a dataset of 9066 labelled messages of at most 140 bytes, that was collected using the Twitter API from six accounts per language. They are accounts of institutions and users known to only contain messages of a specific language. This operation was done for six languages that are German, English, Dutch, Spanish, French, and Italian. Using the last three languages is challenging, because they have similar words and N-gram patterns. Table 5 shows the distribution of the languages.

Language tweets	Original size	After preprocessing
English (en)	1505	1457
French (fr)	1551	1505
Dutch (nl)	1430	1368
Italian (it)	1539	1448
German (de)	1479	1442
Spanish (es)	1562	1508
Total	9066	8728

Table 5: LIGA dataset

5.3 Annotated Twitter Sentiment Dataset

Narr et al. [NHA12] have collected a dataset of tweets to use them in their work, where they make language independent sentiment analysis. The tweets have been human-annotated with sentiment labels by 3 Amazon's Mechanical Turk workers each. There are 12597 tweets in 4 languages: English, German, French and Portugese. As it can be seen in Table 6, most of tweets are in English. All tweets are also annotated with labels needed for sentiment analysis, but this feature of the dataset was not used in the present work.

Language tweets	Original size	After preprocessing
English (en)	7200	7019
German (de)	1800	1781
French (fr)	1797	1784
Portuguese (pt)	1800	1717
Total	12597	12301

Table 6: Annotated Twitter sentiment dataset

5.4 Non-Latin Dataset

Bergsma et al. [BMB+12] have collected tweets by users following language-specific Twitter sources, and used the Twitter API to collect tweets from users who are likely to speak the target language. They have collected tweets for nine languages Arabic, Farsi, Urdu, Hindi, Nepali, Marathi, Russian, Bulgarian, Ukrainian. This dataset was obtained with the help of Paul McNamee, who made it accessible on his website. It is more challenging in having languages that use the Cyrillic, Arabic, and Devanagari alphabets. The last two alphabets do not have capital letters and can not be converted to lower case. Table 7 shows the distribution of the languages in this dataset. Bergsma et al. not only provide

Language tweets	Original size	After preprocessing
Farsi (fa)	4878	3736
Arabic (ar)	2428	1858
Urdu (ur)	2389	2124
Hindi (hi)	1214	1067
Marathi (mr)	1157	1060
Nepali (ne)	1681	1576
Ukrainian (uk)	631	500
Bulgarian (bg)	1886	1764
Russian (ru)	2005	1618
Total	18269	15303

Table 7: Non-Latin dataset

tweets in non-Latin scripts, but also implement two LID approaches and compare them with state-of-the-art competitors (TextCat, Google CLD, langid.py). Their first approach is a discriminative classifier that uses the tweet text and metadata (user name, location, and links). The second approach is based on the prediction by partial matching and uses the PPM-A variant [CW84]. They devided their dataset in three groups, according to the used alphabets. Their two new approaches outperform other competitors and have the best results from 96% to 98.3% for different groups of languages. To achieve this results, the PPM approach was trained on documents from Twitter and Wikipedia. The discriminative approach used N-grams from tweets and their metadata together.

5.5 Dataset from the Work of Carter et al.

Dataset of tweets from the work of Carter et al. [CWT13] contains tweets in English, Dutch, French, German, and Spanish. Per language it has 1000 tweets, but after preprocessing and removing duplicates only 600-700 tweets remained. This dataset is not used in the evaluation part, because all of its languages are also found in the LIGA dataset, but there they are represented with more tweets per language.

6 Evaluation

The purpose of the present work is to compare the language identification approaches and their parameters. To accomplish it, these approaches were implemented in the Java programming language and evaluated with different parameters. The source code of the LIGA approach was found and adapted to be used among other methods, with different N-gram lengths and with the logarithmic frequencies as described in Section 2.2.3. All comparisons are illustrated with the graphs and the corresponding tables used for their generation can be found on the CD that comes with this work.

In this chapter F_1 measure is used to compare language identification approaches. It is calculated for each language and then averaged for the dataset. $\boldsymbol{F_1}$ is defined as the harmonic mean of precision and recall [Rij79]. They are calculated as shown in Equations 8 and 9 from the following values:
TruePositives is the number of tweets, for which the considered language is correctly identified.
FalsePositives is the number of tweets incorrectly identified as belonging to the considered language.
FalseNegatives is the number of tweets belonging to the considered language that are not correctly identified.

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{8}$$

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{9}$$

The F_1 measure is calculated as shown in Equation 10.

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{10}$$

The questions, stated in the introduction, are answered using the datasets from the Leipzig Corpora Collection [QRB06] as a training set. This collection has the corpora in sizes of 100000, 300000 and 1 million sentences for all 19 languages, that are used in the datasets of tweets. These sentences are randomly selected from the newspaper texts or texts from the web. Foreign language material is removed.

