LANGUAGE IDENTIFICATION IN TEXTS

Tommi Jauhiainen

Doctoral dissertation, to be presented for public discussion with the permission of the Faculty of Arts of the University of Helsinki, in Auditorium XII, University Main Building, on the 28th of May, 2019 at 12 o'clock.

ISBN 978-951-51-5130-8 (paperback)
ISBN 978-951-51-5131-5 (PDF)
University of Helsinki
Helsinki 2019

Abstract

This work investigates the task of identifying the language of digitally encoded text. Automatic methods for language identification have been developed since the 1960s. During the years, the significance of language identification as an important preprocessing element has grown at the same time as other natural language processing systems have become mainstream in day-to-day applications.

The methods used for language identification are mostly shared with other text classification tasks as almost any modern machine learning method can be trained to distinguish between different languages. We begin the work by taking a detailed look at the research so far conducted in the field. As part of this work, we provide the largest survey on language identification available so far (Publication 1).

Comparing the performance of different language identification methods presented in the literature has been difficult in the past. Before the introduction of a series of language identification shared tasks at the VarDial workshops, there were no widely accepted standard datasets which could be used to compare different methods. The shared tasks mostly concentrated on the issue of distinguishing between similar languages, but other open issues relating to language identification were addressed as well. In this work, we present the methods for language identification we have developed while participating in the shared tasks from 2015 to 2017 (Publications 2, 3, and 4).

Most of the research for this work was accomplished within the Finno-Ugric Languages and the Internet project. In the project, our goal was to find and collect texts written in rare Uralic languages on the Internet (Publication 6). In addition to the open issues addressed at the shared tasks, we dealt with issues concerning domain compatibility and the number of languages. We created an evaluation set-up for addressing short out-of-domain texts in a large number of languages. Using the set-up, we evaluated our own method as well as other promising methods from the literature (Publication 5).

The last issue we address in this work is the handling of multilingual documents. We developed a method for language set identification and used a previously published dataset to evaluate its performance (Publication 7).

Preface and Acknowledgements

It started in 2015. I had somehow managed to miss the shared task concentrating on discriminating between similar languages that had been held in 2014 at one of the COLING workshops. I cannot pinpoint the exact time I became aware of its existence, but by the 2nd of February 2015, I had downloaded the DSL dataset from GitHub and noticed that it was incomplete. Perhaps I was the first one trying to re-use it after the shared task? I e-mailed Liling Tan, who was indicated as the corresponding author for the dataset, and 30 minutes later she had fixed the GitHub page and I was on my never-ending path "just trying to see how my LI method fares with close languages". With Krister and Heidi,[1] we decided to participate in the 2015 edition of the shared task, and I guess we did quite well, my method being beaten just by a bunch of SVMs. We were supposed to present our poster at the workshop in Hissar, but we never got there due to an unfortunately timed Lufthansa strike cancelling all European flights. I had been looking forward to chatting with Marcos Zampieri, the main organizer of the series of these language identification shared tasks to date, as we were supposed to share a shuttle from Sofia to Hissar. Meeting Marcos was delayed by three years. In hindsight, meeting Marcos at that time might have shaved off a year or two from the publication date of a certain survey article as well.

In late 2010, I was faced with two possible futures. An interesting leadership position had been opened at the National Library of Finland and the directors of my department, Kristiina and Annu, had decided to invite me for a job interview on the 4th of November. A few weeks earlier, I had handed over the almost final version of my master's thesis, where I sketched out my language identification method, to Professor Koskenniemi. Kimmo had liked it a lot and, by off chance, had met with Kristiina just days before my job interview and, among other things, had shortly discussed my thesis as well. On Monday the 8th, my colleagues (and I?) were informed that I had been selected to the new managerial position and would commence in it in three weeks time, more or less. I submitted the final version of the thesis the day after the announcement and got back a draft of the thesis review by Atro and Kimmo on Thursday the same week. I remember sitting down on a sofa in the Metsätalo basement after Anssi's lecture on automata theory to read their review. They wanted me to write an article about my language identification method as soon as possible and suggested that I should try to submit it by the ACL deadline in December. I read through the review many times, but I guess I never go to a job interview without already having decided to really want the job, so I was committed to a leadership career and language identification would have to wait.

It all began in late 2007. Krister was hosting a session at the language technology research seminar on thesis possibilities regarding open morphological and lexical resources. My bachelor's thesis was already almost done, and I was open for new ideas.

[1] Dramatis personæ: Krister Lindén, Heidi Jauhiainen, Kristiina Hormia, Annu Jauhiainen, Kimmo Koskenniemi, Atro Voutilainen, and Anssi Yli-Jyrä.

During the session, I became enthused by the idea of collecting material for an openly available Finnish sentence corpus from the Internet and decided that it was what I wanted to do for my master's thesis. Later, I sat down with Kimmo to present my idea about collecting texts from the Internet and Kimmo asked something like: "But how do you know when a text is written in Finnish?" A question that I have ever since strived to answer and to which this current thesis is still just a partial response.

Since starting my journey on language identification, I have become hugely indebted to a great number of people. Heidi, Kimmo, and Krister have persistently stood by me from the beginning to the present day and this thesis would not exist without any one of them. Most of the work which has been done for this thesis has been conducted as part of the "Finno-Ugric Languages and the Internet" project funded by the Kone Foundation. Without the four-year personal grant from the Foundation, it would not have been possible for me to detach myself from a position at the National Library long enough to really start reinvestigating language identification. In addition to the Kone Foundation itself, I thank especially Jussi-Pekka Hakkarainen and Jack Rueter for introducing me to the Foundation's language programme as well as for all their help during the project. I am also indebted to Kristiina Hormia for granting me leave of absence in order to pursue my scientific ambitions.

I am very grateful for the valuable comments of the preliminary examiners of this thesis, Nikola Ljubešić and Gregory Grefenstette. Without their input, I would not be nearly as satisfied with the manuscript as I currently am. I also thank Professor Jörg Tiedemann for his comments on the manuscript. I am also grateful for all the support and encouragement I have received from my colleagues at the various departments of the University of Helsinki. I am afraid I have been blessed with so many of you that you are too numerous to be mentioned here as are my other friends for whose support and friendship I am also eternally thankful.

Lastly, I would like to thank my family for their love and support through thick and through thin.

Contents

1. Introduction

1.1 Language Identification of Digital Text

Automatic methods for language identification of digital text have been developed since the 1960s (Publication 1). During the years, its significance as an important preprocessing element has grown at the same time as other natural language processing systems have become mainstream in day-to-day applications. In order, for example, to perform machine translation on a piece of text, the language to be translated from must be known. Without some sort of language identification system, the users have to indicate the language of the text manually. Google translate is an example of a system where language identification has been incorporated.

The methods used for the task of language identification are mostly shared with other classification tasks as almost any modern machine learning method can be trained to distinguish between different languages (Publication 1). However, some of the otherwise very successful new machine learning methods, such as deep neural networks, have not been able to surpass the more traditional approaches in language identification as quickly as in other classification tasks (Çöltekin and Rama [2016], Gamallo et al. [2016], and Medvedeva et al. [2017]). Furthermore, the task of language identification is far from being completely solved as is evidenced by, for example, the results from the series of shared tasks related to language identification of close languages, dialects, and language variants (Zampieri et al. [2014], Zampieri et al. [2015b], Malmasi et al. [2016], Zampieri et al. [2017], and Zampieri et al. [2018]). Publications 2, 3, and 4 of this dissertation describe our project's participation in these shared tasks from 2015 to 2017. Each task included a closed and an open track. On the closed tracks, the participants were only allowed to use the material provided by the task organizers. On the open tracks, they were allowed to use any material that they had at their disposal. In Publications 2, 3, and 4 we focus especially on the Discriminating between Similar Languages (DSL) shared task.

In addition to dealing with very similar languages, there are other open issues in language identification. Some of these issues, which would benefit from further research, are briefly introduced in the following section.

The Need for Surveys One of the challenges in researching language identification has been the fact that the task can be seen as falling into many different branches of science. There has not been a comprehensive survey that introduces previous research. Due to the lack of a proper survey, many experiments have been conducted several times and the work of others has gone unnoticed. As part of this thesis, we provide the largest survey on language identification available to researchers so far (Publication 1).

Generative vs. Discriminative Language Identification Classification methods, including those used for language identification, can be roughly divided into two categories: generative and discriminative (Ng and Jordan [2002]). In generative classification, each language is modelled on its own and then the model is used to calculate the probability for the text to be identified, independently of other possible language models. In discriminative language classification, the differences between the languages are modelled and then the differences are used to directly calculate the probability of the text being written in some language. Most methods include properties from both.

1.2 Open Issues

The intended application determines the attributes that need to be taken into account when developing or choosing a language identification method for a language identifier. The exact definition of the constraints determines the difficulty of the task itself. The handling of many of these constraints, like the number or closeness of the languages, is considered an open issue especially when taken to extremes. Some of these constraints can make the task difficult on their own, and more so, when added together. In this section, we list those open issues and challenges in language identification research, that have been tackled in one or more articles included in this thesis. The following subsections do not form an exhaustive list of open issues and some more are considered, for example, by Hughes et al. [2006], Xia et al. [2009], Lui [2014], and Malmasi and Dras [2017].

Low Corpora Quality The quality of corpora can be measured by the correctness of their annotations; however, determining the correctness of an annotation indicating the language used can be difficult as even human annotators sometimes have disagreements (Zaidan and Callison-Burch [2014]). Depending on the other issues being investigated, the quality of these language annotations can be a hindering factor in the training and testing of language identifiers (Publications 2 and 7). Even if a corpus is supposed to be only in only one language, it can include shorter or longer passages in other languages. Using corpora becomes problematic if the language annotation is not done on the same level[2] when compared with the intended use. For example, the language annotation can be correct on a paragraph level, but it may still include individual sentences or words in other languages.

Small Amount of Training Material There are several empirical studies suggesting that modern machine learning methods work best when they are trained on large amounts of training data (Alex [2008], Bergsma et al. [2012], King et al. [2014], Malmasi et al. [2015], Malmasi and Dras [2015a], Adouane [2016], and Malmasi and Zampieri [2016]). The amount of training material available to train the language models for a language identifier can sometimes be very small, for example only a few kilobytes (Vatanen et al. [2010]). Even when the amount of data is very small, some methods still produce reasonably accurate identifications, while others do not (Vogel and Tresner-Kirsch [2012], King and Abney [2013], and Ljubešić and Kranjcić [2014]).

[2] These levels could be, for example: corpus, text, paragraph, sentence, or word.

Out-of-Domain Texts The concept of “domain” is widely used in language identification and related literature. Wees et al. [2015] note that even in the field of domain adaptation, the concept is not unambiguously defined and that interpretations commonly neglect the fact that *topic* and *genre* are different properties of text. In this work, we define a domain to be a property of any given text, combining the topic(s) and the genre(s) of the said text. In addition, it can also include information about other properties that make a text similar or dissimilar from other texts, such as the possible idiolect(s) or even dialect(s) used in the text.

Time and again in the language identification literature, the training data is said to be either in-domain or out-of-domain when compared with the test data (*e.g.* Ljubešić and Toral [2014], Kocmi and Bojar [2017], Li et al. [2018], and Zampieri et al. [2018]). However, we have observed that there are widely varying degrees of domain difference. The degree of domain difference between the training and the test data can be either planned or unplanned and it is set when the dataset is generated. For example, if the training data consists of texts in a completely different topic than the test data, the degree of domain difference is probably greater than when the texts are from the same topic. In addition, the text could be from the same journal or written by the same authors, which would increase the “in-domainness” factor. In an extreme in-domain case, a single text can be divided between the training and the test sets. Classifiers can be more or less sensitive to the domain differences between the training and the testing data depending on the machine learning methods used (Blodgett et al. [2017]).

Close Languages, Dialects, and Language Variants The task of language identification is less difficult if the set of possible languages does not include very similar languages. If we try to discriminate between very close languages or dialects, for example Bosnian and Croatian, the task becomes increasingly more difficult (Tiedemann and Ljubešić [2012]). The line between languages and dialects is not easy to draw, as the distinction can be political. The same methods that are used in language and dialect identification are used in discriminating between language varieties, which are not usually considered even different dialects, such as Brazilian and European Portuguese (Zampieri and Gebre [2012] and Zampieri et al. [2018]).

Short Texts The identification of language in long texts, such as complete documents, has been considered as a solved problem in the past (Hammarström [2007]). When we are dealing with short texts, for example tweets, the task becomes more difficult (Grefenstette [1995], Vatanen et al. [2010], and Ljubešić and Kranjcić [2015]). In Publication 5, we evaluate several language identification methods using different test text lengths. The results of the evaluation indicate that some, but not all, methods can identify a language from as short a sequence as five characters even when the number of languages to be considered is in the hundreds.

Large Number of Languages It has been well-established that the greater the number of languages to choose from, the harder the language identification task becomes (Majliš [2012], Rodrigues [2012], and Brown [2012, 2014]). Dealing with a large number of languages is an open issue as not all identification methods scale up to greater numbers, even though they might produce very good results with a few languages (Majliš [2012] and Publication 5). Only a small minority of available language identification methods have been evaluated using more than 100 languages.[3]

Unseen Languages Supervised language identification methods require training data on the languages that are to be classified. However, in a real world setting, a language identifier is prone to come into contact with languages it has not been trained to deal with (Xia et al. [2009]). Many articles describe evaluations of off-the-shelf language identification tools where the tools are applied to languages that are not in their repertoire. The ability to detect unseen languages is still a rarity among methods used for language identification.

Multilingual Documents Traditionally, most of the language identification literature concentrates on the identification of monolingual documents (Publication 1). When compared with the language identification of a monolingual document, the task of distinguishing between the individual languages of a multilingual document is more difficult (Lui et al. [2014] and Publication 7). The degrees of multilingualism in a document can range from paragraph level to single words or even to parts of words.

1.3 Organization of the Thesis

In the main Sections 2–5, we will go through the open issues and introduce the research we have conducted concerning each issue.

1.4 Publications

This section provides a short introduction to the publications included in this dissertation. For each publication, the contributions of the author are listed. The publications are not presented in chronological order, but in an order in which the contents of the articles would best be presented in a monograph.

1.4.1 List of Publications

1. Tommi Jauhiainen, Marco Lui, Marcos Zampieri, Timothy Baldwin, and Krister Lindén. Automatic Language Identification in Texts: A Survey. *(submitted to JAIR 10/2018)*, 2018c

2. Tommi Jauhiainen, Heidi Jauhiainen, and Krister Lindén. Discriminating Similar Languages with Token-based Backoff. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects, LT4VarDial '15*, pages 44–51, Hissar, Bulgaria, 2015b

[3] See Table 16 on page 50.

3. Tommi Jauhiainen, Krister Lindén, and Heidi Jauhiainen. HeLI, a Word-Based Backoff Method for Language Identification. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 153–162, Osaka, Japan, 2016

4. Tommi Jauhiainen, Krister Lindén, and Heidi Jauhiainen. Evaluating HeLI with Non-Linear Mappings. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 102–108, Valencia, Spain, 2017b

5. Tommi Jauhiainen, Krister Lindén, and Heidi Jauhiainen. Evaluation of Language Identification Methods Using 285 Languages. In *Proceedings of the 21st Nordic Conference on Computational Linguistics (NoDaLiDa 2017)*, pages 183–191, Gothenburg, Sweden, 2017a. Linköping University Electronic Press

6. Heidi Jauhiainen, Tommi Jauhiainen, and Krister Lindén. The Finno-Ugric Languages and The Internet Project. *Septentrio Conference Series*, 0(2):87–98, 2015a. ISSN 2387-3086. doi: 10.7557/5.3471

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

1.4.2 Author's Contributions and Introduction to Publications

Publication 1: Automatic Language Identification in Texts: A Survey During the last 50 years, automatic language identification of text has emerged as a separate field of study related to general text categorization. Especially within the last few years, the amount of relevant research has continued to increase. Despite the ongoing interest in the subject, the field was lacking a comprehensive survey article. Many researchers have been reinventing, reexperimenting, and reevaluating language identification methods without being aware of the work that has already been done. Publication 1 is a comprehensive survey article and a much needed companion to every researcher dealing with language identification. For the survey, we collected information from over 400 articles dealing directly with automatic language identification of text. In order to describe the various features and methods used in language identification in a unified way, we created a mathematical notation that could be used to rewrite many, if not all, of the mathematical formulas used in the surveyed articles.

This survey article is a combination of two previously written unpublished survey manuscripts. Especially Sections 4-6 (on pages 6 to 42 of Publication 1) were taken from a manuscript prepared for journal publication earlier by me and my supervisor Krister Lindén. The mathematical notation introduced in Section 4 is a product of a long co-operative process between me and my supervisor. For Sections 5 and 6, I did the actual surveying work: gathering the relevant articles and reading through them. I wrote the first versions of the method and the feature descriptions, as well as of the transformed equations found in the surveyed articles (during 2013–2018). The other survey manuscript had been prepared by Marco Lui, Marcos Zampieri, and Timothy Baldwin. In December 2017, I took the main responsibility for combining and updating the two 64-page manuscripts (Lui, Zampieri, and Baldwin manuscript was from 1/2015 and our manuscript from 1/2017). The updating and combining work led to a manuscript which had over 170 pages which I then edited down to less than 100 pages in March 2018. Further editing in co-operation with Baldwin during 2018 led to the currently available version. Section 3 was originally from the manuscript by Lui, Zampieri, and Baldwin, but it was heavily rewritten by me.

My contributions: The first contribution is *the survey* itself, the second contribution is *the mathematical notation*, and the third contribution is *the transformation of the original method descriptions into that notation*.

Publication 2: Discriminating Similar Languages with Token-based Backoff This workshop article is the first article describing the language identification method that I have been developing since my master's thesis. The method is best explained in Publication 3, but Publication 2 was the first time it was published. The article describes how we, for the first time, used the token based backoff method in the DSL shared task (Zampieri et al. [2015b]) to distinguish between a set of close languages and language varieties. The languages were divided into 6 groups.[4] We used a two-tiered approach to language identification, in which the language groups were identified first and then the individual languages were identified within the groups. The parameters for the language identification method were separately optimized for each language group when the individual languages were identified. On a separate track of the DSL 2015 shared task, the test set included additional unseen languages and we experimented with methods for their detection. For this article, I did the design and development of the methods used for language identification, their implementations in Java or Python, and designed and ran the identification experiments for the shared tasks. I was responsible for most of the text in the article.

My contributions: The first contribution is *the language identification method* itself, the second contribution is *the method for unseen language detection*, and the third is *the application of both methods in the shared task* of the workshop.

Publication 3: HeLI, a Word-Based Backoff Method for Language Identification This workshop article is the main article describing the *HeLI*[5] language identification method, which was previously explained in less detail in Publication 2. We won the second place in four tracks of the shared task. Identification of Arabic dialects was experimented with in addition to the DSL 2016 set of languages. The

[4] The language groups and the individual languages for the DSL shared tasks from 2015 to 2017 are listed in Table 13 on page 44.

[5] HeLI is an abbreviation/name for the "Helsinki Language Identifying method".

language model generation software was written in Java for the first time and the Java implementation of the HeLI method was rewritten. As part of the article, the software was published in GitHub as open source. For this article, I did the design and development of the language identification methods, their Java implementations, and designed and ran the identification experiments for the shared tasks. I was directly responsible for most of the text in the article. A poster was produced by the authors and presented at the workshop by me and Heidi Jauhiainen.

My contributions: The first contribution of this article is *the set of complete mathematical formulas* which are used to describe the HeLI language identification method, the second contribution is *the open source publication of the implementations*, and the third contribution is *the set of identification experiments on the Arabic dialects* as part of the shared task of the workshop.

Publication 4: Evaluating HeLI with Non-Linear Mappings This article describes our third participation in the VarDial workshop series. We experimented with some variations of the HeLI method, especially using different non-linear mappings proposed by Brown [2014]. We found that one of these mappings, the *Gamma* function, has a very similar effect on identification performance as the penalty value that was already a part of the HeLI method, thus not being able to improve the results. However, with the use of the *Loglike* function, we were able to slightly improve the performance on the development set and even more so on the test set. For this article, I did the design and development of the language identification methods, their implementation in Java, and also performed the identification experiments for the shared tasks. I wrote most of the text in the article. A poster was produced by the authors and presented at the workshop by Krister Lindén.

My contributions: The contribution of this article is *the evaluation of the non-linear mappings previously proposed by Brown [2014]* when used with the HeLI method.

Publication 5: Evaluation of Language Identification Methods Using 285 Languages This article describes research where we aimed to evaluate the most promising of the available language identification methods in an out-of-domain situation for as many languages as possible. A small survey of existing electronic text corpora was conducted while trying to find two different text sources for as many languages as possible. In addition, we created new text corpora for those rare languages in which existing corpora were not available by locating and downloading material from the Internet. In the end, we had an evaluation set for 285 languages.[6] Unfortunately, many of the web pages used in the creation of the corpus are under copyright and the corpus as a whole cannot be published. We evaluated our implementation of the HeLI method together with two existing language identifiers and

[6] The list of the languages and the links to the sources of their training, development and test material are listed on the web page: `http://suki.ling.helsinki.fi/LILanguages.html`. Some of the extremely rare Uralic languages might have data from only one text, thus making the test situation more in-domain in their case.

our implementations of four other methods. The other methods are presented using the unified mathematical notation. For this article, I did the design and development of the language identification methods, implemented them in Java, and designed and ran the identification experiments. I wrote most of the text in the article and gave a presentation at the conference.

