Data-driven Language Typology

Atte Hinkka

Master's thesis
UNIVERSITY OF HELSINKI
Department of Computer Science

Helsinki, November 27, 2017

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty Faculty of Science	Laitos — Institution — Department Department of Computer Science

Tekijä — Författare — Author
Atte Hinkka

Työn nimi — Arbetets titel — Title
Data-driven Language Typology

Oppiaine — Läroämne — Subject
Computer Science

Työn laji — Arbetets art — Level Master's thesis	Aika — Datum — Month and year November 27, 2017	Sivumäärä — Sidoantal — Number of pages 56

Tiivistelmä — Referat — Abstract

In this thesis we use statistical n-gram language models and the perplexity measure for language typology tasks. We interpret the perplexity of a language model as a distance measure when the model is applied on a phonetic transcript of a language the model wasn't originally trained on. We use these distance measures for detecting language families, detecting closely related languages, and for language family tree reproduction. We also study the sample sizes required to train the language models and make estimations on how large corpora are needed for the successful use of these methods.

We find that trigram language models trained from automatically transcribed phonetic transcripts and the perplexity measure can be used for both detecting language families and for detecting closely related languages.

ACM Computing Classification System (CCS):

- **Information systems → Language models;**
- **Computing methodologies → Natural language processing;**

Avainsanat — Nyckelord — Keywords
language typology, language models, phonetic transcriptions

Säilytyspaikka — Förvaringsställe — Where deposited
Kumpula Science Library

Muita tietoja — Övriga uppgifter — Additional information

Acknowledgements

I thank my thesis supervisor, professor Hannu Toivonen, for all the collaboration over the years, his valuable teaching on how to think scientifically, and especially for his supervision of and contribution to this thesis. I thank Dr. Mark Granroth-Wilding for his supervision and camaraderie in the work within the DLT project. Thirdly I thank Leo Leppänen for his valuable feedback on this thesis. I'm also indebted to the rest of the Discovery group where this work was conducted in for all their support. Thanks!

My journey in the world of science started in the now-defunct Biomine project, which is still, in a sense, my scientific home. For the enduring friendship and support I thank the pundits and today's cycling buddies Lauri Eronen, Petteri Hintsanen, and Kimmo Kulovesi.

My work on the Digital Language Typology project, and the research for this thesis was funded by the Academy of Finland.

Lastly I thank my friends and family for all their support during the writing of this thesis.

–A, November 2017

Contents

1 Introduction

Language typology is an approach to the study of languages that has three different goals. It is interested in how languages can be categorized, what features are common between languages, and why are these features common [Cro03, p. 1–2]. Typology studies are commonly centered around somewhat small details in language such as studying the different modern language realizations of an old word, or comparing how a grammatical case behaves within a group of related languages. The detail, or facet could be within the grammar or in the morphology, or some other part of the language. The process often starts from analysis of these details and then leads onto generalizations within or across different languages to come up with classifications and taxonomies.

In this thesis we propose an automatic method for language typology that is applicable to speech corpora and doesn't require human preparation of data. We develop methods that rely on *language similarity* to make hypotheses on how the languages are related. Our methods are based on statistical *language models*, which we use to model language surface features. For us, in this case, the surface is represented by *phonetic transcriptions*. Our approach is based on the idea of interpreting a language model's performance on another language as a similarity measure between the two languages. Our research hypothesis is that the better a language model trained on one language performs on another language, the more similar the languages are, and that a high similarity between two languages is an indicator that the languages may be related. In this thesis the term *distance* is used in place of dissimilarity, conceptually the inverse of similarity, when discussing the similarity of languages.

Our main research subject, phonetic transcriptions, are symbolic descriptions of speech. More specifically to this thesis, they are automatically transcribed texts from the Europarl parallel translation corpus [Koe05], which is a collection of parliament speech transcripts of the European Parliament. We use an automatic tool to transcribe the text corpus to a phonetic transcription corpus. Our chosen modeling tool, statistical language models, are probability distributions over sequences of words, which are used to predict the next word given previous words [MS99, p. 191]. Instead of words, we model language on the level of phonemes, because we want to focus on the very surface of speech, on features that are found within words, within syllables. We chose to use statistical language modeling because it provides an intuitive way of thinking in terms sequences of items instead of relying on a wide range of language specific language resources such as part of speech taggers, morphological analyzers, and treebanks.

Data-driven, or *corpus-driven* means that an approach is based on the idea that the data itself should be the sole source of hypotheses [MH12, p. 6]. This is in contrast to computational linguistics methods that are built for proving pre-existing hypotheses. In addition to not centering on pre-existing hypotheses, the fact that our methods don't use external language resources makes them more data-driven than methods that would utilize such resources as the resources would direct the methods. While our methods don't rely on pre-existing hypotheses, they can still be used to test pre-existing hypotheses. For example, we may have ideas of how a set of languages are related to each other, and the distance information extracted by our methods could be used to back up a family relationship hypothesis.

We don't study morphology, grammar, geographical or social aspects of language. We only hypothesize language relationships on the basis of distance measures. Naturally our methods can be used for e.g. sociolinguistics or dialectological research given applicable corpora. Of established corpus research methods, closest to ours is *lexicostatistics*, which is the statistical study of variation of words in corpora [Emb00]. Lexicostatistics is a good method for analyzing written corpora, but its applicability to inter-language studies is questionable. For further discussion on lexicostatistics, see section 2.1.

An unsolved problem in the use of phonetic transcriptions in inter-language comparisons such as ours is the differing meaning of phonetic symbols between languages. This manifests itself in multiple ways. It may be that for the same sound a similar, but a little bit different symbol is used in another language (or the transcription of that language). It may also be that the phonology of languages differs so that two or more symbols are used in one language instead of just one in the other, i.e. the phonology of the former language is more complex with regards to this sound than in the latter language. Some of these differences may be genuine differences between languages, which should be accounted for, but some others may only be accidental differences caused by the chosen transcription policy. We call this problem the *common phoneme set problem.* For further background on the problem, Port and Leary [PL05] go through the issues with phonology as a formal system, and the issue is formulated in a familiar setting to this thesis by Ellison and Kirby [EK06].

Language models have been used to calculate language distances before, but these studies have been done on text. We use phonetic transcriptions instead of text and believe the approach is an improvement on the earlier attempts because we don't rely on any ad hoc *normalization* methods. In earlier studies the text has commonly been normalized to some kind of a simpler Latin alphabet [BPK92, Kit99, GPA17] by removing at least some letter diacritics to improve language modeling performance. The motivation behind this is the assumption that removing diacritics makes the languages comparable and doesn't significantly change the meaning of the letters in terms of phonology. This may make sense in cases where the identity of the letter does not change when diacritics are removed, but it presents a new set of problems when the identity of the letter is tied to the diacritic. An example of a potentially successful normalization is the case of *acute* in Portuguese where it signifies word stress (a-á), and hence the normalization only removes the stress. Conversely, there are cases like the use of *diaeresis* in Finnish where the letter with a diacritic stands for a whole another letter and sound altogether (a vs ä). Normalization is more thoroughly discussed in Section 2.

This thesis work was done within the *Digital Language Typology* project[1], where the aim is to produce computational methods for language typology. The main analysis target of the project is in the analysis of speech sequences, focusing on *prosody* and other phonetic sequences. The project's target languages *Tundra and Forest Nenets*, *Nganasan* and *North Sami*. These languages have been studied before, but they don't have as wide a range of external resources as the more studied, more widely spoken languages most efforts in computational linguistics have focused on. Both the low-resourced target languages and the focus on speech were motivators for our chosen methods. In addition, given that half of the world's languages are likely to become extinct within the next century [Kra92], languages that commonly don't have the language resources available, the development of methods that are viable on these languages is more current than ever.

[1] Program brochure available at `http://www.aka.fi/globalassets/32akatemiaohjelmat/digihum/hanke-esitteet/vainio-digihum.pdf`

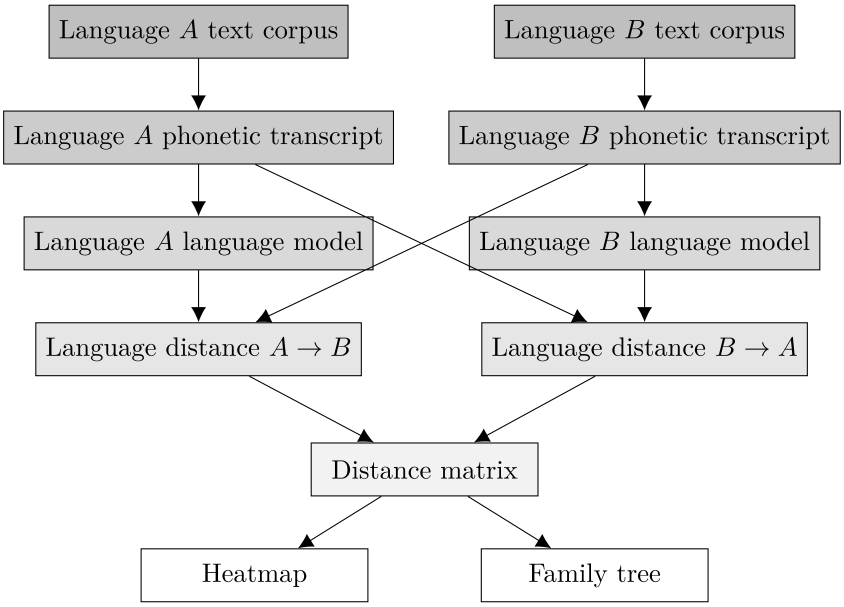

Language A text corpus
Language B text corpus
Language A phonetic transcript
Language B phonetic transcript
Language A language model
Language B language model
Language distance $A \rightarrow B$
Language distance $B \rightarrow A$
Distance matrix
Heatmap
Family tree

Figure 1: Analysis pipeline describing corpora preprocessing, model training, computation of distances using foreign corpora and visualization.

1.1 Computational Language Typology

We want to develop methods for computational language typology on speech. We take a step further from analyzing the distribution of phonemes in a language. We utilize methods and concepts from speech recognition and language identification to compute distance measures between languages. Previously language typology has relied mostly on human work, manual analysis of language features such as grammar or morphology, and then generalizing, classifying and finding universals based on the findings.

We approach typology from the point of view of language similarity. To measure similarity, we use a performance measure of language models. Our hypothesis is that a language model's *perplexity* measure can be interpreted as a similarity measure between languages when the training language of a language model and the corpus being measured are from different languages (see Figure 1). Perplexity, a measure of surprisedness of a language model is defined in Section 3.3. As we want to find out how our method works on speech, we use automatic phonetic transcriptions of text corpora of different languages and model them on the level of phonemes instead of on the level of characters or words.

The main reason for choosing to work with statistical language models is their applicability to generic sequences of symbols and independence from supporting language resources. In other words, limiting ourselves to language models enables us to work on languages that don't have a wide variety of resources available. The ability to handle generic symbol sequences also makes it possible to handle representations of speech other than the standard IPA notation [Int99] phonetic transcriptions used in this thesis.

We use statistical language models, more specifically *n-gram models*. Phonetic transcriptions are treated as abstract symbol sequences where each phoneme or phone is represented by a symbol. N-gram models are language models that have a probability assigned for each sequence of n symbols, i.e. they use a context of $n-1$ to predict the next symbol. Using these probabilities, a perplexity score describing how surprised the language model is is computed between the model and a foreign corpus.

Language similarity as understood in this thesis does not provide direct answers to why languages are similar. This is in contrast to a traditional language typology approach, which is based on the conceptualization of language features and basing the actual relationship analyses on these feature similarities.

Further, to analyze the distances in terms of how they correspond to known language classifications, we use a variety of visualizations and other analysis methods. Statistical analysis gives us a view of how the distances are distributed, and heatmaps are useful for seeing language family patterns. In addition, family trees describe the distances in a more data-driven fashion, not predefined by our own hypotheses.

Original work in this thesis is the use of language models on phonetic transcriptions. Information-theoretic measures have been used before by Kita et al [Kit99] and by Gamallo et al [GPA17], but both work on text. Rama and Singh [RS09] use a Brahmi script for Indian languages that has *"almost one-to-one correspondence between letters and phonemes"*, but it isn't a phonetic writing system in the same sense IPA is.

Given a speech sequence, a phonetic transcription, we model it using language models and use a commonly used performance measure, perplexity, to measure the degree of surprise of a model when applied to a speech sequence from a foreign language. We study the model's learning constraints to gauge whether the methods could be applicable to languages with few and small corpora. We compute language distances to detect known language families and test whether we can detect closely related languages. We also run tests to determine how large learning and test samples our methods require. The methods seem to work reasonably well to for both detecting language families and for detecting closely related languages, although the differing phoneme sets between languages pose a problem to our method. The methods are not applicable to very small corpora, but don't require big data corpora either.

The rest of this thesis is structured as follows: Section 2 provides look into how things are done without computers and then goes through similar and related computational methods for producing language family trees. In section 3 we formally define the methods used in the experiments. Section 4 describes how the methods work with different training and test data sizes, and probe how the methods work on language family detection and on detecting closely related languages. Section 5 concludes the thesis with an overview of the work, and how this line of study should be continued.

2 Background

Comparative linguistics is a branch of linguistics that studies whether languages are related by comparing them [Ant89, p. 20, 310–321]. Objective of these studies is often to come up with a hypothesis on how the languages are related and how they developed over time from other languages. What language family is a language part of? Are some languages clearly exceptionally closely related while between some other languages systematic similarities don't seem to exist? In historical linguistics language families are often represented with family trees while dialect studies often use maps to describe and highlight the geographical aspects of language variation.