The latest collections of the news in all languages with 100000 sentences are taken each to generate training data. These collections are preprocessed like datasets of tweets described before. In the Leipzig Corpora Collection there are also lists of the most frequent words (**MFW**) for all used languages. These lists are used in the next chapter.

It is also possible to separate the datasets of tweets in training and evaluation parts and use them in the comparisons. However, in this case should be ensured, that the tweets in the training and evaluation parts are written by different users, because one user can more likely write similar tweets or utilise the same words in multiple tweets. This was not considered, when the datasets were obtained and preprocessed. Therefore, such comparisons were not made.

6.1 Comparison of the Word-based Methods

The first question, that is answered in the present work is the comparison of known word-based modelling methods for language identification. The words in the lists from Leipzig Corpora Collection that occur two or less times are removed. The internal frequencies of MFW are not used at first in the classification. Only the occurrences of the words in the lists are considered. For the short words approach (**SWA**) with maximal word length of 3 characters were taken only lists with up to 550 words. Longer lists can not be generated for some languages from Leipzig Corpora Collection. For the SWA with maximal word length of 4 letters such lists are up to 1662 and for 5 letters up to 3800 words.

As it can be seen in Figure 5, word-based methods do not work equally well on different datasets. For the LIGA dataset and the Twitter Sentiment dataset the average F_1 measure is shown in range from 80% to 100% and for the rest two datasets in range from 60% to 100% for the better visibility. On the LIGA dataset the frequent words approach (**FWA**) correctly identifies the language of almost all tweets. Using only 500 MFW 97.64% of tweets are correctly identified and the best average F_1 98.94% is achieved using 2750 words.

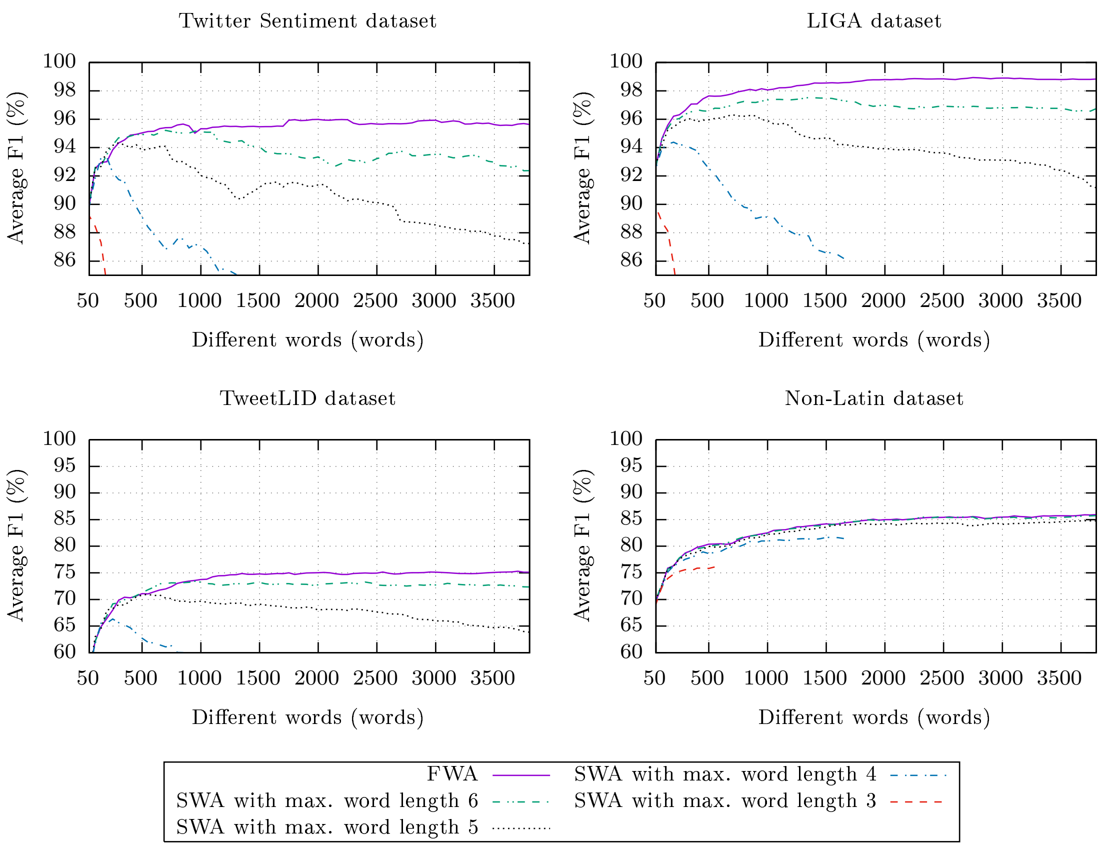

Figure 5: Comparison of the word-based methods without considering word frequencies

However, the best result for the TweetLID dataset is only 75.31% with 3700 used words.

