PhD Dissertation

International Doctorate School in Information and Communication Technologies

DISI - University of Trento

Data Driven Models for Language Evolution

Antonella Delmestri

Advisor:

Prof. Maurizio Marchese

University of Trento

Co-Advisor:

Prof. Nello Cristianini

University of Bristol

April 2011

Abstract

Natural languages that originate from a common ancestor are genetically related, words are the core of any language and cognates are words sharing the same ancestor and etymology. Cognate identification, therefore, represents the foundation upon which the evolutionary history of languages may be discovered, while linguistic phylogenetic inference aims to estimate the genetic relationships that exist between them.

In this thesis, using several techniques originally developed for biological sequence analysis, we have designed a data driven orthographic learning system for measuring string similarity and we have successfully applied it to the tasks of cognate identification and phylogenetic inference.

Our system has outperformed the best comparable phonetic and orthographic cognate identification models previously reported in the literature, with results statistically significant and remarkably stable, regardless of the variation of the training dataset dimension. When applied to phylogenetic inference of the Indo-European language family, whose higher structure does not yet have consensus, our method has estimated phylogenies which are compatible with the benchmark tree and has reproduced correctly all the established major language groups and subgroups present in the dataset.

Keywords

Cognate identification, phylogenetic inference, language evolution, substitution matrices, *PAM-like* matrices.

Acknowledgements

I would like to thank all the following people who, in different ways, have contributed to the realisation of this thesis.

- My advisor *Maurizio Marchese* and my co-advisor *Nello Cristianini*, for helping me to pursue this project and bring it to completion.

- The members of my advisory committee, *Andrea Sgarro* and *Liviu Dinu*, for their thorough review of this dissertation.

- *Clinical Trial Service Unit* at Oxford University and my colleagues *Christina Davies*, *Michael Lay*, *Philip Morris* and *Paul McGale*, for their continued support and useful suggestions.

- *Grzegorz Kondrak* for making available his version of the test dataset, *Brett Kessler* for commenting on his lists, *Quentin Atkinson* for supplying and commenting on the Hittite and Tocharian lists, *Geoff Nicholls* for providing some of his papers and datasets, *Martijn Wieling* for his comments and material on the *PHMM* model.

- *Ana Fortun* for her contribution to the linguistic-inspired substitution matrix, *Alain Thomas* and *Dario Brancato* for their linguistic advice.

- *Adam*, my partner, for his amazing support, deep understanding, extraordinary patience and invaluable help in reviewing my written English.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Language is a defining feature that distinguishes modern humans from all other species, is a carrier of culture and plays a key role in communication. Because of its central function in human evolution, language has always aroused a high level of interest and much debate among scholars of different disciplines in the sciences and humanities. The analogy of language evolution with species evolution [3] has generated a growing attention in the scientific community as a result of the extraordinary progress of computational molecular biology in the field of genomes. Because of this close analogy, the study of language evolution is being increasingly and successfully explored using techniques developed in evolutionary biology [53, 52, 87, 109, 6].

Computational historical linguistics studies the evolution of language, and aims to establish the existence and degree of genetic relationships between speech varieties. It provides an interdisciplinary approach that involves fields as diverse as linguistics, computer science, artificial intelligence, molecular biology, statistics and mathematics, to list a few.

The main objective of this thesis has been the investigation of evolutionary biological models and their application to the study of language evolution, using a machine learning, data driven approach.

1.1 Computational historical linguistics

Languages originating from a common ancestor are genetically related. Historical linguistics aims to build phylogenies of these languages [53, 52] and to reconstruct as far as possible their common ancestors or *proto-languages* [77], in the absence of historical records.

The evolution of language may be analysed through its phonological, lexical and morphological changes, generally represented as a set of features, called characters [106]. Common lexical characters used in historical linguistics are strict or genetic *cognates* [80], which are words deriving from a common ancestor and sharing the same etymological origin [77]. Cognates originate from a *vertical* transmission and do not include borrowings, which are words loaned from other languages through a *horizontal* transmission [138]. However, in many areas of natural language processing, the term *cognates* has a wider meaning and also comprises loans [80]. As vertical and horizontal transmissions are both significant, cognates and borrowings play crucial, but different, roles in the investigation of language evolution [106].

Genetic cognates have the advantage of exhibiting the same character state only as a result of an evolutionary relationship and not because of parallel developments or back mutations [106]. For this reason, the study of genetic cognate words provides evidence of historical relationships between languages and may be used to identify genetic relationships between speech varieties and to infer phylogenies.

The synergy between cognate identification and phylogenetic inference, both representing very promising applications of computational historical linguistics, may contribute to the tracing of language evolution and to the investigation of the origin of language. In this thesis, we have focussed on the exploration of this synergy through the application of evolutionary biological models to the cognate identification problem and their deployment in the task of linguistic phylogenetic inference.

1.2 Cognate identification and phylogenetic inference

In order to make a contribution to the fascinating and intricate problem of language evolution, we have investigated the fields of cognate identification and phylogenetic inference, as the latter depends on the former, with respect to the Indo-European language family.

Several different approaches to the cognate identification problem have been proposed in the literature and phonetic or orthographic methodologies have been applied, as well as learning algorithms or manually-designed procedures [76, 86, 83, 80]. In automatic cognate identification, as in computational molecular analysis, strings may be successfully studied by inexact string matching techniques, which allow their similarity to be measured and their optimal alignment to be found. Global or local alignment algorithms, widely used in biological sequence analysis [41], usually consist of a substitution matrix and a procedure that finds the optimal pairwise alignment. The significance of the resulting alignment depends greatly on the chosen scoring scheme [56].

Phylogenies are evolutionary trees and phylogenetic inference aims to estimate the genetic relationships between taxa, which in principle may be species, languages or other entities [46]. In computational historical linguistics, a phylogenetic tree represents one hypothesis about the evolutionary relationships among groups of languages, based upon similarities and differences in their characters [106]. Methods for linguistic phylogenetic inference estimate the evolutionary history of languages using information that is generally coded in a distance matrix or in a character matrix. Depending on this coding, methods are classified as *distance-based* methods or *character-based* methods and most of them are guaranteed to reproduce the true evolutionary tree under certain conditions. The evaluation of linguistic phylogenetic inference is very difficult because the true evolutionary history is not generally entirely known, even for the best understood language families. The "*Compatible resolution*" and "*No missing subgroups*" criteria are considered essential and desirable, respectively [106], when evaluating linguistic phylogenetic estimations.

1.3 Proposed solution

We have designed a new learning system for measuring string similarities, inspired by biological sequence analysis, and we have applied it to the tasks of cognate identification and phylogenetic inference. We have developed our proposal using data in orthographic format based on the Roman alphabet. However, it may easily be adapted to any alphabetic system, including the phonetic alphabet, if data were available. The system consists of three main modules, each presenting an original aspect:

- The first module is a pairwise aligner that performs sensible global alignments on cognate pairs and prepares a meaningful training dataset, guided by a novel linguistic-inspired substitution matrix. This 26-by-26 matrix aims to represent the a priori likelihood of transformation between pairs of characters in the Roman alphabet and tries to code well-known systematic sound changes left in written Indo-European languages. This component is necessary because there are no databases of aligned cognate words available for linguistic studies.

- The second module is a generator of substitution matrices that we have implemented using several techniques, including *Maximum Likelihood*, *Absolute Frequency Ratio*, *Pointwise Mutual Information* and *PAM-like*. For the latter, which had the superior performance, we have developed a new technique inspired by the *Point Accepted Mutation* (*PAM*) method. This was designed by *Margaret Dayhoff* and co-workers [30, 31, 32] and is widely used for amino acid sequence analysis.

- The third module is a pairwise aligner that measures the similarity between words by using the generated substitution matrices and a novel family of parameterised string similarity measures. The similarity measures derive from different normalisations of a generic scoring algorithm and take into account the similarity of each string with itself with the aim of eliminating, or at least reducing, the bias due to different string length.

We have successfully applied this learning system to the task of cognate identification and phylogenetic inference.

To test the ability of our string similarity measuring system in the task of cognate identification, we have evaluated the likelihood that two words with the same meaning from two different languages were cognates, by calculating their similarity score. We have sorted the scores computed for each language pair, taking into account the alphabetic order when more than one word pair has shown the same rate. To evaluate the accuracy of the system in identifying correctly cognate words, we have not used a threshold, because this may be influenced by the type of application, the method used and the degree of language relatedness [80]. Instead we have utilised the *11-point interpolated average precision* [90], which is an evaluation metric originally built for ranking computation in the field of *Information Retrieval*. This measure has frequently been used in the field of cognate identification by other studies, with which we wanted to make a direct comparison.

We have also applied this learning system for measuring string similarities to the task of phylogenetic inference, in order to test its efficacy against documented aspects of the Indo-European language family. We have utilised the similarity scores between word pairs to calculate similarity scores between language pairs. We have then converted these similarity scores into distance scores, which we have employed in *distance-based* methods to estimate phylogenies.

1.4 Innovative aspects

Our main contributions to the study of language evolution are:

1. The development of a new learning system for measuring word similarity that has been inspired by biological sequence analysis. The system benefits from several original features: a linguistic-inspired substitution matrix to align globally the training dataset, a scoring matrix generator to learn substitution parameters and a novel family of string similarity measures to improve the alignment and rate of word pairs. In particular, for learning *PAM-like* matrices we have developed a technique similar to the *PAM* method, designed by *Dayhoff et al.* [30, 31, 32], which is one of the gold standard in amino acid sequence analysis.

2. The successful application of this learning system to the task of cognate identification. Indeed, the system has shown its superior performance and higher consistency across different language

pairs, when evaluated against the best comparable phonetic and orthographic models previously proposed in the literature [76, 87, 83].

3. The assessment of the system's robustness, regardless of the dimension of the training dataset. The system has been tested by increasing the training dataset dimension by a factor of approximately 100, which implied extending the number of Indo-European languages by a factor of approximately 13. The results have been impressively stable and have shown no relevant difference in the performance.

4. The investigation of the statistical significance of our results when compared with earlier proposals and with each other. The outcome has shown, with strong and good evidence, that our results are more accurate than those previously reported in the literature and that the training dataset dimension does not influence their statistical significance.

5. The application of the proposed methodology to the task of phylogenetic inference in order to test its effectiveness against recognised aspects of the Indo-European language family, whose higher structure is still controversial. Our results have reproduced correctly all the established major language groups and subgroups present in the dataset and have shown to be compatible with the Indo-European benchmark tree. In doing this, our outcome has successfully met the required linguistic evaluation criteria and, in addition, it has included some of the supported higher-level structures.

These results have been presented in the following publications:

- *Delmestri A.*, *Cristianini N.*, "String Similarity Measures and PAM-like Matrices for Cognate Identification", *Bucharest Working Papers in Linguistics*, vol. XII, no. 2, pp. 71-82, 2010.

- *Delmestri A.*, *Cristianini N.*, "Robustness and Statistical Significance of PAM-like Matrices for Cognate Identification", *Journal of Communication and Computer*, vol. 7, no. 12, pp. 21-31, 2010.
- *Delmestri A.*, *Cristianini N.*, "Linguistic Phylogenetic Inference by PAM-like Matrices", Submitted.

1.5 Structure of the thesis

Chapter 1 provides a brief overview of this thesis. Firstly, it introduces computational historical linguistics as a research field and it presents cognate identification, together with phylogenetic inference, as key problems in the study of language evolution. It then gives a concise description of the string similarity measuring system that we have designed using a machine learning, data driven approach, inspired by biological sequence analysis. Finally, it describes our main contributions to the state of the art in the fields of cognate identification and phylogenetic inference, together with our related publications.

Chapter 2 reports the state of the art in both the fields of cognate identification and phylogenetic inference. For the former, it reviews the most authoritative studies proposed in the literature, covering orthographic and phonetic methods, as well as learning systems and static procedures. For the latter, it describes the more interesting *distance-based* and *character-based* methods introduced by other scholars to date. It also presents the two main datasets recommended for linguistic studies.

Chapter 3 introduces the problems of cognate identification and phylogenetic inference as fundamental fields in computational historical linguistics. The cognate identification problem is expressed as an approximate string matching problem, that may be studied by a string distance or string similarity approach. The task of calculating the distance or similarity between two strings is shown to be closely related to the problem of finding their optimal alignments. Particular attention is given to the similarity approach, which is the standard in biological sequence analysis and has been used in our investigation. Global and local string alignment algorithms are explained and the crucial role that substitution matrices play in them is highlighted. *PAM* matrices and *BLOSUM* matrices are briefly described as the most significant amino acid scoring schemes used in bioinformatics. Chapter 3 continues with the presentation of the phylogenetic inference problem, together with a classification of several *distance-based* and *character-based* methods. The difficulty of evaluating linguistic phylogenetic estimations is explained and evaluation criteria are provided. Finally, the Indo-European language family is introduced as the target of our phylogenetic estimation. Its particular role in the field of historical linguistics is addressed, which motivates our investigation.

Chapter 4 proposes a new learning system for measuring string similarity as a solution to both the cognate identification and the linguistic phylogenetic inference problems. The main architecture is described, together with the innovative aspects of our proposal. A novel linguistic-inspired substitution matrix is used to align sensibly the training dataset. Several techniques are utilised to learn substitution parameters, including *Maximum Likelihood*, *Absolute Frequency Ratio*, *Pointwise Mutual Information* and *PAM-like*, which has been inspired by the *PAM* method used in biological sequence analysis. A new family of parameterised string similarity measures is employed to improve the alignment and rate of string pairs. Finally, the application of the learning system to the tasks of cognate identification and phylogenetic inference is explained in detail.

Chapter 5 presents the experimental results in the tasks of cognate identification and phylogenetic inference. For the former, the datasets used are described, together with the results produced by our system. *PAM-like* models are recognised as the most successful and their robustness is assessed towards the variation of the training dataset dimension. *PAM-like* methodology results are compared with others reported in the literature, their outstanding and highly consistent performance is highlighted and their statistical significance verified. Chapter 5 also describes the experimental results in inferring phylogenies, together with the datasets used and the proposed experimental design. The compatibility with the Indo-European benchmark tree, and the correct reproduction of all the established major language groups and subgroups present in the dataset, is assessed.