The *comparative method* is based on the comparison of sets of features between languages [Leh93, p. 31]. This comparison often results in hypotheses on how the language has developed itself over time, how it might have diverged from another earlier language, and what are the systematic similarities between these languages. In addition to knowing how the languages are related, there is often also interest in the timing, into when the languages diverged. Originally the comparative method has been work of well-read scholars, but lately there has been interest in computational solutions to the same problems [McM03]. One can also look at family tree production from a purely computer science perspective as the *language tree reconstruction* problem, which frames the problem as reproduction of a manual method as computational, not as producing new information about the research subject. While one could frame the work in this thesis as such, the aim is not only to reproduce, but to come up with a method that works without the language resources previous methods need and is generalizable to use cases beyond text.

A central differentiator between different computational typology methods is what they use as the distance and/or timing measure. *Glottochronology* is based on *morpheme decay*, which is a theory inspired by radioactivity. *Evolutionary methods* are based on finding the most likely hypothesis for divergence and divergence times. *Edit distance* is based on the number of edit operations needed to modify a sequence to another. Information theoretic methods, such as the one employed in this thesis are based on *entropy*. These methods will be discussed in detail in the following subsections.

Another differentiator are the inputs, which are most commonly either

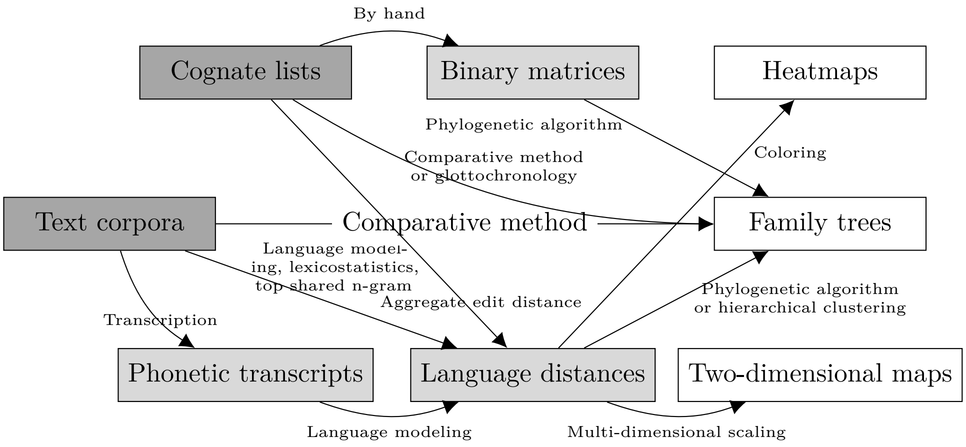


Figure 2: Overview of related methods and what kind of inputs are needed.

text corpora or hand-built cognate lists. Furthermore, the results of these methods can be represented visually in a number of ways, not just as a family tree. See Figure 2 for an overview.

2.1 Cognate Methods

A gateway between traditional manual comparative linguistics and computational methods are *cognates*. Cognates are words that a language has retained over history, and shares with its sibling languages and predecessors. When looking at a family of languages and their shared cognates, the cognates are said to have a common etymological origin. It is commonly understood that every cognate has developed gradually from a proto-form, in different languages in different ways, meaning their later forms may differ significantly. An example of a cognate is the English word *night*, which has counterparts in other Indo-European languages: *Nacht* (German), *natt* (Swedish), *noc* (Polish), and *noche* (Spanish). These all have origin in Proto-Indo-European word *nókʷts [Wik17a]. Another word category of importance are *loan words*, which are as the name suggests, loaned by some other mechanism, often through contact to another language.

Swadesh word lists [Swa50] are commonly used lists in cognate studies which collect universal concepts together to form a dictionary of sorts, which can be used for comparative study of languages based on shared vocabulary. These word lists sparked a long string of studies looking at language similarities by comparing the surface representation of cognates

English	**I**	**you** (singular)	**this**	**who**	**what**
German	Ich	du	dieses	wer	was
Dutch	ik	jij / je	deze / dit	wie	wat
Danish	jeg	du	denne / dette	hvem	hvad
Swedish	jag	du	denna, den här / detta det här	vem	vad

Table 1: Example five word Swadesh list [Wik17c, Wik17b].

between languages and analyzing their change over time. Cognate studies are based on the idea that language families share common concepts, which are represented by the same or similar words. The higher the number of shared words or the similar the words, the more similar the languages are expected to be. Table 1 describes a very simplified Swadesh list for a limited set of Germanic languages. It's quite evident Swedish and Danish are quite similar according to this list. German and Dutch form a pair as well, while English shares less words with the others when one only looks at this table of spellings superficially.

Of cognate methods, *lexicostatistics* studies the proportion of shared words between languages and treats the proportion of shared words as a similarity measure. *Glottochronology* attempts to construct a family, or a divergence tree of languages using lexicostatistics and models language change as divergence from an original proto-language. Glottochronology is based on the concept of *language decay*, which is strongly inspired by the decay mechanisms in radiocarbon timing methods used in archeology [Swa52]. Basic estimation formula in glottochronology is $t = \frac{log(c)}{2log(r)}$ where t is time, c is the percentage of shared cognates and r is the retention rate per thousand years [BFG05]. Glottochronology is considered by many historical linguists to be a problematic method due to two main reasons: the assumption that there is a universal set of words is flawed, and that the basic tenet of a constant rate of word retention through time has been proven to not hold universally [Cam98, p. 180–184]. The criticism towards universal word lists, or Swadesh lists holds for other cognate- and word list based methods as well.

While early scholars used morpheme decay, a term adapted from radioactivity as a metaphor and mechanism for modeling language change, in recent years the focus has turned to computational biology and evolution, and

Language	*foot* (Eng.)
English	foot
German	Fuß
Finnish	jalka
Estonian	jalg

Language	Feature #1	Feature #2
English	1	0
German	1	0
Finnish	0	1
Estonian	0	1

Table 2: An example cognate list and a corresponding binary matrix encoding two cognates as distinct features: Germanic foot (#1) and a Finnic foot (#2).

especially to *phylogenetics*, the study of evolutionary relationships between biological entities [LSV09, p. 16–18]. The hypothesis behind using phylogenetic methods is that language change can be thought of as an evolutionary process and hence methods for analyzing natural evolutionary processes would also work for language study. Evolutionary methods are based on coming up with hypotheses for the order of branchings in a family tree, and estimating the time between branchings. Depending on the method, the heuristic may be maximum likelihood based, i.e. coming up with the likeliest branching hypothesis, or in the case of Bayesian methods a similar heuristic, but based on Bayesian statistics. Evolutionary methods rely on human-coded binary matrices that map shared cognates between languages. To use these algorithms, cognates or other shared features need to be coded into separate features. In Table 2 cognate words are coded into two features. In biology these binary matrices would either code shared *phenotypes* (observable traits), or shared parts of the genome such as individual nucleotides. Lehtinen's master's thesis [Leh09] contains a good overview on the subject of using evolutionary modeling methods on language.

2.2 Edit Distance

Edit distance [Kru83] (in most cases this is synonymous to Levenshtein distance [Lev66]) has been used in computational linguistics to compare words. A common method is take a list of words as the representation of a language and compute edit distance between each word pair. An aggregate language distance can be calculated from these word-pair distances by either summing, averaging or otherwise combining them. The method has been used successfully for dialect studies [NH97], and also for reproducing family trees [PS08]. An advantage of using edit distance is that one doesn't need to construct binary matrices by hand. These methods have been tried and tested for text and certain limited phonetic codings, but generally phonetic comparison beyond symbol comparisons is a difficult problem [Kes05].

2.3 Language Modeling

Language models have been used for a number of purposes: *speech recognition* [JBM75], *language identification* [LE80], *information retrieval* [PC98], but also for reproducing language trees [Kit99]. While one could generalize all statistical approaches to language as a kind of language modeling, we focus on statistical language models as they are commonly understood in the NLP community. We focus on models that are used for estimating the probability of the next item, or predicting the next item given certain number of preceding items.

The use of language models consists of two distinct phases: *learning phase*, and *test phase*. In learning phase the language model is trained by feeding it linguistic sequences from the training corpus. The model keeps track of the sequences encountered and when the training is completed, the model's *parameters* are estimated using an estimation method. These parameters can be thought of as the end result of learning. In addition to parameters there are *hyperparameters* associated with a language model. Hyperparameters control how the language model estimates its actual parameters. In practice hyperparameters are either given from the outside by the programmer or found by a process of optimization where multiple different hyperparameter values are tried and the best given some performance measure is chosen. The test phase contains everything the language model is actually used for. We use the models only for perplexity computation, but an example of another use case would be the prediction of next items given a context.

A central term to language models is *vocabulary*. In a traditional word-based language model, vocabulary is the set of words the language model was trained with. In a phoneme-based language model the vocabulary is the set of phonemes in the transcription. *Vocabulary normalization* is a commonly used method to improve the performance of language models. In the case of character-based language modeling, normalization has been more specifically motivated by practical issues such as different scripts, information theory [Kit99], and phonetics [GPA17]. Given the variety of scripts used across languages and the multitude of character encodings to further complicate the landscape, normalization can be seen as a necessity. However, how normalization between written scripts is done, is not simple as there are often multiple possible transliteration systems. Vocabulary normalization can also be seen as a learning optimization for language models as one can learn the model parameters with less data or more accuracy if the vocabulary size is reduced. This is generally known as *vocabulary selection*, and the solutions to it differ by domain. In the case of Gamallo et al [GPA17] whose work is similar to ours, vocabulary was selected by hand using phonetic criteria.

A central problem in statistical modeling is data sparsity. Because of the large relative frequency differences between linguistic items such as words or phonemes in a language, it is hard to build a representative statistical model for rare linguistic items. In practice this statistical constraint means that often the training data set doesn't contain enough data to train the language model to handle the rare items well. The poor handling of rare items can either cause the rare items to be misrepresented by the model with regards to their relative frequency or not handling the rare items at all because they were not part of the training set. While the first problem could in principle be ignored, the latter one needs to be addressed because our chosen performance measure requires us to estimate the probabilities for all contexts and phonemes.

Of different language model performance measures, we use the *perplexity* [BJM83], which was developed for speech recognition and is conceptually compatible with how humans tend to think of understanding foreign languages. For example, a native speaker of Finnish would likely understand quite a bit of spoken Estonian, but would be surprised or *perplexed* by some of the words or forms. We use perplexity in a similar manner in our attempts to quantify how different languages are.

2.4 Phonetic Transcriptions and Vocabularies

Phonetic transcriptions are symbolic representations of speech. There are many transcription systems, most at least originally developed to handle one language well. For example, for Uralic languages there exists the *Uralic Phonetic Alphabet* (UPA) [Set01]. A phonetic transcription can be more or less exact. *Narrow transcriptions* are more exact estimates of what is pronounced, and aim to describe the exact *phones*. *Broad transcriptions* (also known as phonemic transcriptions) are less exact and in their broadest (or least exact) sense only aim to disambiguate on the level of phonemes, instead of defining the exact phone used.

According to the *phonemic principle* [Swa34], each language has its own unique set of phonemes, which are all and everything that is needed to express that language in speech. Additionally, when one hears a foreign language, one hears it in the phonemes of their own native language, not in the phonemes of the foreign language. In the case of using a language-independent transcription such as IPA and interpreting the symbols as universal symbols for the sounds, the phonemic principle is rather strongly violated as the use of the symbol within a language only makes sense within that exact language. This is commonly known as the *common phoneme set problem*, or the lack of a *common phonetic space* as discussed by Ellison et al [EK06].

The common phoneme set problem can be understood in two different ways. In computer science and information theory terms we can treat it as an accidental mismatch between language model vocabularies, or rather the symbols used within them, which should be fixable by finding correspondences between phonemes between languages. In other words, thinking opportunistically that the phonemes represent something similar enough that using these correspondences doesn't change the meaning of the phonemes in a way that would invalidate the use cases. If we take a phonetic and linguistic viewpoint, the differences between phoneme sets are either systematic phonetic shifts, which can be accounted for if we want to track something beyond them, or alternatively they are something for which there is no clear correspondence for, which is the case when the languages are not very closely related.

2.5 Visualizing Relationships

While our methods can compute distances between languages, the distances are rarely descriptive just by themselves. They can be used for comparisons, but to actually get something out of the distances, they need to be processed somehow to a representation more compatible with human perception. Examples of these representations are tree diagrams and heatmaps. Trees or *dendrograms* are typical representations because they succinctly describe families and categories (see Figure 3 for an example.) Heatmaps rely on

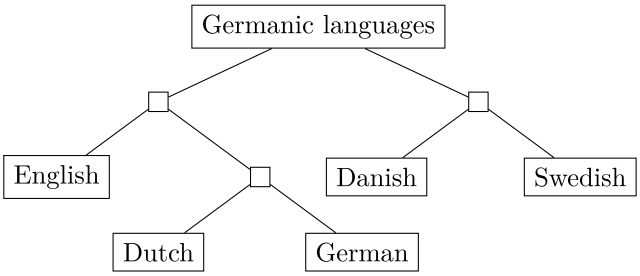


Figure 3: Example language tree of Germanic languages.

color to bring out both patterns and outliers in a standard grid, which makes it possible to distinguish relative differences between items not visible on dendrograms the same way.

Dendrograms are used to represent the results of hierarchical clusters and their most common use case is to show the relationships between languages. Often *language families* and language relationships are described in tree format to show how a language came to be from a predecessor language. Examples of these language families are Germanic languages, or Slavic languages. Languages and language families can be thought to have been born by branching out from an earlier language, i.e. there is a somewhat clear single common origin in time and place. Languages can also be born out of language contact in a way that makes it hard to pinpoint a single origin, these languages are called creoles. While there is ample discussion in this thesis and in current research on how languages can be studied similarly to biological organisms, one cannot always expect to draw similar dendrograms as in computational biology out of languages because languages don't change only at birth, or by themselves gradually, but also through contact with their neighbors. In other words, due to language contact, a more suitable representation of language history would be a directed and cyclic graph, not a tree. However, to produce such a graph one would have to identify and quantify not just language distances, but in a more granular manner the different forms of influence between languages.