My contributions: The first contribution of this article is *the collection and curation of text corpora for 285 languages*, the second contribution is *the implementation of four other language identification methods*, and the third contribution is *the extensive evaluation and analysis of all the considered methods.*

Publication 6: The Finno-Ugric Languages and The Internet Project This article introduces the Kone foundation-funded project "The Finno-Ugric Languages and The Internet". Most of the research for all of the publications included in this thesis was done within the framework of this project. One of the major goals of the project was to use web crawling in order to find and collect web pages containing texts written in under-resourced Uralic languages. A language identifier used in a web crawling environment faces issues with the speed of identification, unseen language detection, as well as with handling multilingual documents. For this article, I did the design and development of the language identification methods, implemented them in Java, and designed and ran the identification experiments. I was responsible for writing the second section of the article, but also contributed to all the other sections. A poster was produced by the authors and presented at the workshop by Heidi Jauhiainen and me.

My contributions: The first contribution of this article is *the implementation of a production version of the language identifier* capable of serving a web crawler system while the crawling is ongoing, and the second contribution is *the detailed analysis of the identification performance within the Uralic language group.*

Publication 7: Language Set Identification in Noisy Synthetic Multilingual Documents In language set identification, the aim is to identify the set of languages used in a multilingual text. For this article, we developed a language set identification method that can be used with existing language identification methods. We used it with the HeLI method and achieved very high accuracy on a previously published dataset (Lui et al. [2014]). As part of the research, we did a detailed error analysis and noticed some problems with the quality of the dataset. For this article, I did the design and development of the language identification methods, implemented them in Java, and conducted the identification experiments. I wrote most of the text in the article and gave a presentation at the conference. In addition to the oral presentation, we prepared a poster which was presented at the conference by me and Heidi Jauhiainen.

My contributions: The first contribution of the article is *the language set identification method*, the second contribution is *the evaluation of that method in a previously* *published dataset*, and the third contribution is *the error analysis* pointing out the problems with the existing dataset.

2. Overview

"Those who cannot remember the past are condemned to repeat it."

George Santayana, Reason In Common Sense (1905)

2.1 The Need for Surveys

Most scientific articles include a section dedicated to related work, where the authors give a summary of what has been done before in the field or subfield of the article. A survey article is a dedicated document, where earlier findings from a given area of interest have been collected. If the field in question has dedicated surveys worth mentioning, they can be reviewed in the previous research paragraphs of the research articles. On the other hand, if there are no surveys in the field, every researcher has to conduct some kind of a survey on their own for their articles, and of course, for their research as well. Having a decent survey article in the field helps researchers to catch up with the situation in the field and find the most relevant articles relating to the specific problem that they are beginning to investigate (Oard et al. [2011]).

Some fields have useful surveys, like the survey of machine learning in automated text categorization by Sebastiani [2002] or the survey of smoothing techniques by Chen and Goodman [1999]. One of the most objective ways to measure the success of a survey article, or of any article, is the number of citations it attracts from the surrounding scientific community.[7] Even good surveys do get outdated as time goes by, but often they will continue to be a much needed source of information for research in the field and might never become completely obsolete.[8] A good survey can be followed by later surveys continuing from the time that the first one ended, without needing to repeat the earlier research.

2.2 Previous Surveys in Language Identification

This section is a short survey into the previous surveys themselves. In the following paragraphs, we are referring to the number of "relevant" research articles the previous survey articles introduce. In this context, as a relevant research article we consider articles directly discussing the automatic identification of the language of digital text. Many other articles are indeed relevant to the field and to research as well.

Muthusamy and Spitz [1997] wrote a page-long sub-section of the language identification research so far. It is basically an index pointing to previous research (13 relevant articles: 1965–1994) and does not go into any detail about the methods used in language identification. They mention the identification of languages using

[7] In Google Scholar, Sebastiani [2002] has 9,138 and Chen and Goodman [1999] 3,326 citations, as of April 2019.

[8] In Google Scholar, Sebastiani [2002] has 588 and Chen and Goodman [1999] 231 citations in articles published in 2017.

non-Latin and non-alphabetical scripts as the next challenge for written language identification. Additionally, the sub-section discusses detecting the language directly from document images, which is a problem related more to optical character recognition than to the language identification of already digitally encoded text.

Juola [2006] provides a two-page introduction to language identification. The work of Muthusamy and Spitz [1997] is listed in the bibliography with seven other relevant articles (1988–2001). He gives a compact description of the language identification task and compares it with other similar tasks. This introduction cannot be considered a comprehensive survey article because it mentions only one of the dozens of articles dedicated to language identification published during the seven years prior to its own publication.

Hughes et al. [2006] review the previous research in language identification and identify outstanding issues: rare languages, unseen languages, sparse training data, multilingual documents, standard corpora for evaluation, evaluation criteria, preprocessing, non-Latin scripts, exotic encodings, length of text, and the use of linguistic content. Their four-page review refers to around 15 relevant articles (1988–2005). This article is the most cited survey article for language identification with its almost 80 citations in Google Scholar as of April 2019.

Shashirekha [2014] gives an overview of automatic language identification from written texts in four pages. She lists some of the existing challenges, methods, and tools that are related to language identification. She does not mention any of the previous survey articles but refers to 14 more recent (2004–2014) relevant articles as well as the most cited language identification article by Cavnar and Trenkle [1994].[9] As challenges, she lists many of the issues we have been working on over the years, namely the length of text, text quality, different encodings, multilingual documents, shared vocabulary, unseen languages, and closely related languages.

The 12-page long journal article by Garg et al. [2014] is the first one declaring itself to be a survey of language identification of text. Like Shashirekha [2014], Garg et al. [2014] failed to mention the earlier survey works by Muthusamy and Spitz [1997] and Juola [2006]. Otherwise, they have surveyed a greater number of relevant work than those before (over 30 articles: 1994–2013). From those articles, they have gathered methods used for language identification and explain some of them using text and diagrams. They list additional information, like identification performance, about the evaluations and experiments from those articles.

The book chapter by Zampieri [2016] discusses the task of automatic language identification. Within the 18 pages, he refers to over 40 articles (1988–2014) providing some details of the research presented in them. He does not refer to any of the earlier survey articles either.

The most recent addition to the family of survey articles is the article by Qafmolla [2017]. She gives a brief overview of both the spoken and the written language

[9] Cavnar and Trenkle [1994] had 2,000 citations in April 2019.

identification methods. The article refers to around ten relevant articles (1994–2013) but does not mention any of the previous survey articles.

What we can learn from the survey articles presented in this section is that there is seemingly no comprehensive survey article available for language identification. A survey article can hardly be called great if it fails to mention any of the older survey articles and/or is itself not mentioned by the newer ones. Apart from the work by Hughes et al. [2006], the survey article by Garg et al. [2014] is the only one that has really attracted some attention in the field, gaining 14 citations so far.[10]

2.3 Tale of a Survey

For our part, our survey began as a "Previous Research" section of a larger research article, which was a combination of the early versions of Publications 1, 3, and 5. In July 2013, we already had a list of over two hundred relevant articles and a little over a year later they were presented in a twenty-page section (with ten extra pages in the references section) summarizing the features and the methods used in language identification. In hindsight, it should not have been a surprise that the reviewers suggested submitting the section as a separate survey article. Separating the survey from the research began in November 2015, and the last updated version of that manuscript, which dates from January 2017, has 45 pages plus eighteen pages for the 353 references. It turned out that the years 2014 and 2015 were especially active in the field of language identification, with around a hundred new relevant articles.

Concurrently with the comprehensive survey being prepared by me and Krister Lindén, another group had formed with a similar aim. Marco Lui had written an excellent survey section for his PhD dissertation "Generalized Language Identification" (Lui [2014]). His literature review section was almost 70 pages long and the c. 220 references for the whole dissertation took another 20 pages. We were aware of the work, and we aimed to concentrate on doing a broad survey of the features and methods used in the literature, with exact mathematical formulas, so that our own survey would not duplicate too much of the work in Lui's more discussion-centered survey. After finishing his PhD, Lui and his supervisor Timothy Baldwin had teamed up with Marcos Zampieri to produce a concise survey article for the field of language identification. Both our groups had separately decided that there was a need for one.

In late November 2017, our two groups became aware of each other and the decision to join forces came quickly. It turned out that we each had a 64-page manuscript, including references. As we already had aimed to complement rather than duplicate Lui's literature review and as both manuscripts needed a lot of updating, we ended up with a combined manuscript of almost 180 pages. We edited a shortened, one hundred page version from the comprehensive one by April 2018 and it was subsequently uploaded to the arXiv e-print service.[11] The shorter version can be considered

[10] I have a list of over 200 relevant articles published in 2016–2018, which gives some indication of the size of the field.

[11] `https://arxiv.org/abs/1804.08186`

quite comprehensive even though it does not list every possible article ever published on the subject. We have surveyed all the features and methods used in language identification and we refer to the first and latest publications where they have been considered for language identification.[12] The survey which is submitted as part of this dissertation is the version currently in peer review at JAIR.[13]

Designing a survey is a compromise between readability, comprehensiveness, and time. When writing a comprehensive survey, for the first time, in a field where the number of articles grows more than linearly with time, it is very hard to finish the survey without it becoming outdated before it is published. This is mostly why we decided to publish the early version in arXiv as soon as possible.

2.4 Describing Features and Methods

The surveyed articles usually contain descriptions of the features and methods used in the experiments presented in each article. The descriptions can be very short, for example just mentioning a well-known machine learning technique (*e.g.* Ciobanu et al. [2018]), or they can be exceedingly long in cases where they are describing more original work (*e.g.* Butnaru and Ionescu [2018]). There is a multitude of ways to describe the features and methods used in the articles. Sometimes the authors just use words to describe how something is done, sometimes they draw diagrams to help the written descriptions, and then sometimes they use mathematical equations in order to make sure that the exact way something was done could be understood by the reader. There is also the possibility of including pseudocode, which relates to equations, but could be harder to read and usually takes up precious space in the article.

As there are quite a number of different ways to write mathematical equations, it is not at all clear what notation should be used. While surveying the previous research it became clear that the variations in mathematical notation hinder the easy understanding of the equations themselves. We wanted to describe the methods in the survey using equations, but we did not want to explain the notations used in the original articles as most researchers had used their own notation or a notation borrowed from some other field. This is why we decided to create a unified notation, by which we would be able to describe many different kinds of language identification methods. It is, of course, our hope that other researchers might find our notation usable for describing new methods in the future. We have used this notation in Publications 1, 3, 4, and 5. In the following Section 2.5, we construct a merged version of the "On notation" sections of those Publications and discuss how the notation was used in the Publications.

[12] Until the end of 2017.

[13] *The Journal of Artificial Intelligence Research*: https://www.jair.org

2.5 On Notation

A corpus C consists of individual tokens u which may be words or characters. A corpus C is a finite sequence of individual tokens $u_1, ..., u_{l_C}$. The total count of all individual tokens u in the corpus C is denoted by l_C. In a corpus C with non-overlapping segments S, each segment is referred to as C_S, which may be a short document or a word or some other way of segmenting the corpus. The number of segments is denoted as l_S.

A feature f is some countable characteristic of the corpus C.[14] When referring to all features F[15] in a corpus C, we use C^F and the count of all features is denoted by l_{C^F}. A set of unique features in a corpus C is denoted $U(C)$.[16] The number of unique features is referred to as $|U(C)|$. The count of a feature f in the corpus C is referred to as $c(C, f)$. If a corpus is divided into segments S, the count of a feature f in C is defined as the sum of counts over the segments of the corpus, i.e. $c(C, f) = \sum_{S=1}^{l_S} c(C_S, f)$. Note that the segmentation may affect the count of a feature in C as features do not cross segment borders.

A frequently-used feature is an n-gram, which consists of a contiguous sequence of n individual tokens. An n-gram starting at position i in a corpus is denoted $u_{i,...,i-1+n}$, where positions $i + 1, ..., i - 1 + n$ remain within the same segment of the corpus as i. If $n = 1$, f is an individual token. When referring to all n-grams of length n in a corpus C, we use C^n and the count of all such n-grams is denoted by l_{C^n}.[17] The count of an n-gram f in a corpus segment C_S is referred to as $c(C_S, f)$ and is defined by Equation 1:

$$c(C_S, f) = \sum_{i=1}^{l_{C_S}+1-n} \begin{cases} 1 & \text{, if } f = u_{i,...,i-1+n} \\ 0 & \text{, otherwise} \end{cases} \tag{1}$$

The set of languages is G and l_G denotes the number of languages. A corpus C in language g is denoted by C_g. A language model O based on C_g is denoted by $O(C_g)$. The features given values by the model $O(C_g)$ are the domain $dom(O(C_g))$ of the model. In a language model, a value v for the feature f is denoted by $v_{C_g}(f)$. When identifying the language of a text M in an unknown language, for each potential language g a resulting score $R(g, M)$ is calculated.

In Publications 3 and 5, we used the notation "$u_{i,...,i-1+n}$" for n-grams in the introduction of the notation, but we also used the notation "u_i^n" when describing how individual words t were scored. We defined "u_i^n" as "n-grams of characters u_i^n, where $i = 1, ..., l_t - n$, of the length n". In Publication 4, we used "u_i^n" notation

[14] For example, a certain n-gram or word.

[15] F includes all features of the same type as f, for example the same length n-grams.

[16] The set of unique features of the type F would be $U(C^F)$. In the future versions of this introduction to notation, we should probably include examples for easier comprehension.

[17] In most of the cases, l_{C^n} has the same meaning as l_{C^F} and should probably be omitted from the general introduction in the future and only introduced when especially needed.

in both the introduction and the article itself. The inconvenience with "$u_{i,\ldots,i-1+n}$" is the length, which is evidenced, for example, in Equation 14 of Publication 1 describing the Absolute Discounting smoothing technique:

$$P_{C_g}(u_i|u_{i-n+1,\ldots,i-1}) = \frac{c(C_g, u_{i-n+1,\ldots,i}) - D}{c(C_g, u_{i-n+1,\ldots,i-1})} + \lambda_{u_{i-n+1,\ldots,i-1}} P_{C_g}(u_i|u_{i-n+2,\ldots,i-1}) \quad (2)$$

In Publications 3 and 4, we use $R_g(M)$ instead of $R(g, M)$. In Publications 3, 4, and 5, $U(C)$ is said to refer to unique tokens in a corpus, but in Publication 1 we say that it refers to features as it can then be used in a more general way, with a token being just one type of feature. In Publication 1, we changed the u in Equation 1 to f for the same reason.

2.6 On The Equivalence of Methods

Yanofsky [2011] introduces a three-tiered classification where programs implement algorithms and algorithms implement functions. Functions always produce exactly the same results from exactly the same inputs. As examples of functions, Yanofsky [2011] gives the *sort* and the *find max* functions. In our publications, we have considered two language identification methods to be the same if they produce identical results from any input. Thus, what we call a "method", is called a "function" in the tiers presented by Yanofsky [2011]. Table 1 gives descriptions of the tiers in the context of language identification.

Our term	**Yanofsky**	**Description**
Method	Function	description of the procedure to identify the text using features f so that the procedure always produces the same results from the same input
Algorithm	Algorithm	well-defined computational procedure that implements a method
Program	Program	an implementation of an algorithm in a programming language

Table 1: Definitions of the terms method/function, algorithm, and program.

The algorithmic descriptions of some of the methods presented in the surveyed articles can be completely different. Sometimes the descriptions also leave room for interpretation on how to implement them. When are two algorithms different then? The question is considered in detail from many different points of view by Blass et al. [2009], but they do not provide any easy answers or definitions. Cormen et al. [1990] simply define an algorithm to be any well-defined computational procedure. As an example of two different algorithms, Yanofsky [2011] gives the *mergesort* and the *quicksort*, that implement the function *sort*. However, the exact definition of an algorithm is left for future work by Yanofsky [2011].

2.7 The Babylonian Confusion

In this section, we present one of the simpler methods used for language identification. The method we have chosen is the sum of relative frequencies using character n-grams or words as features. We use the method to showcase the problem with different notations or the lack of them. We reproduce some of the equations using the original notations from the articles (Equations 5–8), as well as quote the descriptions.

When calculated using the relative frequency, we define the value v of the feature f in the corpus C_g as in Equation 3

$$v_{C_g}(f) = \frac{c(C_g, f)}{l_{C_g^F}} \tag{3}$$

where $l_{C_g^F}$ is the count of all features of the same type[18] as f in the corpus C_g. We define the sum of values as in Equation 4.

$$R_{sum}(g, M) = \sum_{i=1}^{l_{M^F}} v_{C_g}(f_i) \tag{4}$$

where f_i is the ith feature found in the unknown text to be identified, also known as the mystery text M. The language with the highest score is the winner.

The first to use the sum of relative frequencies for language identification were Souter et al. [1994]. They did not use equations in order to formulate the method they used, defining it in words instead. First they define a table containing the relative frequencies of character bigrams as "*... the frequencies for each language represented as a percentage of the total number of bigraphs read in the training sample of that language.*" The language identification method is defined as "*For the bigraph and trigraph-based recognisers, quite a naive statistical approach was adopted. After each graph was read in, the table of percentages for each language was consulted, and the percentages simply added to a running total for each language.*"

Llitjós [2001, 2002] and Llitjós and Black [2001] define the probability $P(\text{trigram}|L)$ as the (Laplace smoothed) relative frequency of the trigram in language L. The Probability of the mystery text "input" for language L is calculated as in Equation 5

$$P(L|\text{input}) = \sum_{\text{input trigram}} \frac{C(\text{trigram})}{\sum_{\text{input trigram}'} C(\text{trigram}')} P(\text{trigram}|L) \tag{5}$$

where $C(x)$ is the number of times x occurs. This is the sum of relative frequencies normalized by the length of the mystery text. The length of the mystery text is equal to all languages L, which means that the normalization does not affect the ordering when the languages are ordered by the probability $P(L|\text{input})$. Llitjós

[18] The type can be, for example, the same length n-grams, n-grams of any length, suffixes, words, or POS tags.

[2001, 2002] calls this method a "*variation of the algorithm presented in Cavnar and Trenkle [1994], which only takes trigrams into account, as opposed to* n*-grams from* $n = \{1, 2, 3, 4\}$*, and assigns probabilities to the languages.*" However, the rank order method presented by Cavnar and Trenkle [1994] could really be considered to be further away from the sum of relative frequencies than, for example, the Naive Bayes (NB). We would be hard pressed to call this a variation of the method by Cavnar and Trenkle [1994] as the only commonality is the use of character n-grams.

Poutsma [2002] uses character trigrams with the sum of relative frequencies. He defines the probability $P(f|L)$ as the relative frequency of trigram f in language L and the sum as in Equation 6

$$\max P(L|D) = \max \sum_{f \in D} P(f|L) \tag{6}$$

where D is the document to be identified. He then continues to use the method with Monte Carlo sampling.

Ahmed et al. [2004] re-invent the same method as a "*new classification technique*" called Cumulative Frequency Addition (CFA). They give the following equation for the relative frequency $F_I(i, j)$:

$$F_I(i, j) = \frac{C(i, j)}{\sum_i C(i, j)} \tag{7}$$

where $C(i, j)$ is the i^{th} n-gram in the j^{th} language and $\sum_i C(i, j)$ is the sum of the counts of all the n-grams in language j. Ahmed et al. [2004] do not actually say how the score is calculated from the relative frequencies, perhaps relying on the quite descriptive name. Later, Babu and Kumar [2010] compare the CFA method, citing Ahmed et al. [2004], to the Neural Network (NN) and the rank order methods, but do not include any real description of the CFA method itself.

Qu and Grefenstette [2004] used character trigrams to identify names using the sum of relative frequencies of trigrams. They define the method using just words: "*... the trigrams for each list were then counted and normalized by dividing the count of the trigram by the number of all the trigrams ... we divide the name into trigrams, and sum up the normalized trigram counts from each language. A name is identified with the language which provides the maximum sum of normalized trigrams in the word.*"

Kastner et al. [2005] evaluate the sum of values method with character 4-grams against their own method. They rely on words to define the method: "*The probability that a tetra-gram identified a particular language was computed for all tetra-grams across all languages. A testing document was scored based on the sum of probabilities of the tetra-grams it contained.*" They do not define what they exactly mean with the probability in the description, so in theory the probabilities could be something else than relative frequencies, though we deem it unlikely.