Chapter 6 provides a detailed description and critical analysis of those successful studies in the field of cognate identification that share with our investigation an orthographic learning approach and that our system has outperformed.

Chapter 7 summarises the work presented in this thesis. It describes the achievements reached by our investigation and highlights the advancements to the state of the art. It also reports our conclusions and outlines our future plans.

Chapter 2

State of the art

In this chapter we review several studies in the fields of cognate identification and phylogenetic inference. Our aim is to provide a solid background of knowledge, but also to show the research history and its progression in these areas over time. For this reason, whenever possible, we follow the chronological order of the proposals. We apologise to the many valuable authors whom, owing to space limitations, we are not able to mention.

2.1 Cognate identification

Words are strings belonging to a natural language and cognates are words sharing the same ancestor and etymology. Cognate identification has been successfully applied to several tasks of computational historical linguistics, including dialectology [72, 103, 140, 141], phylogenetic inference [42, 53, 115, 52, 113, 104, 105, 123, 16, 40, 118] and *proto-language* reconstruction [22, 23, 76, 107, 77]. Moreover, it has been beneficially utilised in many different areas of natural language processing, where the term *cognates* has a wider connotation and also comprises borrowings [80]. These areas include semantic word clustering [1], bilingual lexicography [15, 67], machine translation [57, 82], lexicon induction [88, 75, 122, 97], parallel corpora sentence alignment [124, 21, 94, 96], parallel corpora word alignment [134, 78], cross-lingual information retrieval [111] and confusable drug name detection [81].

A number of different approaches to the cognate identification problem have been proposed, and orthographic or phonetic models have been used as well as learning systems or static procedures. In Sections 2.1.1 and 2.1.2 we will review the most relevant methods presented in the literature, which frequently focus on goals unrelated to historical linguistics.

2.1.1 Orthographic methods

The *Dice's similarity coefficient* [39] is a similarity measure reinvented several times in different fields, which has been frequently utilised in cognate identification. It was originally known as *Czekanowski's similarity index* [26], from the statistician and linguist who proposed it in 1913 to analyse the similarity between two samples, in order to create numerical taxonomies. *Czekanowski* applied his index to samples of phonemes and words in text corpora of different languages. In 1945, the index was introduced to the field of biological communities classification by *Dice* [39], when it assumed its current name. In 1948, it was applied to the study of ecological communities classification by *Sørensen* where it assumed the name of *Sørensen's similarity coefficient* [129]. More recently, it has also been used in the field of information retrieval [89].

In *stringology*, where a *bigram* is an ordered and contiguous pair of characters, the *Dice's similarity coefficient* (*DICE*) [39] of two strings S_1 and S_2 is defined as the ratio of twice the number of bigrams shared by the two strings to the sum of bigrams present in each string:

$$DICE(S_1, S_2) = \frac{2 * b_{12}}{b_1 + b_2} \tag{2.1}$$

where b_1 and b_2 are the numbers of bigrams in S_1 and S_2, respectively, and b_{12} is the number of shared bigrams between S_1 and S_2. Bigrams give some information about the sequence of the characters in a word that single characters do not. When compared with longer *n-grams*, which are ordered and contiguous substrings of n characters, bigrams give less information about the character proximity, but do not miss the occurrence of shorter substrings. *DICE* [39] can assume real values in the range [0,1], where 0 means no similarity and 1 maximum similarity.

Adamson and Boreham [1] pioneered one of the first attempts to identify automatically cognate words based on their orthographic form. They used *DICE* [39] as an association measure to cluster automatically sets of words from a chemical database into semantically related groups. *DICE* [39] was not chosen for its absolute value, but to order significantly the word pairs so that the higher the coefficient, the stronger the association between the words. By defining different thresholds for *DICE* [39], the authors found one that allowed a successful identification of clusters containing related words. The roots of the chemical element names contained in the word sample made the experiment significant in the field of genetically related word identification.

Simard et al. [124] made another early effort to identify automatically broad cognate words based on their orthography. Their research was functional to the task of sentence alignment in bilingual corpora, which are corpora composed of a source text along with a translation of that text into a different language. When the two different versions of the same text are finally aligned, they are called a *bitext* [60]. The authors started from the assumption that cognate words are more likely to be used as mutual translations than other pairs of words. Following the *exact string matching* approach [56], they proposed a method, later called *Truncation*, which consisted of a binary measure of broad cognateness between two words based on their matching prefix. If two words shared a common prefix at least four characters long, they were considered cognates, otherwise they were not. An obvious limitation of this approach is shown by those words that share the first four characters, but are not cognates (*false positive*), and by those words that do not share the first four characters, but are actually cognates (*false negative*).

Church [21] used *Truncation* [124] to align bilingual corpora at the character level. He assumed that characters matched across languages if they participated in broad cognates.

McEnery and Oakes [94], working in the task of sentence alignment in parallel corpora, made an attempt to identify broad cognate words between English, French and Spanish. They applied several string matching techniques, including *DICE* [39], *Truncation* [124] and a variation of the *Damerau-Levenshtein distance* (*DL*) [27], which they called *Dynamic Programming* (*DP*):

$$DP(S_1, S_2) = 1 - \frac{DL(S_1, S_2)}{\max(|S_1|, |S_2|)} \tag{2.2}$$

For each method, the authors divided the word pairs into bands according to their scores and set a threshold for them. They calculated a value of 0.9 for *DICE* [39] and a length of 8 for *Truncation* [124] in order to obtain 95% accuracy. They also discovered that *DICE* [39] performed better than the *DP* technique, which they proposed.

Brew and McKelvie [15] presented an application for lexicography in the task of word-pair extraction from multilingual corpora. In order to identify English and French broad cognates, they evaluated several methods, including *DICE* [39] and five variants of it, which changed how the bigrams were defined and weighted. We have reviewed here those variations that have been used successively in the literature.

XDICE is a variant of *DICE* [39], which is applied on *extended bigrams*, consisting of the standard bigrams plus the ordered letter pairs produced from trigrams, without considering the middle letter:

$$XDICE(S_1, S_2) = \frac{2 * xb_{12}}{xb_1 + xb_2} \tag{2.3}$$

where S_1 and S_2 are two strings, xb_1 and xb_2 the number of *extended bigrams* in S_1 and S_2, respectively, and xb_{12} the number of shared *extended bigrams* between S_1 and S_2.

XXDICE is an extension of *XDICE*, where the contribution of each shared *extended bigram* is not simply 2, but consists of the following normalisation, and *pos* is a function that returns the position of an *extended bigram* in a string:

$$\frac{2}{1 + (pos(xbigram_1) - pos(xbigram_2))^2} \tag{2.4}$$

The authors did not specify how to match the bigrams, if they are not unique within a word [79].

LCSA, as we have named it, is another similarity measure between two words defined as the ratio of the length of their *Longest Common Subsequence* (*LCS*) [137] to the Average length of the two strings. The normalisation prevents bias towards longer words:

$$LCSA(S_1, S_2) = \frac{2 * |LCS(S_1, S_2)|}{|S_1| + |S_2|} \tag{2.5}$$

The authors established a threshold for the similarity measures under test and demonstrated that *XXDICE* outperformed all the others, reaching a high precision in detecting English-French cognates.

Melamed [95], in the task of N-Best translation lexicons induction, introduced a cognate filter, setting a threshold for the *Longest Common Subsequence Ratio* (*LCSR*) of two words, which he defined as the ratio of the length of their longest common subsequence, to the length of the longest word. The normalisation again prevents bias towards longer words:

$$LCSR(S_1, S_2) = \frac{|LCS(S_1, S_2)|}{\max(|S_1|, |S_2|)} \tag{2.6}$$

Melamed [96] identified broad cognates by setting a threshold for *LCSR* in the task of bitext alignment via pattern recognition at the sentence level. The system produced mappings and alignments for a large corpus of French-English bitexts. The author suggested that his approach could be extended to the phonetic level, if phonetic transcriptions of the source texts were provided.

Tiedemann [134] proposed the automatic construction of three weighted string similarity measures, which could be used to identify broad cognates in bilingual corpora. His approach aimed to learn the recurrent spelling changes between candidate cognates, using bitext co-occurrence in Swedish-English parallel corpora. These two languages represent a good example of etymologically related languages presenting a different way of spelling. The training set consisted of reference lexicons and the test set of bilingual word pairs. The first approach identified character mappings, the second vowel and consonant subsequence mappings, and the third non-matching pair mappings. The first two methods measured matching co-occurrences by *DICE* [39], defined in terms of frequencies for each pair in the list of set mappings, and used it in the string matching functions. The third method, that outperformed the others, measured co-occurrence of non-matching pair mappings, such as Swedish *ska* and English *c* using a variant of *LCSR* [95]. The introduction of the learning aspect in the task of cognate identification represented a very significant contribution of this study.

Mann and Yarowsky [88] explored ways of using cognate pairs to create a translation lexicon and proposed an automatic induction of it via bridge languages. In order to detect broad cognate words, they experimented with three variations of the *alphabet-weight edit distance* [56] with modified costs for edit operations. In the first model, called *Levenshtein-Vowel*, the substitution operation weights between vowels were manually modified. On the contrary, the other two models were adaptively trained and the alphabet weights were learnt from a training set by a stochastic transducer, proposed by *Ristad and Yianilos* [116]. The second model, called *Levenshtein-All*, was trained on all the languages considered, while the third, called *Levenshtein-Single*, was trained on each language pair. The authors compared these three models against the *Levenshtein distance* [84] and two other methods introduced in the field of speech recognition: the stochastic transducer by *Ristad and Yianilos* previously mentioned [116] and a *Hidden Markov Model* (*HMM*) proposed by *Jelinek* [69]. The model *Levenshtein-Single* outperformed all the others in the task of cognate identification and confirmed the effectiveness of learning substitution alphabet weights.

Kondrak and Dorr [81] investigated orthographic and phonetic similarity in the task of confusable drug names identification. They proposed a family of n-gram similarity measures called *n-SIM* as a generalisation of *LCSR* [95]. *BI-SIM* and *TRI-SIM* represented the longest common subsequence of bigrams and trigrams, respectively. They also presented a family of n-gram distance measures, called *n-DIST*, as a generalisation of the *edit distance* [84] normalised by the length of the longer string. *BI-DIST* and *TRI-DIST* represented the *edit distance* between subsequences of bigrams and trigrams, respectively. The authors also introduced a generalisation of *Truncation* [124], which they called the *PREFIX coefficient.* They defined it as the length of the *Longest Common Prefix* (*LCP*) between two words, normalised by the length of the longer of the two words, to obtain a real value in the range [0,1]:

$$PREFIX(S_1, S_2) = \frac{|LCP(S_1, S_2)|}{\max(|S_1|, |S_2|)} \tag{2.7}$$

Kondrak and Dorr tested all these measures of similarity between strings plus several others, including *ALINE* [76], which is a phonetic aligner that will be discussed in Section 2.1.2. They showed that the similarity measure *BI-SIM* outperformed all the others on a test dataset containing orthographically and phonetically similar drug names. *ALINE* [76] achieved the greater accuracy on a test dataset including only phonetically similar pairs.

Mackay [86] developed a cognate identification orthographic learning system using *Pair Hidden Markov Models* (*PHMMs*) [41]. *Mackay and Kondrak* [87] tested and compared this system with several other methods, including ALINE [76], showing its superior accuracy in identifying broad cognates. These last two studies will be reviewed and discussed in detail in Chapter 6. The *PHMM* that performed better is called hereinafter only *PHMM.*

Kondrak [78] focussed on identifying broad cognate words in orthographic format in the task of word alignment in bitexts. The author proposed a variant of *LCSR* [95], called *Longest Common Subsequence Formula (LCSF).* This is a similarity measure designed to avoid, or at least mitigate, the bias towards both longer and shorter words that no normalisation, or normalisation by the length of the longer word, may produce. *LCSF* between two strings S_1 and S_2 is defined as:

$$LCSF(S_1, S_2) = \max\ (-\log\ (\begin{pmatrix} n \\ k \end{pmatrix} \begin{pmatrix} n \\ k \end{pmatrix} p^k), 0) \tag{2.8}$$

where n is the length of the longer word, k is the length of the *LCS* [137] between S_1 and S_2 and p is the probability of a match of two randomly selected letters. The author compared the accuracy in detecting broad cognates of several similarity measures, including *PREFIX* [81], *DICE* [1], *LCSR* [95] and *LCSF*, the latter consistently outperforming the others.

Inkpen et al. [67] employed thirteen orthographic similarity measures to identify automatically cognates in French and English for learning aid purposes. The similarity measures included *PREFIX* [81]; *DICE* [1]; *TRIGRAM*, which was defined as *DICE*, but worked on trigrams instead of bigrams; *XDICE* and *XXDICE* [15]; LCSR [96]; *NED*, the *edit distance* [84] normalised by the length of the longer string; a variation of the *SOUNDEX* system [58] that, after reducing all strings to a *Soundex code* of one letter and three digits, removes the zeros, truncates the resulting strings to four characters and returns the *edit distance* between two codes; *BI-SIM*, *TRI-SIM*, *BI-DIST* and *TRI-DIST* [81]. The authors collected word pairs from different sources and trained several machine learning classifiers from the *Weka package*[1], a Java open source collection of machine learning

[1] `http://www.cs.waikato.ac.nz/ml/weka/`

algorithms for data mining tasks. They tested the similarity measures and the machine learning classifiers on a test dataset and showed that many of the similarity measures reached good accuracy, outperforming the learning methods. *XXDICE* [15] achieved the better results.