A central concept behind reconstructing language history in a tree form are proto languages. For example, in the case of Indo-European languages linguists have reconstructed a language called Proto-Indo-European, of which the cognates shared by all Indo-European languages should in theory be derived from. This is to say, there is an expectation of a lineage of languages, shifts, and changes over time. An example of a family tree such as this is in Figure 3 showing a subset of the Germanic language family, which is a part of the Indo-European language family. The tree contains a subset of West Germanic languages in the left branch where Dutch and German are closer to each other than they are to English. The right branch of the tree contains North Germanic (or Scandinavian) languages Danish and Swedish.

2.6 Prior Work on Computing Language Distances

In this section we go through prior work related to computing language distances, and language clustering or text categorization.

Closest to our work and hence relevant to this thesis is the work done by Kita [Kit99]. The work utilizes n-gram language models on language clustering on text corpora. Major difference to our work is the use of a Kullback-Leibler divergence inspired distance measure which differs from perplexity in its way of taking into account language models for both languages when determining the pairwise distances. Gamallo et al [GPA17] study the same problem, but similarly to our work rely on perplexity as the distance measure.

Batagelj et al [BPK92] were one of the earliest to tackle a language family tree reproduction problem. They use a variety of different edit distances to cluster languages using word distances and sum of word distances as the language distance score. Petroni et al [PS08] somewhat similarly compute normalized word distances via edit distance between cognate lists, averaging these distances into a language pair distance, which is then further converted into a language divergence timing estimate to produce family trees. Both normalize the character set to the English alphabet. Batagelj et al use a limited word list of only 16 words while Petroni et al use the Swadesh list prepared by Dyen et al [DKB92].

Cavnar and Trenkle [CT94] propose a method for text categorization using n-gram frequency statistics, specifically frequency distribution profile, or the order of most common n-grams. While their work focuses on categorization, not on the distances themselves, the method or a variation of it has been used by others [AM11, RS09] as it can be used to derive a distance measure between languages via the rank-order difference between n-gram frequency profiles.

Abramov et al [AM11] introduce multiple methods for computational language typology of differing complexities and compare them to each other. They classify languages using a network analysis method on graphs (Global Syntactic Dependency Networks) extracted from treebanks. They compare the results to an n-gram based method building on top of the work by Cavnar and Trenkle [CT94] and use the most common n-grams as features to be classified by the genetic algorithm. Abramov et al state that the n-gram method is simple and relatively effective at reproducing trees, but typologically less useful due to its opaqueness with regards to language features. In addition to the network and n-gram methods they apply a quantified typology method [AL73] using tree bank data which looks at multiple levels of the language at once (morphology, syntax etc). The method works on morphological complexity, dependency structure of sentences, centrality of predicates, and both relative sentence depth and width in dependency trees.

There have been three approaches to normalization: normalization to an arbitrary script, normalization to a phonetically motivated script and the study of an almost phonetic script to start with. Both Kita and Gamallo et al normalize their text corpora into a form of Latin script. In case of non-Latin corpora this involves a transliteration step from the origin script such as Cyrillic to Latin. In addition to this, Kita removes diacritics from characters, practically ending up with the letters of the English alphabet. Gamallo et al normalize the Latin script into 34 symbols (10 vowels, 24 consonants), which they say is a phonological one containing common sounds, consonant palatalizations, and different vowel articulations. Thus Kita uses an ad hoc vocabulary normalization method while Gamallo et al motivate their chosen normalization phonologically. While Gamallo et al have normalized their Latin script into a phonological one, Rama et al [RS09] use Indian written scripts, which can be thought to be almost phonological in nature.

Ellison et al [EK06] produce family trees by constructing a distance matrix intra-language between a set of cognates and then compute inter-language distances between distance matrices. The authors clearly formulate the common phoneme set problem and present a method that is immune to the problem because it doesn't compare the lexical representations of cognates over languages. Intra-language cognate distances are motivated by psychological models implying that word similarity is related to how easily words are confused with each other.

Gray et al [GA03] use Bayesian phylogenetic methods on binary cognate matrices to reproduce a family tree and timing estimates for the Indo-European language family. Notable in their work is the use of prior knowledge in priming the Bayesian process. Similarly to Gray et al, Lehtinen et al [LHK^{+}14] use Bayesian methods, but they produce *split graphs* from cognate data, and work on Uralic languages. Split graphs are an alternative to family trees introduced by Bryant et al [BFG05] as an alternative to trees and full-blown graphs which doesn't force one to cluster unrelated languages together, and still isn't quite as complex as a full-blown graph. In addition to producing split graphs, Lehtinen et al study the effect of loan words in the analysis by using separate word lists of known cognates and known loan words and compare the graphs produced.

Based on this overview we note that there have been multiple takes on both language distance computation and language clustering. The four main types of methods are either information-theoretic/statistical, edit distance based, tree-bank based, and cognate based. Our approach is information-theoretic because we wanted a method that isn't tied to any one particular type of speech sequence, and because we ultimately want to provide an automated alternative to complement a previously largely manual process.

3 Methods

Our experimental pipeline starts by transcribing a text corpus of each language into a corresponding phonetic transcription, followed by training a language model on each transcription. The language models are then applied on phonetic transcriptions of all languages to compute pairwise distance measures between all languages. The distance measure used, perplexity, is in essence the average degree of surprise for each item in the phonetic transcription. For further analysis, the distances are visualized as heatmaps and dendrograms, and by other means. For an overview see Figure 1.

Because we apply language models on language corpora they were not trained on, the sets of phonemes for the model and the corpus it's applied on never perfectly overlap. In addition to this, due to data sparsity, we may either be estimating parameters for rare phonemes poorly or can even be missing them from the model altogether due to the training sample not having them.

The omission of an individual phoneme from a model prevents the estimation of the probabilities of that phoneme using the model. In this case the only way we can compute a perplexity for the transcription is by ignoring the phoneme. Ignoring unknown phonemes is not a viable option as then some of the statistical properties of the model are not respected. In our case the effects of ignoring the problem are quite drastic as the phoneme sets of languages differ, and this would not then be reflected by the perplexity measure. The other effect is that all the contexts the missing phoneme appears would be ignored. In other words, if a phoneme is missing from one language and is present in the other, the distance measure and the language model must handle this difference, not ignore it.

To mitigate both foreign vocabulary item and data sparsity issues in a generic way we utilize two techniques: *smoothing* and *interpolation*. Smoothing is the process of shifting part of the *probability mass* of phonemes encountered during training more evenly amongst all the phonemes. This makes it possible to shift part of the probability mass also to unknown items so that their probability can be estimated. Interpolation is the use of multiple different language models, in our case, the use of trigram, bigram and unigram models together to reduce the impact of missing items and improve the estimation performance on rare phonemes.

As the distances between languages are highly dependent on the choice of phonetic symbols used in phonetic transcriptions, we want to find out if the language models capture something in addition to just phoneme set differences. We use the *cross entropy* distance measure for comparing the phoneme sets to each other and compare these distances to the ones obtained with language models and the perplexity measure. We utilize cross entropy as a distance measure between the phoneme frequency tables, or unigram distributions as it defines a practical distance measure between two discrete random distributions. Like the language model distance, cross entropy distance is asymmetric. Due to the non-central use of this measure, other distance measures for comparing unigram distributions were not evaluated.

3.1 Phonetic Transcription

We use *eSpeak* [2] speech synthesizer to produce phonetic transcriptions using IPA symbols. Text corpora are run as plaintext through eSpeak and after that converted into a sequence of symbols where both utterance boundaries and word boundaries are abstracted into special *word boundary items* to simplify the analysis of the language models. A number of other transcription post-processing steps are taken in addition to abstracting over word boundaries.

Both primary and secondary word stresses are removed from the transcription. In addition to removing word stresses, special compound phonemes consisting of a glottal stop and a vowel such as ʔa are split into two sequential phonemes, ʔ and a. We also considered doing the same for retracted vowels present in the French transcription represented as ə- or ɑ-, but decided not to do that because we deemed them to just be a part of the way eSpeak transcribes French. One could argue that these are one of the accidental transcription differences, an anomaly in transcription narrowness. Motivated by this narrowness in French transcription one could then opt to remove the retractions. This would broaden French transcription to be more in line with other languages.

Long vowels stay as they are, i.e. when there is a sequence like aː, it is treated as a single phoneme, even though typographically it consists of two separate IPA characters in eSpeak output. We also remove phonemes such as *(el)* and *(em)* which are present in eSpeak output. Their exact meaning in

[2] http://espeak.sourceforge.net/, version 1.48.03 on Mac OS X.

the output isn't known. Given that there is not only one kind of a phonetic transcription, or even two, it is rather clear that any automatic transcription system is bound to produce transcriptions of slightly varying broadness for different languages. This is true even when using a standard transcription system, as the broadness of a language's transcription is defined from within the language itself; broadness is not a universal property.

3.2 N-gram Models

N-gram language models are statistical language models that predict the next item in a sequence given $N-1$ previous items. N-gram models are analogous to Markov chains, but we use the term n-gram model as that is more customary within the NLP community. For each context of size $N-1$, the model provides a probability distribution which describes the probability of transitioning to different phonemes. The probability distributions are estimated during the training phase of the model, and they are derived using *maximum likelihood estimation* (MLE) [Sch04]. MLE is a method for estimating the parameters of a statistical model to maximize its *likelihood*, the probability of obtaining the data given the parameterized probability model. In our case likelihood is maximized by counting the occurrences of sequences of N items and their contexts, and using their relative frequencies within the context as individual transition probabilities:

$$\hat{p}(item_b \,|\, item_a) = \frac{count(item_a, item_b)}{count(item_a)},$$

where $\hat{p}(item_b \,|\, item_a)$ is the estimate for probability of $item_b$ given that is preceded by $item_a$.

For example, given sequence A, A, B, B, the bigram ($N = 2$) probability distribution for context A would be calculated for A as $p(A \,|\, A) = \frac{c(A,A)}{c(A)} = \frac{1}{2}$, for B as $p(B \,|\, A) = \frac{c(A,B)}{c(A)} = \frac{1}{2}$; for context B as $p(B \,|\, B) = \frac{c(B,B)}{c(B)} = \frac{1}{2}$, and $p(A \,|\, B) = \frac{c(B,A)}{c(B)} = \frac{0}{2}$.

To find out how different n-gram models work, different models and their combinations were tried. *Unigram model* is a probability distribution of individual symbols over the learning data. *Bigram model* takes context into account by looking at the previous item in a sequence to predict the next. Trigram model extends the observed context to two items, i.e. given two symbols, it provides a probability distribution of items that should follow.

Additive smoothing is used to handle unseen symbols (see Section 3.2.1 for definition) and linear interpolation is utilized to handle unseen transitions between symbols (see Section 3.2.2 for details). Trigram, bigram and unigram models are used in conjunction in an interpolated model setup to estimate the probabilities, and in addition to this, additive smoothing is used on the unigram model to estimate probabilities for items not encountered during training. Hyperparameters for linear interpolation and additive smoothing were not optimized programmatically. They were hand-tuned to lower the perplexity values up to some extent. There are likely more optimal parameters, especially if different parameters values were used for different languages.

3.2.1 Additive Smoothing

A central property expected by the perplexity distance measure from the language model is that it can estimate the probability of each trigram, not just the ones it was trained on. To maintain this property for phonemes not part of the training data set, we use additive smoothing, a mechanism to shift part of the probability mass in the model to these phonemes. This can also be phrased as handling items that are not part of the model's vocabulary. For example, given a learning set of A, A, B, B and a unigram model with items A, B, C we would end up with probabilities $p(A) = \frac{2}{4} = \frac{1}{2}$ and $p(B) = \frac{2}{4} = \frac{1}{2}$, but $p(C)$ would be 0, which is incompatible with the perplexity measure. Similarly, the estimated probabilities for all the cases where the zero-estimated is part of the context would be 0.

Additive smoothing (or Laplace smoothing) [Lid20] is used on probability models to smooth over values where we know the probabilities should not be zero. Additive smoothing works by adding a certain constant (commonly called *pseudocount*) to every item's count in the learning set in order to distribute a small amount of probability mass to every item regardless of whether all the values were encountered in the learning set or not. Additive smoothing probability estimator function is defined as

$$p(item) = \frac{Count_{item} + pseudocount}{Count_{allitems} + Size_{vocabulary} \cdot pseudocount}.$$

Using additive smoothing and a pseudocount $PC = 1$ the derived

pseudocount	A	B	C
2	0.400	0.400	0.200
1	0.429	0.429	0.142
0.5	0.454	0.454	0.091
0.1	0.488	0.488	0.023

Table 3: Probabilities assigned with different pseudocounts.

probabilities would be $p(A) = \frac{2+PC}{4+3*PC} = \frac{3}{7}$, $p(B) = \frac{2+PC}{4+3*PC} = \frac{3}{7}$, and $p(C) = \frac{PC}{4+3*PC} = \frac{1}{7}$, which would shift part of the probability mass to item C. Generally, the higher the pseudocount, the more of the probability mass is shifted to unknown items, as seen on Table 3.

Throughout our tests we used a value of 0.1 for the pseudocount. Initially a default of 1 was used as advocated by Lidstone [CG96, Lid20], but it was reduced to 0.1 to obtain better results. After trying a number of other (lower and higher) values and not seeing a big difference in results, we settled with it. It is not entirely clear how to tune this hyperparameter in our setup as the objective of smoothing differs from single language modeling. If we were tuning the hyperparameter for use within one language, for example for use in speech recognition, the optimization would be rather straightforward. In our case, as there are multiple languages involved, the assumptions that the learning set vocabulary and test set vocabulary overlap in a similar way, and that the sizes of vocabularies are similar, don't apply.