This dataset has Spanish, Catalan, Basque and Galician language. It is difficult to distinguish between them. A half of tweets in Basque are often not identified and many tweets in Galician are identified as Spanish or Catalan tweets. Spanish tweets are often identified as Catalan. This may be because of the similar words in these languages, that also belong to the same language family. The other two datasets have shown moderate results, that are not so good as the results of the LIGA dataset.

Figure 5 shows, that in average all SWA perform worse than FWA. Removing too long words does not bring improvements and the results of all SWA are decreasing, starting from 300-600 words for different datasets. That may result from the noise in the last parts of the lists, where abbreviations and words from another languages can be found. The internal frequencies of the words are not considered and therefore this noise is not eliminated. Using more than 2000 words from the lists from the Leipzig Corpora improve results only for the Non-Latin dataset and other datasets have the best result around this amount of words.

If the internal frequencies of MFW are also utilized, the results will be better compared to the word-based approaches without considering frequencies, as it can be seen in Figure 6.

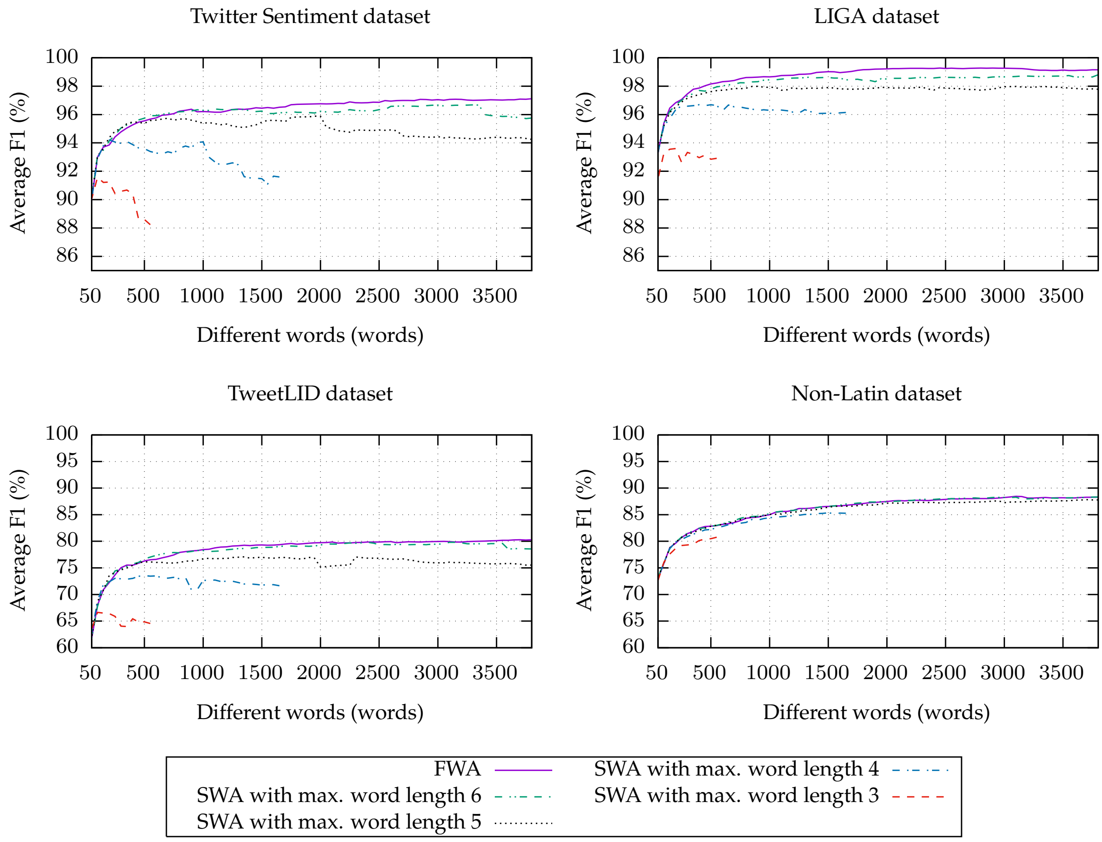

Figure 6: Comparison of the word-based methods considering word frequencies

In this comparison, the cumulative frequencies classifier is used, because in most cases for the word-based approaches it has better results, compared to other classifiers, as it is shown in Figure 7. The best result for the LIGA dataset is 99.28% using the FWA with 2450 words. In comparison to the approaches, that do not use frequencies, the results of the FWA for the TweetLID dataset are in average 5% better, the results of the Twitter Sentiment dataset and the Non-Latin dataset are 2-3% better and the results of the LIGA dataset are slightly improved.