Kondrak [79] presented a formal definition of the families of *n-SIM* and *n-DIST* similarity measures previously proposed by *Kondrak and Dorr* [81]. He also provided *dynamic programming* algorithms [11] for their computation. The author tested these measures against the corresponding standard unigram similarity measures to evaluate their effectiveness on three different word-comparison tasks: the identification of strict cognates, broad cognates, and confusable drug names. The results suggested that the n-gram measures outperformed their unigram equivalents.

Kondrak and Sherif [83], working on orthographic data, developed four different learning models of a *Dynamic Bayesian Network* (*DBN*) [48]. They also evaluated and tested a group of other phonetic and orthographic algorithms, including *ALINE* [76] and *PHMM* [87]. One of the *DBN*, called hereinafter only *DBN*, outperformed all the other systems including *PHMM* [86], but not significantly. This work will be reviewed and examined in more detail in Chapter 6.

Cysouw and Jung [25] experimented an iterative process of multi-gram alignment between words in order to identify broad cognates from large parallel corpora in orthographic format. They utilised automatically extracted, semantically equivalent word pairs in English, French, Spanish, Portuguese, Russian and Hunzib, which is a Caucasian language. The algorithm considered all possible multi-gram pairs, up to four characters long, between cognate candidates, and *DICE* [39] was computed for each of the multi-gram pairs based on their incidence in the whole list of word pairs. In order to align and evaluate the cognate candidates, the authors proposed an extension of the *Levenshtein distance* [84], that included mappings of up to four length multi-grams and used the previously calculated *DICE* [39] as a cost function. The alignments that were found, were utilised to infer iteratively a new cost function, until it reached stabilisation. The authors also tested the method on a random variation of the dataset and it succeeded in recognising noise from broad cognates. One interesting aspect of this method is that it is orthography-independent and can be applied to graphemes written with different alphabets (e.g. Roman and Cyrillic alphabets) without the need of transliteration.

2.1.2 Phonetic methods

Guy [57], following the phonetic approach, developed *COGNATE* in an early attempt to develop a correspondence-based system for the identification of broad cognates in bilingual word-lists, for the task of machine translation. The author worked on bilingual lists of phonetically transcribed word pairs and, by identifying probable sound correspondences, estimated the likelihood that the words of each pair were cognate. A variant of the χ^2 statistic [108] was used on the phoneme correspondences discovered, to calculate correspondence probabilities. For each word pair, the alignment that maximised the sum of the correspondence probabilities was found and the alignment score was transformed into a cognate estimation by an empirical formula. The author did not provide any quantitative evaluation of his system, which was tested subsequently by *Kondrak* [77].

Kessler [72] pioneered in the task of measuring phonetic distance between dialects. He compared several methods including the *Levenshtein distance* [84] and the *alphabet-weight Levenshtein distance* [56], where the alphabet was the set of phones, i.e. the atomic phonetic characters. He considered the Irish Gaelic dialects, which were represented by phonetic word lists provided by *Wagner* [136], each containing about 50 concepts. In order to detect cognate words based on their phonetic transcription, the author proposed the *feature string comparison* approach that associated arbitrarily discrete ordinal values, scaled between 0 and 1, to each of the twelve phonetic features recognised. The distance between any two phones was calculated as the difference between the averages of all twelve feature values and these distances were used in the computation of the *alphabet-weight Levenshtein distance* [56]. The author found that the basic *Levenshtein distance* [84] outperformed the more sophisticated variant based on features comparison.

Covington [22] evaluated phonetic distances in an attempt to align cognate candidates for historical linguistic comparison. He developed a guided search algorithm for finding probable correct alignments between two words, presented in a broad phonetic transcription, on the basis of their surface form, without looking for sound laws or phonological rules. The author distinguished three types of phonetic segments: vowels, glides (i.e. *w, y*) and consonants. He manually assigned penalties for substitutions by a trial and error procedure on a dataset of 82 concepts in several languages derived from the *Swadesh lists* [132] provided by *Ringe* [114]. On this dataset, the algorithm proved to be able to align successfully, challenging word pairs in several languages, such as Spanish-French, English-German and English-Latin, where the best alignment was considered the one with the lowest total penalty. An evident limitation of the algorithm, acknowledged by *Covington* himself, was that it did not use phonetic features that would have been beneficial, even if vocalicity and vowel length were implicitly considered.

Covington [23] extended his algorithm to perform the challenging task of multiple string alignment, as opposed to pairwise string alignment. The author tested the new algorithm on data from several languages and the results were reasonable, considering it was one of the first attempts at multiple string alignment in computational historical linguistics.

Nerbonne and Heeringa [103] followed the phonetic approach in the task of measuring phonetic distances between Dutch dialects. They compared fourteen variants of the *Levenshtein distance* [84]. The simpler two were based on phones, while the more complex twelve required the phoneme decomposition into vectors of phonetic features. For the latter group, the authors experimented with weighting each feature by information gain and with three ways of calculating the distances between phonemes: the *Manhattan distance* [36], the *Euclidean distance* [36] and the distance based on the *Pearson correlation* [36]. Moreover, on both groups, they tested the benefit achieved by utilising diphthongs of one or two phones. The authors normalised the absolute distance between two words, by the length of the longer word. The distance between two dialects was then calculated as the sum of the *Levenshtein distances* [84] between two lists of corresponding words, composed of approximately 100 items from 40 Dutch dialects. This created a 40-by-40 symmetric matrix, which was then processed by a *distance-based* clustering algorithm using the *Ward*'s method [68] for the visualisation of a dendrogram, which accorded well with dialectal scholarship. The authors found that, in the task of measuring dialect relatedness, the more accurate of the tested methods was based on vectors of non-weighted features, whose comparison was better evaluated by the *Manhattan distance* using diphthongs of two phones.

Somers [127] proposed a special algorithm for the automatic analysis of children mis-articulations in the field of speech therapy. The algorithm was an aligner of children phonetic segments with the adult model. The author implemented and tested three versions of the algorithm. The three procedures were based on different substitution cost computation, which used binary articulatory features, perceptual features and multivalued features, respectively. The first version, called *CAT* proved to be the more accurate. The author tested *CAT* on the *Covington*'s test dataset [22] and the results, in terms of accuracy, were comparable with those achieved by the *Covington*'s algorithm [22].

Oakes [107] developed *JAKARTA*, a set of phonetic-based programs, that represented and performed automatically several steps of the *comparative method* [2], in order to achieve *proto-language* reconstruction. *JAKARTA* contained a phonetic aligner, which aimed to discover regular sound changes between historically related languages. The author identified three phonetic features: place, manner, and voicing, and assigned multiple values to the former two and a single value to the latter. He considered numerous possible sound changes, including lenition, fortition, assimilation, dissimilation, apocope, syncope, epenthesis and prothesis [18]. He assigned to all of them a uniform cost of *1*, while to the other substitutions, insertion and deletion operations, he gave a cost of *2*. *Oakes* considered two words to be cognate if their *edit distance* [84] was below a certain threshold, regardless of the words length. He calculated the threshold by examining the distances between cognate and non-cognate pairs in four Indonesian word lists, which he used to test the programs as well. The reconstructions found by *JAKARTA* were compared with the corresponding ones in the linguistic literature and the results were satisfactory. The choice of using the same development and test dataset represented an obvious limitation of this approach, which might have had a lower performance on a different test dataset, as demonstrated [77].

Kondrak [76] developed *ALINE*, a manually-designed algorithm for phonetic sequence alignment. The aligner represented phonetic segments as vectors of feature values and calculated their similarity through a local alignment procedure, performed by *dynamic programming* [11]. The twelve phonetic features were weighted according to their salience, which was established manually by trial and error. For example, the most significant features, which are *Place* and *Manner*, were assigned higher weights than less important features, like *High* and *Long*. The numerical values of each feature were based on data reported in the literature and aimed to reflect the distances between vocal organs during verbal emission. The author tested *ALINE* against the *Covington* algorithm [22] using the dataset employed by *Covington* [22] and found that *ALINE* outperformed the other in terms of accuracy and efficiency in cognate alignment. The author also compared the alignments produced by *Somers* with *CAT* [128] on the same dataset and found that *ALINE* produced more accurate alignments.

Kondrak [77] presented techniques and algorithms for automatically performing various stages of language reconstruction and evaluated them against several other methods. The test dataset was composed of the English, German, French, Latin and Albanian 200-word *Swadesh lists* [132] provided by *Kessler* [73] and arranged in ten pairs. He compared against *ALINE* [76] several similarity-based methods for cognate identification, including *JAKARTA* [107], *Truncation* [124], *DICE* [39] and *LCSR*

[96]. *ALINE* outperformed all the other approaches, whose accuracy, from lowest to highest, was ordered as follows: *DICE*, *JAKARTA*, *Truncation*, *LCSR* and *ALINE*. *Kondrak* also compared *COGNATE* [57] with *JAKARTA* [107] and several other models. The outcome showed that *JAKARTA* and *COGNATE* achieved a similar, but low, accuracy in cognate identification.

Wieling et al. [140] followed the *Mackay*'s approach [86, 87] and applied a *Pair Hidden Markov Model* (*PHMM*) to the task of dialect comparison. The authors also employed a variation of the *Levenshtein distance* [84], where vowels could not match with consonants and vice-versa. They gave the same weight to all the edit operations of substitution, insertion and deletion, and they did not normalise the final score. The authors trained a *PHMM* with Dutch dialect data and they tested the two methods on the same data, as they wanted to determine the sound distances on the basis of their data. The results produced by the *PHMM* were very similar to those achieved by the variation of the *Levenshtein distance* [84], but the training computational time of the *PHMM* was very expensive.

Kondrak [80] investigated the identification of cognates and recurrent sound correspondences. He tested several phonetic methods on a dataset composed of the 200-word *Swadesh lists* [132] of English, German, French, Latin and Albanian provided by *Kessler* [73]. His best result was achieved combining *ALINE* [76] with a sound correspondence-based method trained using a six-language development dataset, including Italian, Spanish, Romanian, Polish, Russian and Serbo-Croatian. This dataset was extracted from the orthographic Comparative Indo-European corpus by *Dyen et al.* [42] and then manually transcribed into a phonetic notation. This system improved the performance of *ALINE* [76] in terms of accuracy in broad cognate identification, but did not outperform the orthographic systems *PHMM* [87] and *DBN* [83] previously described.

Wieling et al. [141] evaluated several pairwise alignment methods on phonetic strings. They used a large corpus of corrected gold standard pairwise alignments extracted from Bulgarian dialect data, in order to compare three variants of the *Levenshtein distance* [84] with the *Pair Hidden Markov Model* (*PHMM*), proposed by *Mackay* [86]. *PHMM* was also tested by *Mackay and Kondrak* [87] and utilised by *Wieling et al.* [140]. The first *edit distance* variation, called *VC-sensitive Levenshtein algorithm*, did not allow alignments of vowels with consonants and vice-versa. The second variation, called *Levenshtein swap algorithm*, was an extension of the first and also allowed two adjacent characters to swap. The third variant, called *Levenshtein PMI algorithm*, used *Pointwise Mutual Information* (*PMI*) [21] to learn sound distances from pairwise alignments. *PHMM* and *PMI* were trained with the same data used for the test in order to determine the sound distances on the basis of those data. The authors evaluated the four methods with respect to the quality of the alignments produced and all the algorithms correctly aligned approximately 95% of the pairs. The *Levenshtein PMI algorithm* presented the lower percentage of incorrect alignments, while *PHMM* showed the lower error rate of misaligned segments, but, as usual, its training computational time was very high.

2.2 Phylogenetic inference

Recent decades have seen a large number of studies developing and employing phylogenetic techniques to investigate the evolution of language. We have reviewed some of the more interesting results especially regarding the Indo-European language family, which is the most intensively studied, but we have mentioned studies involving other language families as well.

Depending on the information coding, methods for linguistic phylogenetic inference may be classified as *distance-based* methods or *character-based* methods. *Distance-based* methods include the *UPGMA* [126] and the *Neighbor-Joining* algorithms [119, 130], while *character-based* methods comprise *Maximum Parsimony*, *Maximum Compatibility*, *Maximum Likelihood* and *Bayesian Inference* [46]. Investigations comparing different methods include *Nakhleh et al.* [100], *Barbançon et al.* [9], *Wichmann and Saunders* [139].

2.2.1 Distance-based methods

Dyen et al. [42] collected an Indo-European dataset described in Section 2.3, made a lexicostatistical classification [132] of the 84 languages included in the monograph and calculated the percentage of cognates shared by each language pair, creating an 84-by-84 distance matrix. In order to estimate a phylogeny, the authors developed a non-standard clustering algorithm. It belonged to the family of pair-group methods [46], like *UPGMA* [126], and was adapted to deal with lexicostatistical percentages. The phylogenetic tree proposed was not compatible with the benchmark tree of the Indo-European language family: it reproduced all the established major Indo-European branches with the exclusion of the Indo-Iranian clade.

Ellison and Kirby [43] calculated a word similarity measure within each of the 95 languages extracted from the digital version of the Indo-European dataset by *Dyen et al.* [42]. They called it *lexical metric* and defined it as a distribution of confusion probabilities, based on the *Levenshtein distance* [84] normalised by the average length of the words. The divergence between two languages was defined as the divergence of their lexical metrics and calculated as the geometric path between the two distributions, creating a distance matrix. In order to root the tree, they added a random outgroup, which was a questionable choice, and used *Neighbor-Joining* [119, 130] to build a phylogeny. This tree was not compatible with the benchmark tree of the Indo-European language family, even if it showed correct groupings for many languages.

Serva and Petroni [123] applied the *Levenshtein distance* [84] normalised by the length of the longer word, to 50 language pairs extracted from the Indo-European dataset by *Dyen et al.* [42]. For each language pair they compared 200 words with the same meaning and they computed the average of these *edit distances* in order to create a 50-by-50 matrix of language distances. The authors transformed this distance matrix into a time distance matrix following the *glottochronology* approach [132]. They imposed established time distances to the system with the aim of providing a phylogenetic tree topology with absolute time scales. Finally, they inferred a rooted phylogenetic tree using *UPGMA* [126]. The proposed tree topology satisfied the "*No missing subgroups*" criterion, but violated some compatibility requirements for phylogenetic estimation [106]. The same methodology was applied by *Petroni and Serva* [110] to the Austronesian language family. This method was expanded by *Blanchard et al.* [13] to represent geometrically the relationships between languages belonging to both the Indo-European and Austronesian language families.