McNamee [2005] uses the sum of relative frequencies with words. He uses vectors to define the method, explaining it in words and with an example using values from a table he presents. Each language model is a frequency-ordered vector of words and "*the percentage of the training data attributed to each observed word*". The sentence to be identified was also a vector of words and "*To compare the two vectors I used the inner product-based on the words in the sentence and the 1,000 most common words in each language.*"

Bosca and Dini [2010] experiment with a "*Pure Corpus Based*" method which turns out to be the sum of relative frequencies with words defined as follows: "*The guess confidence value consists in the normalized sum of term frequencies.*" It is possible that they use the same method with characters or character n-grams, but the method they used is quite vaguely defined and could really be almost anything: "*languages are evaluated comparing language model trained using textual contents from language specific corpora. The guess confidence represents the distance of the input text from a specific language model.*"

Tromp [2011] and Tromp and Pechenizkiy [2011] mention Ahmed et al. [2004] as their inspiration when presenting their graph-based n-gram method called LIGA. The algorithm is presented in a little over 2 pages using mathematical notation, figures, and descriptive text.[19] When the method is analyzed, it comes down to being the sum of relative frequencies using character tri- and quadri-grams. Later, LIGA was used by Vogel and Tresner-Kirsch [2012], who give a more compact description of the method. Later, Patel and Desai [2014], Abainia et al. [2016], and Moodley [2016] partly reproduce the original description by Tromp [2011] and Tromp and Pechenizkiy [2011]. We evaluated the method in Publication 5 and defined it using Equations 3 and 4.

Majliš [2011, 2012] and Majliš and Žabokrský [2012] define the same method, calling it the *YALI*[20] algorithm, in words and examples: "*The probability of each 4-gram is computed using the training data and only the first 100 are preserved. These probabilities are normalized to sum up to 1. During detection, the input text is preprocessed and divided into 4-grams. Scores for each language are summed up and the language with the highest score is the winner.*"

King et al. [2015] used the method in word-level language identification and explained calculating the relative frequencies as follows: "*(6) Tally the number of tokens for each n-gram type; (7) For each type, divide the number of its tokens by the total number of tokens in the training set*". Then in the actual testing phase: "*Then for each word, we search the English dictionary for each n-gram's probability, add these, and divide by the number of n-grams in the word to obtain an average n-gram probability for the word, which we take to represent the probability that the word is English. The process is repeated for Latin, and the English and Latin probabilities are compared, based on formula*":

[19] The presentation is far too long to be re-presented here.

[20] "Yet Another Language Identifier"

$$lg = \text{argmax} \sum_{\text{ngram}} P(\text{ngram}) \tag{8}$$

Martadinata et al. [2016] use relative frequencies of words in sentence level language identification, explaining it in the following way: "*After we have all the frequencies, the frequency will be converted into probabilities. The probabilities are based on the number of occurrences of the word divide with the number of occurrence of all word that occurs on the corpus. ... The language probabilities for the sentence are the sum of all probability on every word.*" Martadinata et al. [2016] mention that this was the technique implemented by Grefenstette [1995], but Grefenstette [1995] defines his word-based technique as a product of relative frequencies: "*The probability that a sentence belongs to a given language is taken as the product of the probabilities of each token.*"

What we have shown in this subsection is merely a small glimpse of the numerous ways to generate method descriptions in the literature. It is often time consuming to figure out what exactly the authors meant when writing the articles.

3. Language Identification

> *"In today's world of ever increasing written collections it requires a certain level of expertise to properly identify and file the material according to its language."*
>
> Morton David Rau, Language Identification by Statistical Analysis (1974)

3.1 Generative vs. Discriminative Language Identification

Rubinstein and Hastie [1997] divide classification methods into informative and discriminative classifiers. *Generative*[21] classifiers aim to model the underlying phenomenon and classification is done by calculating a probability for the observations using the model of each class. *Discriminative* classifiers do not try to model the phenomenon itself, but are instead modeling the class boundaries or the class probabilities directly. As examples of generative classifiers, Rubinstein and Hastie [1997] list Fisher Discriminant Analysis, Hidden Markov Models, and Naive Bayes and as examples of discriminative classifiers Logistic Regression, Neural Networks, and Generalized Additive Models.

Ng and Jordan [2002] use Logistic Regression as an example of a discriminative classifier and Naive Bayes as an example of a generative classifier. Empirically experimenting with the two methods, they show that if the amount of training material is large enough, the discriminative classifier usually attains better results. However, in many cases the generative classifier obtains better results when the amount of training data is small.

In Publication 2, we first published the basic version of the language identifier method that we now call HeLI. The basic idea of the method was already sketched out in my Master's thesis (Jauhiainen [2010]). In Publication 2, we refer to the method as token-based backoff, which is a descriptive name as the method relies on word-based tokenization of text. In general terms, the HeLI method belongs to the group of generative language identification methods. In the following Section 3.2, we give a synthesis of the descriptions of the HeLI method originally presented in Publications 2, 3, 4, and 6.

3.2 The HeLI Method

The basic idea of this method is that each word is given a score for each known language, and the text, whatever the length, is given the average of the scores of the words. For each word, the more specific language models are tried first, and if they cannot be applied, the method backs off to more general language models, *e.g.* from words to longer character n-grams and from longer character n-grams to shorter character n-grams. The models to be used are decided upon their performance on

[21] We follow Ng and Jordan [2002] and refer to informative classifiers as generative classifiers.

the development set. If only word-based models are used, the basic HeLI method is nearly equal to the product of the relative frequencies method used, for example, by Grefenstette [1995].

The variations of the method have included different ways of calculating the scores for the models, different preprocessing schemes (to lowercase or not, filtering non-alphabetic characters or not), and using different models (word n-grams could be used as well, though they have not helped in the experiments conducted so far). In the following paragraphs, we are reproducing the description of the HeLI method using the unified notation introduced in Section 2.5.

The goal is to correctly guess the language $g \in G$ in which the monolingual mystery text M has been written, when all languages in set G are known to the language identifier. In the method, each language $g \in G$ is represented by several different language models only one of which is used for every word t found in the mystery text M. The language models for each language are: a model based on words[22] and one or more models based on character n-grams from one to n_{max}. Each model used is selected by its applicability to the word t under scrutiny. The basic problem with word-based models is that it is not really possible to have a model with all possible words. When we encounter an unknown word in the mystery text M, we back off to using the n-grams of the size n_{max}. The problem with long n-grams is similar to the problem with words: if the n is high, there are too many possible character combinations to have reliable statistics for all even from a reasonably large training corpus. If we are unable to apply the n-grams of the size n_{max}, we back off to shorter n-grams. We continue backing off until character unigrams, if needed.

A development set is used for finding the best values for the parameters of the method. The three parameters are the maximum length of the used character n-grams (n_{max}), the maximum number of features to be included in the language models (cut-off c), and the penalty value for those languages where the features being used are absent (penalty p).[23] The penalty value has a smoothing effect in that it transfers some of the probability mass to unseen features in the language models.

The task is to select the most probable language g, given a mystery text M, as shown in Equation 9.

$$argmax_g P(g|M) \tag{9}$$

$P(g|M)$ can be calculated using Bayes' rule, as in Equation 10.

$$P(g|M) = \frac{P(M|g)P(g)}{P(M)} \tag{10}$$

[22] There can be several models for words, depending on the preprocessing scheme.

[23] In the DSL 2015 shared task, we used a version where each language group had their separate optimized penalty value.

In Equation 10, the a priori probability for the mystery text $P(M)$ is the same for all the languages $g \in G$ and can be omitted when calculating $argmax$. Also, we assume that all languages have equal a priori probability, so that $P(g)$ can be omitted as well, leaving us with Equation 11.

$$argmax_g P(g|M) = argmax_g P(M|g) \tag{11}$$

We approximate the probability $P(M|g)$ of the whole text through the probabilities of its words $P(t|g)$, which we assume to be independent as in Equation 12.

$$P(M|g) \approx P(t_1|g)P(t_2|g)...P(t_{l_M}|g) \tag{12}$$

We use the relative frequencies of words and character n-grams in the models for language g for estimating the probabilities $P(t|g)$.

The training data is preprocessed in different ways to produce different types of language models. The most usual way is to lowercase the text and tokenize it into words using non-alphabetic and non-ideographic characters as delimiters.[24] It is possible to generate several language models for words using different preprocessing schemes and then use the development material to determine which models and in which back-off order are usable for the current task.

The relative frequencies of the words are calculated. A space character is added to the beginning and the end of each word, even if it was not there originally.[25] Then the relative frequencies of character n-grams from 1 to n_{max} are calculated inside the words, so that the preceding and the following space-characters are included. The n-grams are overlapping, so that for example a word with three characters includes three character trigrams. Word n-grams are not used in this method, so all subsequent references to n-grams in this section refer to the n-grams of characters.

The c most common n-grams of each length and the c most common words in the corpus of a language are included in the language models for that language.[26] We estimate the probabilities using relative frequencies of the words and character n-grams in the language models, using only the relative frequencies of the retained tokens, as in Equation 13.

$$P(f|g) \approx \frac{c(C'_g, f)}{l_{C'_g}} \tag{13}$$

[24] The most notable exception being the various apostrophes considered to be parts of words in many written languages, and which should therefore be treated similarly to alphabetic characters.

[25] In the experiments of the 2015 shared task (Publication 2), we used a special character to mark the beginning and the end of sentences instead of the space character, as was done a year earlier by Goutte et al. [2014].

[26] We have experimented with separate cut-off values c for different n-gram lengths and words, but found that the compromise of having a shared value does not have a considerable effect on the performance either way. A shared value is more practical when using a development set to optimize the parameters.

The relative frequencies are then transformed into scores using 10-based logarithms.[27] The derived corpus containing only the word tokens retained in the language models is called C'. $dom(O(C'))$ is the set of all words found in any of the models of languages $g \in G$. For each word $t \in dom(O(C'))$, the values $v_{C'_g}(t)$ for each language g are calculated, as in Equation 14

$$v_{C'_g}(t) = \begin{cases} -\log_{10}\left(\frac{c(C'_g,t)}{l_{C'_g}}\right) & \text{, if } c(C'_g,t) > 0 \\ p & \text{, if } c(C'_g,t) = 0 \end{cases} \tag{14}$$

where $c(C'_g, t)$ is the number of words t and $l_{C'_g}$ is the total number of all words in language g. If $c(C'_g, t)$ is zero, then $v_{C'_g}(t)$ gets the penalty value p.

The derived corpus containing only the n-grams retained in the language models is called C'^n. The domain $dom(O(C'^n))$ is the set of all character n-grams of length n found in any of the models of languages $g \in G$. The values $v_{C'^n_g}(u)$ are calculated in the same way for all n-grams $u \in dom(O(C'^n))$ for each language g, as shown in Equation 15

$$v_{C'^n_g}(u) = \begin{cases} -\log_{10}\left(\frac{c(C'^n_g,u)}{l_{C'^n_g}}\right) & \text{, if } c(C'^n_g,u) > 0 \\ p & \text{, if } c(C'^n_g,u) = 0 \end{cases} \tag{15}$$

where $c(C'^n_g, u)$ is the number of n-grams u found in the derived corpus of the language g and $l_{C'^n_g}$ is the total number of the n-grams of length n in the derived corpus of language g. These values are used when scoring the words while identifying the language of a text.

When using n-grams, the word t is split into overlapping n-grams of characters u^n_i, where $i = 1, ..., l_t - n$, of the length n. Each of the n-grams u^n_i is then scored separately for each language g in the same way as the words.

If the n-gram u^n_i is found in $dom(O(C'^n_g))$, the values in the models are used.[28] If the n-gram u^n_i is not found in any of the models, it is simply discarded. We define the function $d_g(t, n)$ for counting n-grams in t found in a model in Equation 16.

$$d_g(t,n) = \sum_{i=1}^{l_t-n} \begin{cases} 1 & \text{, if } u^n_i \in dom(O(C'^n)) \\ 0 & \text{, otherwise} \end{cases} \tag{16}$$

[27] Using sum of logarithms instead of directly multiplying relative frequencies is a necessary algorithmic detail as current computers are unable to handle numbers with arbitrary precision.

[28] For the third submission of the DSL 2015 shared task, we used a special multiplier for the values of the character n-grams, which were found in only one of the languages within a language group. The multiplier was used for the Balkan group of languages and for the Spanish varieties in the submission, but it was tested with the other groups only after the shared task.

When all the n-grams of the size n in the word t have been processed, the word gets the value of the average of the scored n-grams u_i^n for each language, as in Equation 17

$$v_g(t,n) = \begin{cases} \frac{1}{d_g(t,n)} \sum_{i=1}^{l_t-n} v_{C_g'^n}(u_i^n) & \text{, if } d_g(t,n) > 0 \\ v_g(t,n-1) & \text{, otherwise} \end{cases} \tag{17}$$

where $d_g(t,n)$ is the number of n-grams u_i^n found in the domain $dom(O(C_g'^n))$. If all of the n-grams of the size n were discarded, $d_g(t,n) = 0$, the language identifier backs off to using n-grams of the size $n-1$. If no values are found even for unigrams, a word gets the penalty value p for every language, as in Equation 18.

$$v_g(t,0) = p \tag{18}$$

The mystery text is preprocessed in the same way as the training text to match the language model used. After this, a score $v_g(t)$ is calculated for each word t in the mystery text for each language g. If the word t is found in the set of words $dom(O(C_g'))$, the corresponding value $v_{C_g'}(t)$ for each language g is assigned as the score $v_g(t)$, as shown in Equation 19.

$$v_g(t) = \begin{cases} v_{C_g'}(t) & \text{, if } t \in dom(O(C_g')) \\ v_g(t, min(n_{max}, l_t+2)) & \text{, if } t \notin dom(O(C_g')) \end{cases} \tag{19}$$

If a word t is not found in the set of words $dom(O(C_g'))$ and the length of the word l_t is at least $n_{max} - 2$, the language identifier backs off to using character n-grams of the length n_{max}. In case the word t is shorter than $n_{max} - 2$ characters, $n = l_t + 2$. For creating the n-grams, a space character is added to the beginning and the end of each word, even if it was not there originally.[29]

The whole mystery text M gets the score $R_g(M)$ equal to the average of the scores of the words $v_g(t)$ for each language g, as in Equation 20

$$R_g(M) = \frac{\sum_{i=1}^{l_{T(M)}} v_g(t_i)}{l_{T(M)}} \tag{20}$$

where $T(M)$ is the sequence of words and $l_{T(M)}$ is the number of words in the mystery text M. Since we are using negative logarithms of probabilities, the language having the lowest score is returned as the language with the maximum probability for the mystery text.[30]

[29] In the DSL 2015 shared task we used a special character to mark the beginning and the end of sentences.

[30] In the second and third submissions for the DSL 2015 shared task, we gave a positive *ad-hoc* bonus for Bosnian, choosing it over Croatian in cases where the score difference was 0.01 or smaller.

3.3 Performance of the HeLI Method

In the second workshop in 2015, we obtained fourth place in the closed track of DSL shared task test set A. More information about the repertoire of the languages and the accuracies for individual languages can be found in Section 5.1 and especially Table 13 on page 44. In 2015, we designed and implemented several small modifications to the basic HeLI method in order to gain improvements in accuracy as mentioned in the previous Section. The results from the track are reproduced in Table 2. The measure used to rank the submissions in the shared task was the accuracy of the identifications. We participated in the shared task using the team name "*SUKI*". The HeLI method was overcome by Support Vector Machines (SVM) used by the teams *MAC*, *MMS*, and *NRC*. Less accurate results were provided by teams using Prediction by Partial Matching (PPM5C, team *Bobicev*), Logistic Regression (LR, team *MMS*), Naive Bayes (NB, team *MMS*), Maximum Entropy (ME, team *BRUniBP*), SVM (team *PRHLT*), Logistic Classifier (LG, team *PRHLT*), and Bayesian Net (team *NLEL*).

Method (*Team*)	**Features used**	**Accuracy**
SVM ensemble (*MAC*)[31]	ch. *n*-grams {2,4,6}, word *n*-grams 1-2	95.5
SVM ensemble (*MAC*)	ch. *n*-grams 1-6, word *n*-grams 1-2	95.4
SVM (*MAC*)	ch. *n*-grams 1-6, word *n*-grams 1-2	95.3
SVM (*MMS*)[32]	TF-IDF[33] ch. *n*-grams 2-7	95.2
SVM ensemble (*NRC*)[34]	ch. *n*-grams 2-6, word *n*-grams 1-2	95.2
SVM (*NRC*)	ch. *n*-grams 2-6, word *n*-grams 1-2	94.8
HeLI (*SUKI*)	**ch. *n*-grams 1-8, words**	**94.7**
PPMC5 (*Bobicev*)[35]	Markovian ch. *n*-grams 1-6	94.1
LR (*MMS*)	TF-IDF ch. *n*-grams 2-7	94.1
NB (*MMS*)	ch. 5-grams	94.1
ME (*BRUniBP*)[36]	(Markovian?) ch. *n*-grams 1-4, word *n*-grams 1-2	93.7
SVM (*PRHLT*)[37]	skip-gram word embeddings	92.7
LG (*PRHLT*)	sentence vectors	92.7
LG (*PRHLT*	skip-gram word embeddings	92.1
Bayesian Net (*NLEL*)[38]	ch. *n*-grams, words	85.6
? (*INRIA*)	?	83.9
Bayesian Net (*NLEL* with bug)	ch. *n*-grams, words	64.0

Table 2: The accuracies attained using different methods on the DSL 2015 test set A. The results attained using the HeLI method are bolded.

[31] Malmasi and Dras [2015b]

[32] Zampieri et al. [2015a]

[33] Product of term frequency and inverse document frequency.

[34] Goutte and Léger [2015]

[35] Bobicev [2015]

[36] Ács et al. [2015]

[37] Franco-Salvador et al. [2015]

[38] Fabra-Boluda et al. [2015]

In the DSL 2016 shared task, our language identification system reached the 2nd position[39] in both the closed and open submissions without any modifications to the basic HeLI method. We also published an open source implementation of the program implementing the method.[40] The results from the track are reproduced in Table 3.

In 2016, many teams were interested in experimenting with the use of the various Neural Network-based deep learning methods that had become very efficient in other classification tasks. No results using Neural Networks had been submitted in 2015. The *tubasfs* team set out to evaluate the use of deep Neural Networks, but ended up winning the shared task using SVMs instead (Çöltekin and Rama [2016]) and did not submit any results with the Neural Networks as their performance on the development set was too low. Team *GW_LT3* submitted results using NNs.[41] Teams *mitsls* and *Uppsala* used Convolutional Neural Networks (CNN). Team *andre/clac* evaluated several NN variants and ended up submitting results using a CNN with a Bidirectional Long Short Term Memory (LSTM). Team *ResIdent* used Deep Residual Networks (ResNet).

Team *UPV_UA* used Kernel Discriminant Analysis (KDA) with string kernels. Team *PITEOG* used a Chunk-Based Language Model (CBLM), which is similar to PPM, as well as an implementation of the Expectation-Maximization (EM) algorithm for words. Team *XAC* used Gradient Boosting (GB) and Random Forests (RF) in their submitted runs, but achieved significantly better results using an NB classifier with the TF-IDF weighting on the same dataset after the shared task (0.902 accuracy). Team *Citius_Ixa_Imaxin* experimented with their dictionary-based language identifier, *Quelingua*, using the sum of Inverse Ranking (IR) with words.

As can be seen in Tables 2 and 3, the HeLI method was ranked as the best performing generative method in both. It was overshadowed only by the much more discriminating SVMs, performing better than many other discriminative methods.

[39] We were ranked shared first place as the results were not statistically different according to the organizers (Malmasi et al. [2016]).

[40] `https://github.com/tosaja/HeLI`

[41] Perhaps traditional Multilayer Perceptron (MLP).

[42] Çöltekin and Rama [2016]

[43] Zirikly et al. [2016]

[44] Goutte and Léger [2016]

[45] Herman et al. [2016]

[46] Cianflone and Kosseim [2016]

[47] Barbaresi [2016]

[48] Adouane et al. [2016]

[49] McNamee [2016]

[50] Ciobanu et al. [2016]

[51] Gamallo et al. [2016]

[52] Bjerva [2016]

[53] Franco-Penya and Sanchez [2016]

[54] Belinkov and Glass [2016]

Method (*Team*)	Features used	Accuracy
SVM (*tubasfs*)[42]	ch. n-grams 1-7	0.894
HeLI (*SUKI*)	**ch. n-grams 1-6, words**	**0.888**
LR (*GW_LT3*)[43]	ch. n-grams 2-6, word n-grams 1-3	0.887
SVM (*nrc*)[44]	ch. 6-grams	0.886
KDA (*UPV_UA*)	strings	0.883
CBLM (*PITEOG*)[45]	chunks	0.883
NB (*andre/clac*)[46]	ch. 7-grams	0.885
NB (*andre/clac*)	ch. 8-grams	0.883
GB (*XAC*)[47]	stats., morph. crit., ch 5-grams, word n-grams 2-4	0.879
SVM (*ASIREM*)[48]	ch. 4-grams	0.878
PPM (*hltcoe*)[49]	max. 5th order Markovian ch. n-grams	0.877
SVM (*ASIREM*)	words	0.872
RF (*XAC*)	stats., morph. crit., ch 5-grams, word n-grams 2-4	0.870
EM (*PITEOG*)	words	0.866
LR (*UniBucNLP*)[50]	TF-IDF word n-grams 1-2	0.865
SVM (*HDSL*)	ch. n-grams, word n-grams	0.853
NB (*Citius_Ixa_Imaxin*)[51]	words	0.853
NN (*GW_LT3*)	ch. n-grams 2-6	0.850
ResNet (*ResIdent*)[52]	byte embeddings	0.849
NB (*eire*)[53]	word 2-grams	0.838
CNN (*mitsls*)[54]	characters	0.831
CNN (*Uppsala*)	words	0.825
CNN-LSTM (*andre/clac*)	characters	0.785
Sum of IR (*Citius_Ixa_Imaxin*)	words	0.776
SVM (*eire*)	words	0.585

Table 3: The accuracies attained using different method and feature combinations on the DSL 2016 test set A, closed track. The results attained using the HeLI method are bolded.