All SWA work better than without considering their frequencies and for the Non-Latin dataset they perform as good as the FWA, with the exception of the SWA with maximal word length of 3 characters. The SWA have one of the best of their values for most of the datasets using 500 MFW and retain this value with more words.

Figure 7 shows that for most of the datasets and amounts of words, the FWA with the Cumulative Frequency Addition Classifier (**CFAC**) performs better than with other classifiers and reaches its plateau with 2000 different words. The average F_1 measure of the FWA with the CFAC is 1-3% better than with the Naive Bayesian Classifier (**NBC**) and 1-6% better than with the Rank-order Statistics Classifier (**RSC**) for the most datasets. However, for the LIGA dataset the NBC and the CFAC work quite the same and only 1% better than RSC. For this dataset the NBC outperforms the CFAC starting with 1950 words.

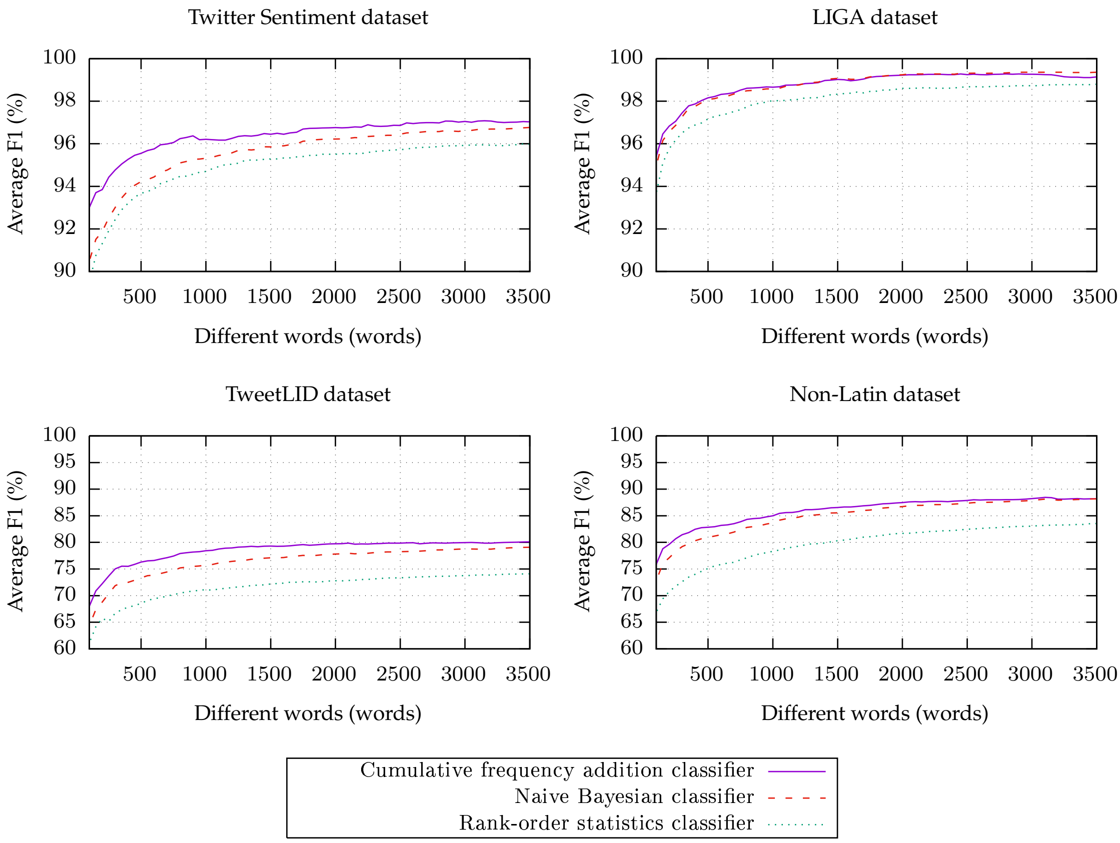

Figure 7: Comparison of classifiers for the frequent words approach

As it can be seen from the results described before, the short words approach used by

Greffenstette [Gre95] and Prager [Pra99] does not bring any improvements to the language identification of tweets for the used datasets compared to the FWA. For most of the datasets, increasing the amount of used words to more than 2000 words, does the results only slightly better. The best results for all datasets are shown in Table 8.