Brown et al. [16] developed *Automated Similarity Judgment Program* (*ASJP*) aiming to perform a large-scale classification of languages by calculating their lexical similarity following a lexicostatistical approach [132]. They used 100-word *Swadesh lists* [133] from 245 globally distributed languages, with the objective of expanding their database to all the world's languages. They used the *Neighbor-Joining* [119, 130] algorithm to generate the phylogenetic trees. The list dimension was subsequently reduced to 40 more stable lexical elements for the achievement of better results, and the database was expanded to 900 languages [64]. The algorithm used to determine whether or not words were likely to be cognate, was changed by *Bakker et al.* [8]. They employed the *Levenshtein distance* [84], as proposed by *Serva and Petroni* [123], but with a double normalisation. Firstly, they divided the *edit distance* by the length of the longer word and then divided this quantity by the averaged normalised *Levenshtein distance* among the words with different meaning. *ASJP* presented non-uniform performance, passing both evaluation tests for some language families and failing both for others [106].

Downey et al. [40] estimated phylogenies for the Sumbanese language family using *ALINE* [76] to produce a distance matrix that was then processed by *distance-based* methods [46]. In order to control the bias due to different string length, the authors normalised the algorithm score by the *Arithmetic mean* of *ALINE* [76] applied to rate each string with itself. *Downey et al.* utilised both *UPGMA* [126] and *Neighbor-Joining* [119, 130] to estimate phylogenetic trees. The proposed evolutionary trees were close to the historical reconstruction, especially the phylogeny built by *UPGMA* [126], which satisfied the "*No missing subgroups*" criterion, but violated some compatibility requirements for phylogenetic estimation.

2.2.2 Maximum Parsimony

Gray and Jordan [53] made one of the first attempts to apply biological phylogenetic methods to historical linguistics. They encoded the presence or absence of 5,185 lexical characters of cognateness for 77 Austronesian languages in a binary matrix and they employed a *Maximum Parsimony* analysis [46] that produced a single most parsimonious tree. The topology of this tree supported the *express-train model* of Austronesian expansion [37] and showed considerable agreement with traditional linguistic groupings, even if the tree violated the "*Compatible resolution*" criterion [106].

Rexová et al. [113] used *Maximum Parsimony* and greedy consensus trees [46] on the comparative Indo-European corpus by *Dyen et al.* [42] focussing on the impact of the character encoding. They employed three different methods of character encoding, creating a standard multi-state matrix, an altered multi-state matrix and a binary matrix. The study showed substantial dissimilarities between the two multi-state matrices and the binary matrix, including different tree rooting. This suggested that the binary encoded data matrix produced less reliable trees than those created employing the multi-state matrices.

2.2.3 Maximum Compatibility

Ringe et al. [115] prepared an Indo-European, dataset described in Section 2.3, and used *Maximum Compatibility* [46] to estimate the phylogenetic tree of the Indo-European language family. They utilised lexical, morphological and phonological characters from 24 Indo-European languages and the *Kannan and Warnow* algorithm [70], which runs in polynomial time. They assigned weights to characters, which made the model very dependent on the linguistic choice. The authors rooted the tree by hand, after examination of the unrooted tree produced by *Maximum Compatibility* [46], and their methodology passed the two evaluation criteria of phylogenetic inference [100].

2.2.4 Bayesian analysis

Gray and Atkinson [52] estimated the language-tree divergence times for the Indo-European language family, suggesting a root age of Indo-European of between 7,800 and 9,800 Before Present (BP), consistent with the Anatolian theory of Indo-European origin. In order to aid the estimation of older language relationships, they added to the 84 Indo-European speech varieties included in the monograph of *Dyen et al.* [42], three extinct Indo-European languages, Hittite, Tocharian A and Tocharian B, reaching a total of 87 languages. Based on cognate judgments from this extended corpus, they produced a binary matrix of 2,449 lexical characters indicating the presence or absence of words in each cognate group. This binary matrix was then examined using *Maximum-Likelihood* models, *Bayesian Markov Chain Monte Carlo* (*MCMC*) analysis and rate-smoothing algorithms to produce a majority-rule consensus tree [46]. This model allowed homoplasy, i.e. back mutation or parallel evolution. It supported polymorphism, i.e. the presence of multiple words in one language for a given meaning, coded as multiple states for that character in one language. The proposed tree topology satisfied the two criteria required by *Nichols and Warnow* [106] for phylogenetic estimation, while the dating failed the calibration criterion. The *Gray and Atkinson* method was subsequently extended [7, 5, 4] and also applied to study the Bantu language family [112, 63].

Nicholls and Gray [104, 105] applied to language evolution a stochastic model, first introduced by *Huson and Steel* [66], and dated the Indo-European language family at about 8,000-9,000 BP. The model implemented *Dollo Parsimony* principles, used *Bayesian* phylogenetic inference [46] and *MCMC* algorithms [61] to generate a sample distribution of trees, and screened them using constraints before producing a consensus tree. The authors used encoded multi-state lexical characters from the *Ringe et al.* dataset [115] and from the *Dyen et al.* corpus [42, 52], extended with Hittite, Tocharian A and Tocharian B. They ran several analyses considering 3 subsets of the first dataset and 6 subsets of the second. They found that age estimations of the root were uniform across all analyses, whereas the topologies were not reliable. This model did not allow homoplasy and supported polymorphism. On the other hand, it could not handle missing data and so the analyses were necessarily limited to those characters shown in all speech varieties, discarding some languages that presented too much missing data.

Ryder and Nicholls [118] extended the *Nicholls and Gray* method [105] to handle missing data, using binary encoding of cognate classes as lexical traits from the *Ringe et al.* dataset [115]. They also gave an analysis of the *Dyen et al.* corpus [42, 52], extended with Hittite, Tocharian A and Tocharian B, in the paper supplement. They estimated the date of the Proto-Indo-European language around 7,100-9,800 BP.

2.3 Indo-European linguistic datasets

The corpora prepared by *Dyen et al.* [42] and by *Ringe et al.* [115] for the Indo-European language family are recommended datasets for linguistic studies [106]. They differ in many aspects, including the number of languages considered, their dating and the types of characters reported.

The Comparative Indo-European corpus by *Dyen et al.* [42] provides lexical data in the form of 200-word *Swadesh lists* [132] of universal, non-cultural and stable meanings from 84 contemporary Indo-European speech varieties. In it, each word is presented in orthographic format without diacritics, using the 26 letters of the Roman alphabet. The data are grouped by meaning and cognateness, which is reported as *certain* or *doubtful*. The digital version of the dataset covers 95 languages, of which only 84 were considered accurate enough to be included in the monograph.

The dataset of *Ringe et al.* [115] is provided in two versions, unscreened and screened, both containing phonological, morphological and lexical characters for 20 extinct and 4 existing Indo-European languages. These speech varieties have been chosen to represent the 12 major subgroups of Indo-European languages through their oldest and best documented languages in each branch. The screened dataset is produced from the unscreened version by removing all characters that clearly exhibited homoplasy. The data are provided in three matrices, each corresponding to a character type. A multi-state coding is used to describe each of the phonological, morphological and lexical characters included.

Chapter 3

Language evolution

Languages that are genetically related originate from a common ancestor. The estimation of their evolutionary relationships is the primary aim of historical linguistics. The evolution of languages may be studied through their phonological, lexical and morphological changes and cognate words are frequently used lexical characters. For this reason, cognate identification is one of the principal tasks of historical linguistics, together with phylogenetic inference, which seeks to represent these genetic relationships through evolutionary trees.

In this chapter, we present the problems of cognate identification and phylogenetic inference as strategic and promising fields for computational historical linguistics. Because of the close analogy between language evolution and species evolution, we focus on evolutionary biological techniques, that can be successfully borrowed from that field and applied to our context.

We choose and introduce the Indo-European language family as the target of our investigation for the high significance and particular role it has in the field of historical linguistics.

3.1 Introduction

Charles Darwin in "*On the Origin of Species*" [28] had a premonitory vision about the analogy between language evolution and species evolution:

> "*If we possessed a perfect pedigree of mankind, a genealogical arrangement of the races of man would afford the best classification of the various languages now spoken throughout the world.*"

This hypothesis states that languages and genes each form genealogical relationships among populations, and that the co-evolution of languages and populations should make these genealogies *resemble* each other. However, invasions could sometimes have altered this pattern [117]. The analogy between language evolution and species evolution is now widely accepted, as it is based on evidence from the fields of linguistics, archaeology and genetics. The latter has been recently enhanced by data from *mitochondrial DNA*[1] [19] and *Y-chromosome*[2] [71], which suggest the "*Out of Africa*" model. In this theory, *modern Homo sapiens*, or *Homo sapiens sapiens*, originated in East Africa about 200,000 years ago from some non-human ancestor and spread from there, replacing more archaic human populations, such as Neanderthals [117]. This theory was also suggested by *Charles Darwin* in "*The Descent of Man*" [29]. Natural selection [28] and neutral evolution, proposed by *Motoo Kimura* [74], may be considered the main mechanisms driving species evolution as well as language evolution.

[1] Mitochondrial DNA is passed down from mother to offspring. "*Mitochondrial Eve*", supposed to have lived between 150,000 and 250,000 years ago, is the most recent common ancestor of all humans alive today.

[2] DNA in Y-chromosomes is passed down from father to son. "*Y-chromosomal Adam*", supposed to have lived between 60,000 and 90,000 years ago, is the most recent common ancestor of every living man's Y-DNA.

With an erect body carriage, a highly developed brain and a descended larynx, *modern Homo sapiens* developed over time the capacity to acquire and use language. If *Homo sapiens sapiens* originated in Africa and then spread to all the other continents, then it is reasonable to suppose that all the world's languages share a common origin. This fascinating hypothesis is confirmed by some indication of global linguistic similarities that allow linguists to group languages into families and superfamilies [117], even if the debate remains highly controversial.

The analogy of language evolution with species evolution has generated a growing interest in the scientific community following the amazing progress of computational molecular biology in the field of genomes. The successful application of bioinformatic techniques in the field of language evolution has started to show exciting opportunities, as well as making significant contributions per se.

However, despite the analogy between language evolution and species evolution, there are some noteworthy differences. Firstly, languages evolve much faster then genomes [117]. Secondly, genetic sequences can be very long, reaching millions of characters for nucleic and amino acid sequences, while linguistic strings are not more than ten characters long on average. Finally, computational molecular biology can rely on a vast range of high quality databases (e.g. UniProt[3]), sequence database search tools (e.g. FASTA[4] and BLAST[5]) and sequence alignment programs (e.g. Clustal[6]). For historical linguistics only a few, and not always accurate, datasets are available [42, 73, 115].

[3] `http://www.uniprot.org/`

[4] `http://www.ebi.ac.uk/Tools/fasta/`

[5] `http://blast.ncbi.nlm.nih.gov/`

[6] `http://www.clustal.org/`

3.2 Linguistic background

Historical linguistics studies the evolution of language, which can be viewed as a series of states threaded along the dimension of time [121]. *Comparative linguistics* is a branch of historical linguistics that aims to establish the existence and degree of genetic relationships between two or more languages. It seeks to build phylogenetic language trees and to reconstruct as far as possible their common ancestors or *proto-languages*, in the absence of historical records. The foundation for all the approaches in comparative linguistics is "*L'arbitraire du signe*" or "*de Saussure's First Principle*" [121]. It states that *any* word used to represent *any* concept in *any* language is arbitrary, with a few exceptions including borrowings, onomatopes and nursery words. This leads us to assume that, if words grouped by their semantic meaning show relatedness, it cannot be by chance.

The problem of how to establish genetic relationships among languages is still the subject of much debate among the linguistic scholars. There are two main methodologies, both subject to some criticism, but both helpful and successful in showing genetic connections between languages. They both analyse semantically related words in order to identify cognate words that exhibit the same character state only as a result of an evolutionary relationship [106], and may provide evidence of historical relationships between languages.

- The *comparative method* [2], developed over the last two centuries, is a central procedure in historical linguistics. It studies sound recurrent correspondences in semantically matching morphemes in order to detect connections among languages. The main theoretical principle supporting the *comparative method*, and confirmed by a huge amount of evidence, is the "*Regularity hypothesis*", which states that sounds develop regularly in a phonetic environment. The *comparative method* consists of the following activities: potential cognate lists compilation, regular sound correspondence determination, complementary distribution sounds sets discovery, proto-phonemes reconstruction and typological consistency check of the rebuilt system. The *comparative method* is nowadays widely accepted, even if it has sometimes been criticised for not being accurate enough, as some phonological changes are not as regular as stated [73].

- The *multilateral comparison* or *mass lexical comparison*, developed by *Greenberg* [54] and supported by *Ruhlen* [117], proposes language classification based on the number of surface similarities between groups of semantically related words. Even if the revolutionary conclusions reached by *Greenberg* have been increasingly accepted by the linguistic community, his method of long-range comparison has been sharply criticised by many linguists for its lack of rigour. *Kessler* [73] believes that the *multilateral comparison*'s ability to detect connections decreases faster than the *comparative method*'s capacity, as the divergence time between languages increases.

Related to the *comparative method*, *lexicostatistics* [42] is a mathematical measure based approach for the construction of linguistic phylogenies that involves the quantitative comparison of cognates. This method aims to assess quantitative language relatedness and consists of the following tasks: word list creation, cognate identification, lexicostatistic percentage calculation and phylogenetic tree creation. One of its fundamental assumptions is that some words that form the basic core of vocabulary are more resistant than others to changes and loans. Because of this, they remain better preserved over time. This word set includes lower numerals, pronouns, body parts, objects of nature and basic activities.