3.4 Modified Versions of the Method

We have experimented with different ways of calculating the values $v_{C'_g}$ for the features f. In this Section, we take a closer look at those techniques we have used in the shared tasks: Additive Smoothing, TF-IDF, and non-linear mappings.

Additive Smoothing Additive Smoothing is one of the most commonly used methods for smoothing. It is also referred to as Laplace or Lidstone smoothing and has been used in language identification by, for example, Dunning [1994], Adams and Resnik [1997], Vatanen et al. [2010], and Franco-Penya and Sanchez [2016]. In our HeLI method, we have used the penalty value p to perform a sort of Additive Smoothing. Before Publication 2, we experimented with using Lidstone smoothing in the models instead of the penalty values, but it produced somewhat poorer results. The value $v_{C'_g}(f)$ using Lidstone smoothing is calculated as in

$$v_{C_g'}(f) = \frac{c(C_g', f) + \lambda}{l_{C_g'} + |U(C_g')|\lambda} \tag{21}$$

where λ is a smoothing parameter usually set between 0 and 1.[55]

TF-IDF For the 4th edition of the DSL shared task, we were interested in evaluating the use of the TF-IDF weighting scheme to calculate the value $v_{C_g'}(f)$. We were inspired by the successful use of the TF-IDF weighting by Barbaresi [2016] a year earlier. He was able to significantly boost the accuracy of his language identifier after the 3rd edition of the DSL shared task using the TF-IDF in calculating the probabilities for the languages. Adouane [2016] also found that the TF-IDF weighted n-grams worked better with Support Vector Machines (SVM) than simple frequencies when discriminating between Arabic dialects. We calculated the TF-IDF as in Equation 22

$$v_{C_g'}(f) = c(C_g', f) log \frac{l_G}{df(C_G', f)} \tag{22}$$

where $df()$ is defined as in Equation 23. Let l_G be the number of languages in a language-segmented corpus C_G'. We define the number of languages in which a feature f appears as the document frequency df of f as

$$df(C_G', f) = \sum_{g=1}^{l_G} \begin{cases} 1 & \text{, if } c(C_g', f) > 0 \\ 0 & \text{, otherwise} \end{cases} \tag{23}$$

We used the $v_{C_g'}(f)$ values from Equation 22 instead of relative frequencies in Equations 14 and 15, but we were unable to come even close to the accuracy of our original method.

Non-Linear Mappings Brown [2014] experimented with five different language identification methods, modifying them to use two non-linear mappings: the Gamma and the Loglike functions. We experimented with applying these two non-linear mappings to the relative frequencies used by the HeLI method. In addition to the mappings, we still used the penalty value p for smoothing. Both functions have a variable, *gamma* or *tau*, which is decided using the development set.

When Brown [2014] used the gamma function in his experiments, he was able to reduce the errors made by his own language identifier by 83.9% with 1,366 languages and by 76.7% with 781 languages.[56] The value $v_{C_g'}(f)$, which replaces the relative frequency, using the Gamma function is calculated as in Equation 24.

[55] This equation would replace the Equations 14 and 15.

[56] The error percentages were reduced from 13.309% to 2.136% and from 11.879% to 2.770%, respectively.

$$v_{C'_g}(f) = \Big(\frac{c(C'_g, f)}{l_{C'_g}}\Big)^{\gamma} \tag{24}$$

The values calculated using Equation 24 are equal to the original relative frequencies when γ equals 1. We experimented with the Gamma function using the development set of DSL 2017 shared task. It would seem that the penalty value p and the γ variable have at least partly the same effect. If we fix one of the values, we are able to reach almost or exactly the same results by varying the other as can be seen in Table 4, where p is optimized for different γ values to produce the best results. As no combination reduced the error rate at all on the development set, we did not submit identifications to the shared task using the Gamma function.

Recall	**Gamma γ**	**Penalty p**
0.9105	*0.5*	*3.3*
0.9102	0.7	4.6
0.9103	0.8	5.3
0.9105	**1.0**	**6.6**
0.9104	1.2	7.9
0.9104	1.3	8.6
0.9105	*1.5*	*9.9*
0.9104	1.7	11.2

Table 4: Table showing how different combinations of the penalty p and Gamma γ give almost the same recall on the DSL 2017 development set when γ is fixed and the penalty p re-optimized. The bolded result is equal to the unmodified HeLI method.

With the Loglike function, Brown [2014] was able to reduce the error rate of his own language identifier by 83.8% with 1,366 languages and 76.7% with 781 languages.[57] The value $v_{C'_g}(f)$ using the Loglike function is calculated as in Equation 25.

$$v_{C'_g}(f) = \frac{\log(1 + 10^{\tau}\frac{c(C'_g,f)}{l_{C'_g}})}{\log(1 + 10^{\tau})} \tag{25}$$

Using the Loglike function, we managed to achieve a tiny recall improvement (from the original 91.05% to 91.09%) with the DSL 2017 development set. Even though the improvement was far from being statistically significant, we did submit identification results from the test set to the shared task using the Loglike function. On the test set, the improvement was less tiny when compared with the original HeLI method.

[57] The error percentages were reduced from 13.309% to 2.146% and from 11.879% to 2.772%, respectively.

The accuracy on the test set rose from 90.54% attained by the original method to 90.99% using the Loglike function.[58] The official results from the DSL 2017 closed track are reproduced in Table 5. The measure used to rank the submissions in the shared task was the weighted F-score.

Method (*Team*)	**Features used**	**F**
SVM (*CECL*)[59]	ch. & POS tag *n*-grams, global statistics	0.927
SVM (*mm_lct*)[60]	TF-IDF ch. *n*-grams 1-6, word *n*-grams 1-2	0.925
NB (*XAC_Bayesline*)[61]	TF-IDF ch. *n*-grams	0.925
SVM (*tubasfs*)[62]	TF-IDF ch. *n*-grams 1-7, word *n*-grams 1-3	0.925
LR (*gauge*)	ch. *n*-grams 1-6	0.916
SVM + NB (*cic_ualg*)[63]	ch. *n*-grams 3-6, words	0.915
HeLI with Loglike (*SUKI*)	**ch. *n*-grams 1-7, words**	**0.910**
NB + CNN + MLP (*timeflow*)[64]	ch. *n*-grams, word embeddings	0.907
NB (*cic_ualg*)	ch. *n*-grams 3-6, words	0.907
HeLI (*SUKI*)	**ch. *n*-grams 1-8, words**	**0.905**
NB + MLP (*timeflow*)	ch. *n*-grams, word embeddings	0.903
Perplexity, voting (*Citius_Ixa_Imaxin*)[65]	ch. 5-7 and word 1-3 *n*-grams	0.902
Perplexity (*Citius_Ixa_Imaxin*)	words	0.901
CBOW NN (*mm_lct*)	ch. *n*-gram 1-5 embeddings	0.900
NB (*bayesline*)	ch. 4-*n*-grams (primarily?)	0.889
NB + CNN (*timeflow*)	ch. *n*-grams, word embeddings	0.887
Perplexity (*Citius_Ixa_Imaxin*)	ch. 7 *n*-grams	0.879
LSTM NN (*deepCybErNet*)		0.202

Table 5: The weighted F-scores attained using different methods on the DSL 2017 test set. The results attained using the original HeLI method and the HeLI method with Loglike function are bolded.

3.5 To Discriminate or Not

In generative probabilistic modeling, it is possible to calculate the probability of a given text using the language models irrespective of the other languages being considered. Adding new languages to generative systems only requires modeling the languages to be added, but in a system using a discriminative classifier, all the languages have to be re-trained. With discriminative models, new borders for the remaining languages also have to be learned when removing languages from the repertoire.

[58] The test set was balanced, so the recall and accuracy were equal. The error reduction comparable with those calculated by Brown [2014] was *c.* 5%.

[59] Bestgen [2017]

[60] Medvedeva et al. [2017]

[61] Barbaresi [2017]

[62] Çöltekin and Rama [2017]

[63] Gómez-Adorno et al. [2017]

[64] Criscuolo and Aluísio [2017]

[65] Gamallo et al. [2017]

Some of the commonly used techniques, such as setting the parameters using a development corpus, can move otherwise generative classifiers towards the discriminative side. This is true for the basic HeLI method as well. As the penalty value p is determined using a development corpus, it is optimized to discriminate between the languages in that corpus. The language models used with the language identifier are also determined by the results obtained with the development corpus and therefore the process is in this sense discriminative. If we leave out, or add, languages after the optimization on the development corpus has been conducted, we might not be using the optimal selection of language models or the optimal penalty value anymore. In the DSL 2015 shared task, we obtained the best accuracy when we introduced an additional discriminative multiplier for character n-grams found in only one language. It is possible to modify all generative language identification methods to use more or less complicated discriminative elements. Whether it improves the results or not depends on the task at hand as shown by Ng and Jordan [2002].

4. The Data

"If you put into the machine wrong figures, will the right answers come out?"

Question asked Charles Babbage regarding the Difference Engine (ante 1864)

4.1 Low Corpora Quality

The META-SHARE[66] network of language data repositories (Piperidis [2012]) lists 929[67] language resources tagged with "text" and "corpus." The annotation level of the corpora varies with the original intended use of the said corpora. Annotating corpora manually is labor-intensive and thus time consuming and expensive (Tomanek et al. [2007]). There is an understandable trend to try to avoid the manual phase by using automated annotation. Automated language identification is an important part of the preprocessing pipeline in automated corpus creation (Quasthoff et al. [2006]). During our research, we have noticed that many of the corpora used for Natural Language Processing (NLP) research are somewhat inaccurate in language annotation, which creates problems when very high accuracies are reached. For example, some of the sentences in the DSL 2016 test set were incorrectly identified even over language group borders. These "sentences" are listed in Table 6. There are very few lines that could be considered to belong to the intended language by any means. Furthermore, there is only one real sentence among the "sentences" in Table 6. That sentence was annotated to be Spanish and the HeLI method identified it as Portuguese. According to Google Translate it is actually Galician, which is probably true as the web service is able to translate it into perfect English and it mentions the Galician government.[68] Galician is a less used language spoken around the border of Spain and Portugal, closely related to both, but more similar to Portuguese (Simons and Fennig [2018]). It is natural that a language identifier which has not been trained on the Galician language thinks that the sentence might be either Spanish or Portuguese.

The quality or accuracy of a corpus used determines the upper limit of the accuracy that can be reached when machine learning methods are trained or tested using it. For example, if every 100th sentence is tagged with a wrong language in a test corpus, it is not very useful when trying to reach percentages above or near 99%. When we were constructing the out-of-domain evaluation setting for Publication 5, we decided to use relatively short texts, averaging 15,000 characters, from as reliable a source as possible. We used short texts in order to be able to control the quality of the test set better. We inspected all of them manually, but of course we were not experts in all of the 285 languages, so the cleaning was most likely not perfect.

[66] http://metashare.ilsp.gr

[67] As of April 2019.

[68] Google Translate cannot translate the same sentence into perfect English using the Spanish language model. The Portuguese model otherwise does well, but the "Estratexia Xuventude" named entity is not translated.

Sentence	**HeLI**	**DSLCC**
Copyright: Project Syndicate/Institute for Human Sciences, 2011. www.project-syndicate.org	id	sr
Slask - Hanover 3:5 (1:3)	id	sr
6. Simon and Garfunkel – ”Mrs. Robinson”	id	sr
10. Xavier Florencio (Bouygues) m.t.	fr-FR	es-ES
4. Alexander Noren (SUE) 276 (69+70+68+69)	fr-FR	es-ES
6. J.-M. Latvala (Ford Focus) a 1.06,4	fr-FR	es-ES
. Vaughn Taylor (USA) 269 (67-65-67-70)	pt-BR	fr-FR
BRÉSIL : J. Cesar - Maicon, Lucio, Juan, M. Bastos - Elano, G. Silva, F. Melo - Kaka - Robinho, L. Fabiano.	pt-BR	fr-FR
Zovko - Bogomolov Jr. 4-6, 2-6	pt-PT	hr
Este é, como sinalou, un dos obxectivos da Estratexia Xuventude 2013 que promove o Goberno galego e que xorde da confianza nas ≪moitas posibilidades≫ que ofrecen os novos artistas da comunidade.	pt-PT	es-ES
O B R A Z L O Ž E NJ E	pt-BR	sr
.Chris Di Marco (USA) 208 (71-70-67)	id	fr-FR
Espanyol - Cordoba 4-2 (Vazquez 9, 20, 88 Vila 35 / Aguilar 40, Diaz 49), 5-4	es-ES	hr
9. (9) Juan Martin Del Potro (Arg) 3180	es-AR	bs
10. Michael Schumacher 46 8 1 0 1 12 0 12 0 0 2 2 0 6 2 0	bs	es-ES

Table 6: “Sentences” which were “incorrectly” identified by our language identifier over language group borders in DSL 2016 shared task.

As can be seen in Table 6, using automated language annotation is likely to introduce annotation errors, which would still be easy for a human to detect. Automatic corpora creation for language identification development creates a need for automatic annotation error detection, which has been done in NLP for other areas (Dickinson [2015]). Tufis and Irimia [2006] suggest that the corpora could be re-tagged with a language model learned from it.[69] Following this example, the language identifier itself could be used to clean the corpora it has been trained on. If the corpora quality is very bad and the accuracy of the language identifier therefore low, then the cleaning might turn out to make the situation worse. In both cases, though, the language identifier could be used to mark those parts of the corpora that might need some checking by a human annotator. It is a clear sign of a problem if the language identifier identifies part of its own training corpus to be in another language than it is annotated with. If the identification accuracy is otherwise good, it is a strong hint that the identified part is incorrectly annotated. With generative language identification methods, the probability given by the method could be used to detect annotation errors. If the probability of a sentence to be in the annotated language is very low, then it can be suspected to be somehow unusual, possibly partly or completely written in a different language. However, if we are dealing with very similar languages

[69] Tufis and Irimia [2006] call this biased tagging.

or dialects, then sometimes even longer sentences can be truly ambiguous making automated annotation error detection very difficult.

For the DSL 2016 open track,[70] we collected additional linguistics resources from the Common Crawl[71] corpus. We used all the web pages from the domains corresponding to the language variants on the track. The repertoire of the languages and varieties included in the DSL 2016 shared task are listed in Table 13 on page 44. Several *ad-hoc* techniques were used in subsequent steps which are listed in Table 7. Similar techniques have been used, for example, in pre-processing the text destined for the Leipzig Corpora Collection (Quasthoff et al. [2006]). After each step, we evaluated the accuracy of language identification on the DSL development set. In the end, we arrived at the accuracy of 85.56%, which was slightly higher than the 85.09% obtained using only the DSL training set. Our results on both the closed and the open tracks can be seen in Table 8.[72]

Technique	**Accuracy**
Original accuracy before improvements	49.86%
Minimum line length (25 characters)	51.08%
Lines must include one of the top 5 characters and one of the top 5 words (of at least 2 characters) of the language (in the DSL training data)	62.42%
Lines must start and end with characters that start and end sentences in the DSL training data & No duplicate lines & May not include characters not in the DSL training data for any language	68.34%
External language identifier must identify Canadian or French lines as French	69.19%
Language identifier trained on the DSL training data must identify lines with corresponding languages	74.66%
HeLI parameter re-optimization	80.93%
Segmentation into sentences with minimum sentence length of 25 characters & Re-identification with language identifier using DSL training data	83.15%
No duplicate sentences & HeLI parameter re-optimization	84.90%
Add DSL training data & HeLI parameter re-optimization	85.56%

Table 7: *Ad-hoc* techniques used to improve the suitability of the corpus to be used as training material for a language identifier in DSL 2016 shared task open track.

[70] On the open tracks, the participants were allowed to use any material that they had at their disposal.

[71] http://commoncrawl.org/

[72] The results from the closed track for all the participating teams are displayed in Table 3 on page 27.

4.2 Small Amount of Training Material

Many articles describing language identification experiments include a table called a "learning curve", showing how the identification accuracy improves when the amount of training data increases (*E.g.* Bergsma et al. [2012], Brown [2012], Ljubešić and Kranjcić [2014], Goutte et al. [2016], and Malmasi and Zampieri [2016].). Machine learning techniques seem to be "data hungry" and the increase in the amount of training data usually leads to increased classification accuracy (Obermeyer and Emanuel [2016]). There are many exceptions to this rule of thumb, and the effect of adding data depends on the classification method used as well as the quality and domain compatibility of the data to be added (Schohn and Cohn [2000]). Also, it is not clear that adding more training data treats the respective languages equally. Ljubešić et al. [2007] present a chart where the precision and recall of identifying Croatian documents within a bilingual Croatian-Serbian corpus is given as relative to the Croatian training set size. The harmonic mean of precision and recall clearly begins to get lower when the training set gets larger than 400,000 characters. They fail to mention, however, what happens to the combined accuracy of the two languages.

The open tracks of the DSL shared tasks gave the participants the possibility to use any other text sources at their disposal to train the language models for their language identification methods. We gathered the accuracy information from DSL shared task reports from 2014 to 2016 and the gains from using external corpora for language identification can be seen in Table 8. When comparing the differences in results with and without the use of external corpora, it is clear that it is not self-evident that more training material always means better accuracies.

As can be seen in Table 8, in two out of three instances we were able to improve the results by using external corpora in addition to the one provided by the organizers. In those two cases, our results on the closed track were very low and thus easily improvable. For the DSL 2016 A-set, our results were slightly worse for the open set than for the closed one.

As mentioned earlier, for the 2016 DSL open track, we collected additional data for the training of the language identifier increasing the amount of training data by *c.* 8,700% (Publication 3). Table 9 lists the languages, sizes of their training data, as well as the results obtained on the open track. If we examine the Malay-Indonesian pair, we can see that adding more data decreases the recall of Indonesian by 0.4%, but at the same time the recall for Malay goes up by the same percentage. As can be seen in Table 9, the additional training data did not have a clear positive effect in any of the language groups concerned. With the French pair, the recall went down for both varieties, but in every other group the recall for some of the varieties increased and decreased for others.

Team	Test set	Closed accuracy	Open accuracy	Difference
NLEL	A 2015	64.0%	91.8%	27.8%
SUKI	**B2 2016**	**64.2%**	**79.6%**	**14.6%**
SUKI	**B1 2016**	**68.8%**	**82.2%**	**13.4%**
NRC	B1 2016	91.4%	94.8%	3.4%
NRC	B2 2016	87.8%	90.0%	2.2%
Citius	A 2016	85.3%	87.1%	1.8%
Citius	B2 2016	68.6%	69.2%	0.6%
NRC	A 2015	95.2%	95.7%	0.5%
NRC	A 2016	88.6%	89.0%	0.4%
NRC	B 2015	93.0%	93.4%	0.4%
PITEOG	B1 2016	80.0%	80.0%	0.0%
SUKI	**A 2016**	**88.8%**	**88.4%**	**-0.4%**
PITEOG	B2 2016	76.0%	72.8%	-3.2%
UniMelb	2014	91.8%	88.0%	-3.8%
Citius	B1 2016	70.8%	66.4%	-4.4%
UMich	2014	93.2%	85.9%	-7.3%

Table 8: Differences in accuracy with (open track) and without (closed track) the use of external corpora for different teams in DSL test sets from 2014 to 2016. Our results are bolded.

Language	Original	Final	Increase	Rec. closed	Rec. open
Bosnian	620,000	6,100,000	900%	78.1%	74.1%
Croatian	740,000	10,500,000	1,300%	84.6%	85.9%
Serbian	690,000	13,300,000	1,800%	91.4%	91.7%
Malay	510,000	8,600,000	1,600%	99.2%	99.6%
Indonesian	670,000	35,800,000	4,200%	99.4%	99.0%
Portuguese (Br)	790,000	265,300,000	33,500%	94.0%	91.8%
Portuguese (Pt)	720,000	14,100,000	1,900%	94.3%	95.7%
Spanish (Ar)	830,000	28,300,000	3,300%	85.9%	82.9%
Spanish (Mx)	620,000	51,700,000	8,200%	83.2%	91.5%
Spanish (Sp)	900,000	47,100,000	5,100%	70.5%	71.2%
French (Fr)	700,000	241,500,000	34,400%	93.3%	87.3%
French (Ca)	570,000	14,200,000	2,400%	91.6%	89.7%
All	*8,350,000*	*736,500,000*	*8,700%*	*88.8%*	*88.4%*

Table 9: The languages and varieties used in the sub-task 1 of the DSL 2016 shared task. The sizes, in tokens, of the original provided corpus and the final corpus including the material from the Common Crawl. The recalls obtained in the closed and in the open tracks are shown in the two last columns.