Dataset	Approach	Words	Classifier	Average F_1
Twitter Sentiment	Frequent words approach	3800	Cumulative frequency addition classifier	97.11%
LIGA	Frequent words approach	2900	Naive Bayesian classifier	99.36%
TweetLID	Frequent words approach	3700	Cumulative frequency addition classifier	80.28%
Non-Latin	Frequent words approach	3100	Cumulative frequency addition classifier	88.46%

Table 8: Comparison of the best word-based approaches

For most of the datasets the best performance is achieved with the FWA considering the internal frequencies of the MFW and using the CFAC. Only for the LIGA dataset the NBC outperforms other classifiers. The amounts of used words that maximize the performance vary from 2900 to 3800.

6.2 Comparison of the N-gram-based Methods

The N-grams can be used instead of the words. They can be of a different length. Many researchers ([Gre95], [TP11], [VTK12]) use the N-grams of the length 3, known as trigrams. Figure 8 shows the comparison of the lengths of the N-grams for three methods, that use them. These approaches are described in Section 2.2. The N-gram approach is shown in this figure with three different classifiers.

Only for the Non-Latin dataset 2 of 5 approaches have the best values with the N-gram length 3. The N-grams of the length 4 have the highest results and starting with the length 5 the average F_1 is decreasing for most of the approaches and datasets. The best results among all others have the LIGA dataset with the N-grams of the length 5 and the improved graph-based N-gram approach - 99.71% of the tweets are correctly identified. The results of this dataset are described below in more detail.

In contrast to the word-based methods, where the CFAC dominates other classifiers, the NBC works in average better than other classifiers for the N-gram approach. For the graph-based N-gram approaches is used only the CFAC, because this classifier was initially implemented by Tromp and Pechenizkiy.

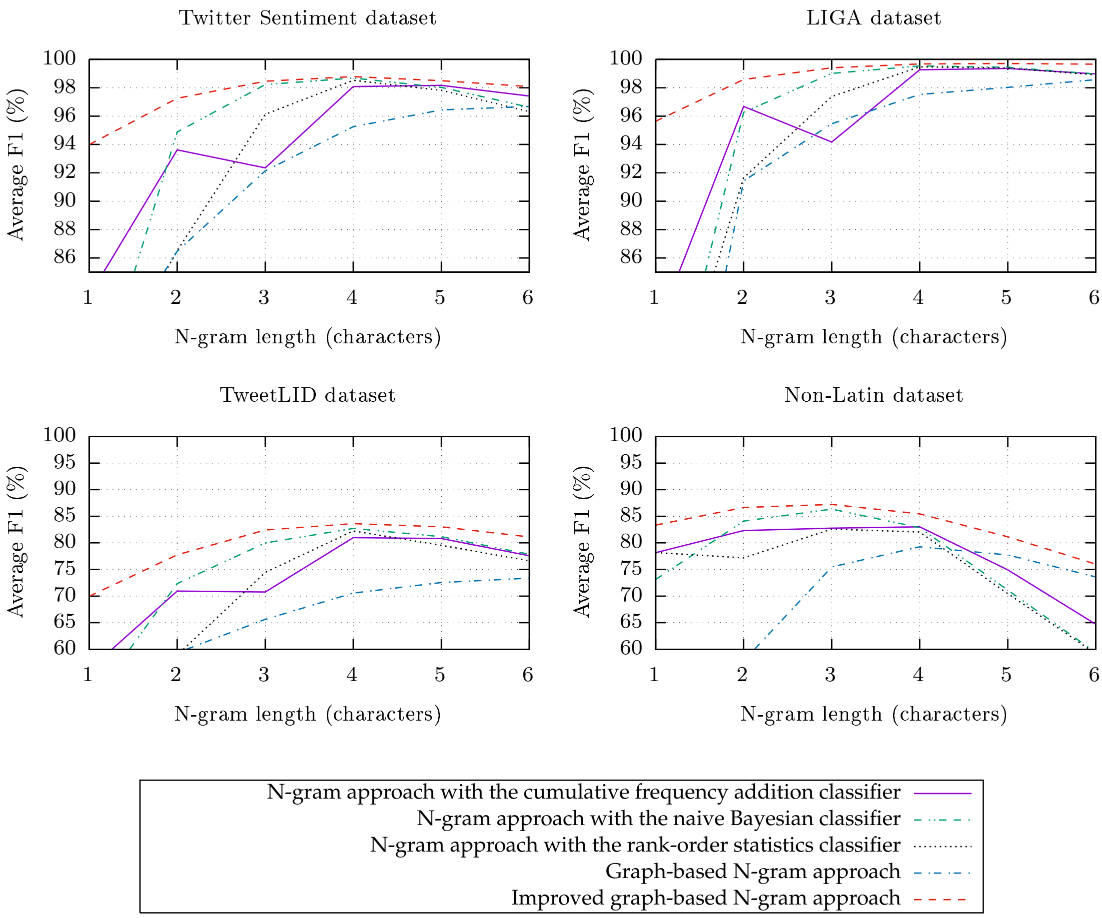

Figure 8: Comparison of the N-gram length

For all datasets except the Non-Latin dataset the N-gram approach with the CFAC has an anomaly with the N-grams of length 3, when the average F_1 value is decreased. This is not common for any other considered classifier.