Glottochronology [42] is an application of lexicostatistics seeking to date language divergence. It attempts to estimate the length of time since two or more languages diverged from a common ancestral *proto-language*, under the assumption of a constant rate of change in the fundamental vocabulary. This method was proposed by *Swadesh* [131] based on an analogy with the use of carbon dating for measuring the age of organic materials. In order to conduct his experiments, *Swadesh* prepared a list of 200 universal and non-cultural words, which he considered the "*intimate*" part of any vocabulary [132]. After more research, *Swadesh* [133] proposed a new list, which he recognised as being even more general and stable. It contained 100 words only, collected mainly from the previous list, but with the addition of some new words. The *Swadesh lists* are reported and documented in Appendix A.

Even if *Swadesh*'s work has been sharply criticised by linguists, *Swadesh lists* have been, and continue to be, widely used as datasets in computational historical linguistics, and more generally in natural language processing. Ironically, it has been through the use of *Swadesh lists*, of over one third of all the world's languages, that a recent investigation [6], run by biological evolutionists, has reinforced the indication against the *glottochronology* approach. This study suggested that, while languages evolve slowly most of the time, when dividing they show rapid bursts of evolution, accounting for between 10% and 33% of the total divergence among fundamental vocabularies.

3.3 Computational applications

Identification of cognates, discovery of phonetic similarities, detection of recurrent sound correspondences, language reconstruction and phylogenetic inference are laborious and time-consuming activities, traditionally executed manually by linguistic experts.

In the last two decades, computational methods have been increasingly applied to historical linguistic tasks to add processing power, computational rigour and statistical significance to the field, as well as to provide novel hypotheses to be critically examined and tested by linguistic experts. Surprisingly, also textual statistical analysis is able to identify relationships among languages by studying their shared statistical properties [135]. Indeed, there are still many language families that need to be studied, language relationships that may possibly be shown and many controversies that wait to be solved.

The two main applications of computational techniques to historical linguistics that have shown to be promising are:

- *Cognate identification*, which aims to recognise phonetic similarity, orthographical similarity and regular sound changes. All these features may show clear evidence and degree of language relatedness, guide to *proto-language* reconstruction in the absence of historical records, and allow phylogenetic inference.

- *Phylogenetic inference*, which aims to estimate the genetic relationships between languages based upon similarities and differences in their characters and to represent them with evolutionary trees.

3.4 Cognate identification

Languages that originate from a common ancestor are genetically related. For example, the Romance family includes all the languages that descended from Latin and gradually diverged from it over time, including Italian, Spanish, Portuguese, French and Romanian.

The study of language relatedness has been historically based on the detection of *cognates*, words that derive from the same predecessor and share an identical etymological origin, from Latin "*cognatus*" ← "*cum*" + "*gnatus*", meaning "*born together*". For example, the words Italian *fiore*, Spanish and Portuguese *flor*, French *fleur*, Romanian *floare* are all cognate. They derive from Latin *flos/floris*, with the accusative form *florem*, which means *flower*. Less obvious is that the English word *flower* is not part of this group of cognates, as it is considered a borrowing from Old French. Even less evident is that German *Blume*, Dutch *bloem*, Swedish *blomma*, and Danish *blomst* are part of the same group of cognates introduced above. They derive from a word that belonged to an extinct *proto-language* called *Proto-Germanic*, which did not leave any historical evidence, but did release signs of its existence in all the Germanic daughter languages. By studying this cognate group across all the Germanic languages, linguists have been able to reconstruct a supposed common ancestor, proposed as **blo-s-*, where the asterisk indicates a reconstruction and not a documented word. Considering all the cognate words believed to belong to this group, linguistic scholars made another step forward, reconstructing the proto-word for the hypothetical ancestor of Latin, *Proto-Germanic* and all the other European and Indian languages, called *Proto-Indo-European*. The proposed reconstructed root for this group of cognates is **bhlo-*.

It should be more apparent now why cognate identification represents the foundation for discovering the evolutionary history of languages. The fact that the quantity and the similarity of cognates between related languages are non-increasing monotonic functions of time, may also lead to the estimation of divergence time [44, 5, 4, 55]. The importance of *proto-language* reconstruction derives from the relatively recent invention of writing systems, dating back about 5,000 years, and from the lack of writing evidence for some recent or also current languages. In fact, on top of the approximately 7,000 living languages [85], many thousands more are considered extinct, and many of them have not left any explicit proof of their nature or even of their existence [117].

3.4.1 String matching

Cognate words ultimately are strings and for this reason they can be successfully studied by string matching techniques. A *string* S is an ordered list of characters from an alphabet $\mathcal{A}$ written contiguously from left to right [56]. For any string S, $S[i..j]$ is the *substring* of S that contains the contiguous characters of S starting at position i and ending at position j. In particular, $S[1..i]$ is the *prefix* of string S that ends at position i. $S[i..n]$ is the *suffix* of S that starts at position i, where n is the length of the string S. A *subsequence* of S is formed by ordered, but not necessarily contiguous, characters of S. In computational molecular biology, biological strings are usually referred to as *sequences.* When comparing two characters in strings, the characters *match* if they are equal, otherwise they *mismatch* [56]. The two main approaches to the computation of string matching are:

- *Exact string matching* [56], which aims to find exact matches on strings and substrings;
- *Inexact or approximate string matching* [58, 101], which focuses on finding matches on subsequences, meaning that some errors are acceptable in valid string matches.

Depending on the type of application, one approach may be more suitable than the other. Word processor applications, system word utilities (e.g. *grep* on Unix, Windows, etc), digital telephone directories, digital dictionaries and thesauri are only a small subset of all the tasks that use *exact string matching* algorithms [56]. On the other hand, *inexact string matching* [58, 101] is the basic approach in computational molecular biology and in many fields of natural language processing. In particular, historical linguistics and bioinformatics share the need to model and discover active mutational processes, through string comparison.

In *inexact string matching* [58, 101], the strategy frequently used for subsequence comparison is *dynamic programming* [11], which is a general computational method of solving complex problems. It breaks them into sub-problems, which are simpler to solve, and uses their solutions to find an answer to the main problem. The three essential components of *dynamic programming* [11] are the recurrence relation, the tabular compilation and the trace back. The bottom-up scheme is generally preferred to the top-down one, because it minimises the number of recursive calls and so provides higher efficiency [56].

By adopting the *inexact string matching* approach [58, 101] to determine the relatedness of two strings, it is possible to either measure their *distance*, evaluating how distant the two strings are from each other, or to measure their *similarity*, calculating instead how similar the two strings are [56]. The distance method leads to a minimisation problem, because it aims to find the minimum distance between two strings, while the similarity method guides towards a maximisation problem, as it seeks to find the maximum similarity between two strings.

3.4.1.1 String distance

Given an alphabet $\mathcal{A}$, where $\mid \mathcal{A} \mid \geq 2$, and the set Σ of all finite strings over $\mathcal{A}$, a distance function $\mathrm{D} : \Sigma \text{ x } \Sigma \rightarrow \Re$ is called a *metric* [120], if it holds the metric axioms, $\forall$ S_1, S_2, $S_3 \in \Sigma$:

1. $\mathrm{D}(\mathrm{S}_1, \mathrm{S}_2) \geq 0$	Non negativity
2. $\mathrm{D}(\mathrm{S}_1, \mathrm{S}_2) = 0 \leftrightarrow \mathrm{S}_1 = \mathrm{S}_2$	Self-identity axiom
3. $\mathrm{D}(\mathrm{S}_1, \mathrm{S}_2) = \mathrm{D}(\mathrm{S}_2, \mathrm{S}_1)$	Symmetry
4. $\mathrm{D}(\mathrm{S}_1, \mathrm{S}_2) \leq \mathrm{D}(\mathrm{S}_1, \mathrm{S}_3) + \mathrm{D}(\mathrm{S}_3, \mathrm{S}_2)$	Triangle inequality

Table 3.1: The metric axioms

The *edit distance* or *Levenshtein distance* [84] is the most classic formalisation of the notion of distance between two strings and it was first discussed in the field of coding theory. It is defined as the minimum number of edit operations necessary to transform one string into another, where the edit operations allowed are deletion, insertion and substitution of a character. Deletion and insertion are frequently referred to as *indel* operations. The *edit distance* d is a metric and holds [101]:

$$0 \leq d(S_1, S_2) \leq \max(|S_1|, |S_2|). \tag{3.1}$$

The *edit distance* associates a unitary cost to any edit operation and a zero cost to any match. It has received considerable attention, because it can be easily generalised and applied to a wide range of disciplines, such as computational biology, signal processing and text processing to mention a few. The algorithm to calculate the minimum distance has a remarkable history of multiple independent discovery and publications in different areas [120] and this is an indication of its crucial and determinant role. Moreover, many variations of the *edit distance* have been proposed over time using different approaches, such as *dynamic programming* algorithms, automata based algorithms, bit parallelism algorithms and filtering algorithms. Detailed overviews have been presented by *Hall and Dowling* [58] and by *Navarro* [101].

Wagner and Fisher [137] developed a bottom-up *dynamic programming* algorithm to calculate the *edit distance* between two strings S_1 and S_2. It involves the use of an $(n{+}1)$-by-$(m{+}1)$ matrix and is $\mathcal{O}(n * m)$ both in time and space, where n and m are the lengths of the two strings.

The base conditions of the *Wagner and Fisher* algorithm state that i characters must be deleted to convert i characters of S_1 to zero characters of S_2, and j characters must be inserted to convert zero characters of S_1 to j characters of S_2:

$$\begin{array}{ll} d(i,0) = i; & \forall\ i\colon\ 0 \le i \le n \\ d(0,j) = j; & \forall\ j\colon\ 0 \le j \le m \end{array} \tag{3.2}$$

The algorithm recurrence relation, $\forall i : 0 \le i \le n$ and $\forall j : 0 \le j \le m$, establishes a recursive relationship between the value of $d(i,j)$ and the values of $d(i-1,j)$, $d(i,j-1)$ and $d(i-1,j-1)$:

$$d(i,j) = min \begin{cases} d(i-1,j) + 1; \\ d(i,j-1) + 1; \\ d(i-1,j-1) + t(i,j); \end{cases} \tag{3.3}$$

$$\text{where} \quad \begin{array}{ll} t(i,j) = 0 & \text{if } S_1[i] = S_2[j] \\ t(i,j) = 1 & \text{if } S_1[i] \neq S_2[j] \end{array}$$

The *edit distances* are inserted in the $(n+1)$-by-$(m+1)$ matrix, one row or one column at a time, starting for the smallest possible values of i and j, which are progressively increased. String S_1 corresponds to the vertical axis, while string S_2 corresponds to the horizontal axis of the matrix. The values of adjacent cells differ by one at most, and the upper-left to lower-right diagonals are non-decreasing. The *edit distance* is the value $d(n, m)$ in the bottom right cell of the tabular representation.

Table 3.2 shows the *tabular computation* of the *edit distance* between the words Italian *fiore* and Spanish *flor*, as an example. The pointers from one cell to another show the step or the steps minimising the edit operations for that particular cell. There is only one path that minimises the distance between the two words, which is shown in bold, and the *edit distance* is $d(fiore, flor) = 2$.

$d(i,j)$			*f*	*l*	*o*	*r*
		0	1	2	3	4
	0	**0**	← 1	← 2	← 3	← 4
f	1	↑ 1	↖ **0**	← 1	← 2	← 3
i	2	↑ 2	↑ 1	↖ **1**	↖← 2	↖← 3
o	3	↑ 3	↑ 2	↖↑ 2	↖ **1**	← 2
r	4	↑ 4	↑ 3	↖↑ 3	↑ 2	↖ **1**
e	5	↑ 5	↑ 4	↖↑ 4	↑ 3	↑ **2**

Table 3.2: Example of tabular computation of the *edit distance* with one optimal alignment

Because the *edit distance* of two strings can be represented by an alignment minimising the number of mismatches and *indels* [56], once the value of the *edit distance* has been computed, it is possible to recover an *optimal alignment* by tracing back the arrows in $\mathcal{O}(n + m)$ time. For example, the optimal alignment between the words Italian *fiore* and Spanish *flor* can be computed from the *dynamic programming* Table 3.2, as shown in Table 3.3, where mismatches and *indels* are in bold.

f	***i***	*o*	*r*	***e***
f	***l***	*o*	*r*	**-**

Table 3.3: Example of one optimal alignment produced by the *edit distance*

Note that there may be more alignments that present the same minimum number of edit operations. For example, in the case of the words Italian *fiore* and French *fleur*, the tabular computation in Table 3.4 shows that there are more possible optimal alignments that represent the same *edit distance*, which is $d(fiore, fleur) = 4$.

$d(i,j)$			*f*	*l*	*e*	*u*	*r*
		0	1	2	3	4	5
	0	**0**	← 1	← 2	← 3	← 4	← 5
f	1	↑ 1	↖ **0**	← **1**	← 2	← 3	← 4
i	2	↑ 2	↑ 1	↖ **1**	↖← **2**	↖← 3	↖← 4
o	3	↑ 3	↑ 2	↖↑ 2	↖ **2**	↖← **3**	↖← 4
r	4	↑ 4	↑ 3	↖↑ 3	↖↑ 3	↖ **3**	↖ **3**
e	5	↑ 5	↑ 4	↖↑ 4	↖ 3	↖←↑ 4	↖↑ **4**

Table 3.4: Example of tabular computation of the *edit distance* with several optimal alignments

Table 3.5 shows the possible optimal alignments between the words Italian *fiore* and French *fleur* calculated by the *edit distance*, where mismatches and *indels* are in bold. String alignments will be analysed in more detail in Section 3.4.2.

f	***i***	***o***	***r***	***e***
f	***l***	***e***	***u***	***r***

f	***i***	***o***	**-**	*r*	***e***
f	***l***	***e***	***u***	*r*	**-**

f	***i***	**-**	***o***	*r*	***e***
f	***l***	***e***	***u***	*r*	**-**

f	**-**	***i***	***o***	*r*	***e***
f	***l***	***e***	***u***	*r*	**-**

Table 3.5: Examples of several optimal alignments produced by the *edit distance*

Numerous other variations of the *edit distance* have been proposed in the literature, some are its simplifications, some are its extensions and in this section we will mention those which are the most relevant.