Currently, the ISO 639-3 standard includes language codes for 7,858 languages.[73] *Ethnologue* lists 7,097 living spoken languages, of which around a third are considered endangered (Simons and Fennig [2018]). As of this writing, the Crubadan project[74] has entries for 2,228 languages (Scannell [2007]), which still leaves almost 5,000 languages unaccounted for. Brown [2014] was able to collect text data for 1,366 languages, utilizing mostly Bible translations, Wikipedia, and the Europarl corpus.

Publication 6 describes the early phases of the Finno-Ugric Languages and the Internet project which was active from 2013 to 2019. In the project, our aim was to find web pages with textual content written in the less used Uralic languages on the Internet (Publication 6). Using these texts, we wanted to create language-specific text corpora for these languages. The ISO 639-3 has separate language codes for 40 languages belonging to the Uralic group, of which three, Hungarian, Finnish, and Estonian are majority languages in their respective countries. We have been searching for texts written in the remaining 37 under-resourced languages. So far, we have been able to find texts in all but one and the links to these texts can be found from our Wanca service.[75] From those 36 languages, 11[76] do not currently have Crubadan entries.

One of the questions we had to tackle was how much training data we need in order to identify new text in a context where little training data is available for some languages (like the rare Uralic languages) and huge amounts for others (like English or German). In the beginning of the project, we were able to collect training material manually for 34 rare Uralic languages, but for a few of those languages the amount of material was very low. For example, for the Kemi Sami language, we had only one text source (Zorgdrager [2017]).[77] We thought it necessary to include it in our language repertoire, as there is always a tiny possibility that somebody, somewhere, digitizes a previously unknown text in that language. The results of the evaluations of Publication 5 indicate that at least the HeLI method is not seriously affected by the size differences between the training data of different languages. In the evaluations, the HeLI method was able to reach over 90% recall and precision for all the 285 languages[78] at 65 characters.

4.3 Out-of-Domain Texts

The concept of "domain" is widely used in language identification and related literature. Wees et al. [2015] note that even in the field of domain adaptation, the concept

[73] `https://iso639-3.sil.org/code_tables/639/data` (as of 15.10.2018).

[74] `http://crubadan.org`

[75] `http://wanca.fi/wanca/`

[76] Kamas, Kemi Saami, Livvi-Karelian, Ludian, Mator, Nganasan, Pite Saami, Selkup, Ter Saami, Tundra Enets, and Ume Saami.

[77] `https://en.wikipedia.org/wiki/Kemi_Sami_language`

[78] The list of the languages and the manually discovered sources for their training, development and test material are listed on the web page: `http://suki.ling.helsinki.fi/LILanguages.html`.

is not unambiguously defined and that interpretations commonly neglect the fact that *topic* and *genre* are different properties of text. In this work, we have defined a domain to be a property of any given text, combining the topic(s) and the genre(s) of the said text. In addition, it can also include information about other properties that make a text similar or dissimilar from other texts, such as the possible idiolect(s) or even dialect(s) used in the text.

Time and again in the language identification literature, the training data is said to be either in-domain or out-of-domain when compared with the test data (*e.g.* Ljubešić and Toral [2014], Kocmi and Bojar [2017], Li et al. [2018], and Zampieri et al. [2018]). However, we have observed that there are widely varying degrees of domain difference. The degree of domain difference between the training and the test data can be either planned or unplanned and it is set when the dataset is generated. For example, if the training data consist of texts in a completely different topic than the test data, the degree of domain difference is probably greater than when the texts are from the same topic. In addition, the text could be from the same journal or written by the same authors, which would increase the "in-domainness" factor. In an extreme in-domain case, a single text can be divided between the training and the test sets.

Classifiers can be more or less sensitive to the domain differences between the training and the testing data depending on the machine learning methods used (Blodgett et al. [2017]). Lui and Baldwin [2011] discuss the effect of domain in language identification. They experiment with different source combinations for the training and the test sets, showing that usually an in-domain (from the same source) training clearly produces better results.

If the training and the development data are extracted from the same source or domain, it will not be clear to what extent the language identifier learns the domain in question instead of the language. Li et al. [2018] point out that most real-world situations require the use of (domain) heterogenous corpora for learning.

There are ways to pre-process text data in order to make it more general in domain. In addition to the standard test set A, the 2015 version of the DSL shared task included a second test set, the test set B, where all the named entities in the running text had been transformed to the same "#NE"-tag. "Blinding" or anonymizing the named entities in this way was an attempt to avoid the topic bias in classification and to examine the influence which proper names have on language identification (Zampieri et al. [2015b]). When we were preparing our system for the test set B of the DSL 2015 shared task, we did not try to blind the named entities from the training and development sets[79] to re-train the language identifier. Instead, we merely removed the named entity tags from test set B and used the same methods as with test set A. Our submission obtained the second place on the track where the named entities were blinded. We collected the group average accuracies from those tests in Table 10, where the effect of blinding the named entities can be seen. The differences would

[79] The training and the developments sets were the same for both test sets.

seem to indicate that the named entities clearly have an effect when distinguishing between the European and South American variants of Spanish or Portuguese, as the average accuracies are much higher with the named entities (test set A) than without them (test set B). The notable thing is that the difference is not so big for the Balkan group of languages (Bosnian, Croatian, and Slovak). One possible explanation for this is the physical distance between the users of the languages and dialects. The Balkan group of languages is used within a few hundred kilometers and all the countries have a common border with each other. It is natural that they discuss more of the same named entities than those residing in different continents.

Language/variety group	Test set A	Test set B	Difference	Error increase
Bosnian, Croatian, Slovak	87.7%	86.3%	-1.4%	+11.4%
Indonesian, Malaysian	99.7%	99.3%	-0.4%	+133.3%
Czech, Slovak	99.8%	99.9%	+0.1%	-50.0%
Portuguese (Brazilian/European)	92.4%	88.3%	-4.1%	+53.9%
Spanish (Argentine/Castilian)	90.4%	86.1%	-4.3%	+44.8%
Macedonian, Bulgarian	99.9%	99.9%	0%	0%
Unseen languages	98.2%	96.5%	-1.7%	+94.4%

Table 10: Accuracies with (test set A) and without the named entities (test set B), and their differences.

The third iteration of the DSL shared task in 2016 included test sets B1 and B2, which were defined to be out-of-domain social media data. They proved to be tweets by single users and the language was defined as the language most used by the tweeter. The tweets, however, included a great deal of non-lingual material and text in languages[80] not present in the training material. Together with the formatting differences of the tweets, the presence of unknown languages made the language identification of the test sets a completely different kind of task, not merely an out-of-domain task.

[80] At least English, but possibly other languages as well.

5. The Hard Contexts

> *"A language is a dialect with an army and navy"*
>
> –anonymous Bronx high school teacher (1943 or 1944)
> remark after a lecture given by Max Weinreich

5.1 Close Languages, Dialects, and Language Variants

According to Haugen [1966], there are two dimensions to the use of the words "language" and "dialect." The first one is the *structural* dimension, where linguists consider the genetic relationship between languages and dialects. The second one is the *functional* dimension, where sociologists emphasize how, where, and by whom the languages and dialects are used. These two dimensions are intertwined in general use, and there are no universally used definitions concerning the difference between a language and a dialect, at least none that could be measured from the texts themselves. In language identification, we generally measure structural differences between languages and dialects, but we also employ functional differences if the given division of languages is more function-based. Functional differences can be measured, for example, by the use of different content words, *e.g.* names of people and places. In our work, we have generally taken the definitions of languages from the ISO 639-3 standard[81] and used the three letter language codes for the languages issued by the standard. Using the language divisions of the ISO 639-3 standard can sometimes be frustrating as they can be based on either structural or functional considerations.

Automatic language identification can be considered easy if the languages are not closely related (Kosmajac and Keselj [2018]). Generally, the more similar the languages are, the more difficult the task of language identification becomes. The first to experiment with language identification for structurally very similar languages were Sibun and Reynar [1996], who had Croatian, Serbian, and Slovak as part of their language repertoire. Another good example of a pair of close languages is the Bokmål variation of Norwegian when compared with Danish. In his experiments, Prager [1999] noticed that the two Norwegian dialects Bokmål and Nynorsk are further away from each other than Bokmål is from Danish. Table 11 gives as examples some translations of the first sentence of the first article of the Universal Declaration of Human Rights. It gives some indication of how difficult it is to distinguish between close languages and how languages create "dialect" continuums. Some differences between translations are partly due to the translators choice of words. When the Finnish translation is compared to any of the Germanic translations, no common vocabulary is to be found. The most difficult to distinguish are usually those language variants used in different countries that are not normally considered to be even separate dialects in either a structural or functional sense, such as European and Brazilian varieties of Portuguese (Zampieri and Gebre [2012]).

[81] https://iso639-3.sil.org

Alla människor äro födda fria och lika i värde och rättigheter.	Swedish
Alle menneske er fødde til fridom og med same menneskeverd og menneskerettar.	Norwegian, Nynorsk
Alle mennesker er født frie og med samme menneskeverd og menneskerettigheter.	Norwegian, Bokmål
Alle mennesker er født frie og lige i værdighed og rettigheder.	Danish
Alle minsken wurde frij en gelyk yn weardigens en rjochten berne.	Frisian
Alle menslike wesens word vry, met gelyke waardigheid en regte, gebore.	Afrikaans
Alle Menschen sind frei und gleich an Würde und Rechten geboren.	German
All de Minschen sünd frie un gliek an Wüürd un Rechten baren.	Saxon, Low
All Mënsch kënnt fräi a mat deer selwechter Dignitéit an dene selwechte Rechter op d'Welt.	Luxembourgish
All human beings are born free and equal in dignity and rights.	English
Aw human sowels is born free and equal in dignity and richts.	Scots
Kaikki ihmiset syntyvät vapaina ja tasavertaisina arvoltaan ja oikeuksiltaan.	Finnish

Table 11: The beginning of the first article of the Universal Declaration of Human Rights in several Germanic languages and in Finnish.

The similarity of languages, as expressed in the texts themselves, could perhaps be used to define whether languages are different or not. Unsupervised clustering could be used to divide the texts into separate groups and then the methods for language identification could be used to decide into how many groups, or languages, the texts could be divided. On the one hand, the groups which can be distinguished with over 99% average accuracy in a sentence (see Table 13), could be considered different languages (Indonesian-Malaysian, Czech-Slovak, and Macedonian-Bulgarian). On the other hand, the average accuracy when discriminating between Croatian, Bosnian, and Serbian is very similar to the accuracy between Spanish variants.

During the Finno-Ugric Languages and the Internet project, we did not harvest texts written in the three most used Uralic languages: Hungarian, Finnish, and Estonian, as several corpora for these languages already exist (Publication 6). However, we could not ignore these three languages, as their presence on the Internet makes finding some of the less represented ones especially difficult. Some of the rare Uralic languages are very similar to widely used languages, for example Tornedalen Finnish[82] to Finnish (Publication 6). Also, some of them are very close to each other, for example the pair of Finnic languages spoken in north-western Russia: Ludic and Livvi-Karelian. In order to create language specific text corpora, we needed to find a way to distinguish between these languages. The wish to be able to distinguish between very close languages is in part the reason for our interest in dialect and language variety identification.

[82] Tornedalen Finnish is considered a separate language by the ISO 639-3 standard. Structurally it is very similar to Finnish, and is mostly used in the northern regions of Sweden.

In many of the well-resourced languages, like Arabic and English, the written form of the language is standardized even if spoken dialects are in use. For many of these languages, the emergence of social media has brought these spoken dialects more into written form (Al-Badrashiny and Diab [2016], Eisenstein [2017]). Being aware of the dialectal variation within a well-resourced language is important when trying to identify rare languages closely related to it. Finnish is also a good example of such a well-resourced language. Finns in fact have a long tradition of dialectal literature (Mielikäinen [2004]). At the end of the 1990s, a trend began to translate well-known comic books into different Finnish dialects and for example Donald Duck has been translated into over ten Finnish dialects (Piippo et al. [2017]). In addition to Tornedalen Finnish, Kven is also considered to be a separate language from Finnish by the ISO 639-3 standard (Simons and Fennig [2018]). When Finnish is written in a dialectal form, it sometimes seems to be orthographically closer to Tornedalen or Kven Finnish than to official written Finnish. Table 12 lists an excerpt from the Bible (Luke 2, 1-3) in official written Finnish,[83] in Tornedalen Finnish,[84] and in the Finnish dialect from Rauma.[85] The words that could be considered official written Finnish in the two latter excerpts are bolded. The excerpt in Tornedalen Finnish has more vocabulary that could be considered official written Finnish than the written Finnish dialect from Rauma.

The problem with using language models generated from official written Finnish to distinguish between written dialectal Finnish and some of the close languages led us to generate additional language models for Finnish using dialectal Finnish sources. Collecting Finnish dialects from the Internet in a similar way as we have collected the Uralic languages could be an interesting endeavor for the future.

The problem of distinguishing between close languages, dialects, and language variants has been tackled in a series of shared tasks as part of VarDial workshops since 2014 (Zampieri et al. [2014], Zampieri et al. [2015b], Malmasi et al. [2016], Zampieri et al. [2017], Zampieri et al. [2018], and Zampieri et al. [2019]). The shared tasks started with the DSL shared task, which was part of the first four workshops in 2014–2017. Later, new specialized shared tasks were added. In 2016, a shared task for Arabic Dialect Identification (ADI) was included in the workshop and it was subsequently reorganized in 2017 and 2018. German Dialect Identification (GDI) debuted as part of the 2017 workshop and was re-run in 2018 and 2019. DSL has not been included as a task in VarDial after 2017, but new tasks for Indo-Aryan Language Identification (ILI) and Discriminating between Dutch and Flemish in Subtitles (DFS) in 2018, as well as Cuneiform Language Identification (CLI) and Discriminating between Mainland and Taiwan Variation of Mandarin Chinese (DMT) in 2019 were added.

[83] http://raamattu.fi/1992/Luuk.2.html

[84] https://keskustelu.suomi24.fi/t/11135014/jouluevankeliumi-mean-kielela

[85] http://nappablog.blogspot.com/2013/12/jouluevankeliumi-rauman-gialel.html

Finnish (ISO 639-3: *fin*)
1. Siihen aikaan antoi keisari Augustus käskyn, *että koko valtakunnassa oli toimitettava verollepano.*
2. Tämä verollepano oli ensimmäinen *ja tapahtui Quiriniuksen ollessa Syyrian käskynhaltijana.*
3. Kaikki menivät kirjoittautumaan veroluetteloon, kukin omaan kaupunkiinsa.
Tornedalen Finnish (ISO 639-3: *fit*)
1. ***Siihen*** *aikhaan tapahtu* ***se***, *ette* ***keisari*** *Aykystykseltä* ***tuli määräys*** ***koko*** *mailmale* ***mennä*** *manttaalihuuole.*
2. ***Tämä*** *manttaalihuuto* ***oli*** *ensimäinen* ***ja se*** *tapahtu* ***silloin*** *ko Kviriniys* ***oli Syyrian maaherra***.
3. ***Ja*** *silloon kaikin menit, itte kuki omhaan kaupunkhiin manttaalihuuole.*
Finnish dialect from Rauma (ISO 639-3: *fin*)
1. Siihe aikka anno keisar ***Augustus*** *oorderi,* *ett* ***koko*** *valdkunnas täyty ruvet kokkoma* ***vero***.
2. ***Tämä*** *verongokkominen ol ensmäine lukkuas* ***ja*** *tapadus sillon go* ***Syyria*** *ol* ***Kyreniuksen*** *gomenos.*
3. ***Kaikki*** *käveväkki sitt skriivaamas puumerkkis verorullaha, jokane omas kaupungisas.*

Table 12: Excerpt from the Bible (Luke 2, 1-3) in official written Finnish, Tornedalen Finnish, and the Finnish dialect from Rauma. The words that could be considered official written Finnish in the two latter excerpts are bolded.

Discriminating between Similar Languages (DSL) shared task The first DSL shared task was organized as part of the workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects (VarDial) in 2014 (Zampieri et al. [2014]). The dataset used in the first DSL shared task and its creation is described by Tan et al. [2014]. The datasets of the later workshops were constructed in the same way, but are different in content.[86] Furthermore, the repertoire of the languages evolved over the years. The languages and language groups considered in the three DSL shared tasks[87] that we participated in are listed in Table 13, together with the best accuracies that we obtained for each language on the closed tracks. In all of the shared tasks, the datasets were balanced so that each language had an equal number of sentences: 18,000 for training, 2,000 for development, and 1,000 for testing. For the actual testing, the development sets could be used for training purposes in combination with the training sets.[88]

The language identification methods used by other participants in the DSL shared tasks are described in articles included in the workshop proceedings, summarized by the overview article of each shared task (Zampieri et al. [2014], Zampieri et al. [2015b], Malmasi et al. [2016], and Zampieri et al. [2017]). Language identification of the

[86] The datasets can be downloaded from `http://ttg.uni-saarland.de/resources/DSLCC/`

[87] We participated in the 2nd, 3rd, and 4th iterations of the DSL shared task during the years 2015–2017.

[88] We used the development set for training in 2016 and 2017, but not in 2015.

Language	Code	DSL 2015	DSL 2016	DSL 2017
Croatian	hr	88.8%	84.6%	81.7%
Bosnian	bs	82.8%	78.1%	80.1%
Serbian	sr	91.6%	91.4%	94.9%
		87.7%	**84.7%**	**85.6%**
Malaysian	my	99.6%	99.2%	97.9%
Indonesian	id	99.8%	99.4%	98.3%
		99.7%	**99.3%**	**98.1%**
Czech	cz	99.9%		
Slovakian	sk	99.7%		
		99.8%		
Portuguese	pt-PT	90.1%	94.3%	93.4%
Brazilian Portuguese	pt-BR	94.6%	94.0%	95.8%
		92.4%	**94.2%**	**94.6%**
Spanish	es-ES	91.3%	70.5%	87.7%
Argentinian Spanish	es-AR	89.5%	85.9%	81.4%
Mexican Spanish	es-MX		83.2%	
Peruvian Spanish	es-PE			89.4%
		90.4%	**79.9%**	**86.2%**
Bulgarian	bg	99.7%		
Macedonian	mk	99.8%		
		99.8%		
French	fr-FR		93.3%	93.6%
Canadian French	fr-CA		91.6%	90.3%
			92.5%	**92.0%**
Iranian Farsi	fa-IR			95.7%
Afghan Farsi	fa-AF			93.5%
				94.6%
Unseen language	*xx*	98.2%		
Our ranking		4th	2nd	7th

Table 13: Languages and the identification accuracies we obtained in the DSL shared tasks from 2015 to 2017. The average accuracies for each subgroup are in bold.

languages belonging to the sub-groups of the DSL tasks has also been researched independently of the shared tasks. Distinguishing between Malay and Indonesian was studied by Ranaivo-Malançon [2006]. Language identification of South-Slavic languages has been researched by Ljubešić et al. [2007], Tiedemann and Ljubešić [2012], Ljubešić and Kranjcić [2014], and Ljubešić and Kranjcić [2015]. Zampieri et al. [2012], Zampieri [2013], Zampieri et al. [2013], and Maier and Gómez-Rodríguez [2014] investigated Spanish dialect identification. Zampieri and Gebre [2012] concentrated on Portuguese dialect identification. The possibilities of distinguishing between French dialects were researched by Zampieri et al. [2012] and Zampieri [2013].

In the DSL 2015 shared task, we used all the language models we generated for HeLI to distinguish between Indonesian and Malaysian, including tokenizing the words by just whitespaces and not removing the non-alphabetic characters. This meant that we had the number characters and their formatting as part of words and character *n*-grams, which was a feature found important by Ranaivo-Malançon [2006] as the use of commas and periods with numbers differs between the two languages. In later editions, we filtered out the non-alphabetic or non-ideographic characters, which could partly explain the decrease in accuracy from 2015 to 2016 and 2017 when Malaysian and Indonesian are concerned.

The DSL 2016 shared task included two extra test sets, B1 and B2, for Bosnian, Croatian, Serbian, Brazilian Portuguese, and European Portuguese. The extra test sets were considered to be out-of-domain in relation to the training and the development set. The test sets consisted of tweets so that several tweets from a single user were merged together to be processed as one text (Malmasi et al. [2016]). We achieved second place in both of the open tracks for the tweet test sets, but that was mainly because only four teams submitted results on the open tracks. Five out of fourteen teams reached better accuracies on the closed tracks than we did on the open tracks.

Arabic Dialect Identification Arabic dialect identification has witnessed an increase in interest in recent years. The first attempt at identifying Arabic dialects is described by Dasigi and Diab [2011]. Since then, it has been researched, for example by Elfardy and Diab [2013], Zaidan and Callison-Burch [2014], and Ali et al. [2016].