We could optimize pseudocount for each language pair separately, but there's a risk of choosing a too high value when the vocabularies are highly divergent. This would result in an over-smoothed model where genuine differences between vocabularies are not reflected. There's no single obvious way to optimize for a one pseudocount value for all models, either. For example, if we tried to optimize the average language distance measure, we might end up with over-smoothed models again as we would likely end up optimizing away the highest language distances with highly divergent vocabularies. Even with a more conservative approach genuinely good distance measures between language pairs close to each other could be negatively affected if we tried to optimize distance measures where our model doesn't work very well.

We did not experiment with other smoothing methods. The assumption is that the choice of smoothing technique isn't currently the main bottleneck with regards to the quality of the results, and thus trying out other methods wouldn't have necessarily improved the results.

The frequency of phonemes seems to follow a kind of a log-linear relationship with the rank. The most common phonemes are significantly more common than the least common phonemes. One could also characterize the phoneme frequency distribution to also somewhat follow the Zipf distribution [Zip32], although Zipf generally estimates the relatively common (rank between 10-30) phonemes to be less frequent than they are in our data, which can clearly be seen on Figure 4.

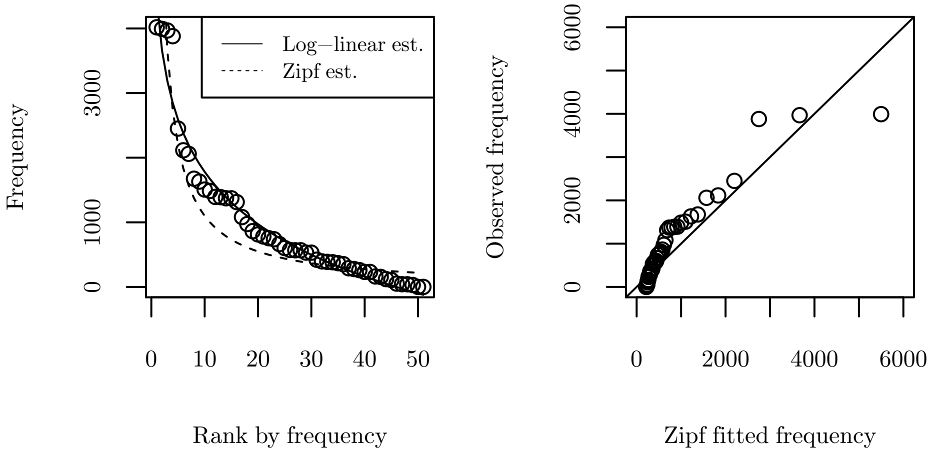


Figure 4: Frequency distribution of English phonemes and their fit against the Zipf distribution.

Even when we know the distribution of phonemes, we cannot assume the exact same implications and mitigation effects with regards to data sparsity as for cases where the training and testing language are the same. The languages may be using differing phonemes altogether, i.e. a common phoneme in one language may be missing from another, which means the n-gram contexts this phoneme appears aren't handled gracefully. A more capricious case is when the languages in question use a phoneme in completely different roles. A common phoneme in first language may be relatively uncommon in a second language, and may also be used in different contexts from the first language. Some of these differences are caused by genuine differences between phonologies, but some may be more accidental differences in transcription.

3.2.2 Interpolation

Unigram models typically have between 50 and 60 unique items, bigram models 1000-1200 unique transitions, and trigram models 8000-10000 unique transitions. In order to fully train a model, the training data set needs to be large enough. Trigram model requires the largest training data set, followed by bigram and then unigram. See Section 4.1 for more details. Interpolation, as its name implies estimates parameter values by calculating a kind of a consensus estimation between multiple models so that we can mainly rely on the specific trigram model, but also use the information in the simpler unigram and bigram language models when the trigram model cannot estimate a probability for a context.

Linear interpolation (or Jelinek-Mercer interpolation) [JM80] is an interpolation method for n-gram language models, which works on the assumption that n-gram models with lower n-parameters will more likely have a defined, or well-estimated probabilities for rare item sequences not found from the higher order model. Linear interpolation works by estimating the probabilities over multiple n-gram models. We use the term *interpolating trigram model* to refer to a language model that interpolates between trigram, bigram and unigram models. Probabilities for this model are estimated as $\hat{P}(w_i \mid w_{i-2}w_{i-1}) = \lambda_1(w_i \mid w_{i-2}w_{i-1}) + \lambda_2(w_i \mid w_{i-1}) + \lambda_3(w_i \mid w_i)$, where λ_1 is weight of the trigram model, λ_2 the weight of the bigram model, λ_3 the weight of the unigram model, and $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

We didn't try to optimize the interpolation hyperparameters (λ_1, λ_2 and λ_3), but settled for values $\lambda_1 = 0.7$, $\lambda_2 = 0.2$, and $\lambda_3 = 0.1$ early on. Some minor modifications to the hyperparameters were tried, but the results were not conclusive. Both interpolation and smoothing hyperparameter optimization share the same conceptual difficulties with regards to the decision on which distance measure or measures to minimize. See Section 3.2.1 for discussion on the conceptual issues with hyperparameter optimization.

3.3 Perplexity Measure

Perplexity [BJM83] is a measure used in the speech recognition community to describe how surprised a system is when it encounters a certain input. The simple way to understand the measure's value is that a perplexity value of k means that one is as surprised on average as one would have been if

Item	Count	Probability
A	2	0.33
B	2	0.33
C	1	0.167
D	1	0.167

Table 4: Frequency table for sequence A, A, B, B, C, D.

one were to have had to guess between k equally probable choices at each predictable phoneme. In other words, with a k value of 10, one had to choose from 10 equally probable choices on each phoneme in the measured sequence. Formally perplexity is defined as

$$Perplexity = b^{-\frac{1}{N}\sum\limits_{i=1}^{N}\log_b q(x_i)},$$

where b is the base and is customarily, and in our tests, 2, and $q(x_i)$ is the estimated probability for the item in the language model.

For example, given a learning sequence of A, A, B, B, C, D, we would get the unigram model frequency Table 4. Given the frequency table, we would calculate the perplexity for a test sequence A, B, D as $2^{-\frac{1}{3}(log_2 0.33+log_2 0.33+log_2 0.167)} = 3.8$.

An alternative distance measure to perplexity for our use case would have been Juang & Rabiner's Kullback-Leibler divergence based method [JR85], which was used by Kita [Kit99] in their experiments, which were similar to ours. We did tests with the measure, but they are not included in this thesis as the results did not significantly differ from the ones done with perplexity. Perplexity was chosen over the measure used by Kita as it is better known, simpler, and in our case performs similarly.

3.4 Cross Entropy Measure

Cross entropy is a distance measure between two arbitrary probability distributions. In information theory terms, when entropy can be thought of as the average coding cost of a random variable when coded against its own probability distribution, cross entropy is the average uncertainty when the random variable is coded against some other probability distribution. Entropy can be interpreted as a measure of surprisedness [MS99, p. 73–76], somewhat

Variable	A	B	C	D
Probability	0.1	0.2	0.3	0.4

(a) Distribution p'.

Variable	A	B	C	D
Probability	0.1	0.4	0.4	0.1

(b) Distribution q'

Figure 5: Discrete probability distribution p' and q'.

similarly to perplexity. We use cross entropy to compute distances between unigram distributions: how surprised are we in the context of one language's unigram distribution when encountering a foreign unigram distribution.

In the case of discrete distributions, cross entropy is defined as

$$H(p,q) = -\sum_{x} p(x) \cdot log_2 q(x),$$

where $p(x)$ is the probability according to distribution p and $q(x)$ the probability according to distribution q.

For example, given the discrete probability distributions p' in Table 5a and q' in Table 5b, the cross entropy between p' and q' would be $H(p',q') = -(0.1 \cdot log_2 0.2 + 0.2 \cdot log_2 0.4 + 0.3 \cdot log_2 0.4 + 0.4 \cdot log_2 0.1) = 2.22$ while $H(q',p') = -(0.2 \cdot log_2 0.1 + 0.4 \cdot log_2 0.2 + 0.4 \cdot log_2 0.3 + 0.1 \cdot log_2 0.4) = 2.09$. In other words, it takes less bits to code distribution p' given distribution q' than the other way. Note that $H(p',p') = 1.85$, and $H(q',q') = 1.72$.

3.5 Visualizing Language Distances

Language relation hypotheses in related studies have mainly been visualized with family trees, or dendrograms. Their use originates from computational biology, where dendrograms are used for representing relationships between genes, or between organisms. In dendrograms each language belongs to a cluster and these clusters similarly can belong to clusters (see Figure 3 for an example). Characteristic to dendrograms is the expectation of each dendrogram member to have just one closest neighbor and that the further in the tree one goes, the more distant the tree members are. In other words, if a language is equally close to two other languages, in a dendrogram one would still have to be chosen as the closer one. Dendrograms are hierarchical in nature, i.e. they can only express language relationship upwards and downwards in a tree, not sideways or across branches.

Due to the complexities involved with what can be well represented by a tree, an alternative representation with more degrees of freedom is a heatmap. Heatmaps are tables where each language is represented by both a row and a column and the color of the cell in the intersection is the distance between them. In our case the heatmap is asymmetric because there are two distances between each language pair. Heatmaps are especially useful for us because they don't hide the complexity of two different distance measures between language pairs. Heatmaps show the two distance measures separately in a graphical representation and can be used to explain why certain languages behave in unexpected ways when distances are represented by more opaque methods such as trees.

The colors and how they change from one to another are a central part of a heatmap. In order to properly distinguish groups one needs to understand what kind of values require which colors and whether the color gradients should be gradual, continuing or even have jumps in them. We selected colors so that low distances would be easily distinguishable from the rest by constructing the color scale from two gradients, which have a small discontinuation between them. In addition to this the colors were picked so that the heatmap would be monochromatic for better printability and readability without perfect color vision.

4 Experiments

In order to validate the viability and assess the performance of n-gram models and perplexity measure, two sets of experiments were executed. To test the effects of sample size in learning and testing of language models, different sample sizes were evaluated for both tasks. The general assumption is that the larger the training and test sets are, the less the distance measure varies from sample to sample when its variability is measured using a sampling approach. We are especially interested in the effects of sample sizes because of the research project's interest in low-resource languages that are limited both in terms of tools and in terms of available corpora. In order to analyze these languages, we must have methods that work with small corpora.

The general viability of n-gram models is studied by comparing visualizations of distance matrices and distances between languages to linguistics literature and well-known language families. Cross entropy, a distance measure which compares the frequency distributions of phonemes within language model vocabularies is used to study the effects of differing phoneme sets.

We use the Europarl parallel translation corpus, which consists of a collection of translated parliament speeches from the European Parliament. This means that for each language corpus the subject matter is roughly the same. Different languages have different amount of material available due to some countries having joined the EU after the corpus was first published. While the same amount of input data is used for each language for each task in the tests, the corpora are not aligned. Due to the rather peculiar contents of the corpora, political speeches, one can only speculate on how representative are the corpora of the languages overall. However, as the subject matter is the same for all languages and hence also the lexicon, the corpora should be relatively comparable to each other.

4.1 Effects of Sample Size

There are a number of approaches for exploring the effects of sample size and the learning and testing data size requirements. One can approach the problem from a purely statistical point of view and compute the correlation between the parameters of a *fully trained* model and a sample size *constrained* model to gauge how well the sample size constrained model corresponds to a fully trained model. The main benefit of this approach is the immediate applicability of commonly accepted thresholds for statistical significance in the analysis of the results. Main caveat is that there is no simple way of knowing how the statistical correlation corresponds to the model's prediction performance without further studying this relationship.

Then there are the use-case specific metrics. If we were to take family tree reproduction as the main use case for our methods, we could tailor the sample size tests around this and use *cophenetic distance* [SR62] as the distance measure. Cophenetic distance is a distance measure used to compare and calculate distances between trees. In our case, we could utilize cophenetic distance to deem a sample size constrained model to be good enough — or even require it to produce an identical family tree to a fully trained model. If we were to take nearest neighbor discovery as the main use case for our methods, we could focus on the neighbor lists of languages and use a statistical metric such as *Spearman's rank correlation coefficient* [Spe04], which is a method for measuring the correlation between the orders of items, in our case the ideal order of neighbors to the order computed by our method. This would again make it easier to rely on statistical significance for making decisions on whether a learning sample is sufficiently large. Both of these are use case specific measures for language model performance. As we were mainly interested in how perplexity measure behaves, and not focused on either of these use cases, we didn't use these measures.

Instead, we approach the problem by trying to answer the question *what is the probability of having a large enough corpus*, where large enough means that the perplexity measure is within a predefined threshold with high enough certainty. In other words, in order to validate the feasibility of language models for smaller data sets, we use probability as the driving measure for finding large enough sample sizes. For each sample size, we calculate the probability of reaching a distance measure within a threshold when compared to a fully trained model or when compared to the full test data set.

Algorithm 1 Learning sample size

```
Input: $C_f$, $iterations$, $threshold$, where $C_f$ is the full training corpus
Output: $P_{size}$, an associative array of $size \to probability$
  $M_f \leftarrow Train(C_f)$
  $D_f \leftarrow Distance(M_f, C_f)$
  for $size$ in sample sizes do
    $A \leftarrow 0$
    for $i = 0$ to iteration do
      $M_c \leftarrow Train(Sample(C_f, size))$
      $D_c \leftarrow Distance(M_c, C_f)$
      if $|D_c - D_f| <= threshold$ then
        $A \leftarrow A + 1$
      end if
    end for
    $P_{size} \leftarrow \frac{A}{sample\ count}$
  end for
```

The learning sample size algorithm (see Algorithm 1) goes through learning sample sizes in growing order, trains constrained models from sampled training corpora and calculates distances between them and the full testing corpus. The probability of the sample size being large enough is calculated by dividing the number of models having a distance measure within a threshold of the fully trained model by the number of iterations (which can also be thought of as the sample count). The measure tested and its parameters are, or correspond directly, to the distance measure produced by our method itself, which makes it interpretable in the same context.