The *Hamming distance* [59], defined for two strings S_1 and S_2 of the same length, allows substitutions only and was first introduced in error coding theory. This distance is a metric and holds [101]:

$$0 \leq H(S_1, S_2) \leq |S_1| \tag{3.4}$$

For example, the *Hamming distance* of the words Italian *fiore* and French *fleur* is $H(fiore, fleur) = 4$, as it is shown by the optimal alignment without *indels* reported in table 3.6:

f	***i***	***o***	***r***	***e***
f	***l***	***e***	***u***	***r***

Table 3.6: Example of optimal alignment produced by the *Hamming distance*

The *Damerau-Levenshtein distance* [27] between two strings adds to the set of operations allowed by the *Levenshtein distance*, i.e. insertion, deletion and substitution, also the transposition of two characters and was proposed for spelling error corrections.

The *operation-weight edit distance* [56] is a simple, but crucial generalisation of the *edit distance*, where arbitrary weights or costs can be associated to every *indel* operation, to every substitution operation and to every match as well, where no operation is needed.

The *alphabet-weight edit distance* [56] is another simple, but fundamental extension of both the *edit distance* and the *operation-weight* *edit distance.* In this case the weight or cost of any operation varies in accordance with the characters in the alphabet that the operation has to manage. For example, a substitution depends on which character in the alphabet has to be removed and which has to be added, an *indel* operation depends on which character has to be inserted or deleted, a match on which character has matched.

Normalisations of the *edit distance* [84] are frequently useful, in order to obtain values in the range [0,1]. Moreover, since the sizes of strings vary, one edit operation in a short word is much more relevant than one edit operation in a long word. As a consequence, a sensible normalisation may, for example, divide the *edit distance* by the length of the longer of the two strings or by the average length of the two strings.

3.4.1.2 String similarity

Another way to formalise the relatedness of two strings is to measure their similarity instead of their distance, and this is the most frequent approach in computational molecular biology and natural language processing. Numerous string similarity measures have been proposed in the literature in different fields, and we have reviewed those relevant to the task of cognate identification in Section 2.1.

In bioinformatics, given an alphabet $\mathcal{A}$ where $\mid \mathcal{A} \mid \geq 2$, let $\mathcal{A}'$ be an extension of $\mathcal{A}$ with the addition of the character "–" representing a gap. Let $\mathcal{M}$ be a *substitution matrix* $|\mathcal{A}'|$-by-$|\mathcal{A}'|$, which associates a value $\mathcal{M}[A'_i, A'_j]$ to any pair of characters $\mathcal{A}'_i$ and $\mathcal{A}'_j$ belonging to $\mathcal{A}'$, and let Σ' be the set of all finite strings over $\mathcal{A}'$. If S'_1 and S'_2 denote the strings of length n resulting from an alignment AL of the strings S_1 and S_2, the value of this alignment is obtained by summing the pairwise value for each character pair in AL. The *similarity* $\mathcal{S}$ of two strings S_1 and S_2 is defined as the value of the optimal alignment of S_1 and S_2 that maximises the total alignment value [56]:

$$\mathcal{S}(S_1, S_2) = \max_{AL} \sum_{i=1}^{n} \mathcal{M}(S'{}_1(i), S'{}_2(i)) \tag{3.5}$$

The Longest Common Subsequence (LCS) [137] between two strings is a special case of string similarity and it allows insertions and deletions at unitary cost, but not substitutions. A common subsequence is formed by ordered, but not necessarily contiguous, characters present in both words.

For example, the LCS of the words Italian *fiore* and Spanish *flor* is *f-o-r* and $|LCS(fiore, flor)|$ = *3*, as it is shown in Table 3.7 where the *indels* are in bold:

f	***i***	**-**	*o*	*r*	***e***		*f*	**-**	***i***	*o*	*r*	***e***
f	**-**	***l***	*o*	*r*	**-**		*f*	***l***	**-**	*o*	*r*	**-**

Table 3.7: Example of optimal alignments produced by *LCS*

When the cost of a substitution is set at twice the cost of an *indel* operation, and a match has zero cost, the number of mutations needed to convert a string into another, is the sum of their lengths minus twice the length of their *LCS* [56]. In this case, the following relationship exists between the *edit distance* [84] and the *LCS* of two strings S_1 and S_2:

$$d(S_1, S_2) = |S_1| + |S_2| - 2 * |LCS(S_1, S_2)| \tag{3.6}$$

It is worth noting that string similarity is strictly related to the *alphabet-weight edit distance* and it is frequently possible to transform one problem into the other [56]. However, the similarity approach allows local alignment to be performed, while the distance approach does not, as discussed in Section 3.4.2.2. The similarity of two strings S_1 and S_2 as an alignment of them, will be examined in Section 3.4.2, while substitution matrices will be addressed in Section 3.4.3.

3.4.2 String alignments

The task of calculating the distance or the similarity between two strings is closely related to the task of finding their optimal alignment: *dynamic programming* algorithms can perform both tasks [56].

Alignment algorithms usually consist of a scoring scheme for measuring distance or similarity between characters and a procedure for finding the optimal alignments. The significance of any alignment depends greatly on the chosen scoring scheme [56]. Several substitution matrices have been proposed for biological sequence analysis and we will discuss them in Section 3.4.3. In protein sequence analysis:

> "*There are several different types of alignments: global alignment of pairs of proteins related by common ancestry throughout their lengths, local alignments involving related segments of proteins, multiple alignments of members of protein families, and alignments made during data base searches to detect homology.*" [62]

In historical linguistics, because of the scarcity of structured cognate databases, the types of alignments involved are generally global and local alignments between two words, and multiple alignments between a group of words. Even if the small length of the strings could make global alignments apparently more appropriate, local alignment can be useful in order to focus on the word roots, disregarding inflectional and derivational affixes [76, 25].

In this thesis, we have not discussed multiple string alignments, which is a fascinating and challenging topic of our future research plan.

3.4.2.1 Global string alignment

The similarity of two strings S_1 and S_2 as global alignment of them can be calculated by a bottom-up *dynamic programming* algorithm. It is known as *Needleman-Wunsch algorithm* [102] in recognition of the authors who first discussed global similarity, even if the more efficient version generally used is by *Gotoh* [51]. As for the *edit distance* [84], the algorithm involves the use of an $(n{+}1)$-by-$(m{+}1)$ matrix and is $\mathcal{O}(n * m)$ both in time and space, where n and m are the lengths of the two strings S_1 and S_2. If the similarity $\mathcal{S}(i,j)$ is defined as the value of the optimal alignment of the prefixes $S_1[1..i]$ and $S_2[1..j]$ and $\mathcal{M}$ is a substitution matrix, the base conditions of the *Needleman-Wunsch* algorithm are:

$$\begin{aligned} \mathcal{S}(i,0) &= \textstyle\sum_{1\le k\le i} \mathcal{M}(S_1(k), `-`) \qquad && \forall i : 0 \le i \le n \\ \mathcal{S}(0,j) &= \textstyle\sum_{1\le k\le j} \mathcal{M}(`-`, S_2(k)) \qquad && \forall j : 0 \le j \le m \end{aligned} \tag{3.7}$$

The algorithm recurrence relation, $\forall i : 0 < i \le n$ and $\forall j : 0 < j \le m$, establishes a recursive relationship between the value $\mathcal{S}(i,j)$ and the values $\mathcal{S}(i-1,j), \mathcal{S}(i,j-1)$ and $\mathcal{S}(i-1,j-1)$:

$$\mathcal{S}(i,j) = max \begin{cases} \mathcal{S}(i-1,j) + \mathcal{M}(S_1(i), `-`); \\ \mathcal{S}(i,j-1) + \mathcal{M}(`-`, S_2(j)); \\ \mathcal{S}(i-1,j-1) + \mathcal{M}(S_1(i), S_2(j)); \end{cases} \tag{3.8}$$

The similarity between the two strings S_1 and S_2 is the value $\mathcal{S}(n,m)$ in the bottom-right cell of the tabular representation. Storing the pointers while composing the table, as it was shown for the *edit distance* [84], allows any optimal alignment to be built by tracing back any path of pointers from the bottom-right cell (n, m) to the top-left cell $(0, 0)$ in $\mathcal{O}(n + m)$ time. If the two strings are identical, a path along the main diagonal can be drawn in the tabular representation [56].

Table 3.8 shows an example of tabular computation of the similarity between the words Italian *fiore* and Spanish *flor* using global alignment, where the scoring scheme adopted is a 26-by-26 identity matrix on the Latin alphabet, with gap penalties equal to *−1*. The pointers from one cell to another show the step or the steps that maximise the score until that particular cell. The path that maximises the similarity score between the two words is shown in bold and the string similarity is $\mathcal{S}$(*fiore*,*flor*) = *2*.

$\mathcal{S}(i,j)$			f	l	o	r
		0	1	2	3	4
	0	**0**	← - 1	← - 2	← - 3	← - 4
f	1	↑ - 1	↖ **1**	← 0	← - 1	← - 2
i	2	↑ - 2	↑ 0	↖ **1**	↖← 0	↖← - 1
o	3	↑ - 3	↑ - 1	↖↑ 0	↖ **2**	← 1
r	4	↑ - 4	↑ - 2	↖↑ - 1	↑ 1	↖ **3**
e	5	↑ - 5	↑ - 3	↖↑ - 2	↑ 0	↑ **2**

Table 3.8: Example of tabular computation of global alignment using the identity matrix

Table 3.9 shows the optimal global alignment between the words Italian *fiore* and Spanish *flor* computed by tracing back the arrows, where mismatches and *indels* are displayed in bold.

f	***i***	*o*	*r*	***e***
f	***l***	*o*	*r*	**-**

Table 3.9: Example of one optimal global alignment produced using the identity matrix

3.4.2.2 Local string alignment

A local string alignment aims to identify similar regions between two strings, instead of looking at them as a whole. The optimal local alignment of two strings is the alignment of their substrings that presents the highest scoring. This concept becomes particularly helpful when some regions of two strings have accumulated so much noise through mutation that they are no longer alignable [41].

The similarity approach allows local and global alignment of strings to be performed, while the distance approach permits only the latter. To understand this point, let us remember that the distance method aims to find the alignment with the minimum score. Furthermore, by definition, the distance between any two substrings that are not equal is greater than zero. As a consequence, when aligning two strings, the substrings showing minimal distance should be identical, under ordinary scoring schemes [56].

The *dynamic programming* algorithm for solving the problem of local string alignment is known as the *Smith-Waterman algorithm* [125] in recognition of the *Smith-Waterman* paper, but the more efficient version generally used is by *Gotoh* [51]. It uses an $(n{+}1)$-by-$(m{+}1)$ matrix and is $\mathcal{O}(n * m)$ both in time and space, where n and m are the lengths of the two strings. This algorithm is closely related to the *Needleman-Wunsch algorithm* [102] for global alignment introduced in Section 3.4.2.1, but it presents two important differences.

Firstly, it forces a new alignment to start, if the current one has a negative score. This is achieved allowing each cell of the *dynamic programming* table to assume a value of zero, if all the other options have negative values. As a consequence, the maximum similarity score between two strings S_1 and S_2 is never less than zero, and no pointer is recorded unless the score is positive.

The base conditions of the algorithm fill in the top row and left column of the tabular computation with zeros:

$$\begin{aligned} \mathcal{S}(i,0) &= 0 \qquad && \forall i : 0 \leq i \leq n \\ \mathcal{S}(0,j) &= 0 \qquad && \forall j : 0 \leq j \leq m \end{aligned} \tag{3.9}$$

where n and m are the lengths of the strings S_1 and S_2, respectively. The algorithm recurrence relation, $\forall i : 0 < i \leq n$ and $\forall j : 0 < j \leq m$, is enriched with an extra possibility:

$$\mathcal{S}(i,j) = max \begin{cases} 0 \\ \mathcal{S}(i-1,j) + \mathcal{M}(S_1(i), `-`); \\ \mathcal{S}(i,j-1) + \mathcal{M}(`-`, S_2(j)); \\ \mathcal{S}(i-1,j-1) + \mathcal{M}(S_1(i), S_2(j)); \end{cases} \tag{3.10}$$

The second difference with the global alignment algorithm is that the optimal alignment score can be in any cell of the tabular computation and not necessarily in the bottom right corner, because an alignment can end anywhere in the matrix. The tracing back of an optimal alignment starts from the maximum score cell or cells, and ends when a cell with value zero is met and runs in $\mathcal{O}(n+m)$ time.

Table 3.10 shows an example of tabular computation of the similarity between the words Italian *fiore* and Spanish *flor* using local alignment, where the scoring scheme adopted is a 26-by-26 identity matrix on the Latin alphabet, with gap penalties equal to *−1*. The pointers from one cell to another show the step or the steps that maximise the score until that particular cell. The path that maximises the similarity score between the two words is shown in bold and the string similarity is $\mathcal{S}(fiore, flor) =$ *3*.

$\mathcal{S}(i,j)$			*f*	*l*	*o*	*r*
		0	1	2	3	4
	0	**0**	0	0	0	0
f	1	0	↖ **1**	0	0	0
i	2	0	0	↖ **1**	0	0
o	3	0	0	0	↖ **2**	← 1
r	4	0	0	0	↑ 1	↖ **3**
e	5	0	0	0	0	↑ 2

Table 3.10: Example of tabular computation of local alignment using the identity matrix

Table 3.11 shows the optimal local alignment between the words Italian *fiore* and Spanish *flor* computed by tracing back the arrows, where mismatches are displayed in bold.

f	***i***	*o*	*r*
f	***l***	*o*	*r*

Table 3.11: Example of one optimal local alignment produced using the identity matrix

It is worth noting that the optimal local alignment for the words Italian *fiore* and Spanish *flor*, has produced a higher similarity score than the optimal global alignment for the same word pair. This is because it has identified the word root, discarding inflectional suffixes.