In 2016, we participated in the Arabic Dialect Identification (ADI) shared task in addition to the DSL task. The ADI shared task included four Arabic dialects and Modern Standard Arabic. The Arabic texts in the ADI dataset were not natural written language, but they were automatic transcripts generated by speech recognition software as described by Ali et al. [2016]. The training data for the ADI task was not similarly balanced as it was for the DSL task. The five variations of Arabic had different amounts of training data, ranging from 999 sentences for Modern Standard Arabic to the 1,758 sentences of Levantine Arabic. We submitted only one run on the closed track of the task, reaching an F1 score of 0.482%. The result gave us 7th position as can be seen in Table 14.

[89] Malmasi and Zampieri [2016]
[90] Ionescu and Popescu [2016]
[91] Eldesouki et al. [2016]
[92] Adouane et al. [2016]
[93] Zirikly et al. [2016]
[94] Belinkov and Glass [2016]
[95] Ciobanu et al. [2016]
[96] Çöltekin and Rama [2016]
[97] Herman et al. [2016]
[98] Alshutayri et al. [2016]
[99] Guggilla [2016]
[100] Hanani et al. [2016]
[101] McNamee [2016]
[102] Gamallo et al. [2016]

Pos.	Method (*Team*)	Features used	F1 score
1.	SVM ens., mean prob. (*MAZA*)[89]	ch. n-grams 1–6, words	0.513
2.	KDA (*UnibucKernel*)[90]	ch. n-grams 2–7	0.513
3.	SVM (*QCRI*)[91]	ch. n-grams 2–5	0.511
4.	SVM (*ASIREM*)[92]	ch. n-grams 5–6	0.495
	SVM ens., median (*MAZA*)	ch. n-grams 1–6, words	0.494
	SVM ens., voting (*MAZA*)	ch. n-grams 1–6, words	0.492
5.	NN & LR ens. (*GW_LT3*)[93]	ch. n-grams 1–6, words	0.492
	NN (*GW_LT3*)	ch. n-grams 2–6	0.492
6.	CNN ens. (*mitsls*)[94]	characters	0.483
7.	**HeLI (*SUKI*)**	**ch. n-grams 1–8**	**0.482**
8.	SVM (*UniBucNLP*)[95]	ch. n-grams 2–7	0.474
	SVM (*UniBucNLP*)	ch. n-grams 2–6	0.473
9.	SVM (*tubasfs*)[96]	ch. n-grams 1-7	0.473
	SVM (*ASIREM*)	words	0.471
10.	SVM (*HDSL*)	ch. and word n-grams	0.459
11.	EM, 1 iter. (*PITEOG*)[97]	words	0.452
	LR (*GW_LT3*)	ch. n-grams 2–6, word n-grams 1–3	0.448
	CBLM (*PITEOG*)	ch. n-grams	0.447
	CNN (*mitsls*)	characters	0.445
12.	SVM (*ALL*)[98]	ch. 3-grams	0.435
13.	CNN (*cgli*)[99]	word embeddings	0.433
14.	SVM (*AHAQST*)[100]	ch. 3-grams	0.426
	CNN (*mitsls*)	characters	0.418
15.	PPM-A (*hltcoe*)[101]	Markovian ch. n-grams 1–4	0.413
	LSTM ens. (*AHAQST*)	characters	0.412
	SVM (*UniBucNLP*)	ch. n-grams 2–5	0.394
	SVM (*ALL*)	ch. n-grams 1–3	0.387
	SVM (*ALL*)	words	0.384
16.	Quelingua (*Citius_Ixa_Imaxin*)[102]	words	0.382
	EM, 0 iter. (*PITEOG*)	words	0.367
17.	NB (*eire*)[103]	word 2-grams	0.346
	Sum of std. deviations (*AHAQST*)	words	0.341
	NB (*Citius_Ixa_Imaxin*)	words	0.266
18.	DT (J48) (*UCREL*)	words	0.244

Table 14: The weighted F1 scores attained using different method and feature combinations on the ADI 2016 shared task closed track. The results attained using the HeLI method are bolded. A row without position indicates an additional run by a team already positioned higher.

5.2 Short Texts

The length of the text to be identified is one of the major factors affecting the accuracy of identification. Generally, the shorter the text is, the harder it is to identify (Publication 5). How short a text can be in order for its language to be identified

[103] Franco-Penya and Sanchez [2016]

depends strongly also on the presence of other factors. Some of these factors are, for example, how close the other languages are, how many languages there are in the repertoire of the language identifier, and whether the text to be identified is in-domain with the training material or not.

The length of the text is usually measured in characters (Vatanen et al. [2010]), but can be measured, for example, in bytes (Cowie et al. [1999]) or words (Grefenstette [1995]). Hughes et al. [2006] note that most of the research they were aware of concentrated on language identification on the document level. They propose that future work should concentrate on language identification within the documents.

However, one of the earliest needs for language identification was the need to identify the language of individual words, or especially names, for text-to-speech synthesis (Church [1985], Vitale [1991]). The foreign names that can be found in otherwise monolingual texts should still be pronounced according to their language of origin. Church [1985] used the product of relative frequencies method[104] with character trigrams to distinguish between names in 14 languages or language varieties. Identifying the language of individual words in text has been used in speech recognition as well. Häkkinen and Tian [2001] used language identification to automatically generate a recognition grammar for speech recognition. They evaluated two methods, the Markovian character n-grams and a decision tree in name identification between 4 languages (English, Finnish, Spanish, and German). Their decision tree-based language identifier managed to identify proper names with an accuracy of 90.9%.

Mandl et al. [2006] present a table where the accuracies of four language identification methods are compared as a function of the test document size ranging from 25 to 500 characters. The best method they evaluated was NB, which obtained 96.6% accuracy at 25 characters and 99.3% at 50 characters when distinguishing between eight Indo-European languages. In addition to NB, they evaluated the rank order method used earlier by Cavnar and Trenkle [1994] and it reached 93.0% accuracy at 25 characters and 98.2% accuracy at 50 characters. In our evaluations presented in Publication 5, the rank order method[105] had 90.7% accuracy at 25 characters and 97.0% at 50 characters when distinguishing between 285 languages.[106] Mandl et al. [2006] considered the identification of short texts to be especially important for identification of languages in multilingual documents as in their procedure for handling multilingual texts they divided the texts into shorter passages to be separately identified.

Vatanen et al. [2010] present a chart which shows the language identification accuracy in relation to the test sample length when distinguishing between 281 languages. They experimented with test lengths from 5 to 21 characters with 2-character intervals: 5, 7, 9, ..., 21. They ignored word boundaries and thus some of their test texts

[104] See Equations 3 and 20 of Publication 1.

[105] See Equation 22 of Publication 1.

[106] List of the 285 languages is available at `http://suki.ling.helsinki.fi/LILanguages.html`.

might have been partial words. Their results show clearly how identification accuracy rises as the test size increases. NB with Absolute Discounting smoothing, the best method they evaluated, already achieves almost 50% accuracy at 5 characters and almost 85% at 15 characters. Their tests were in-domain tests as the training and test sets were different parts of the same original document. As a dataset, they used the translations of the Universal Declaration of Human Rights (UDHR) for 281 languages.[107] As a source for language models in general, the UDHR corpus is rather small and specialized. For example, the English version contains only 1,553 words, of which 449 are unique, and the 9th most common word is "Article".

For Publication 5, we did extensive evaluations using test text sizes varying from 5 to 150 characters for 285 languages. The macro average F-scores are presented in Table 15. In addition to our HeLI method, the other evaluated language identifiers and methods were the "Whatlang" program which uses variable length byte n-grams from 3 to 12 bytes as its language model (Brown [2013]), NB with Absolute Discounting (NBAD) smoothing (Vatanen et al. [2010]), the LIGA method (Tromp [2011]), the LogLIGA variation of the LIGA method (Vogel and Tresner-Kirsch [2012]), the rank order method by Cavnar and Trenkle [1994], and the product of relative frequencies (King and Dehdari [2008]). The Absolute Discounting method obtained 64% accuracy at 5 characters and 90% at 15 characters. When we constructed the dataset used in our experiments, we aimed at creating an out-of-domain test setting. Where possible, we used data sources for the training and the test sets that were from different domains. We did an additional pass on the much larger training sets making certain that passages from the test sets were not included in them (Publication 5). However, the accuracies for Absolute Discounting were higher than those reported by Vatanen et al. [2010] for the in-domain tests, even though we mostly used the same UDHR documents as a source for our test data. We had manually removed foreign language inclusions from the source of our test data and had noticed that foreign inclusions could be found even in the UDHR collection. The foreign languages present in the UDHR documents could have been partly responsible for the best accuracies obtained by Vatanen et al. [2010] being under 90% at their longest test length of 21 characters, when in our out-of-domain test setting the same method already reached 94%.

It can be seen from the results of our evaluations that sometimes the accuracy for individual languages actually gets lower with longer test texts.[108] This can be due to the fact that the test texts were randomly selected for each length, and thus it is possible that more difficult or more ambiguous texts were selected for the longer lengths than for the shorter ones.

[107] The corpus used by Vatanen et al. [2010] is available at http://research.ics.aalto.fi/cog/data/udhr/.

[108] http://suki.ling.helsinki.fi/NodaEvalResults.xlsx

Length	HeLI	Whatlang	NBAD	LIGA	LogLIGA	C & T	ProdRF
5	63.3	55.1	**64.5**	36.0	48.4	34.6	58.4
10	83.2	78.0	**83.8**	46.6	56.4	66.2	75.4
15	90.2	87.1	**90.7**	55.2	59.6	80.5	82.8
20	**94.0**	91.7	**94.0**	62.0	60.5	87.6	86.4
25	**96.0**	94.2	95.7	67.0	61.2	91.5	88.5
30	**97.2**	95.7	96.8	70.7	61.5	93.7	89.9
35	**98.0**	96.8	97.4	73.9	61.7	95.2	90.7
40	**98.5**	97.4	97.9	76.2	62.0	96.1	91.4
45	**98.9**	97.9	98.2	78.4	62.1	96.8	91.9
50	**99.2**	98.3	98.6	80.1	62.2	97.3	92.3
55	**99.3**	98.6	98.7	81.5	62.3	97.7	92.5
60	**99.5**	98.8	98.9	82.7	62.4	98.1	92.8
65	**99.6**	99.0	99.0	83.7	62.4	98.3	93.1
70	**99.7**	99.1	99.1	84.6	62.4	98.5	93.2
80	**99.8**	99.4	99.2	86.3	62.5	98.8	93.4
90	**99.9**	99.5	99.3	87.4	62.5	99.0	93.7
100	**99.9**	99.6	99.4	88.3	62.5	99.2	93.8
120	**100.0**	99.8	99.6	89.9	62.5	99.4	94.0
150	**100.0**	99.9	99.7	91.6	62.5	99.6	94.2

Table 15: The F-scores calculated from macro averaged recall and precision attained by different methods for different test text lengths (in characters) in the out-of-domain test setting for 285 languages. The best scores for each text length are bolded.

5.3 Large Number of Languages

Generally, the more languages the language identification method has to distinguish between, the harder the task seems to become (Brown [2012], Rodrigues [2012], and Publication 5). It is intuitively understandable that if more classes are added to a classifier, the classifying becomes more difficult. However, this partly depends on the performance measure used. For example, if we are measuring the average accuracy for all languages, the average accuracy may improve in cases where easily distinguishable languages are added to the repertoire. Brown [2014] presents results where the average accuracy is higher for 1,366 languages than for a subset of 781 languages.[109] He attributes this phenomenon to the greater percentage of unannotated multilingual Wikipedia texts present in the smaller corpus. Most of the research in language identification has concentrated on a small number of languages. We have listed references presenting empirical evaluations of language identification methods with more than 100 languages in Table 16.

[109] Originally, most of the language identifiers he evaluates perform better with a smaller language repertoire, but after applying the non-linear mappings all the identifiers have better accuracy with more languages.

Reference	# Lang
Brown [2014]	1,366
Brown [2013]	1,100
Brown [2012]	923
Xia et al. [2009]	c. 600
Rodrigues [2012]	372
King and Dehdari [2008]	300
Publication 7	285
Publication 5	285
Vatanen et al. [2010]	281
Yamaguchi and Tanaka-Ishii [2012]	200+
Cazamias et al. [2015]	200
Chew et al. [2011]	182
Lui [2014]	143
Kocmi and Bojar [2017]	136
Majliš [2011]	122
Jauhiainen [2010]	103

Table 16: Empirical evaluations with more than 100 languages.

Vatanen et al. [2010] experimented with two sets of languages, the smaller containing 50 languages and the larger 281 languages. They compared the rank order method by Cavnar and Trenkle [1994] with NB. They tested several different smoothing methods with NB, and the best performing turned out to be Absolute Discounting smoothing (Ney et al. [1994]). When the average accuracy was calculated over all sample lengths (from 5 to 21 characters), the language identifier using Absolute Discounting smoothing reached 82.2% accuracy with the smaller set of languages and 77.8% with the larger set. Majliš [2012] evaluated five different classification methods with 30, 60, and 90 languages. He presents a chart which clearly shows how the accuracies of all the methods decrease when languages are added to the repertoire.

The evaluation chart presented by Majliš [2012] shows how the performance of a Regression tree-based classifier is affected more than the other evaluated methods when the number of languages is increased. The relative order of the other four methods does not change, but the Regression tree classifier drops from third position to last when the number of languages is increased from 60 to 90. Majliš [2012] does not discuss the phenomenon. In Publication 5, we show that not all language identification methods scale to a larger number of languages. The LIGA and LogLIGA methods were originally tested with a smaller number of languages, and in those evaluations the LogLIGA method was clearly better (Vogel and Tresner-Kirsch [2012]). However, in an evaluation setting equipped with 285 languages, the LogLIGA method fails to distinguish between the languages as well as the original LIGA method (Publication 5).

In our work, the effect of adding languages can also be seen in Table 13 in Section 5.1 on page 44, which shows the identification accuracies we obtained in the different VarDial shared tasks. From 2015 to 2016, the number of Spanish variants was raised from 2 to 3, and as a result the accuracy of the group identification lowered from 90.4% to 79.9%. The test setting and the language identifier are not exactly identical[110] between the shared tasks, but they are similar enough for us to be confident that the reduction in accuracy is in large part due to adding a language variety.

Instead of directly identifying the languages of a text, the identification can be done in a tiered fashion by first distinguishing between larger groups of languages. Language groups can be identified using methods and parameters optimized for the task, after which the languages within the groups are identified using different methods or parameters. This tiered method was used by Goutte et al. [2014] in the first edition of the DSL shared task and we used it in the second edition. In our results from 2015, there was only one sentence that was classified into an incorrect group: one Portuguese sentence was classified as Spanish. In the two following workshops, 2016 and 2017, we did not use a separate tier for group classification. There were 15 incorrect identifications over language groups in 2016, which is much more than in 2015. However, as can be seen in Table 6 on page 33, the 15 incorrect groupings are mostly due to bad corpus quality. In 2017, our method made a total of 28 incorrect groupings, mostly in similar low quality texts as in 2016. In the light of these results, we think that using a separate tier for group identification is not necessary to identify the languages in their groups, at least not for the HeLI method. However, it might be beneficial to do so in practice as then there is the possibility of re-calibrating the parameters within each group. This, in turn, might help the identification within individual language groups.

5.4 Unseen Languages

In supervised machine learning, the language identifier is trained to distinguish between the languages using examples in the training data (Kotsiantis [2007]). All those languages which the language identifier has not been trained in are called unknown or unseen languages. Many language identification methods do not include handling the unseen languages and therefore these methods just guess the unseen language to be one of the languages in their repertoire.

In cases where the language identifier is not used in a controlled setting, it is always possible that it comes across languages that it does not have a language model for. This is especially true for a language identifier which is used in a web crawling pipeline. Xia et al. [2009] used language identification in harvesting instances of Interlinear Glossed Text (IGT) from the Internet. They report that around 10% of their IGT instances were written in an unseen language. We faced the same problem with the language identifier we used in combination with the Heritrix crawler (Mohr et al. [2004]) in the Finno-Ugric Languages and the Internet project. Our language

[110] Each year, previously unseen training and test data was provided to the shared task participants. We also experimented with several variations of the HeLI method in 2015, see Section 3.2.

identifier had language models for 395 languages, but sometimes it encountered texts in an unseen language.

One of the most straightforward ways of implementing unseen language detection in a language identifier is through thresholding (Cowie et al. [1999], Eskander et al. [2014], and Bobicev [2015]). In thresholding, a pre-determined confidence threshold is used to decide whether the classification is successful or not. In order to be used with thresholding, the language identification method should have some way of indicating how confident the predictions are for the languages it knows. This is more problematic for discriminating methods, which can produce a high confidence value for language A based on just the fact that the method is certain that language B has been ruled out.

In Publication 2, we present a method for unseen language detection in which we use thresholding as part of the method. The first part of the unseen language detection method is based on the score $R(g, M)$ produced by the HeLI method.[111] The score depicts a probability that the sentence M is written in language g and it is normalized so that it can be compared between mystery texts of different lengths. A separate threshold is decided for each language g using a development corpus. If the score $R(g, M)$ of the highest scoring language is higher[112] than the threshold for that language, the text is tagged to be written with an unseen language. If the score is lower than the threshold, we continue to the second part of the method. In the second part, we count how many of the lowercased words consisting of alphabetical characters are found in the language models used, $dom(O(C^{'}))$,[113] and how many are not. Then we use the ratio between these two counts as a cut-off, which is again decided individually for each language using the development corpus. If the ratio calculated from the mystery text is higher than the cut-off, the text is tagged as being in an unseen language.

The 2nd edition of the DSL shared task contained a separate category for unseen languages (Zampieri et al. [2015b]). Several unseen languages were grouped under one category and there was no annotation to distinguish between the different languages. We used different optimization schemes for the threshold and the cut-off ratio for our submission to the shared task. In the first scheme, we optimized the parameters so that the language identifier made as few precision errors (identifying something else as the unseen language) with the development set as possible and in the second scheme we aimed to reach the best overall accuracy. The difference in overall accuracy between the two approaches was 0.63% on the test set (93.73% vs. 94.36%).

Another way to deal with unseen languages is to create a separate language model for the unseen language (Adouane [2016] and Adouane and Dobnik [2017]). The unseen language model is trained using data which includes textual material in several

[111] Equation 20 in Section 3.2 on page 24.

[112] Since we are using negative logarithms of probabilities, the language having the lowest score is returned as the language with the maximum probability for the mystery text.

[113] See the definition of $dom(O(C^{'}))$ in Section 3.2.

Aasiankultakissa, (*Catopuma temminckii* eli *Profelis temminckii* eli *Felis temminckii*) on Kaakkois-Aasiassa elävä kissaeläin.

Figure 1: Multilingual example from Finnish Wikipedia of a sentence written in Finnish and Latin.

irrelevant languages. A variation of this method is to train the language identifier with as many languages as possible. If most of the languages are irrelevant for the task at hand, it is not a problem if they gather a lot of junk and material in unseen languages. We used the irrelevant language approach in the production system of the Finno-Ugric Languages and the Internet project. Most of the 395 languages known by our system were irrelevant with regard to the project's aim to collect text material in rare Uralic languages.

5.5 Multilingual Texts

Multilingual texts are texts written using more than one language. The length of a passage in another language needed for the text to be called multilingual varies on a case by case basis. Multilingual text can be, for example, a web page where translations of one paragraph exist in different languages or a foreign language quote inside an otherwise monolingual text. Multilingual language identification is needed for automatic processing of multilingual documents in general, for example, in machine translation or information retrieval (Prager [1999], Ozbek et al. [2006], Mandl et al. [2006], Murthy and Kumar [2006], Hughes et al. [2006], King et al. [2015], and Lui et al. [2014]). Multilingual language identification for corpora creation purposes has earlier been studied by Ludovik and Zacharski [1999]. As an example of a multilingual text, we present a line from Finnish Wikipedia in Figure 1. The example includes 7 words in Finnish and 6 words in Latin.

One area which has recently gained much attention is code-switching (Nguyen and Dogruöz [2013], Giwa and Davel [2013], Solorio et al. [2014], and Mave et al. [2018]). Code-switching happens when words or terms are borrowed from other languages and used as parts of sentences.

Language set identification is the task of determining which languages are present in a document or text segment. Performing language set identification is advantageous when dealing with multilingual texts and the set of possible languages is large. After the language set identification has been conducted, the segmentation of the document by language can be done with a hugely reduced set of languages, thus making it a much simpler task.

Lui et al. [2014] created an openly available corpus for evaluating multilingual language identification, the WikipediaMulti.[114] They used the corpus to evaluate two previously introduced methods for multilingual language identification by Prager

[114] https://people.eng.unimelb.edu.au/tbaldwin/etc/wikipedia-multi-v6.tgz

[1999] and Yamaguchi and Tanaka-Ishii [2012] as well as their own method. In order to provide comparable results, we used this same corpus when evaluating our own method in Publication 7. Later, the same corpus was used by Kocmi and Bojar [2017].