The LIGA dataset has been used in different researches. In the work of Tromp and Pechenizkiy [TP11] using the trigrams and 50% of the LIGA dataset as the training part and 50% as the evaluation part was achieved that with the graph-based N-gram approach 97.5% of tweets were identified correctly and with the N-gram approach 93.1%. In the work of Vogel et al. [VTK12] using the same dataset and parameters with the improved graph-based N-gram approach the result was increased to 99.8%.

The results shown in Table 9 are obtained using the same dataset, but with 338 tweets removed after preprocessing, and with the external data from the Leipzig Corpora Collection used as the training data.

N-gram length	N-gram approach with CFAC	N-gram approach with NBC	N-gram approach with RSC	Graph-based N-gram approach with CFAC	Improved graph-based N-gram approach with CFAC
1	80.79%	70.47%	71.68%	60.15%	95.64%
2	96.68%	96.27%	91.65%	91.43%	98.58%
3	94.17%	99.01%	97.38%	95.46%	99.41%
4	99.27%	**99.54%**	99.47%	97.53%	99.68%
5	99.36%	99.45%	99.39%	98.03%	**99.71%**
6	98.98%	98.97%	98.91%	98.57%	99.66%

Table 9: The comparison of the N-gram-based approaches for the LIGA dataset

These results were also illustrated in Figure 8. The combination of the NBC with the N-gram length 4 is the best for the N-gram approach and outperforms the outcomes from the work of Tromp and Pechenizkiy. However, the results stated in the work of Vogel et al. are not achieved with any approach and parameters.

The best N-gram-based approaches are shown in Table 10. The improved graph-based N-gram approach with CFAC outperforms other approaches for all used datasets and the best N-gram length vary from 3 to 5.

Dataset	Approach	N-gram length	Classifier	Average F_1
Twitter Sentiment	Improved graph-based N-gram Approach	4	Cumulative frequency addition classifier	98.79%
LIGA	Improved graph-based N-gram approach	5	Cumulative frequency addition classifier	99.71%
TweetLID	Improved graph-based N-gram approach	4	Cumulative frequency addition classifier	83.63%
Non-Latin	Improved graph-based N-gram approach	3	Cumulative frequency addition classifier	87.23%

Table 10: Comparison of the best N-gram-based approaches

Finally, in Table 11 the highest average F_1 of word- and N-gram-based approaches are shown. The approaches with the best performance are also listed in this table.

Dataset	Best average F_1 of the word-based approaches	Best average F_1 of the N-gram-based approaches	Best approach
Twitter Sentiment	97.11%	**98.79%**	Improved graph-based N-gram approach with CFAC and N-gram length 4
LIGA	99.36%	**99.71%**	Improved graph-based N-gram approach with CFAC and N-gram length 5
TweetLID	80.28%	**83.63%**	Improved graph-based N-gram approach with CFAC and N-gram length 4
Non-Latin	**88.46%**	87.23%	Frequent words approach with CFAC and 3100 words

Table 11: Comparison of the best approaches

The differences between the best results of the N-gram and word-based approaches are only from 0.35% to 1.68% for all datasets except the TweetLID dataset, for which this difference is 3.35%. The improved graph-based N-gram approach with logarithmic frequencies with CFAC works in many cases better than the N-gram approach and has the best values for 3 of 4 datasets. Its best N-gram lengths vary for different datasets from 4 to 5. On the other hand, the highest average F_1 for the Non-Latin dataset is 88.46% and it is obtained with the FWA with the CFAC and 3100 words.

However, for this dataset in the original work of Bergsma et al. [BMB$^+$12] the best results for their two approaches, described in Section 5.4, vary from 96% to 98.3% for different groups of languages. For the LIGA datasets the highest results of the other researches were described in Section 6.2. For the Twitter Sentiment dataset in the original work was made only language-independent sentiment analysis. The results of the TweetLID dataset can not be compared with the outcomes from the works from the original workshop as in the present work all the tweets with multiple languages or with undefined languages were removed in the preprocessing step.

7 Conclusion

Language identification is an important preprocessing step for many NLP tasks. Some approaches, tending to solve it, identify nearly 100% of texts used for the evaluation, but their performance is strongly decreasing, when the length of the tested text is small. Twitter disposes a huge collection of such small texts, that are called tweets.