3.4.3 Substitution matrices

Substitution matrices, or scoring matrices, are scoring schemes widely used in bioinformatics in the context of protein or nucleic acid sequence alignments, where they have a fundamental role. Indeed, it is sometimes suggested that the scoring matrix is the most critical technical element in a biological sequence alignment system [56].

Given an alphabet $\mathcal{A}$ with $\mid \mathcal{A} \mid \geq 2$, each character of $\mathcal{A}$ is more or less likely to transform into several other characters over time. A *substitution* *matrix* $\mid \mathcal{A} \mid$-by-$\mid \mathcal{A} \mid$ over $\mathcal{A}$ represents the rates at which each character of $\mathcal{A}$ may change into another character of $\mathcal{A}$. These rates, in principle, may be costs, when they signify distances, or may be scores, when they indicate similarities.

For example, a 20-by-20 scoring matrix for protein alignments tries to capture the possible transformation rates of the twenty amino acids[7] that form proteins. Similarly, a 4-by-4 scoring matrix for *DNA* or *RNA* sequences tries to express the possible transformation rates of the four nucleotides[8] that constitute nucleic acids. Each entry $[i, j]$ of these matrices tries to express the likelihood that the i element may be transformed into the j element in a certain amount of evolutionary time.

There are many different ways of constructing a substitution matrix, but the general approach is to collect a sample of verified pairwise alignments, or multiple sequence alignments, and derive from them the substitution parameters using a probabilistic model [56]. Ideally, the values in the substitution matrix should reveal the phenomena that the alignments try to represent and the scores in the matrix should be proportional to the true probabilities of mutations occurring through a period of evolution [56].

When aligning strings, the target is to assign a rate to the alignments that gives a measure of the relative likelihood that the strings are related, as opposed to being unrelated [41]. To compare these two hypotheses, the *log-odds ratio* is considered, which is the logarithm of the ratio of the probability that the sequences are associated, as opposed to being random. In the *related or match model* $\mathcal{M}$, aligned pairs of characters occur with a joint probability, and the probability for the whole alignment

[7] Proteins are made of 20 amino acids: Alanine, Arginine, Asparagine, Aspartic acid, Cysteine, Glutamine, Glutamic acid, Glycine, Histidine, Isoleucine, Leucine, Lysine, Methionine, Phenylalanine, Proline, Serine, Threonine, Tryptophan, Tyrosine, Valine. The alphabet for protein sequence analysis is $\mathcal{A} = \{$A; R; N; D; C; Q; E; G; H; I; L; K; M; F; P; S; T; W; Y; V$\}$

[8] Nucleic acids are made of 4 nucleotides: Adenine, Cytosine, Guanine and Thymine. The alphabet for nucleic acid sequence analysis is $\mathcal{A} = \{$A; C; G; T$\}$

is the product of these joint probabilities. In the *unrelated or random model* $\mathcal{R}$, the probability of the two strings is just the product of the probabilities of each character, because the model assumes that each character occurs independently [41].

If S_1 and S_2 are two aligned strings, $f(S_{1i})$ and $f(S_{2j})$ the frequency of the S_{1i} and S_{2j} character, respectively, and $f(S_{1i}, S_{2j})$ the joint probability that the characters S_{1i} and S_{2j} have derived from some unknown original residue, which may coincide with one or both characters, the *odds ratio* can be expressed as:

$$\frac{P(S_1, S_2 \mid \mathcal{M})}{P(S_1, S_2 \mid \mathcal{R})} = \frac{\prod_i f(S_{1i}, S_{2i})}{\prod_i f(S_{1i}) * \prod_i f(S_{2i})} = \prod_i \frac{f(S_{1i}, S_{2i})}{f\,(S_{1i}) * f(S_{2i})} \tag{3.11}$$

To obtain an additive scoring system, the *log-odds ratio* is considered as the logarithm of this ratio. When properly arranged, these *log-odds ratios* constitute the substitution matrix [41].

Ideally, if the similarity approach is adopted, positive and negative scores should indicate respectively conservative and non-conservative substitutions. Indeed, when two characters are expected to be aligned together in related strings more often than to occur by chance, then the *odds ratio* is greater than one and the score is positive. It is worth highlighting that the rates of identical character substitutions are inversely proportional to their occurrences, because the rarer the character is, the smaller the likelihood to find two of them aligned by chance [56].

In bioinformatics, several scoring schemes have been developed and the most widely used substitution matrices in protein sequence analysis at present are the *PAM* matrices [30, 31, 32] and the *BLOSUM* matrices [62], introduced in Sections 3.4.3.1 and the 3.4.3.2, respectively.

3.4.3.1 PAM matrices

The "*Point Accepted Mutation*" (*PAM*) matrices [30, 31, 32] are a family of amino acid substitution matrices, developed by *Margaret Dayhoff* and co-workers, that encode and summarise expected evolutionary changes of amino acids. An accepted point mutation in a protein is a replacement of one amino acid by another that has been accepted by natural selection and spread to its descendants. The foundation of the *PAM* approach is to obtain substitution rates from global alignments between closely related proteins and then to extrapolate from these data, longer evolutionary divergences. This approach assumes that the frequencies of the amino acids remain constant over time and that the mutational process causing replacements in a unitary interval, operate the same for longer periods. All the *PAMn* matrices are calculated by a *log-odds ratio* of a matrix M^n, where M represents the character substitution probabilities in a unitary time. Consequently, a higher number in the *PAM* family indicates a longer evolutionary distance and a lower sequence similarity [56].

The *PAM* method may also be viewed as a *Markov Model*, [41] where the states correspond to the twenty amino acids, and the state transition probabilities are the only parameters of the system. Every matrix M^n represents the result of n steps of a *Markov chain*. The probability of mutation at each site is independent of the occupants of other sites and of the previous history of mutations.

PAM matrices have proved to be very effective in detecting distant relationships between proteins, finding alignments able to show significant biological phenomena. They have been the standard and sole substitution matrices for amino acid alignments up until the advent of *BLOSUM* matrices [62].

3.4.3.2 BLOSUM matrices

The *BLOck SUbstitution Matrices* (*BLOSUM*) [62] are another successful family of amino acid scoring schemes, developed by *Steven and Jorja Henikoff*. These matrices derive from comparison of sequences extracted from the *BLOCKS* database [56], which contains multiple aligned segments, without gaps, corresponding to the most highly conserved regions of proteins. The *BLOSUM* approach is based on the belief that highly conserved sequence alignments from highly diverged protein sequences lead to accurate substitution score estimates [56]. Following this idea, each *BLOSUM* matrix in the family is built calculating *log-odds ratios* from blocks presenting no more than a certain threshold of similarity. For example, *BLOSUM62* is the matrix built using sequences with no more than 62% similarity. In this way larger numbers in the *BLOSUM* matrix family denote smaller evolutionary distance and therefore higher sequence similarity, which is the opposite of the *PAM* matrix family.

BLOSUM matrices have shown to be very effective in detecting similarities in distant sequences and are the main competitor for the *PAM* matrices [30, 31, 32]. Even if they are not discussed any further in this thesis, their exploitation in computational linguistics and natural language processing, forms part of our challenging future plans.

3.4.4 Cognate identification systems

When distance or similarity between cognates has to be evaluated, the methods applied can be either *orthographic*, where cognates are analysed in their writing form of graphemes, or *phonetic*, where cognates have to be represented in a phonetic notation in order to be examined.

The *orthographic approach* relies on the fact that alphabetic character correspondences represent in some way sound correspondences, as sound changes leave traces in the orthography. It benefits from not requiring any phonetic transcription, the attainment of which is still a very time-consuming and challenging task. In fact, there are only a few phonetic datasets of cognates available for computational linguistic applications. Furthermore, the task of automatic phonetic transcription is still some way from achieving the accuracy required to be used in the field of historical linguistics. The main issue is the variety of phonetic systems present in different languages, and the existence of homophones also within the same language. For example, the Italian word *ape*, meaning *bee*, is pronounced /ape/, while the English word *ape* is pronounced /eip/. Moreover, in non phonetic languages like English, pronunciation is often unpredictable and it is possible to have inside the same language, words with the same spelling, but different pronunciation. A classic example is the infinitive, the simple past and the past participle of the English verb *to read* /riːd/ - *read* /red/ - *read* /red/. For these reasons, phonetic transcriptions are still frequently executed manually [80] with the consequent dependency on linguistic collaboration, possible lack of uniformity and accuracy, and loss of time. On the other hand, *phonetic methods* depend on phonetic transcriptions of texts, but benefit from the phonetic characteristics and features of phonemes, which can be decomposed into vectors of phonetic attributes.

Even if for the task of cognate identification a phonetic approach is supposed to be more accurate than an orthographic one for its understanding of phonetic changes, the debate remains open and a comparative evaluation of several recent results seem to prove the opposite [86, 87, 83, 80].

Another differentiating feature between methodologies applied to the assessment of word relatedness is the ability to adapt, or not, to different contexts. Based on that, evaluation systems can be either *static*, or *active.* A *static system* is based on manually-designed and incorporated knowledge, does not require any supervision and is not able to learn by processing data. On the other hand, an *active system* has the capacity to learn and adjust, but may need supervision.

3.5 Phylogenetic inference

Phylogenies are evolutionary trees and phylogenetic inference aims to estimate the genetic relationships between *taxa*, which in principle may be species, languages or other entities [46]. In linguistics, a phylogenetic tree represents an estimation or hypothesis about the evolutionary relationships among groups of languages, based upon similarities and differences in their characters. For example, Latin evolved into modern Romance languages like Italian, Spanish, Portuguese, French and Romanian. This can be proved by analysing cognate words, which are lexical characters commonly used in historical linguistics. The languages appear as leaves in the tree and are joined together when they are supposed to descend from a common ancestor. Internal nodes represent intermediate, non-documented languages, and tree branch lengths may signify language distances or divergence time, accordingly with the methodology employed.

Phylogenetic trees are generally binary and they are either *unrooted*, when they only represent relationships between languages, or *rooted*, when they also identify a common ancestor. Any unrooted tree can be rooted on any of the internodes in the tree and it is compatible with all the rooted trees that can be built in this way. The number of unrooted binary trees for n languages is equal to $3 * 5 * 7 * \ldots * (2n - 5)$. Because for each unrooted tree there are $2n - 3$ possible rooted trees, the number of rooted trees is $3 * 5 * 7 * \ldots * (2n - 3)$ [46]. A common way to root an unrooted tree is to utilise an *outgroup*, which serves as a reference for determination of the evolutionary relationship among the other nodes. It should be a language considered related to the other languages in the set, but less closely related to any language in the group than they are to each other [106].

Phylogenetic networks are rooted direct graphs that may be used when, together with evolutionary relationships, more complex interactions need to be represented, which may include borrowing, creolisation or language mixture. In this thesis we have focussed on phylogenetic trees only.

3.5.1 Methods for phylogenetic inference

Methods for linguistic phylogenetic inference estimate the evolutionary history of languages using the information available about them. This information is generally coded in a matrix that may be a distance matrix or a character matrix. Depending on this, methods are classified as *distance-based* methods or *character-based* methods [106] and most of them are guaranteed to reproduce the true evolutionary tree, under certain conditions. When a method returns more than one tree having the same best score, a *consensus tree* has to be calculated [46].

Some methods not only aim to infer phylogenetic tree topologies, but also to estimate the dating of language divergence times that depend on the original character data and on various assumptions. The scholars are divided as to whether or not the currently available statistical methodology for dating purposes may be accepted with any degree of confidence in historical linguistics [44, 5, 4, 55].

3.5.1.1 Distance-based methods

Distance-based methods represent a major family of phylogenetic methods, where the initial character matrix is used to statistically calculate a pairwise distance matrix, which is then used to estimate a phylogenetic tree. It has been proved that the amount of information about the phylogeny that is lost in this process is remarkably small and that the estimates of the phylogenies produced by *distance-based* methods are quite accurate [46]. Distances may be considered estimates of the branch lengths separating pairs of languages, where different branches may have different rates of evolution.

A famous class of *distance-based* methods consists of *clustering algorithms*, which apply an algorithm to a distance matrix in order to produce a phylogenetic tree. These methods are very fast and, under certain assumptions, they are guaranteed to perform well. However, their statistical properties are not clear, because they do not optimise an explicit criterion [46]. Two standard clustering algorithms extensively used in phylogenetic inference are *UPGMA* [126] and *Neighbor-Joining* [119, 130].

- **UPGMA** (Unweighted Pair-Group Method with Arithmetic mean) [126] is guaranteed to perform well under the molecular clock hypothesis, which implies that the input distances represent languages that have evolved with a constant rate of evolution. This is a reasonable assumption, which follows the *glottochronology* approach [132], only if the entities are closely related. At each step, *UPGMA* combines together the nearest two clusters into a new cluster. The distance between the new cluster and the others is calculated as the mean distance between the elements of each cluster. The computational cost is $\mathcal{O}(n^2)$, where n is the distance matrix dimension. Clocklike trees are rooted and have an equal total branch length from the root to any leaf [46].

- **Neighbor-Joining** (*NJ*) [119, 130] is a clustering algorithm that is guaranteed to reconstruct phylogenetic trees perfectly, when the pairwise distances are the exact reflection of a tree. *NJ* assumes the minimum evolution criterion for phylogenetic trees and, at each iteration, it chooses the topology that minimises the total branch length. It produces an unrooted tree, that may be rooted by using an outgroup. The computational cost is $\mathcal{O}(n^3)$, where n is the distance matrix dimension.

3.5.1.2 Character-based methods

A language may be described by a vector of character states and a group of languages may be represented by a matrix, where each row symbolises a language and each column signifies a character. *Character-based* methods use a character matrix to estimate a phylogeny.