The method we proposed in Publication 7 is built on the idea of using existing monolingual language identifiers in determining the set of languages for a multilingual text. The basic idea is to slide an overlapping byte window of size x through the text in steps of one byte. The text in each window is sent to a separate language identifier, which gives the most likely language for the window. In our method, there is a variable called "CurrentLanguage," which is first set as the language of the first byte window. CurrentLanguage changes after z consecutive identifications of the byte window give a language different from the current CurrentLanguage as a result. When the whole text has been processed, the set of languages for the text comprises all the languages that have been the CurrentLanguage at some point. The parameters, window size x and change threshold z, are empirically determined using a development set.[115]

The idea of using a window approach in multilingual language identification was earlier proposed by Mandl et al. [2006]. However, they used the number of words as the measure of the size for a window and the language was changed each time a different language (from a selection of 8 languages) was identified for a window. When we are handling noisy documents, the number of languages to be identified is large, or when we are dealing with very close languages, we need to have several frames agreeing on the language change before actually changing the CurrentLanguage. Furthermore, the byte window is applicable for texts in languages where word segmentation is non-trivial, such as Chinese and Japanese.

The used corpus, WikipediaMulti, is a synthetic corpus of multilingual texts made available by Lui et al. [2014]. It consists of three parts each with 44 languages: 5,000 monolingual documents for training, another 5,000 multilingual documents for development, and 1,000 multilingual documents for testing. All the multilingual documents have been generated by randomly concatenating parts of monolingual documents. We used the same macro- and micro-averaged recall, precision, and F-score as performance measures as Lui et al. [2014]. Our best result, micro-averaged F-score of .975, on the development set was achieved using the window size x of 400 bytes and z of 100 times as the change threshold.

Later, Kocmi and Bojar [2017] evaluated their language set identification system with the WikipediaMulti dataset. Their system uses Recurrent Neural Networks (RNN) with Gated Recurrent Units (GRU) for language set identification. In addition, they considered the use of Long Short-Term Memory (LSTM), but LSTM did not provide as good results as the GRUs. It is probable, that Kocmi and Bojar [2017] were not aware of our Publication 7, as they do not refer to our paper.

We reproduce the evaluation results with WikipediaMulti presented by Kocmi and Bojar [2017] in Table 17, where our own results are included on the last line. In

[115] We used x of 400 bytes and z of 100 times when we achieved the best results in Publication 7.

System	P_M	R_M	F_M	P_μ	R_μ	F_μ
SEGLANG	80.9%	97.5%	.875	77.1%	97.5%	.861
LINGUINI	85.3%	77.2%	.802	83.8%	77.4%	.805
LLB 2014	96.2%	96.4%	.957	96.3%	95.5%	.959
LLB 2017	96.2%	96.3%	.961	96.3%	96.4%	.963
LanideNN	96.2%	97.4%	.966	95.4%	97.4%	.964
Jauhiainen et al. 2015	97.7%	97.9%	.977	97.4%	97.9%	.976

Table 17: The macro- (M) and micro- (μ) averaged precision (P), recall (R), and the F-score (F) with different methods in the WikipediaMulti language set identification task.

System	P_μ	R_μ	F_μ
Jauhiainen et al. 2015, 43 languages	97.6%	98.3%	.979
Kocmi and Bojar 2017, 43 languages	96.6%	97.3%	.964
Jauhiainen et al. 2015, 285 languages	97.3%	98.3%	.978
Kocmi and Bojar 2017, 136 languages	94.9%	97.2%	.961

Table 18: The micro- (μ) averaged precision (P), recall (R), and the F-score (F) using pre-trained language models.

the table, “SEGLANG” refers to the system by Yamaguchi and Tanaka-Ishii [2012], “LINGUINI” to a system by Prager [1999], “LLB 2014” to the original results by Lui et al. [2014], “LLB 2017” to the re-training of the Lui et al. [2014] system by Kocmi and Bojar [2017], and “LanideNN” to the system by Kocmi and Bojar [2017] themselves. Kocmi and Bojar [2017] assume that the difference between the original and retrained results obtained using the system by Lui et al. [2014] is due to the Gibbs sampling used in the method. The table clearly shows that our method still attains the best results by every measure.

Kocmi and Bojar [2017] also evaluated the performance of their method using their own training data on the same test set. They had training data for 136 different languages and they report results using all of those languages as well as a reduced set of 43 languages. Earlier, in Publication 7, we did a very similar experiment with our own language identifier using the 285 language models from Publications 5 and 6. The micro-averaged recall, precision, and the F-scores are collected in Table 18. All the results would seem to indicate that our method is still the state-of-the-art in language set identification, clearly providing better results than those obtained by Kocmi and Bojar [2017] using state-of-the-art deep learning methods.

6. Conclusion

This work has investigated the automatic language identification of digital texts. Over the last 50 years, automatic language identification of text has emerged as a separate field of study related to general text categorization. We have shown that language identification is actively being researched in a variety of research fields and that language identification is an important part of many real-world applications employing natural language processing. As a task, language identification has sometimes been branded as solved, but the processing of real-world data in real-world applications has revealed many open issues.

Especially within the last few years, the amount of research related to language identification has continued to increase. Despite the ongoing interest in the subject, the field lacked a comprehensive survey article. As part of this dissertation, we have presented the most comprehensive survey on language identification of digitally-encoded text so far (Publication 1). In addition, we have shown that different ways of describing language identification methods evidenced in the research literature sometimes hinder the re-usability of the said methods. In order to make the situation more coherent, we created a unified notation that can be used to describe features and methods used for language identification.

The need to evaluate and compare different language identification methods in a controlled setting has led to the establishment of a series of shared tasks. The DSL shared task has especially concentrated on distinguishing between close languages, dialects, and language varieties. We developed a language identification method called HeLI and applied it in three consecutive DSL shared tasks (Publications 2, 3, and 4). The HeLI method proved to be very competitive and was ranked in shared first place in the 3rd edition of the shared task (Publication 3).

The goals of the Finno-Ugric Languages and the Internet project led us to evaluate the most promising of the available language identification methods in an out-of-domain situation for as many languages as possible. We created new text corpora for those rare languages in which existing corpora were not available by locating and downloading material from the Internet (Publications 5 and 6). We evaluated the HeLI method together with two existing language identifiers and our implementations of four other methods with a dataset for 285 languages including very rare languages from the Uralic language group. The HeLI method outperformed the other methods and considerably reduced the identification error rate for texts over 60 characters in length.

We presented a method for identifying the set of languages of multilingual documents (Publication 7). We evaluated the method using an existing corpus designed for multilingual language identifier evaluation. Our language set identification method combined with the HeLI language identification method clearly outperforms the other methods that have been evaluated on the dataset to date.

6.1 Future Tasks

One of the issues we have not yet addressed is the segmenting of multilingual documents by language. Our language set identification scheme already detects the approximate position of the language change in a multilingual document, but it should be extended so that it can pinpoint the exact position. We would also like to experiment with word-level language identification in documents where code-switching takes place. Perhaps conditional random fields could be used together with the HeLI method to learn code-switching patterns.

Distinguishing between some of the very close languages, like Serbian, Croatian, and Bosnian, would still seem to be difficult for the HeLI method when compared with some of the more discriminative methods like SVM. We would like to continue investigating the possibility of including some discriminative elements into the method in order to better deal with the remaining difficult cases.

We have already continued to further improve the HeLI method after the research included in this dissertation. We participated in three of the shared tasks included in the VarDial evaluation campaign of 2018 (Zampieri et al. [2018]). We used language model adaptation to adjust the values in the language models to fit the mystery text collection. Our submissions managed to achieve clear first positions in the Indo-Aryan Language Identification and the German Dialect Identification shared tasks (Jauhiainen et al. [2018a] and Jauhiainen et al. [2018b]).

We are also planning to revisit the issue of detecting unseen languages. In the 2018 GDI shared task, our language model adaptation method would have profited from better unseen language detection as there were unseen dialects present within the test set. However, this is a very difficult issue as the line between unseen dialects and out-of-domain texts can be very narrow.

References

Kheireddine Abainia, Siham Ouamour, and Halim Sayoud. Effective Language Identification of Forum Texts Based on Statistical Approaches. *Information Processing and Management*, 52:491–512, 2016.

Judit Ács, László Grad-Gyenge, Thiago Bruno, and Rodriguez de Rezende Oliveira. A Two-level Classifier for Discriminating similar Languages. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects (LT4VarDial)*, pages 73–77, Hissar, Bulgaria, 2015.

Gary Adams and Philip Resnik. A Language Identification Application Built on the Java Client/server Platform. In *Proceedings of the ACL/EACL'97 Workshop on From Research to Commercial Applications: Making NLP Work in Practice*, pages 43–47, Madrid, Spain, 1997.

Wafia Adouane. Automatic Detection of Underresourced Languages: Dialectal Arabic Short Texts. Master's thesis, University of Gothenburg, Gothenburg, Sweden, 2016.

Wafia Adouane and Simon Dobnik. Identification of Languages in Algerian Arabic Multilingual Documents. In *Proceedings of The Third Arabic Natural Language Processing Workshop (WANLP 2017)*, pages 1–8, Valencia, Spain, 2017.

Wafia Adouane, Nasredine Semmar, and Richard Johansson. ASIREM Participation at the Discriminating Similar Languages Shared Task 2016. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 163–169, Osaka, Japan, 2016.

Bashir Ahmed, Sung-Hyuk Cha, and Charles Tappert. Language Identification from Text Using N-gram Based Cumulative Frequency Addition. In *Proceedings of Student/Faculty Research Day*, pages 12.1–12.8, CSIS, Pace University, New York, USA, 2004.

Mohamed Al-Badrashiny and Mona T. Diab. LILI: A Simple Language Independent Approach for Language Identification. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING 2016): Technical Papers*, pages 1211–1219, Osaka, Japan, 2016.

Beatrice Alex. *Automatic Detection of English Inclusions in Mixed-lingual Data with an Application to Parsing.* PhD thesis, The University of Edinburgh, 2008.

Ahmed Ali, Najim Dehak, Patrick Cardinal, Sameer Khurana, Sree Harsha Yella, James Glass, Peter Bell, and Steve Renals. Automatic Dialect Detection in Arabic Broadcast Speech. In *Proceedings of Interspeech 2016*, pages 2934–2938, San Francisco, USA, 2016.

Areej Alshutayri, Eric Atwell, AbdulRahman Alosaimy, James Dickins, Michael Ingleby, and Janet Watson. Arabic Language WEKA-Based Dialect Classifier for Arabic Automatic Speech Recognition Transcripts. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 204–211, Osaka, Japan, 2016.

A. Suresh Babu and P. Kumar. Comparing Neural Network Approach with N-Gram Approach for Text Categorization. *International Journal on Computer Science and Engineering*, 2(1):80–83, 2010.

Adrien Barbaresi. An Unsupervised Morphological Criterion for Discriminating Similar Languages. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 212–220, Osaka, Japan, 2016.

Adrien Barbaresi. Discriminating between Similar Languages using Weighted Subword Features. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 184–189, Valencia, Spain, 2017.

Yonatan Belinkov and James Glass. A character-level Convolutional Neural Network for Distinguishing Similar Languages and Dialects. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 145–152, Osaka, Japan, 2016.

Shane Bergsma, Paul McNamee, Mossaab Bagdouri, Clayton Fink, and Theresa Wilson. Language Identification for Creating Language-specific Twitter Collections. In *Proceedings of the Second Workshop on Language in Social Media (LSM2012)*, pages 65–74, Montréal, Canada, 2012.

Yves Bestgen. Improving the Character Ngram Model for the DSL Task with BM25 Weighting and Less Frequently Used Feature Sets. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 115–123, Valencia, Spain, 2017.

Johannes Bjerva. Byte-based Language Identification with Deep Convolutional Networks. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 119–126, Osaka, Japan, 2016.

Andreas Blass, Nachum Dershowitz, and Yuri Gurevich. When Are Two Algorithms The Same? *The Bulletin of Symbolic Logic*, 15(2):145–168, 2009.

Su Lin Blodgett, Johnny Tian-Zheng Wei, and Brendan O'Connor. A Dataset and Classifier for Recognizing Social Media English. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 56–61, Copenhagen, Denmark, 2017.

Victoria Bobicev. Discriminating between Similar Languages Using PPM. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects (LT4VarDial)*, pages 59–65, Hissar, Bulgaria, 2015.

Alessio Bosca and Luca Dini. Language Identification Strategies for Cross Language Information Retrieval. In *Working notes for LogCLEF2010: the CLEF 2010 Multilingual Logfile Analysis Track*, Padua, Italy, 2010.

Ralf D. Brown. Finding and Identifying Text in 900+ Languages. *Digital Investigation*, 9:S34–S43, 2012.

Ralf D. Brown. Selecting and Weighting N-grams to Identify 1100 Languages. In *Proceedings of the 16th International Conference on Text, Speech and Dialogue (TSD 2013)*, pages 475–483, Plzeň, Czech Republic, 2013.

Ralf D. Brown. Non-linear Mapping for Improved Identification of 1300+ Languages. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 627–632, Doha, Qatar, 2014.

Andrei M. Butnaru and Radu Tudor Ionescu. UnibucKernel Reloaded: First Place in Arabic Dialect Identification for the Second Year in a Row. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 77–87, Santa Fe, New Mexico, USA, 2018.

William B. Cavnar and John M. Trenkle. N-Gram-Based Text Categorization. In *Proceedings of SDAIR-94, Third Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Las Vegas, USA, 1994.

Jordan Cazamias, Chinmayi Dixit, and Martina Marek. Large-Scale Language Classification - Writing a Detector for 200 Languages on Twitter. Stanford course report, 2015.

Çagri Çöltekin and Taraka Rama. Discriminating Similar Languages: Experiments with Linear SVMs and Neural Networks. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 15–24, Osaka, Japan, 2016.

Çagri Çöltekin and Taraka Rama. Tübingen System in VarDial 2017 Shared Task: Experiments with Language Identification and Cross-lingual Parsing. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 146–155, Valencia, Spain, 2017.

Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 13(4):359–394, 1999.

Yew Choong Chew, Yoshiki Mikami, and Robin Lee Nagano. Language Identification of Web Pages Based on Improved N-gram Algorithm. *International Journal of Computer Science Issues*, 8(3):47–58, 2011.

Kenneth Church. Stress Assignment in Letter to Sound Rules for Speech Synthesis. In *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, pages 246–253, Chicago, Illinois, USA, 1985.

Andre Cianflone and Leila Kosseim. N-gram and Neural Language Models for Discriminating Similar Languages. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 243–250, Osaka, Japan, 2016.

Alina Maria Ciobanu, Sergiu Nisioi, and Liviu P. Dinu. Vanilla Classifiers for Distinguishing between Similar Languages. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 235–242, Osaka, Japan, 2016.

Alina Maria Ciobanu, Shervin Malmasi, and Liviu P. Dinu. German Dialect Identification Using Classifier Ensembles. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 288–294, Santa Fe, New Mexico, USA, 2018.

Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. The MIT Press, Cambridge, Massachusetts, 1990.

Jim Cowie, Yevgeny Ludovik, and Ron Zacharski. Language Recognition for Mono- and Multi-lingual Documents. In *Proceedings of the VexTal Conference*, pages 209–214, Venice, Italy, 1999.

Marcelo Criscuolo and Sandra Maria Aluísio. Discriminating between Similar Languages with Word-level Convolutional Neural Networks. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 124–130, Valencia, Spain, 2017.

Pradeep Dasigi and Mona Diab. CODACT: Towards Identifying Orthographic Variants in Dialectal Arabic. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 318–326, Chiang Mai, Thailand, 2011.

Markus Dickinson. Detection of Annotation Errors in Corpora. *Language and Linguistics Compass*, 9(3):119–138, 2015.

Ted Dunning. Statistical Identification of Language. Technical Report MCCS 940-273, Computing Research Laboratory, New Mexico State University, 1994.

Jacob Eisenstein. Identifying regional dialects in on-line social media. In C. Boberg, J. Nerbonne, and D. Watt, editors, *Handbook of Dialectology*. Wiley, 2017.

Mohamed Eldesouki, Fahim Dalvi, Hassan Sajjad, and Kareem Darwish. QCRI DSL 2016: Spoken Arabic Dialect Identification Using Textual Features. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 221–226, Osaka, Japan, 2016.

Heba Elfardy and Mona Diab. Sentence Level Dialect Identification in Arabic. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 456–461, Sofia, Bulgaria, 2013.

Ramy Eskander, Mohamed Al-Badrashiny, Nizar Habash, and Owen Rambow. Foreign Words and the Automatic Processing of Arabic Social Media Text Written in Roman Script. In *Proceedings of The First Workshop on Computational Approaches to Code Switching*, pages 1–12, Doha, Qatar, 2014.

Raül Fabra-Boluda, Francisco Rangel, and Paolo Rosso. NLEL UPV Autoritas participation at Discrimination between Similar Languages DSL 2015 Shared Task. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects (LT4VarDial)*, pages 52–58, Hissar, Bulgaria, 2015.

Hector-Hugo Franco-Penya and Liliana Malmani Sanchez. Tuning Bayes Baseline for Dialect Detection. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 227–234, Osaka, Japan, 2016.

Marc Franco-Salvador, Paolo Rosso, and Francisco Rangel. Distributed Representations of Words and Documents for Discriminating Similar Languages. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects (LT4VarDial)*, pages 11–16, Hissar, Bulgaria, 2015.

Pablo Gamallo, José Ramom Pichel, Iñaki Alegria, and Manex Agirrezabal. Comparing two Basic Methods for Discriminating Between Similar Languages and Varieties. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 170–177, Osaka, Japan, 2016.

Pablo Gamallo, Jose Ramom Pichel, and Iñaki Alegria. A Perplexity-Based Method for Similar Languages Discrimination. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 109–114, Valencia, Spain, 2017.

Archana Garg, Vishal Gupta, and Manish Jindal. A Survey of Language Identification Techniques and Applications. *Journal of Emerging Technologies in Web Intelligence*, 6(4):388–400, 2014.

Oluwapelumi Giwa and Marelie H. Davel. N-Gram based Language Identification of Individual Words. In Philip Robinson, editor, *Proceedings of the 24th Annual*

Symposium of the Pattern Recognition Association of South Africa, pages 15–22, Johannesburg, South Africa, 2013.

Cyril Goutte and Serge Léger. Experiments in Discriminating Similar Languages. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects (LT4VarDial)*, pages 78–84, Hissar, Bulgaria, 2015.

Cyril Goutte and Serge Léger. Advances in Ngram-based Discrimination of Similar Languages. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 178–184, Osaka, Japan, 2016.

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

Cyril Goutte, Serge Léger, Shervin Malmasi, and Marcos Zampieri. Discriminating Similar Languages: Evaluations and Explorations. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia, may 2016.

Gregory Grefenstette. Comparing Two Language Identification Schemes. In *Proceedings of the 3rd International conference on Statistical Analysis of Textual Data (JADT 1995)*, Rome, Italy, 1995.

Chinnappa Guggilla. Discriminating between Similar Languages, Varieties and Dialects using CNN- and LSTM-based Deep Neural Networks. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 185–194, Osaka, Japan, 2016.

Helena Gómez-Adorno, Ilia Markov, Jorge Baptista, Grigori Sidorov, and David Pinto. Discriminating between Similar Languages Using a Combination of Typed and Untyped Character N-grams and Words. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 137–145, Valencia, Spain, 2017.

Harald Hammarström. A Fine-Grained Model for Language Identification. In *Proceedings of Improving Non English Web Searching (iNEWS-07) Workshop at SIGIR 2007*, pages 14–20, Amsterdam, Netherlands, 2007.

Abualsoud Hanani, Aziz Qaroush, and Stephen Taylor. Classifying ASR Transcriptions According to Arabic Dialect. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 126–134, Osaka, Japan, 2016.

Einar Haugen. Dialect, language, nation. *American anthropologist*, 68(4):922–935, 1966.

Ondřej Herman, Vít Suchomel, Vít Baisa, and Pavel Rychlý. DSL Shared task 2016: Perfect Is The Enemy of Good Language Discrimination Through Expectation-Maximization and Chunk-based Language Model. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 114–118, Osaka, Japan, 2016.

Baden Hughes, Timothy Baldwin, Steven Bird, Jeremy Nicholson, and Andrew MacKinlay. Reconsidering Language Identification for Written Language Resources. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, pages 485–488, Genoa, Italy, 2006.

Juha Häkkinen and Jilei Tian. N-gram and Decision Tree Based Language Identification for Written Words. In *Conference Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU 2001)*, pages 335–338, Madonna di Campiglio, Italy, 2001.

Radu Tudor Ionescu and Marius Popescu. UnibucKernel: An Approach for Arabic Dialect Identification based on Multiple String Kernels. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 135–144, Osaka, Japan, 2016.

Heidi Jauhiainen, Tommi Jauhiainen, and Krister Lindén. The Finno-Ugric Languages and The Internet Project. *Septentrio Conference Series*, 0(2):87–98, 2015a. ISSN 2387-3086. doi: 10.7557/5.3471.

Tommi Jauhiainen. Tekstin kielen automaattinen tunnistaminen. Master's thesis, University of Helsinki, Helsinki, 2010.

Tommi Jauhiainen, Heidi Jauhiainen, and Krister Lindén. Discriminating Similar Languages with Token-based Backoff. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects, LT4VarDial '15*, pages 44–51, Hissar, Bulgaria, 2015b.

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

Tommi Jauhiainen, Krister Lindén, and Heidi Jauhiainen. HeLI, a Word-Based Backoff Method for Language Identification. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 153–162, Osaka, Japan, 2016.