The purpose of the present work is to compare the language identification approaches and to find the most effective ones for four datasets of tweets. The short words approach, the frequent words approach, the N-gram approach, the graph-based N-gram approach and the improved graph-based N-gram approach are used for this comparison. The F_1 measure of each approach is calculated for each language and then averaged for the dataset to compare all approaches with their parameters.

At first, the word-based approaches with different parameters and classifiers are compared. All word-based approaches show the best performance with considering internal frequencies of the used words. Alternatively, only the occurrences of the words in the lists could be considered, but in this case the performance is from 2% to 5% worse. The most effective classifier for the word-based methods is the cumulative frequency addition classifier. When it is used, the average F_1 exceeds the results of other classifiers from 1% to 6% for different datasets. The best performance of the word-based methods is achieved using from 2900 to 3800 most frequent words from the Leipzig Corpora Collection.

Subsequently, the comparison of the N-gram-based approaches with different parameters and classifiers is made. For the N-gram approach the naive Bayesian classifier has the highest average F_1 for most of the datasets. It is outperformed by the cumulative frequency addition classifier only for the Non-Latin dataset. For this dataset the best N-gram length for most of the N-gram-based approaches is 3, for the LIGA dataset it is 5 and for the two other datasets it is 4.

Finally, the best approaches are determined for each dataset. All of them use the cumulative frequency addition classifier. For the Non-Latin dataset the best approach is the frequent words approach with 3100 words. As it is the only dataset with the languages having non-Latin alphabets, it can be concluded, that for such languages the frequent words approach is the most effective one among all considered approaches. However, two approaches with better results for this dataset are described in the original work of Bergsma et al. The improved graph-based N-gram approach is the most effective for three other datasets. This approach modifies the graph-based N-gram approach of Tromp and Pechenizkiy with taking the natural logarithm of counts of N-grams and their transitions.

Investigation of the performance of the improved graph-based N-gram approach with other classifiers than the cumulative frequency addition classifier can be recommended for the future works. Moreover, the comparison of different smoothing techniques is advisable for this classifier. As it can be stated from the present work the further researches should consider using not only trigrams, but the N-grams of the length 4 and 5 as in most cases they perform better. The short words approach is not recommended for the following researches as it is always outperformed by the frequent words approach.

References

[ACT04] AHMED, Bashir ; CHA, Sung-Hyuk ; TAPPERT, Charles: Language identification from text using n-gram based cumulative frequency addition. In: *Proceedings of Student/Faculty Research Day, CSIS, Pace University* (2004), S. 12–1

[AGL+11] ACHREKAR, Harshavardhan ; GANDHE, Avinash ; LAZARUS, Ross ; YU, Ssu-Hsin ; LIU, Benyuan: Predicting flu trends using twitter data. In: *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on* IEEE, 2011, S. 702–707

[AMM11] ARAMAKI, Eiji ; MASKAWA, Sachiko ; MORITA, Mizuki: Twitter catches the flu: detecting influenza epidemics using Twitter. In: *Proceedings of the conference on empirical methods in natural language processing* Association for Computational Linguistics, 2011, S. 1568–1576

[BMB+12] BERGSMA, Shane ; MCNAMEE, Paul ; BAGDOURI, Mossaab ; FINK, Clayton ; WILSON, Theresa: Language identification for creating language-specific twitter collections. In: *Proceedings of the second workshop on language in social media* Association for Computational Linguistics, 2012, S. 65–74

[CT94] CAVNAR, William B. ; TRENKLE, John M.: N-Gram-Based Text Categorization. In: *In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, 1994, S. 161–175

[CW84] CLEARY, John G. ; WITTEN, Ian: Data compression using adaptive coding and partial string matching. In: *Communications, IEEE Transactions on* 32 (1984), Nr. 4, S. 396–402

[CWT13] CARTER, Simon ; WEERKAMP, Wouter ; TSAGKIAS, Manos: Microblog language identification: overcoming the limitations of short, unedited and idiomatic text. In: *Language Resources and Evaluation* 47 (2013), Nr. 1, S. 195–215

[Dun94] DUNNING, Ted: *Statistical identification of language*. Computing Research Laboratory, New Mexico State University, 1994

[GGJ14] GARG, Archana ; GUPTA, Vishal ; JINDAL, Manish: A Survey of Language Identification Techniques and Applications. In: *Journal of Emerging Technologies in Web Intelligence* 6 (2014), Nr. 4, S. 388–400

[Goo65] GOOD, Irving J.: *The estimation of probabilities: An essay on modern Bayesian methods*. Bd. 30. MIT press, 1965

[Gre95] GREFENSTETTE, Gregory: Comparing two language identification schemes. In: *Proceedings of the 3rd International Conference on Statistical Analysis of Textual Data (JADT 95)*, 1995, S. 263–268