We studied the effect of test sample size with a similar test to the learning sample test. As described in Algorithm 2, different sized corpora are tested in a sampling manner in growing order against a fully trained model. For each corpora size a distance is calculated and the number of distances within a given threshold is divided by the number of iterations to compute the probability of reliably reaching the threshold.

Algorithm 2 Test sample size

```
Input: $C_f$, $iterations$, $threshold$, where $C_f$ is the full training corpus
Output: $P_{size}$, an associative array of $size \to probability$
  $M_f \leftarrow Train(C_f)$
  $D_f \leftarrow Distance(M_f, C_f)$
  for $size$ in sample sizes do
    $A \leftarrow 0$
    for $i = 0$ to iteration do
      $D_c \leftarrow Distance(M_f, Sample(C_f, size))$
      if $|D_c - D_f| <= threshold$ then
        $A \leftarrow A + 1$
      end if
    end for
    $P_{size} \leftarrow \frac{A}{sample\ count}$
  end for
```

We study the behavior by sampling because the methods themselves are statistical and rely on the distribution of items or item sequences. The iteration count required wasn't extensively studied, it was increased for as long as the results seemed to be unstable from run to run. In the learning tests we used a perplexity threshold of 1.5, within which the perplexity for each iteration has to be within, a cutoff probability of 0.9, the minimum proportion of iterations that have to reach a perplexity within the perplexity threshold for a cutoff to happen, and an iteration count of 200. For test samples tests we used a perplexity threshold of 0.5, a cutoff probability of 0.9 and an iteration count of 1000. All of these parameters were adjusted so that the distance measure would be reasonably close to what a fully trained model would produce and that the tests would finish in a reasonable amount of time. While an increase in the iteration count would improve accuracy, the improvements would be rather small and they would likely not affect the analysis.

Different perplexity thresholds were used for the experiments because of the differing stabilization behaviors between learning and testing. Using a lower threshold on the learning test would have resulted in unpractically large stabilization sample sizes considering the low resource language mindset. Conversely, using the same, higher perplexity threshold for test samples

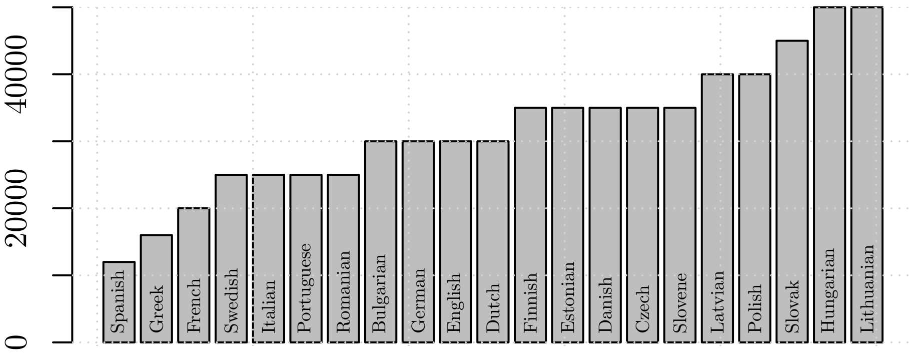


Figure 6: Learn stabilization sample size using the trigram model over all studied languages.

would have sacrificed perplexity accuracy, and wouldn't have given as good an understanding of stabilization behavior. These two conclusions are derived from the general assumption that we have similar sized language samples for different languages. In other words, there is no harm in using a larger test sample as learning behavior is the more likely bottleneck.

While our testing methodology tries to account for variability in the training and test sets by sampling, one should note that our tests only measure the stabilization by testing the model against its own language, not as is done on our method's viability tests where we apply a language model on a language other than the model was trained on.

4.1.1 Sample Sizes in Learning

Based on the experiments results, languages can be roughly divided into three groups in their learning behavior. There are languages with low sample size requirements, a group consisting mostly of Romance languages. The higher end of sample size requirements is dominated by Slavic and Baltic languages. The rest of the languages fall in the middle between 30000 and 40000 phonemes (see Figure 6), which translates to a word count between 5000 and 7000. Hungarian and Lithuanian require larger samples and only stabilize at 50000 phonemes or 8000 words. The languages with low sample requirements are Spanish, Greek and French, which stabilize below 30000 phonemes, Spanish notably at 12000 phonemes, or 2500 words.

To illustrate the differences between languages, we studied the two ex-

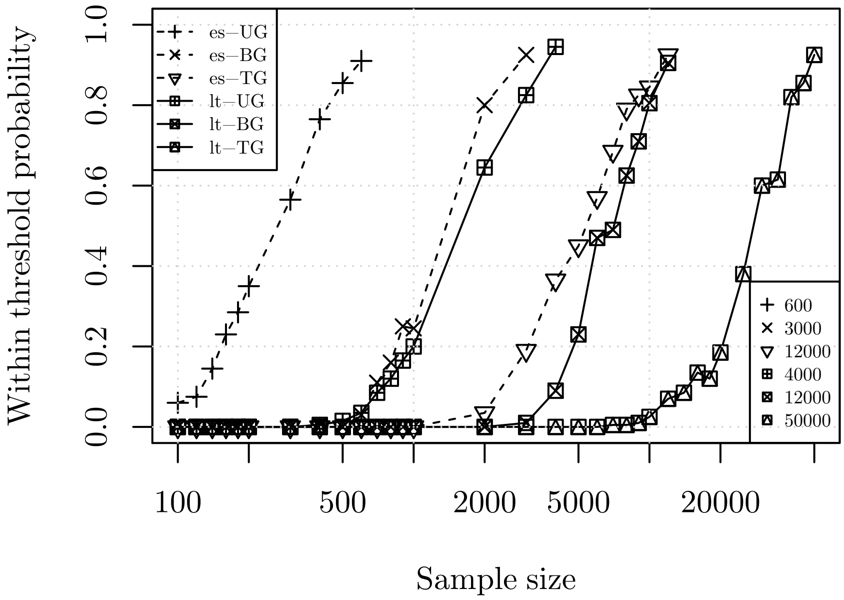


Figure 7: Learning stabilization for Spanish and Lithuanian languages using different models. Stabilization cutoff sizes for each model are shown in lower right corner.

trema languages by learning behavior, Spanish and Lithuanian (see Figure 7.) Spanish bigram model can be trained with a sample of 3000 while Lithuanian unigram model requires 4000 phonemes for training. Similarly, Spanish trigram model and Lithuanian bigram models can both be trained with 12000 phonemes. This shows that while one can make conclusions on the basis of model's complexity, in the sense bigram model generally doesn't require as large a learning sample as a trigram model, the differences between languages are significant and need to be taken into account when estimating the required sample size.

To come up with a way to estimate the learning sample size for a language, we studied whether this could be achieved by extrapolation from the language's phoneme count. While the two extreme languages in their learning behavior, Spanish and Lithuanian, at 40 and 75 phonemes respectively, seemed to support this hypothesis, we didn't find a correlation between the variables. Two counter examples to this are Hungarian and Romanian: Hungarian has 51 phonemes and it stabilizes at 50000 phonemes while Roma-

Threshold	Min	1st Qu.	Median	3rd Qu.	Maximum
1.5	12000	25000	30000	35000	50000
2.0	7000	14000	20000	25000	30000
2.5	5000	12000	12000	16000	25000
3.0	4000	8000	10000	12000	16000
4.0	3000	5000	6000	8000	12000
5.0	2000	4000	5000	6000	8000

Table 5: Learning sample stabilization points for different perplexity thresholds. Each row is a statistical summary over all the languages of the stabilization sample size given a perplexity threshold.

nian has 92 phonemes and it stabilizes at 25000 phonemes. In other words: different languages have differing learning sample requirements and these requirements cannot be estimated by their phoneme counts size alone.

A correlation between a language's phoneme count and its learning stabilization sample size would have made it possible to estimate a learning sample size for languages that were not part of this test. As this was not the case and as we also don't have a single good number for the perplexity threshold, we then studied the stabilization behavior for the whole set of languages with different perplexity thresholds to have a better idea on what kind of sample sizes one needs to use to obtain a certain precision in the perplexity measure. Table 5 gives a statistical overview of the learning stabilization sample sizes with different values of perplexity threshold. In a corpora size restricted research setting, given a sample size between the median and 3rd quartile values one should be able to reach comparable perplexity precision.

4.1.2 Sample Sizes in Testing

In a similar experiment to the learning sample size test we studied how perplexity stabilization behaves with different test sample sizes. Half the languages stabilize with sample sizes of 2000 to 5000 phonemes, which translates to 350 to 900 words as can be seen in Figure 8. Seven languages stabilize at under 1000 phonemes or under 200 words. Bulgarian is a notable outlier because it only stabilizes at 14000 phonemes, or 2500 words. Spanish, similarly to learning sample size test, requires the smallest sample at 500. Along with Spanish in the low sample size requirement group are other Romance

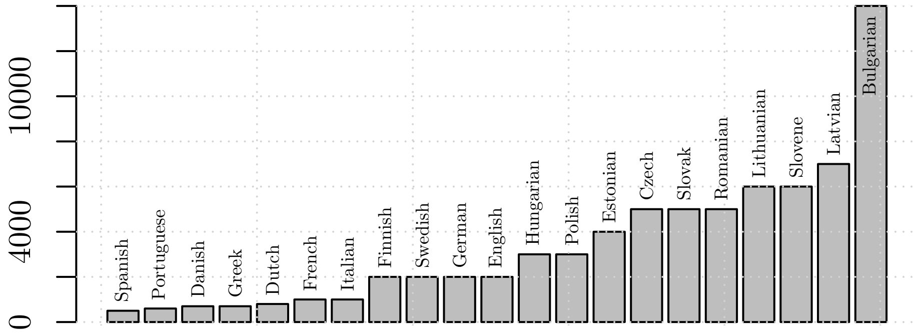


Figure 8: Test stabilization sample size using the trigram model over all studied languages

languages Portuguese, French and Italian. The high end is dominated by Slavic and Baltic languages as Latvian, Slovene, Lithuanian, Czech, Slovak and Bulgarian are in the top seven languages requiring largest samples.

The experiment indicates that while the unigram model can be trained with a small sample, the test sample for the unigram model needs to be rather large compared to the other models. An example of this is Finnish, where the unigram model requires a sample of 14000 phonemes to stabilize, as seen on Figure 9. At the same time, the bigram model requires a sample of only 3000, and the trigram model a sample of only 2000 phonemes. In terms of words this is over 2000 for the unigram model, 450 for the bigram model, and 300 for the trigram model.

The unigram model seems to require a large test set while the trigram model requires considerably more learning data. The bigram model is somewhere between these two for both tasks. We believe that a higher order n-gram model test sample size stabilizes with a smaller sample because the model is simply better at predicting. I.e. given a bigram model and a trigram model, the bigram model is more likely to be highly perplexed by something very rare, which may then heavily affect perplexity and in our tests cause it to not be within the threshold. Another way to look at the problem is the higher variability of a lower order n-gram model's perplexity measure, meaning that when we set a fixed threshold parameter for a test, our tasting favors the models with lower variability.

When training language models, one is often concerned with *overfitting*. Overfitting is a phenomenon where the trained model is too descriptive of

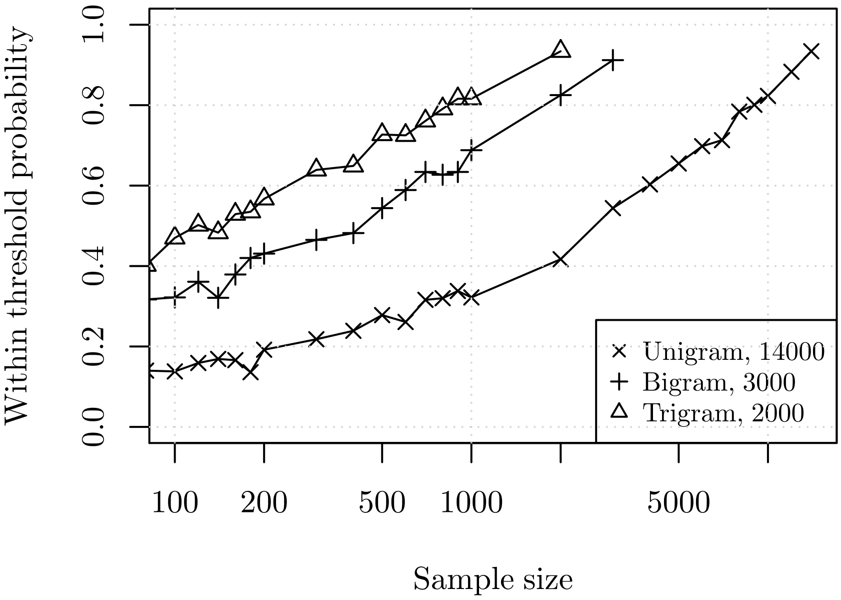


Figure 9: Testing stabilization for Finnish language. Right lower legend shows the stabilization cutoffs for each model type.

the corpus it was trained on. For example, in the case of speech recognition, if one is training a language model to recognize a language but only trains on speakers of one distinct dialect, the model would in fact learn to recognize that dialect, not the language itself. Overfitting is commonly mitigated by making sure the trained model keeps improving during training on a separate *held out data set* from the one it is being trained on.