Tommi Jauhiainen, Krister Lindén, and Heidi Jauhiainen. Evaluation of Language Identification Methods Using 285 Languages. In *Proceedings of the 21st Nordic Conference on Computational Linguistics (NoDaLiDa 2017)*, pages 183–191, Gothenburg, Sweden, 2017a. Linköping University Electronic Press.

Tommi Jauhiainen, Krister Lindén, and Heidi Jauhiainen. Evaluating HeLI with Non-Linear Mappings. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 102–108, Valencia, Spain, 2017b.

Tommi Jauhiainen, Heidi Jauhiainen, and Krister Lindén. Iterative Language Model Adaptation for Indo-Aryan Language Identification. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 66–75, Santa Fe, NM, USA, 2018a.

Tommi Jauhiainen, Heidi Jauhiainen, and Krister Lindén. HeLI-based experiments in Swiss German dialect identification. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 254–262, Santa Fe, NM, USA, 2018b.

Tommi Jauhiainen, Marco Lui, Marcos Zampieri, Timothy Baldwin, and Krister Lindén. Automatic Language Identification in Texts: A Survey. *(submitted to JAIR 10/2018)*, 2018c.

Patrick Juola. Language Identification, Automatic. In *Encyclopedia of Language and Linguistics*, volume 6, pages 508—-510. Elsevier, Amsterdam, Netherlands, 2006.

Charles M. Kastner, G. Adam Covington, Andrew A. Levine, and John W. Lockwood. Hail: A Hardware-Accelerated Algorithm for Language Identification. In Tero Rissa, Steve Wilton, and Philip Leong, editors, *Proceedings of the 2005 International Conference on Field Programmable Logic and Applications (FPL)*, pages 499–504, Tampere, Finland, 2005.

Ben King and Steven Abney. Labeling the Languages of Words in Mixed-Language Documents using Weakly Supervised Methods. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1110–1119, Atlanta, USA, June 2013.

Ben King, Dragomir Radev, and Steven Abney. Experiments in Sentence Language Identification with Groups of Similar Languages. In *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, pages 146–154, Dublin, Ireland, 2014.

Josh King and Jon Dehdari. An N-gram Based Language Identification System. The Ohio State University, 2008.

Levi King, Sandra Kübler, and Wallace Hooper. Word-level language identification in The Chymistry of Isaac Newton. *Digital Scholarship in the Humanities*, 30(4): 532–540, 2015.

Tom Kocmi and Ondřej Bojar. LanideNN: Multilingual Language Identification on Character Window. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Long Papers*, volume 1, pages 927–936, Valencia, Spain, 2017.

Dijana Kosmajac and Vlado Keselj. Slavic Language Identification using Cascade Classifier approach. In *Proceedings of the 17th International Symposium INFOTEH-JAHORINA (INFOTEH 2018)*, East Sarajevo, Bosnia-Herzegovina, 2018. IEEE.

Sotiris B. Kotsiantis. Supervised Machine Learning: A Review of Classification Techniques. *Informatica*, 31:249–268, 2007.

Yitong Li, Trevor Cohn, and Timothy Baldwin. What's in a domain? learning domain-robust text representations using adversarial training. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics — Human Language Technologies (NAACL HLT 2018)*, pages 474–479, New Orleans, USA, 2018.

Nikola Ljubešić and Denis Kranjcić. Discriminating between VERY Similar Languages among Twitter Users. In *Proceedings of the 9th Language Technologies Conference*, pages 90–94, Ljubljana, Slovenia, 2014.

Nikola Ljubešić and Denis Kranjcić. Discriminating Between Closely Related Languages on Twitter. *Informatica*, 39, 2015.

Nikola Ljubešić and Antonio Toral. caWaC - a Web Corpus of Catalan and its Application to Language Modeling and Machine Translation. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014)*, pages 1728–1732, Reykjavik, Iceland, 2014.

Nikola Ljubešić, Nives Mikelić, and Damir Boras. Language Identification: How to Distinguish Similar Languages? In *Proceedings of the 29th International Conference on Information Technology Interfaces (ITI 2007)*, pages 541–546, Cavtat/Dubrovnik, Croatia, 2007.

Ariadna Font Llitjós. Improving Pronunciation Accuracy of Proper Names with Language Origin Classes. Master's thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2001.

Ariadna Font Llitjós. Improving Pronunciation Accuracy of Proper Names with Language Origin Classes. In Malvina Nissim, editor, *Proceedings of the Seventh ESSLLI Student Session*, pages 53–67, Trento, Italy, August 2002.

Ariadna Font Llitjós and Alan W. Black. Knowledge of Language Origin Improves Pronunciation Accuracy of Proper Names. In Paul Dalsgaard, B. Lindberg, Henrik Benner, and Zheng-Hua Tan, editors, *Proceedings of the 7th European Conference on Speech Communication and Technology, 2nd INTERSPEECH Event (EUROSPEECH 2001 Scandinavia)*, pages 1919–1922, Aalborg, Denmark, 2001.

Yevgeny Ludovik and Ron Zacharski. Multilingual Document Language Recognition for Creating Corpora. Technical report, New Mexico State University, 1999.

Marco Lui. *Generalized Language Identification.* PhD thesis, The University of Melbourne, 2014.

Marco Lui and Timothy Baldwin. Cross-domain Feature Selection for Language Identification. In *Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP 2011)*, volume 1, pages 553–561, Chiang Mai, Thailand, 2011.

Marco Lui, Jey Han Lau, and Timothy Baldwin. Automatic Detection and Language Identification of Multilingual Documents. *Transactions of the Association for Computational Linguistics*, 2:27–40, 2014. ISSN 2307-387X.

Wolfgang Maier and Carlos Gómez-Rodríguez. Language Variety Identification in Spanish Tweets. In *Proceedings of the EMNLP'2014 Workshop on Language Technology for Closely Related Languages and Language Variants (LT4CloseLang 2014)*, pages 25–35, Doha, Qatar, October 2014. Association for Computational Linguistics.

Martin Majliš. Large Multilingual Corpus. Master's thesis, Charles University in Prague, Prague, 2011.

Martin Majliš. Yet Another Language Identifier. In *Proceedings of the Student Research Workshop at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 46–54, Avignon, France, 2012.

Martin Majliš and Zdeněk Žabokrský. Language Richness of the Web. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, pages 2927–2934, Istanbul, Turkey, 2012.

Shervin Malmasi and Mark Dras. Automatic Language Identification for Persian and Dari Texts. In *Proceedings of the 14th Conference of the Pacific Association for Computational Linguistics, PACLING'15*, pages 59–64, Bali, Indonesia, 2015a.

Shervin Malmasi and Mark Dras. Language Identification using Classifier Ensembles. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects, LT4VarDial'15*, pages 35–43, Hissar, Bulgaria, 2015b.

Shervin Malmasi and Mark Dras. Feature Hashing for Language and Dialect Identification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 399–403, Vancouver, Canada, 2017.

Shervin Malmasi and Marcos Zampieri. Arabic Dialect Identification in Speech Transcripts. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 106–113, Osaka, Japan, 2016.

Shervin Malmasi, Eshrag Refaee, and Mark Dras. Arabic Dialect Identification using a Parallel Multidialectal Corpus. In *Proceedings of the 14th Conference of the Pacific Association for Computational Linguistics, PACLING'15*, pages 209–217, Bali, Indonesia, 2015.

Shervin Malmasi, Marcos Zampieri, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, and Jörg Tiedemann. Discriminating Between Similar Languages and Arabic Dialect Identification: A Report on the Third DSL Shared Task. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 1–14, Osaka, Japan, 2016.

Thomas Mandl, Margaryta Shramko, Olga Tartakovski, and Christa Womser-Hacker. Language Identification in Multi-lingual Web-Documents. In *Proceedings of the 11th International Conference on Applications of Natural Language to Information Systems (NLDB 2006)*, pages 153–163, Klagenfurt, Austria, 2006.

Puji Martadinata, Bayu Distiawan Trisedya, Hisar Maruli Manurung, and Mirna Adriani. Building Indonesian Local Language Detection Tools Using Wikipedia Data. In Yohei Murakami and Donghui Lin, editors, *Worldwide Language Service Infrastructure*, pages 113–123. Springer, 2016.

Deepthi Mave, Suraj Maharjan, and Thamar Solorio. Language identification and analysis of code-switched social media text. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 51–61. Association for Computational Linguistics, 2018. URL `http://aclweb.org/anthology/W18-3206`.

Paul McNamee. Language Identification: A Solved Problem Suitable for Undergraduate Instruction. *Journal of Computing Sciences in Colleges*, 20(3):94–101, 2005.

Paul McNamee. Language and Dialect Discrimination Using Compression-Inspired Language Models. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 195–203, Osaka, Japan, 2016.

Maria Medvedeva, Martin Kroon, and Barbara Plank. When Sparse Traditional Models Outperform Dense Neural Networks: the Curious Case of Discriminating between Similar Languages. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 156–163, Valencia, Spain, 2017.

Aila Mielikäinen. Liudennus murretutkimuksessa ja savolaismurteisessa kirjallisuudessa. *Virittäjä*, 108(4):508–530, 2004.

Gordon Mohr, Michael Stack, Igor Rnitovic, Dan Avery, and Michele Kimpton. Introduction to Heritrix. In *4th International Web Archiving Workshop*, Bath, UK, 2004.

Avashlin Moodley. Language Identification With Decision Trees: Identification Of Individual Words In The South African Languages. Bachelor's Thesis, University of South Africa, 2016.

Kavi Narayana Murthy and G. Bharadwaja Kumar. Language Identification from Small Text Samples. *Journal of Quantitative Linguistics*, 13(1):57–80, January 2006.

Yeshwant K Muthusamy and A Lawrence Spitz. Automatic Language Identification. In Ronald Cole, Joseph Mariani, Hans Uszkoreit, Giovanni Battista Varile, Annie Zaenen, Antonio Zampolli, and Victor Zue, editors, *Web Edition: Survey of the State of the Art in Human Language Technology*, pages 314–317. Cambridge University Press, Cambridge, UK, 1997.

Hermann Ney, Ute Essen, and Reinhard Kneser. On structuring probabilistic dependences in stochastic language modelling. *Computer Speech and Language*, 8(1): 1–38, 1994.

Andrew Y. Ng and Michael I. Jordan. On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Advances in Neural Information Processing Systems 15 (NIPS 2002)*, pages 841–848, Vancouver, British Columbia, Canada, 2002.

Dong Nguyen and A. Seza Dogruöz. Word Level Language Identification in Online Multilingual Communication. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 857–862, Seattle, USA, 2013.

Douglas W. Oard, Fabrizio Sebastiani, Jonathan Furner, and Gary Marchionini. Publishing Survey Articles on Information Retrieval Topics. *ACM SIGIR Forum*, 45 (1):70–72, 2011.

Ziad Obermeyer and Ezekiel J. Emanuel. Predicting the future - big data, machine learning, and clinical medicine. *The New England journal of medicine*, 375:1216–1219, 2016.

Gorkem Ozbek, Itamar Rosenn, and Eric Yeh. Language Classification in Multilingual Documents. Technical report, Stanford University, 2006.

Shraddha Patel and Vaibhavi Desai. LIGA and Syllabification Approach for Language Identification and Back Transliteration: A Shared Task Report by DA-IICT. In *FIRE '14 Proceedings of the Forum for Information Retrieval Evaluation*, pages 43–47, Bangalore, India, 2014.

Irina Piippo, Johanna Vaattovaara, and Eero Voutilainen. Kieli, tuo viekas seuralainen. *Puhe ja kieli*, 37(1):43–48, 2017.

Stelios Piperidis. The META-SHARE Language Resources Sharing Infrastructure: Principles, Challenges, Solutions. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'2012)*, pages 36–42, Istanbul, Turkey, 2012.

Arjen Poutsma. Applying Monte Carlo Techniques to Language Identification. *Language and Computers*, 45(1):179–189, 2002.

John M. Prager. Linguini: Language Identification for Multilingual Documents. In *Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences (HICSS-32)*, Maui, USA, 1999.

M. A. Nejla Qafmolla. Automatic Language Identification. *European Journal of Language and Literature Studies*, 7(1):140–150, 2017.

Yan Qu and Gregory Grefenstette. Finding Ideographic Representations of Japanese Names Written in Latin Script via Language Identification and Corpus Validation. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004)*, pages 183–190, Barcelona, Spain, 2004.

Uwe Quasthoff, Matthias Richter, and Christian Biemann. Corpus Portal for Search in Monolingual Corpora. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, pages 1799–1802, Genoa, 2006.

Bali Ranaivo-Malançon. Automatic Identification of Close Languages – Case study: Malay and Indonesian. *ECTI Transactions on Computer and Information Technology*, 2(2):126–134, 2006.

Paul Rodrigues. *Processing Highly Variant Language Using Incremental Model Selection.* PhD thesis, Indiana University, 2012.

Y. Dan Rubinstein and Trevor Hastie. Discriminative vs Informative Learning. In David Heckerman, Heikki Mannila, Daryl Pregilbon, and Ramasamy Uthurusamy, editors, *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97)*, pages 49–53, Newport Beach, California, USA, 1997.

Kevin P Scannell. The Crúbadán Project: Corpus Building for Under-resourced Languages. In *Building and Exploring Web Corpora: Proceedings of the 3rd Web as Corpus Workshop*, pages 5–15, Louvain-la-Neuve, Belgium, 2007.

Greg Schohn and David Cohn. Less is More: Active Learning with Support Vector Machines. In Pat Langley, editor, *Proceedings of the Seventeenth International Conference on Machine Learning (ICML '00)*, pages 839–846, Stanford, CA, USA, 2000.

Fabrizio Sebastiani. Machine Learning in Automated Text Categorization. *ACM Computing Surveys (CSUR)*, 34(1):1–47, 2002.

H. L. Shashirekha. Automatic Language Identification from Written Texts - An Overview. *International Journal of Innovative Research in Computer and Communication Engineering*, 2(5):156–160, 2014.

Penelope Sibun and Jeffrey C. Reynar. Language Identification: Examining the Issues. In *Proceedings of the 5th Annual Symposium on Document Analysis and Information Retrieval (SDAIR-96)*, pages 125–135, Las Vegas, USA, 1996.

Gary F. Simons and Charles D. Fennig, editors. *Ethnologue: Languages of the World, Twenty-first edition.* SIL International, Dallas, Texas, 2018. Online version: http://www.ethnologue.com.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Gohneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, and Pascale Fung. Overview for the First Shared Task on Language Identification in Code-Switched Data. In *Proceedings of The First Workshop on Computational Approaches to Code Switching*, pages 62–72, Doha, Qatar, October 2014. URL http://www.aclweb.org/anthology/W14-3907.

Clive Souter, Gavin Churcher, Judith Hayes, John Hughes, and Stephen Johnson. Natural Language Identification using Corpus-Based Models. *Hermes, Journal of Linguistics*, 13:183–203, 1994.

Liling Tan, Marcos Zampieri, Nikola Ljubešić, and Jörg Tiedemann. Merging Comparable Data Sources for the Discrimination of Similar Languages: The DSL Corpus Collection. In *Proceedings of the 7th Workshop on Building and Using Comparable Corpora (BUCC)*, Reykjavik, Iceland, 2014.

Jörg Tiedemann and Nikola Ljubešić. Efficient Discrimination Between Closely Related Languages. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, pages 2619–2634, Mumbai, India, 2012.

Katrin Tomanek, Joachim Wermter, and Udo Hahn. An Approach to Text Corpus Construction which Cuts Annotations Costs and Maintains Reusability of Annotated Data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 486–495, Prague, Czech Republic, 2007.

Erik Tromp. Multilingual Sentiment Analysis on Social Media. Master's thesis, Eindhoven University of Technology, Eindhoven, 2011.

Erik Tromp and Mykola Pechenizkiy. Graph-Based N-gram Language Identification on Short Texts. In *Proceedings of the 20th Annual Belgian Dutch Conference on Machine Learning (Benelearn 2011)*, pages 27–34, The Hague, Netherlands, 2011.

Dan Tufis and Elena Irimia. RoCo_News-A Hand Validated Journalistic Corpus of Romanian. In *Proceedings of the 5th LREC Conference*, pages 869–872, Genoa, Italy, 2006.

Tommi Vatanen, Jaakko J. Väyrynen, and Sami Virpioja. Language Identification of Short Text Segments with N-gram Models. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, pages 3423–3430, Valletta, Malta, 2010.

Tony Vitale. An Algorithm for High Accuracy Name Pronunciation by Parametric Speech Synthesizer. *Computational Linguistics*, 17(3):257–276, 1991.

John Vogel and David Tresner-Kirsch. Robust Language Identification in Short, Noisy Texts: Improvements to LIGA. In Martin Atzmueller and Hotho Andreas, editors, *Proceedings of the 3rd International Workshop on Mining Ubiquitous and Social Environments (MUSE)*, pages 43–50, Bristol, UK, 2012.

Marlies van der Wees, Arianna Bisazza, Wouter Weekamp, and Christof Monz. What's in a Domain? Analyzing Genre and Topic Differences in Statistical Machine Translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*, pages 560–566, Beijing, China, 2015.

Fei Xia, William D. Lewis, and Hoifung Poon. 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 (EACL2009)*, pages 870–878, Athens, Greece, 2009.

Hiroshi Yamaguchi and Kumiko Tanaka-Ishii. Text Segmentation by Language Using Minimum Description Length. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012)*, pages 969–978, Jeju Island, Korea, July 2012.

Noson S. Yanofsky. Towards a Definition of an Algorithm. *Journal of Logic and Computation*, 21(2):253–286, 2011.

Omar F. Zaidan and Chris Callison-Burch. Arabic Dialect Identification. *Computational Linguistics*, 40(1):171–202, 2014.

Marcos Zampieri. Using Bag-of-words to Distinguish Similar Languages: How Efficient Are They? In *Proceedings of the 2013 IEEE 14th International Symposium on Computational Intelligence and Informatics (CINTI)*, pages 37–41, Budapest, Hungary, 2013.

Marcos Zampieri. Automatic Language Identification. In *Working with Text: Tools, Techniques and Approaches for Text Mining*, chapter 8, pages 189–205. Elsevier, 2016.

Marcos Zampieri and Binyam Gebrekidan Gebre. Automatic Identification of Language Varieties: The Case of Portuguese. In *Proceedings of The 11th Conference on Natural Language Processing (KONVENS 2012)*, pages 233–237, Vienna, Austria, 2012.

Marcos Zampieri, Binyam Gebrekidan Gebre, and Sascha Diwersy. Classifying Pluricentric Languages: Extending the Monolingual Model. In *Proceedings of the Fourth Swedish Language Technology Conference (SLTC)*, pages 79–80, Lund, Sweden, 2012.

Marcos Zampieri, Binyam Gebrekidan Gebre, and Sascha Diwersy. N-gram Language Models and POS Distribution for the Identification of Spanish Varieties. In *Proceedings of la 20ème conférence du Traitement Automatique du Langage Naturel (TALN)*, pages 580–587, Sables d'Olonne, France, 2013.

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

Marcos Zampieri, Binyam Gebrekidan Gebre, Hernani Costa, and Josef van Genabith. Comparing Approaches to the Identification of Similar Languages. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects (LT4VarDial)*, pages 66–72, Hissar, Bulgaria, 2015a.

Marcos Zampieri, Liling Tan, Nikola Ljubešić, Jörg Tiedemann, and Preslav Nakov. Overview of the DSL Shared Task 2015. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects (LT4VarDial)*, pages 1–9, Hissar, Bulgaria, 2015b.

Marcos Zampieri, Shervin Malmasi, Nikola Ljubešic, Preslav Nakov, Ahmed Ali, Jörg Tiedemann, Yves Scherrer, and Noëmi Aepli. Findings of the VarDial Evaluation Campaign 2017. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 1–15, Valencia, Spain, 2017.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Ahmed Ali, Suwon Shon, James Glass, Yves Scherrer, Tanja Samardžić, Nikola Ljubešić, Jörg Tiedemann, Chris van der Lee, Stefan Grondelaers, Nelleke Oostdijk, Antal van den Bosch, Ritesh Kumar, Bornini Lahiri, and Mayank Jain. Language Identification and Morphosyntactic Tagging: The Second VarDial Evaluation Campaign. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 1–17, Santa Fe, USA, 2018.

Marcos Zampieri, Shervin Malmasi, Yves Scherrer, Tanja Samardžić, Francis Tyers, Miikka Silfverberg, Natalia Klyueva, Tung-Le Pan, Chu-Ren Huang, Radu Tudor Ionescu, Andrei Butnaru, and Tommi Jauhiainen. A Report on the Third VarDial Evaluation Campaign. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, Minneapolis, USA, 2019. Association for Computational Linguistics.

Ayah Zirikly, Bart Desmet, and Mona Diab. The GW/LT3 VarDial 2016 Shared Task System for Dialects and Similar Languages Detection. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 33–41, Osaka, Japan, 2016.

Nellejet Zorgdrager. The Role of Place-Names in Olof Sirma's two Yoik Texts and their Translations. *Journal of Northern Studies*, 11(1):71–93, 2017.