[Kri13] KRIKORIAN, Raffi: New tweets per second record, and how. In: *Twitter Blogs* 16 (2013)

[McN05] McNamee, Paul: Language identification: a solved problem suitable for undergraduate instruction. In: *Journal of Computing Sciences in Colleges* 20 (2005), Nr. 3, S. 94–101

[MRPS11] Muralidharan, Sidharth ; Rasmussen, Leslie ; Patterson, Daniel ; Shin, Jae-Hwa: Hope for Haiti: An analysis of Facebook and Twitter usage during the earthquake relief efforts. In: *Public Relations Review* 37 (2011), Nr. 2

[NHA12] Narr, Sascha ; Hulfenhaus, Michael ; Albayrak, Sahin: Language-independent Twitter sentiment analysis. In: *Knowledge Discovery and Machine Learning (KDML), LWA* (2012), S. 12–14

[Pra99] Prager, John M.: Linguini: Language Identification for Multilingual Documents. In: *Journal of Management Information Systems*, 1999, S. 1–11

[PS03] Peng, Fuchun ; Schuurmans, Dale: *Combining naive Bayes and n-gram language models for text classification*. Springer, 2003

[QRB06] Quasthoff, Uwe ; Richter, Matthias ; Biemann, Christian: Corpus portal for search in monolingual corpora. In: *Proceedings of the fifth international conference on language resources and evaluation* Bd. 17991802, 2006

[Rij79] Rijsbergen, C. J. V.: *Information Retrieval*. 2nd. Newton, MA, USA : Butterworth-Heinemann, 1979. – ISBN 0408709294

[SCH+94] Souter, Clive ; Churcher, Gavin ; Hayes, Judith ; Hughes, John ; Johnson, Stephen: Natural language identification using corpus-based models. In: *Hermes Journal of Linguistics* 13 (1994), Nr. S 183, S. 203

[SDGS13] Sinha, Shiladitya ; Dyer, Chris ; Gimpel, Kevin ; Smith, Noah A.: Predicting the NFL using Twitter. In: *arXiv preprint arXiv:1310.6998* (2013)

[SOM10] Sakaki, Takeshi ; Okazaki, Makoto ; Matsuo, Yutaka: Earthquake shakes Twitter users: real-time event detection by social sensors. In: *Proceedings of the 19th international conference on World wide web* ACM, 2010, S. 851–860

[TP11] Tromp, Erik ; Pechenizkiy, Mykola: Graph-based n-gram language identification on short texts. In: *Proc. 20th Machine Learning conference of Belgium and The Netherlands*, 2011, S. 27–34

[VTK12] Vogel, John ; Tresner-Kirsch, David: Robust language identification in short, noisy texts: Improvements to liga. In: *Proceedings of the Third International Workshop on Mining Ubiquitous and Social Environments (MUSE 2012)*, 2012, S. 43–50

[WTW13] Williams, Shirley A. ; Terras, Melissa M. ; Warwick, Claire: What do people study when they study Twitter? Classifying Twitter related academic papers. In: *Journal of Documentation* 69 (2013), Nr. 3, S. 384–410

[Zip49] Zipf, George K.: *Human Behaviour and the Principle of Least Effort: an Introduction to Human Ecology*. Addison-Wesley, 1949

[ZSVG+14] ZUBIAGA, Arkaitz ; SAN VICENTE, Inaki ; GAMALLO, Pablo ; PICHEL, José R. ; ALEGRIA, Inaki ; ARANBERRI, Nora ; EZEIZA, Aitzol ; FRESNO, Vıctor: Overview of tweetlid: Tweet language identification at sepln 2014. In: *Tweet-LID@ SEPLN* (2014)

List of Figures

1 Dataflow for N-gram-based text categorization [CT94] 2
2 The graph resulting from the example training set 5
3 The graph resulting from the evaluation text 6
4 Example of the rank-order statistics classifier 9
5 Comparison of the word-based methods without considering word frequencies 17
6 Comparison of the word-based methods considering word frequencies . . 18
7 Comparison of classifiers for the frequent words approach 19
8 Comparison of the N-gram length 21

List of Tables

1 Most frequent words of European languages 3
2 Example of an internal frequencies calculation 7
3 Comparison of the results depending on the article and training model lengths from the work of Cavnar and Trenkle 10
4 TweetLID dataset 13
5 LIGA dataset 14
6 Annotated Twitter sentiment dataset 14
7 Non-Latin dataset 15
8 Comparison of the best word-based approaches 20
9 The comparison of the N-gram-based approaches for the LIGA dataset . . 22
10 Comparison of the best N-gram-based approaches 22
11 Comparison of the best approaches 23