In our case overfitting could be described in three ways: as overfitting to our chosen corpus, overfitting because of an insufficient learning sample size, or overfitting to the model's own language. If we were overfitting to a corpus, the language model would be too focused on parliament speech, speech transcriptions, or to our automatic transcription system. An advantage of the parallel translation corpus is that the subject matter is exactly the same across languages, i.e. overfitting this is not as big a problem. The same applies to speech transcriptions. It would have been useful to try other automatic transcription systems, but this was not possible due to the lack of easily available comparable systems. If we were overfitting because of an insufficient learning sample size, we would be overfitting because the learning sample wouldn't be representative of the corpus as a whole. This we mitigate against with the tests presented here.

If we were overfitting to the model's own language, we would be using more training material that is needed. We would end up with a model that performs very well on its own language, but which would perform very poorly on others, i.e. the distances to others would be disproportionately high. Although self distances seem to be significantly lower for all languages than distances to other languages, the distances to others are still meaningful and the qualitative analysis in Section 4.2 shows that this sort of overfitting has not happened. It would have been better to address this issue in a statistical manner as well, but there is no obvious method to do that as we do not know what the distribution of distances should be like. We can only qualitatively make the judgment that the distances make sense.

In order to understand what kind of perplexity differences make a difference using our methods, we looked at the language pairs with lowest distances. The lowest perplexity differences between languages in our tests are between the closely related languages Czech and Slovak. Using the Czech model the perplexity on a Czech corpus is 8.79 while perplexity for a Slovak corpus is 21.41. Against the Slovak model Slovak corpus gets a perplexity of 8.64 and Czech a perplexity of 19.05. The next closest neighbor to Czech is Greek at 37.60, and for Slovak the next closest neighbor is Greek at 42.62. Given these kind of perplexity differences, one could conservatively hypothesize that a perplexity threshold of 3.0 would be sufficient, giving us a ground rule of 12000 phonemes, or roughly 2000 words for achieving a level of perplexity stability needed to detect these kinds of pairs. It must be emphasized, though, that these kinds of results only follow if the corpora and transcriptions are similar. The sample size required is roughly 4 pages of text or 20 minutes of speech to produce 2000 words at 500 words per page, or 100 words per minute in speech.

As using automatic transcriptions with homogeneous corpora is rather unrealistic for a project such as the Digital Language Typology project, one could hypothesize that with a similar sized vocabulary (50 to 60 items) of speech features at the same rate as in our transcription one would need roughly 20 minutes of speech data. This would mean that the method is not strictly a big data method requiring hundreds of hours of data, and could be applicable to smaller speech corpora. How and what these speech features are and how one can generalize a language into a set of these speech features given speech corpora spoken by different people is intentionally left out of this analysis.

4.2 Language Model Viability and Performance

As language models' main use case has been within areas such as speech recognition, text prediction, and information retrieval, their use as a language distance measure is not a generally well understood problem. The methods used in the analysis of experiment results are rather ad hoc, and rely mainly on comparisons to existing literature within linguistics on language relationships, and statistical analyses.

To gauge how the methods work on languages that are known to be relatively close to each other, we try to detect known language families. To see whether we can detect very closely related languages, we analyze distances in more detail by comparing the perplexity difference of different order n-gram models on language pairs. As from a computer science viewpoint our task is very closely related to prior work on the reproduction of family trees, we also visualize them as trees and analyze these trees.

4.2.1 Detecting Language Families

Three language families were chosen for this test, the test includes Germanic languages Danish, Dutch, English, German, Swedish; Slavic languages Bulgarian, Czech, Polish, Slovak, Slovenian, and Romance languages French, Italian, Portuguese, Romanian, and Spanish.

In order to understand what is a low and what is a high distance, and to understand the overall behavior of the distance measure, we first take a look at the distribution of distances. The distances between languages vary greatly, lowest being the distances from language to itself, where the perplexity values vary from 6.47 for Spanish to 9.15 for Czech. Typical distances between languages are under 2000, roughly half the distances being under 1000 as seen in Figure 10. Closely related languages Czech and Slovak have distances of 23.03 (Czech model, Slovak corpus) and 20.13 (Slovak model, Czech corpus). According to visual inspection, most distances are under 1000, and the distances above 1000 are mostly below 4000, i.e. lower distances are generally more common than higher distances. Within the

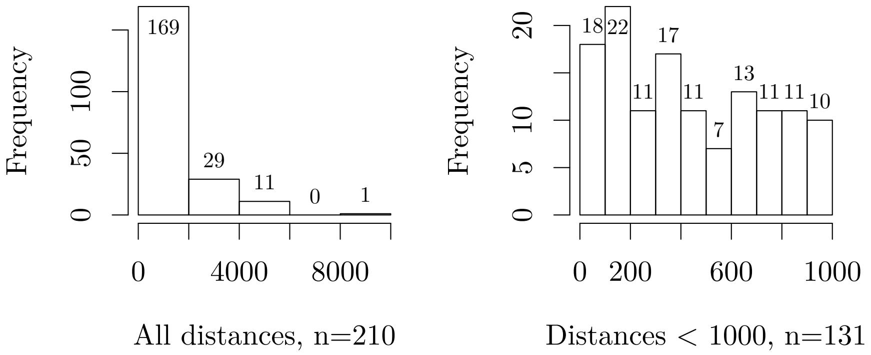


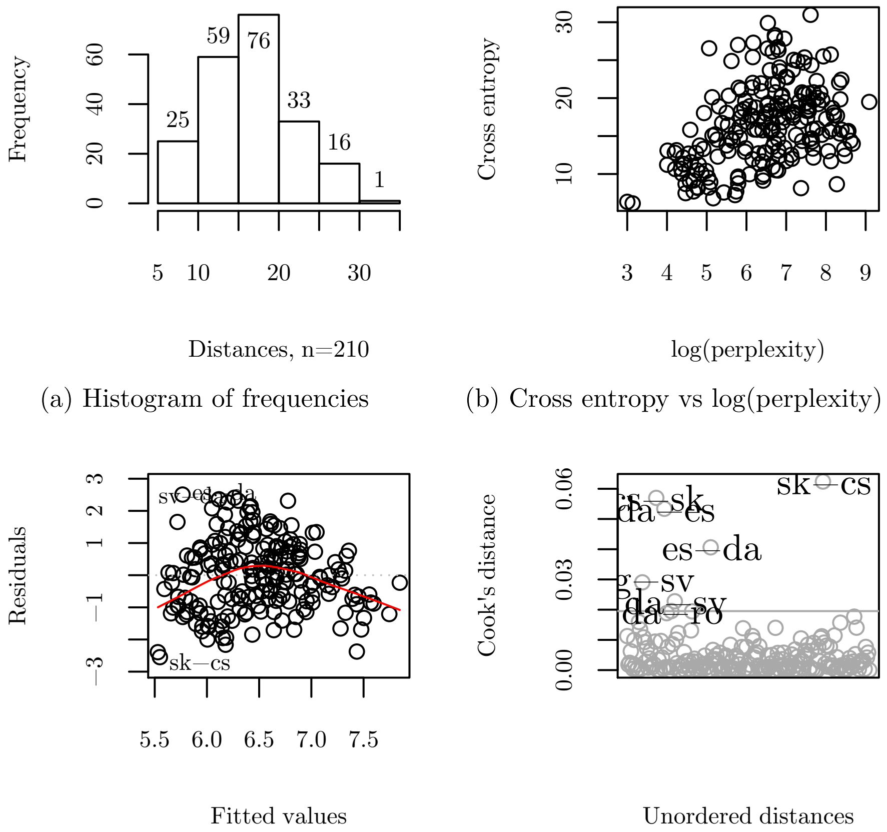
Figure 10: Frequency distribution of distances using the trigram model, self distances excluded, and on the three families data set.

distances under 1000 the distances are more evenly distributed than above 1000, but again the lowest distances are more common than higher distances.

To study the effect of the phoneme distributions on our language models, we take a closer look at the distributions of cross entropy and trigram perplexity, and at how they correlate. We study the correlation between cross entropy and trigram perplexity using Spearman's rank correlation coefficient as the studied distributions were not normally distributed according to Shapiro-Wilkes test, and hence the more commonly used Pearson's correlation coefficient wasn't applicable.

The distributions of cross entropy distances and trigram perplexity distances are quite different from each other, as evident from Figures 10 and 11a. Spearman's correlation coefficient between cross entropy and trigram perplexity is 0.40, which points to a moderate correlation between the variables. To study the correlation, we build a linear model to see how cross entropy functions as a predictor for perplexity. In Figure 11b cross entropy is plotted against the logarithm of perplexity, and we can see that the point cloud is quite wide with large deviations, although a fuzzy right-ascending shape can be seen. In Figure 11c Y axis shows the prediction residuals for the predicted log perplexities. Highlighted in this plot are pairs Spanish-Danish, Swedish-Bulgarian and Danish-Spanish (model-corpus order), which have high residuals, i.e. they are not well predicted by the linear model.

Figure 11d shows the outliers with regards to deviance from mean of *Cook's distance*. Cook's distance tells us how much a data point affects the



(a) Histogram of frequencies

(b) Cross entropy vs log(perplexity)

(c) Residuals vs fitted

(d) Cook's distances

Figure 11: Cross entropy distribution and cross entropy's prediction performance of trigram perplexity using a linear model. Cutoff line for Cook's distance (Plot 11d) is set at four times the mean of Cook's distances. Self distances are excluded, using the three families test setup.

linear model, i.e. the higher Cook's distance a data point has, the more the linear model needs to be changed to take this data point into account. Here the cutoff line is set at four times the mean Cook's distance, and the labeled distances above the cutoff line can then be thought of as a larger group of outliers. Five highest Cook's distances (pairwise, either way) are for Danish-Spanish, Slovak-Czech, Czech-Slovak, Danish-Spanish, Bulgarian-Swedish, Danish-Swedish and Danish-Romanian.

In addition to the correlation coefficient, another way to study the goodness of fit of the model is by plotting the residuals of the linear model, as in Figure 11c, and verifying they are uniformly distributed. In our case

A	B	XE(A → B)	P(A → B)	XE(B → A)	P(B → A)
Czech	Slovak	6.13	20.99	6.34	19.45
Danish	Romanian	24.89	289.63	12.17	2410.27
Danish	Spanish	26.57	149.95	8.66	3923.27
Danish	Swedish	29.90	691.72	14.00	5846.06
Bulgarian	Swedish	27.02	397.64	12.41	4252.13

Table 6: Top five linear regression model outlier language pairs according to their Cook's distance in the three families data set. P stands for perplexity, XE for cross entropy. Here cross entropy is the predictor and perplexity the predicted.

they appear to be somewhat uniformly distributed, although the red line bends downwards on the left and on the right, which would hint that the relationship is somewhat curved: for low and high values of cross entropy perplexity is lower than expected. Even though the model may not be completely representative, we can still use it to dive deeper into the how the variables correlate.

When looking at the outliers with Cook's distances, there are interesting outliers. The cross entropy distance from Spanish to Danish is low at 8.66 (Danish having lowest distance compared to other neighbors of Spanish), but the other way the distance is 26.57, which is one of the highest distances from Danish. Looking at perplexity, Spanish corpus on Danish model gets a perplexity of 149.95, which is low while using the Spanish model on Danish the distance is a very high 3923.27. This relationship seems to be completely asymmetric and indicates quite a strong disagreement between the models on the proximity of Danish and Spanish. As reported on Table 6 other such asymmetric pairs are Danish-Italian, Bulgarian-English, and Bulgarian-Swedish. Czech and Slovak are outliers as well due to their significantly lower pairwise distance, both in terms of perplexity and cross entropy.

This analysis sheds some light on the expected and demonstrated relationship between the distribution of phonemes and perplexity. While the correlation coefficient shows a moderate correlation, there are significant outliers. Danish language in general is part of many of these outlier pairs, as is the known closely related language pair Czech-Slovak. Given these results, we believe that trigram perplexity can partly be explained by the phoneme distribution.

To analyze whether we can actually detect language families, we turn our attention to the heatmap in Figure 12. Without looking at the details, all language groups are rather visible and distinct from other groups on the diagonal. When looking at in-group distances, all Slavic languages except for Polish are relatively close to each other. In Germanic languages Danish is the outlier, having relatively higher distances to other languages in its own family. From Romance languages only French is not as close to the others.

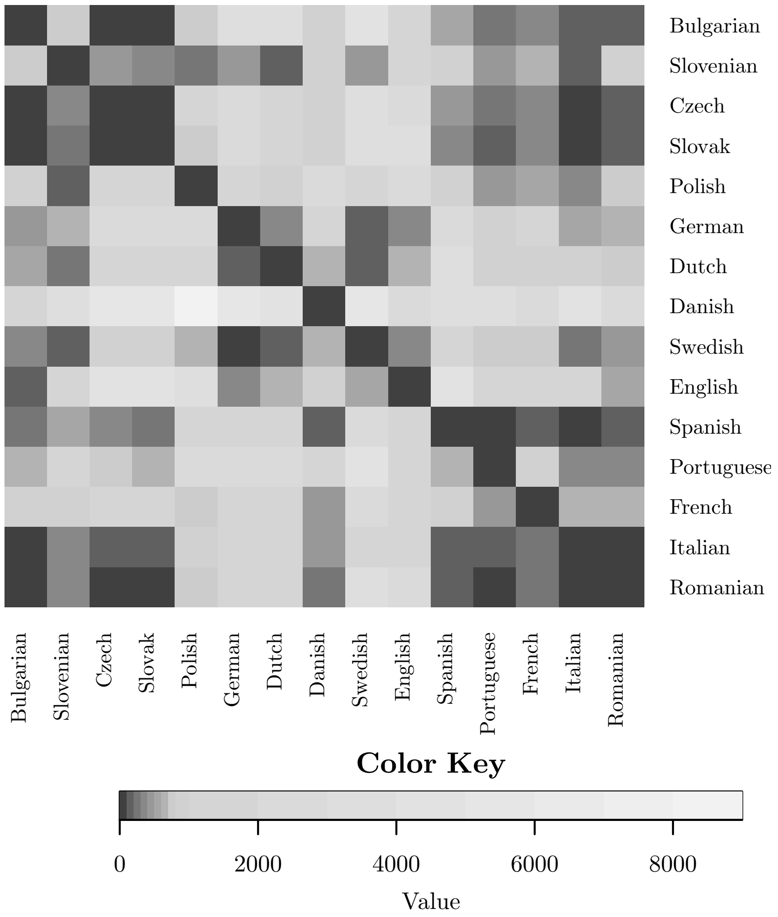


Figure 12: Heatmap of distances. The language of the model is signified by the column and the language of the corpus is signified by the row., i.e. this describes distances both ways.

Taking the out-group perspective, Italian and Romanian are close to all other Slavic languages except for Polish. In relative to other languages, Danish model performs well on the corpora of Romance languages Spanish, French, Italian and Romanian while the models of these languages perform poorly on the Danish corpus. In general, the outliers of these languages, in the sense that they are distant from their own language groups are Danish, French, Polish and Slovene.

To understand why Danish, French and Polish, the least fitting languages in their respective language families, don't fit into their language families as well as other languages, we took a further look at how much of an overlap there is in their phonology with other languages in their families in terms of shared phonemes. Danish language has a curious phonology and clearly differing set of phonemes in our transcription, which explains its distance from its own language family. For example, glottal stop ʔ, guttural r ʁ, and vowels e, ɐ̯, and o only exist in Danish of all Germanic transcriptions. These phonemes together make up 21.4% of the Danish transcription. One has to note, though, that Swedish includes the long forms of vowels e, and o (eː, oː,) but our system considers the short and long forms of vowels as strictly different phonemes.

Polish transcription differs significantly from other Slavic languages. Phonemes ɔ, ɨ, ɛ, ɲ^j, w are either missing or very rare in Bulgarian, Czech and Slovak transcriptions, and they make up 22.9% of phonemes in the Polish transcription. Polish and Slovenian are more similar to each other, sharing the phonemes ɔ, ɛ and ʋ, which are either missing or very rare in the other Slavic transcriptions. Polish and Slovene have relatively low mutual perplexity (Polish-Slovene 256.28, Slovene-Polish 193.11) and seem to not be missing as many phonemes as the other Slavic languages from each other (13.2%, 16.3%). In comparison, Czech transcription is missing 28.8% of the Polish transcription phonemes.

French has a number of common phonemes which don't exist in other Romance transcriptions. The other Romance languages are lacking the guttural r ʁ, nasal vowels ã, ɔ̃, ɛ̃, retracted vowels ə̠, a̠, e̠, and vowel y which only exists in French and Portuguese transcriptions. These phonemes make up 18.5% of phonemes in the French transcription.

Given the results and further analysis of the phonetic spaces, we believe perplexity works relatively well as a distance measure for our use case of detecting language families. The phoneme distributions of the languages affect the trigram model as evident from the correlation between cross entropy and trigram perplexity. Using phonetic transcriptions as it is between languages is problematic partly due to differences in transcription and partly due to small differences between phonologies between languages. By overcoming the hurdles caused by differing phoneme sets, be the reason behind them whichever, one should be able to significantly improve the overall performance of the methods. For example, by mapping Polish phonemes into their closest corresponding phonemes in other Slavic languages the perplexities can be significantly reduced.

We did a limited experiment on this by mapping ɔ to o and ɛ to e, which together make up 16.3% of phonemes in the Polish transcription, and distances to Bulgarian, Czech and Slovakian were reduced from 785.23, 1126.86 and 779.23, to 205.69, 102.77 and 72.19. These reductions are significant and show that there is a window of improvement if the phonetic spaces could be aligned to overcome these symbol set differences. Even though the phonemes mapped are actually different and describe different sounds, in this case we could also think of this mismatch as a byproduct of a too narrow transcription for the use case.

The tests were replicated using a similar bigram model (interpolation between bigram model and an additively smoothed unigram model), and in a parallel set of experiments using an LSTM neural network model, which is not described here. The bigram model's distances were more evenly distributed and generally higher than using the other models. The correlation coefficient between cross entropy and bigram perplexity was 0.40. Between trigram and bigram models the trigram model had slightly higher differences between distances, meaning the language families stood out from the heatmap clearer. The differences were not very big, however.

The LSTM model was able to distinguish Slavic and Germanic language families well, but it couldn't distinguish the Romance family at all. A benefit of the LSTM model was its "binary" behavior in the sense that it clearly worked on some language pairs and not at all on others, i.e. there's less vagueness in analyzing its performance. Furthermore, the LSTM model found that Italian and Romanian were very closely related to almost all languages, somewhat similarly to the trigram model. The trigram model also found Italian and Romanian close to both Slavic and Romance languages. Using the LSTM model Slovenian had low distances to many Germanic languages, but it wasn't as close to other Slavic languages, similarly to the trigram model.

The correlation between cross entropy and LSTM model's perplexity was

A	B	A → B	B → A	Mean	Max
Czech	Slovak	20.99	19.45	20.22	20.99
Estonian	Finnish	57.62	60.81	59.22	60.81
Estonian	Italian	65.97	88.83	77.40	88.83
Bulgarian	Slovak	70.37	96.93	83.65	96.93
Greek	Romanian	97.73	48.35	73.04	97.73
Greek	Spanish	97.87	38.17	68.02	97.87
Bulgarian	Czech	82.21	98.79	90.50	98.79
Estonian	Slovak	104.56	91.88	98.22	104.56

Table 7: Top eight language pairs using the trigram model, ordered by maximum distance.

weak with a Spearman's correlation coefficient of 0.21. Interestingly, the correlation between trigram perplexity and LSTM perplexity was a relatively high 0.57, however, this was not studied further. The major difference between the linear models where cross entropy is the predictor and either trigram perplexity or LSTM perplexity is the predicted are the high outliers of the LSTM model. The most extreme of these outliers is the Finnish model on Danish corpus. The perplexity of Finnish model on a Danish corpus is the highest of all distances we have seen in our experiments: 1.5 million. The perplexities are overall lower for the LSTM model than for the trigram model. The LSTM model used is not further documented in this thesis as it is not the work of the author. The results are reported here to discuss the similarities between the behaviors of trigram and LSTM models.

4.2.2 Detecting Closely Related Languages

As evident from Section 4.2.1 Czech and Slovak languages jump out as outliers due to their low mutual distance. The other known language pair which jumps out, when using the whole set of Europarl languages is Estonian and Finnish, as seen on Table 7. Czech and Slovak languages are both part of the West Slavic language group, forming their own Czech-Slovak subgroup. The languages are largely mutually intelligible and similar in both grammar and vocabulary. Estonian and Finnish are both Finno-Ugric languages, part of a smaller Finnic group of languages.

To shed light on the question whether language modeling is at all useful for detecting closely related languages, a list of closest languages was also calculated using a unigram model. We can notice that the two closest

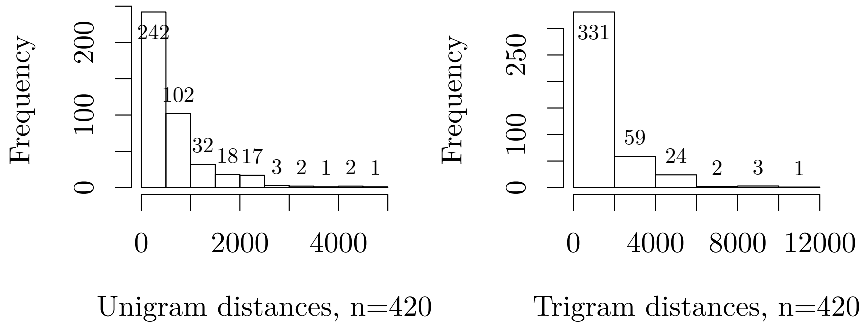


Figure 13: Distance distributions for unigram and trigram models between all languages, excluding self distances.

language pairs Czech and Slovak, and Estonian and Finnish, are again at the top (see Table 8). However, when using the unigram model, the perplexity values are quite a bit closer to each other: the pair Estonian and Italian is suspiciously close to Estonian and Finnish pair. This can be explained in two ways.

The first evident explanation is the distribution of perplexities for both models: unigram model's perplexities are all in all closer to each other and generally lower than the trigram model's (see Figure 13). The distribution of distances seems rather similar, but highest unigram perplexities are around 5000 while trigram perplexities span up to 15000. The other explanation is that the trigram model captures properties of the languages not present in the unigram model and due to the similarity of the languages places them closer to each other than to the other languages in the test.

Generally, the perplexity values for a trigram model are expected to be more extreme than the perplexity values for a unigram model. Trigram model applied on its own language corpus always performs better than a unigram model applied on its own language corpus, i.e. trigram model is able to use its learned contextual information to its advantage when predicting the next item. This is in contrast to cases where it cannot use its contextual information and gets a higher perplexity than a unigram model: it is in a sense a tighter fit to a language. Using this information, one can hypothesize that if a higher order n-gram model performs better than a lower order n-gram model on a foreign language, the languages are also a tighter fit with

A	B	A → B	B → A	Mean	Max
Czech	Slovak	42.75	39.83	41.29	42.75
Estonian	Finnish	52.52	60.35	56.43	60.35
Estonian	Italian	54.36	62.63	58.49	62.63
Estonian	Slovak	71.96	72.84	72.40	72.84
Romanian	Slovak	82.55	73.78	78.16	82.55
Czech	Estonian	85.51	78.32	81.91	85.51
Greek	Romanian	87.56	43.18	65.37	87.56
Bulgarian	Slovak	60.01	88.42	74.22	88.42

Table 8: Top eight language pairs using the unigram model, ordered by maximum distance.

regards to each other: there are properties in the languages that are useful for predicting the other language.

Building on top of this hypothesis, the pair Czech and Slovak expectedly gets a lower distance score using the trigram model than using the unigram model. The Pair Estonian and Finnish stays roughly the same while the rest of the language pairs get higher distance scores in the trigram model, which indicates that the things the higher order model learns don't carry over to other languages. Especially the third language pair in the ranking, Estonian and Italian gets a clearly higher distance measure using the trigram model than using the unigram model.

Using the perplexity-improvement hypothesis, we can deduce that the language models detect the similarity of Czech and Slovak very well. Pair Estonian and Finnish stands out as well, even though distance from Finnish to Estonian in the trigram model is slightly higher than in the unigram model. Other language pairs don't come up in this simplistic ranking at all.

4.2.3 Reproducing Family Trees

When the distance matrix from Section 4.2.1 as seen in Figure 12 is represented as a tree (see Figure 14), or a dendrogram, certain groups pop up. Germanic languages are rather well clustered, except for Danish. Romance and Slavic languages get mixed up in two different main branches with Czech and Slovak, and Slovenian and Polish being grouped together. Romance languages are clustered in an inconsistent manner with either Danish or Slavic languages among them in subtrees. The reason for the relative proximity of Danish and Spanish in this clustering is explained by the Danish model

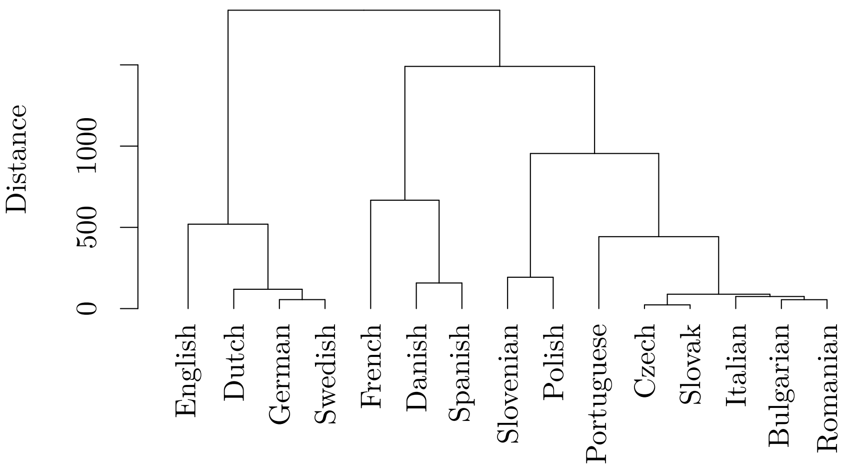


Figure 14: Reconstructed family tree from the three language families distance matrix using UPGMA clustering.

working well on the Spanish corpus.

We use the *Unweighted Pair Group Method with Arithmetic Mean* [Sok58] (UPGMA) hierarchical agglomerative clustering method from the R Statistical Computing system [3], which works from the leaves up, always creating a cluster of the two closest items and ending when everything is in a cluster. It derives new distances from the newly created cluster to items not in the cluster by averaging the distances from cluster members to the items not in the cluster.

In general, we believe our methods work for mutually similar languages, but have trouble with languages less similar. For example, Germanic languages Dutch, German and Swedish behave as we expect, as do the Slavic languages Bulgarian, Czech and Slovak. The same is exhibited by Romance languages Portuguese, Italian and Romanian, although these last groups of three are mixed in the same subtree. When we go further apart from the successfully grouped three-language subtrees, the tree-distance-averaging clustering method cannot find the canonical literature-backed [Cam98, p. 168] tree shape we would like to see.

[3] https://stat.ethz.ch/R-manual/R-devel/library/stats/html/hclust.html

5 Conclusions

In this thesis we analyzed languages using automatic phonetic transcriptions by information-theoretic means: computed distances between languages and treated language model's perplexity measure as a distance measure. We experimented with n-gram models, studied their sample size constraints, studied how languages' phoneme sets and phoneme distributions affect the methods, and finally studied how our results correspond with current literature.

The Digital Language Typology project, which this thesis is a part of, studies low-resourced languages. As the focus of the project is on speech as opposed to text corpora, we needed a framework that would facilitate working with generic symbol sequences and wouldn't rely on any external resources. Language modeling and perplexity as a distance measure were a good match for these criteria.

We made two main sets of experiments. The first of them studied sample size requirements: how big a speech corpus would be needed to properly train a language model and compute a distance measure between a model and a corpus. The second set of experiments studied the feasibility of the approach as a whole: can it produce typological or language relationship hypotheses? The sample size experiment worked by finding how long a sequence of phonemes is required to reliably reach a perplexity value within a given threshold. The feasibility experiment relied on the qualitative analysis of heatmaps and ordered lists of distance measures, and comparisons to linguistic literature. Additionally, statistical analysis was used to illuminate the details of relationships between cross entropy and perplexity to quantify the effect of phoneme distributions on different language distance measures.

We found that n-gram models of lower n, like unigram and bigram models can be trained with smaller training samples than for example trigram models. However, the lower the n, the larger a test set one needs to estimate the model's perplexity on a corpus. The learning behaviors between languages vary greatly. For instance, a Spanish trigram model can be trained with the same sized sample that is required to train a Lithuanian bigram model. The differences between languages in test samples are even more pronounced, with some languages requiring a sample larger by order of magnitude than others. Both learning and test samples behaved similarly: Romance languages typically required small and Slavic languages large samples. We discovered that trigram models can be trained on roughly 20 minutes of speech data given a similar sized vocabulary of speech symbols and a similar rate of symbols per minute as in our corpus. A speech sample of 20 minutes is short compared to the corpora required for other speech systems tasks, for example in the training of speech recognition systems.

We found that language models work on phonetic transcriptions and should similarly work on other speech sequences. The central limitation in the use of language models is the natural lack of a common symbol system between languages: there is no true common phoneme set. The interpretation of symbol equivalence between languages is only an approximation. The effect of differing symbols for corresponding (or near corresponding) sounds between languages are small in cases where the phonemes are not very common, but in case of common phonemes, perplexity measure is heavily affected. If phonemes have different symbols, but are genuinely in similar roles in compared languages, they can be mapped across to significantly lower the perplexity measure between the languages. We also found there to be a moderate correlation between the cross entropy measure between the phoneme distributions of two languages and their trigram perplexity.

Despite the effect of phoneme sets on the n-gram model performance, we were able to detect and analyze language families. We were also able to detect closely related languages by comparing the distances obtained using both bigram and trigram models. For all languages the trigram model performed better than the bigram model on their own language corpora. Only on related language pairs Czech-Slovak and Estonian-Finnish was trigram perplexity almost the same or better than bigram perplexity. This is in contrast to most language pairs, where the trigram model typically performs clearly worse than the bigram model.

We know from prior work that language models work for similar tasks as ours when applied on text, and we show that they also function for phonetic transcripts. It is, however, an open research question how they work on other speech sequences. As evident from our work, the vocabulary, or the set of symbols in the sequences being studied is in key role in determining how one can go beyond a single language, or a single method of transcription, be the transcription a phonetic one or some other representation of speech.

It can be said that our current approach cannot achieve what traditional typology does, because it doesn't provide a detailed analysis of what are the distinguishing features between languages. When thinking in terms of n-gram models, the language model type employed in this thesis, more detailed analyses, for example on the sources of perplexity (what things are unexpected), or on the sources of clarity (what things were expected) are possible. However, the clearest future continuation for this work is finding information-theoretic methods for bridging the gaps caused by the common phoneme set problem.

This work pioneered the use of automatic phonetic transcriptions on multiple data-driven language typology tasks and showed how language models can be used in conjunction with perplexity as a language distance measure and how language distances can be used to make language relationship hypotheses. We identified the issues caused by differing phoneme sets between languages and studied the effects of this in relation to the results of our methods. We showed that language models and perplexity are a useful tool, but also acknowledge that their usefulness is limited especially because one cannot easily explain their results. It would require further studies and method development to open up the models to inspection and to properly highlight the sources of perplexity between languages.

The unigram, bigram, trigram models, and distance computation code were written by the author and will be made available in source form [4].

[4] https://github.com/guaq/data-driven-language-typology

References

[AL73] Altmann, G. and Lehfeldt, W.: *Allgemeine Sprachtypologie.* Wilhelm Fink, München, 1973.

[AM11] Abramov, Olga and Mehler, Alexander: *Automatic Language Classification by means of Syntactic Dependency Networks.* Journal of Quantitative Linguistics, 18(4):291–336, 2011.

[Ant89] Anttila, Raimo: *Historical and Comparative Linguistics.* John Benjamins Publishing, 1989, ISBN 978-9027235572.

[BFG05] Bryant, D., Filimon, F., and Gray, R.: *Untangling our past: languages, trees, splits and networks.* The evolution of cultural diversity: A phylogenetic approach, pages 67–83, 2005.

[BJM83] Bahl, L.R., Jelinek, F., and Mercer, R.L.: *A Maximum Likelihood Approach to Continuous Speech Recognition.* IEEE Transactions on Pattern Analysis and Machine Intelligence, (2):179–190, 1983.

[BPK92] Batagelj, Vladimir, Pisanski, Tomaž, and Keržič, Damijana: *Automatic Clustering of Languages.* Computational Linguistics, 18(3):339–352, 1992.

[Cam98] Campbell, Lyle: *Historical Linguistics: An Introduction.* Edinburgh University Press, Edinburgh, 1998.

[CG96] Chen, Stanley F and Goodman, Joshua: *An empirical study of smoothing techniques for language modeling.* In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 310–318. Association for Computational Linguistics, 1996.

[Cro03] Croft, William: *Typology and Universals.* Cambridge University Press, 2003, ISBN 9780521004992.

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

[DKB92] Dyen, Isidore, Kruskal, Joseph B, and Black, Paul: *An Indoeuropean Classification: A Lexicostatistical Experiment.* Transactions of the American Philosophical Society, 82(5):iii–132, 1992.

[EK06] Ellison, T. Mark and Kirby, Simon: *Measuring Language Divergence by Intra-lexical Comparison.* In *Proceedings of the 21st*

International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics, pages 273–280. Association for Computational Linguistics, 2006.

[Emb00] Embleton, Sheila: *Lexicostatistics/glottochronology: from Swadesh to Sankoff to Starostin to future horizons.* In Renfrew, C., McMahon, A., and Trask, L. (editors): *Time Depth in Historical Linguistics*, pages 143–165. The McDonald Institute for Archaeological Research, Cambridge, 2000.

[GA03] Gray, R.D. Russell D and Atkinson, Q.D. Quentin D: *Language-tree divergence times support the Anatolian theory of Indo-European origin.* Nature, 426(6965):435–439, nov 2003.

[GPA17] Gamallo, Pablo, Pichel, José Ramom, and Alegria, Iñaki: *From language identification to language distance.* Physica A: Statistical Mechanics and its Applications, 484:152–162, 2017.

[Int99] International Phonetic Association: *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet.* Cambridge University Press, 1999.

[JBM75] Jelinek, Frederick, Bahl, Lalit, and Mercer, Robert: *Design of a Linguistic Statistical Decoder for the Recognition of Continuous Speech.* IEEE Transactions on Information Theory, 21(3):250–256, 1975.

[JM80] Jelinek, Fred and Mercer, Robert L.: *Interpolated estimation of Markov source parameters from sparse data.* In *Proceedings, Workshop on Pattern Recognition in Practice*, pages 381–397, Amsterdam, 1980. North Holland.

[JR85] Juang, B H and Rabiner, Lawrence R: *A Probabilistic Distance Measure for Hidden Markov Models.* AT&T Technical Journal, 64(2):391–408, 1985.

[Kes05] Kessler, Brett: *Phonetic comparison algorithms.* Transactions of the Philological Society, 103(2):243–260, 2005, ISSN 1467-968X.

[Kit99] Kita, Kenji: *Automatic Clustering of Languages Based on Probabilistic Models.* Journal of Quantitative Linguistics, 6(2):167–171, 1999.

[Koe05] Koehn, Philipp: *Europarl: A Parallel Corpus for Statistical Machine Translation.* In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86. AAMT, AAMT, 2005.

[Kra92] Krauss, Michael: *The world's languages in crisis.* Language, 68(1):4–10, 1992.

[Kru83] Kruskal, Joseph B: *An Overview of Sequence Comparison: Time Warps, String Edits, and Macromolecules.* SIAM Review, 25(2):201–237, 1983.

[LE80] Li, K and Edwards, T: *Statistical models for automatic language identification.* In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'80.*, volume 5, pages 884–887. IEEE, 1980.

[Leh93] Lehmann, Winfred P.: *Theoretical Bases of Indo-European Linguistics.* Routledge, London, 1993.

[Leh09] Lehtinen, Jyri: *Language change as an evolutionary process.* Master's thesis, University of Helsinki, Faculty of Arts, Department of General Linguistics, 2009.

[Lev66] Levenshtein, V. I.: *Binary codes capable of correcting deletions, insertions, and reversals.* Soviet Physics—Doklady 10, 707–710. Translated from Doklady Akademii Nauk SSSR, pages 845–848, 1966.

[LHK$^+$14] Lehtinen, J., Honkola, T., Korhonen, K., Syrjänen, K., Wahlberg, N., and Vesakoski, O.: *Behind family trees.* Language Dynamics and Change, 4(2):189–221, 2014.

[Lid20] Lidstone, G.: *Note on the general case of the Bayes–Laplace formula for inductive or a posteriori probabilities.* Transactions of the Faculty of Actuaries, 8:182–192, 1920.

[LSV09] Lemey, P., Salemi, M., and Vandamme, A.M.: *The Phylogenetic Handbook: A Practical Approach to Phylogenetic Analysis and Hypothesis Testing.* Cambridge University Press, 2009, ISBN 9781139478618.

[McM03] McMahon, April and McMahon, Robert: *Finding Families: Quantitative Methods in Language Classification.* Transactions of the Philological Society, 101(1):7–55, 2003.

[MH12] McEnery, Tony and Hardie, Andrew.: *Corpus Linguistics: Method, Theory and Practice.* Cambridge University Press, 2012, ISBN 9780521838511.

[MS99] Manning, Christopher D. and Schütze, Hinrich: *Foundations of Statistical Natural Language Processing.* MIT Press, Cambridge, MA, USA, 1999, ISBN 0-262-13360-1.

[NH97] Nerbonne, John and Heeringa, Wilbert: *Measuring Dialect Distance Phonetically.* In *Proceedings of the Third Meeting of the ACL Special Interest Group in Computational Phonology (SIGPHON-97)*, pages 11–18, 1997.

[PC98] Ponte, Jay M and Croft, W Bruce: *A language modeling approach to information retrieval.* In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281. ACM, 1998.

[PL05] Port, Robert F and Leary, Adam P: *Against Formal Phonology.* Language, 81(4):927–964, 2005.

[PS08] Petroni, Filippo and Serva, Maurizio: *Language distance and tree reconstruction.* Journal of Statistical Mechanics: Theory and Experiment, 2008(08), 2008.

[RS09] Rama, T and Singh, AK: *From Bag of Languages to Family Trees From Noisy Corpus.* Proceedings of the RANLP 2009, pages 355—-359, 2009.

[Sch04] Scholz, F. W.: *Maximum Likelihood Estimation.* Encyclopedia of Statistical Sciences, 2004.

[Set01] Setälä, E.N.: *Über Transskription der finnisch-ugrischen Sprachen: Historik und Vorschläge.* In *Finnish-Ugrische Forschungen*, pages 15–52. Helsingfors, Leipzig, 1901.

[Sok58] Sokal, Robert R: *A statistical method for evaluating systematic relationship.* University of Kansas Science Bulletin, 28:1409–1438, 1958.

[Spe04] Spearman, C.: *The Proof and Measurement of Association between Two Things.* The American Journal of Psychology, 15(1):72–101, 1904.

[SR62] Sokal, Robert R. and Rohlf, F. James: *The Comparison of Dendrograms by Objective Methods.* Taxon, 11(2):33–40, 1962.

[Swa34] Swadesh, Morris: *The Phonemic Principle.* Language, 10(2):117–129, 1934.

[Swa50] Swadesh, Morris: *Salish Internal Relationships.* International Journal of American Linguistics, 16(4):157–167, 1950.

[Swa52] Swadesh, Morris: *Lexico-Statistic Dating of Prehistoric Ethnic Contacts: With Special Reference to North American Indians and Eskimos.* Proceedings of the American Philosophical Society, 96(4):452–463, 1952.

[Wik17a] Wikipedia: *Cognate – Wikipedia, the free encyclopedia*, 2017. `https://en.wikipedia.org/w/index.php?title=Cognate&oldid=801516595`, [Online; accessed 21-Sep-2017].

[Wik17b] Wiktionary: *Appendix:Danish Swadesh list – Wiktionary, the free dictionary*, 2017. `https://en.wiktionary.org/w/index.php?title=Appendix:Danish_Swadesh_list&oldid=47206280`, [Online; accessed 7-Sep-2017].

[Wik17c] Wiktionary: *Appendix:Swadesh lists – Wiktionary, the free dictionary*, 2017. `https://en.wiktionary.org/w/index.php?title=Appendix:Swadesh_lists&oldid=47165671`, [Online; accessed 7-Sep-2017].

[Zip32] Zipf, George Kingsley: *Selected studies of the principle of relative frequency in language.* Harvard University Press, 1932.