- **Maximum Parsimony** (*MP*) [46] is a non-parametric statistical method, whose target is to find an unrooted tree that requires the minimum number of evolutionary changes to describe the observed data. It may find several trees with the same best score. *MP* does not guarantee to produce the true tree because of the "*Long Branch Attraction*" [46]. This phenomenon occurs when the rates of evolution are very different on different branches of the true tree. In this case, *MP* considers closely related those lineages that evolve rapidly, regardless of their true evolutionary relationships. *MP* may be weighted, when different weights are assigned to different characters, or unweighted. Finding an *MP* tree is an *NP-complete* problem [50] and for this reason *MP* analyses are frequently performed using heuristics. Generally, these may find only local optima, rather then global optima, and anyway be very time-consuming.

- **Maximum Compatibility** (*MC*) [46] is a non-parametric method, which aims to find an unrooted tree that presents the maximum number of compatible characters to illustrate the observed data. Being compatible here means evolving without any homoplasy, i.e. without back mutation or parallel evolution. When a tree has all of the characters compatible, it is called a *perfect phylogeny. MC* may be weighted or unweighted and may find several trees with the same best score. The problem is *NP-complete* [14] and there are no highly accurate heuristics available. However, if the maximum number of states per character is bounded, then it is possible to find a solution in $\mathcal{O}(2^{2r} * n * k^2)$ where r is the maximum number of states per character, n is the number of leaves, and k is the number of characters [70].

- **Maximum Likelihood** (*ML*) methods [46] are based on explicit parametric models of character evolution and they aim to estimate the tree and the parameters that maximize the likelihood of the observed data, under the chosen evolutionary model. *ML* is statistically consistent and generally produces very good estimates of the phylogenetic tree, but it is *NP-hard* [20].
- **Bayesian methods** [46] are also based on explicit parametric models of character evolution. Their objective is to estimate a consensus tree, or sometimes the maximum posterior probability tree, of a posterior probability distribution on the space of the model trees, calculated from an initial tree and the observed data. *Bayesian methods* generally produce very good estimates of the phylogeny, but their computational time is extremely expensive. *Markov Chain Monte Carlo* (*MCMC*) algorithms [61] are frequently used to calculate an approximate posterior distribution of the trees instead. Initial prior parameters or *priors* may allow the inclusion in the evolutionary analysis of evidence available from other fields, such as genetics, anthropology and archaeology. However, the results should be examined considering both their sensitivity to the *priors* used and the reliability of the *MCMC* approximation of the tree probabilities [65].

3.5.2 Evaluation of phylogenetic inference

The evaluation of phylogenetic estimations is very difficult because the true evolutionary history is not generally fully known, even for the best understood language families. The choice of both data and phylogenetic inferring methodology significantly impact the phylogenetic estimation. The following criteria, proposed by *Nichols and Warnow* [106], should be a necessary and crucial requirement of any phylogenetic estimation, when using data from a well-known language family.

The "*Compatible resolution*" criterion requires that the inferred tree is compatible with the benchmark tree, meaning that the established subgroups should not be mixed, even if they may not be completely resolved. That may happen when the data are not sufficient to provide a complete resolution or when a consensus tree is used.

The "*No missing subgroups*" criterion requires that the estimated tree includes all the established subgroups and it is strictly stronger than the first criterion, and for this reason is considered desirable, but not essential.

The "*Calibration*" criterion is essential for models that estimate dating. It requires that a method is tested on one or more datasets and, if the inferred dates are not close enough to the established dates, the model has to be calibrated on the known dates.

3.6 The Indo-European language family

The Indo-European is one of the most intensively studied language families [38] and it is significant in the field of historical linguistics, as it possesses one of the longest recorded histories. There are only a few hundred languages belonging to this family, however, they are spoken by more than 45% of the global world population [85]. All languages are supposed to be descendants of a common ancestor, the *Proto-Indo-European*, and the basic subgroups are very well established. They include the extinct Anatolian and Tocharian and the contemporary Albanian, Armenian, Celtic, Germanic, Greek, Italic, Baltic and Slavic, grouped together into a Balto-Slavic clade, Indo-Aryan and Iranian, linked together to form an Indo-Iranian clade [106].

The origin of the Indo-European language family still represents one of the most recalcitrant problems of historical linguistics [38]. The relatively small number of languages and distinct branches contained in the Indo-European family, suggests that the reconstruction of Indo-European origins is complicated by large extinctions of its speech varieties, following the expansion of a few dominant subgroups [38].

The higher-order subgrouping of the Indo-European language family remains controversial [115], but the initial split into Anatolian versus all the others is linguistically well sustained. Moreover, some phylogenies have more support than others, including a radial phylogeny, one where Celtic departs very early, one that groups Balto-Slavic and Indo-Iranian together, or Armenian with Greek or Celtic with Italic [106].

We have applied our investigation to the Indo-European language family in an attempt to make a contribution to the problem of its first-order subgrouping, which has never reached any consensus [115].

Chapter 4

A string similarity measuring system

In recent decades, computational linguistics, and computational historical linguistics in particular, has aroused much interest in the scientific community. Cognate identification and phylogenetic inference represent key fields in the investigation of language evolution and have been successfully approached by various computational techniques. A number of different attempts to the cognate identification problem have been proposed including orthographic and phonetic systems, as well as learning or static procedures. In the field of phylogenetic inference, *distance-based* and *character-based* methods have been investigated as well.

In this chapter, we present a new orthographic learning system for the measurement of string similarity, that combines and adapts several techniques developed for biological sequence analysis to the natural language processing environment. Many of the ideas discussed in previous chapters are integrated into this new proposal that we successfully apply to both the fields of cognate identification and phylogenetic inference. For the former, we calculate word similarities and for the latter we compute language similarities, then transformed into language distances, to allow the estimation of phylogenetic trees with *distance-based* methods.

4.1 Architecture

In order to study word relatedness, we have designed a new learning system following the similarity approach, which is considered the standard in biological sequence analysis and frequently used in natural language processing. Similarity allows local alignment, as well as global alignment, to be performed and it leads to the maximisation problem of finding the highest scoring alignment or alignments of two strings [56].

Following this idea, we have developed a new orthographic learning system for measuring string similarity that, inspired by biological sequence analysis, consists of the three main modules described below. Each of them includes an original aspect:

- A global pairwise aligner, which sensibly aligns cognate pairs and prepares a meaningful training dataset, guided by a novel linguistic-inspired substitution matrix, described in Section 4.2. This 26-by-26 matrix aims to represent the a priori likelihood of transformation between each character of the Roman alphabet into another and tries to code well-known systematic sound changes left in written Indo-European languages.

- A generator of scoring matrices, which learns substitution parameters using several techniques, including *Maximum Likelihood*, *Absolute Frequency Ratio*, *Pointwise Mutual Information* and *PAM-like*, discussed in Section 4.3. For the latter, which has performed the best, we have developed a new technique inspired by the *PAM* method, introduced in Section 3.4.3.1. Designed by *Margaret Dayhoff* and co-workers [30, 31, 32], it is widely used for amino acid sequence analysis.

- A pairwise aligner, which, in order to measure the similarity between strings, benefits from the generated substitution matrices and from a novel family of parameterised string similarity measures, explained in Section 4.4. The similarity measures derive from different normalisations of a generic scoring algorithm and take into account the similarity of each string with itself, in the aim of eliminating, or at least reducing, the bias due to different string length.

Our proposal has been developed using data in orthographic format based on the Roman alphabet. However, it may easily be adapted to any alphabetic system, including the phonetic alphabet, if data were available.

4.2 A linguistic-inspired substitution matrix

Many learning techniques in bioinformatics take advantage of biological sequences, aligned by experts, available in organised databases. The first challenge we have faced in this study has been the lack of such resources of data for computational historical linguistic studies.

In order to generate automatically a sensibly aligned training dataset, we have prepared a linguistic-inspired substitution matrix in the belief that systematic phonetic changes leave their traces in the orthography of written languages.

We have considered the 26 letters of the Latin alphabet and we have produced a symmetric 26-by-26 matrix containing the a priori likelihood of transformation between each character of the alphabet into another for the Indo-European family. We have given a value of *2* to all the elements of the main diagonal, because it is likely that a character preserves itself. We have assigned a value of *0* to all the character transformations considered "*possible*", a value of *−3* to all the character transformations evaluated "*impossible*" and a gap penalty of *−1* for insertion and deletion, to avoid possible overlaps between two *indels* and an "*impossible*" match. For example, we have considered "*impossible*" that character '*A*' may change into character '*B*', while we have classified as "*possible*" that character '*A*' may transform into character '*E*'. For the classification of the character conversions evaluated "*possible*", we have considered several regular sound changes, including *vowel shift chain*, *Grimm*'s and *Verner*'s laws, *Centum-Satem division*, *rhotacism*, *assimilation*, *dissimilation*, *lenition*, *fortition* and *L-vocalisation* [2].

We have then used this matrix to perform global pairwise alignments by the *Needleman-Wunsch* algorithm [102, 51] on the cognate pairs of the training dataset in order to lay the foundations of a meaningful learning process. If the aligner for a word pair has found more than one optimal alignment with the same rate, it has chosen one of them through the alternate tracing back shown in brackets (↖←↑, ↖↑←, ←↖↑, ←↑↖, ↑←↖, ↑↖←). In doing this, we have aimed to eliminate possible bias caused by always giving priority to the same conditional predicates in the algorithm, and therefore assuring a more balanced learning process.

In Appendix B, we will provide the linguistic-inspired substitution matrix, together with an explanation of the choices we made and several examples of orthographic changes classified by linguistic motivations. This matrix has been proposed and discussed in [33, 34].

4.3 Substitution matrices

We have already discussed in Section 3.4.3 substitution matrices and their significance in biological sequence analysis. Scoring schemes are crucial for the performance of any string similarity measuring system and for this reason we have focussed on them, developing a generator of substitution matrices, which may employ different techniques.

Firstly, we have collected a sample of global pairwise alignments, obtained with the aid of the linguistic-inspired matrix, and we have derived from them substitution parameters using different probabilistic models. We have used several learning techniques on our training dataset in order to infer increasingly complex scoring matrices.

Each scoring matrix has been produced in two versions, one for the Roman alphabet and one for an extension of it including the gap, with the aim of understanding how to best manage gap penalties.

The proposed substitution matrices have then been utilised to measure word similarity, employing global and local alignment.

4.3.1 Maximum Likelihood matrices

A simple statistical method for inferring a scoring matrix from aligned data is to apply the *Maximum Likelihood* (*ML*) criterion [41]. *ML* for a model $\mathcal{M}$, estimates the values of the parameters Θ that make the dataset $\mathcal{D}$ as likely as possible. Formally:

$$\Theta^{ML} = \underset{\Theta}{argmax} \quad P(\mathcal{D}|\Theta, \mathcal{M}) \tag{4.1}$$

ML has the desirable property of being consistent, in the sense that the parameter values used to generate the dataset are also the values that maximise the likelihood. On the other hand, it does have the limitation of producing poor results, if the data are scarce [41].

Given a dataset of aligned words from an alphabet $\mathcal{A}$, with $\mid \mathcal{A} \mid \geq 2$, the *ML* estimate of the parameter $\Theta(\mathcal{A}_i, \mathcal{A}_j)$ is the observed relative frequency $f(i, j)$ of the character $\mathcal{A}_i$ being transformed into the character $\mathcal{A}_j$:

$$f(i, j) = \frac{\#(\mathcal{A}_i, \mathcal{A}_j)}{\sum_{k,h} \#(\mathcal{A}_k, \mathcal{A}_h)} \tag{4.2}$$

4.3.2 Absolute Frequency Ratio matrices

Another simple statistical method for estimating a substitution matrix from aligned data is to use absolute frequencies [108], instead of relative frequencies.

Given a set of aligned words from an alphabet $\mathcal{A}$, with $\mid \mathcal{A} \mid \geq 2$, the *Absolute Frequency Ratio* (*AFR*) of the character pair $(\mathcal{A}_i, \mathcal{A}_j)$ is the observed absolute frequency of character $\mathcal{A}_i$ being transformed into character $\mathcal{A}_j$, divided by the absolute frequency of character $\mathcal{A}_i$ and character $\mathcal{A}_j$:

$$f(i,j) = \frac{\#(\mathcal{A}_i, \mathcal{A}_j)}{\#\mathcal{A}_i * \#\mathcal{A}_j} \tag{4.3}$$

4.3.3 Pointwise Mutual Information matrices

Another statistical method for estimating a scoring matrix from aligned data, is the *Pointwise Mutual Information* (*PMI*). This measure derives from the *Mutual Information*, which was originally introduced by *Fano* [45] in the field of information theory. *PMI* has been applied to various disciplines, including lexicography [21] and dialectology [141].

PMI is a measure of association between two events described by discrete probability distributions and it is defined as the *log-odds ratio* of the joint probability of observing two events together, to the marginal probabilities of observing them independently.

If two events X and Y have probability distributions $p(X)$ and $p(Y)$, respectively, and joint probability $p(X,Y)$, their *PMI* is defined as:

$$PMI(X,Y) = \log_2 \frac{p(X,Y)}{p(X) * p(Y)} \tag{4.4}$$

PMI holds the following properties [90]:

1.	$PMI(X,Y) = 0$	$\leftrightarrow$	X,Y independent
2.	$PMI(X,Y) = -\log_2 p(X)$	$\leftrightarrow$	X,Y perfectly dependent
3.	$PMI(X,Y) = PMI(Y,X)$	$\forall$	X,Y

Given a dataset of aligned words from an alphabet $\mathcal{A}$, with $\mid \mathcal{A} \mid \geq 2$, in order to infer *PMI* substitution scores, we have calculated the relative frequencies of each character $\mathcal{A}_i$ and of the transformation of it into $\mathcal{A}_j$ in the training dataset. Each entry $PMI(i,j)$ of the matrix has been obtained by the *log-odds ratio* of the joint relative frequencies of the two characters $\mathcal{A}_i$ and $\mathcal{A}_j$, over the product of their disjoint relative frequencies:

$$PMI(i,j) = \log_2 \frac{f(i,j)}{f(i) * f(j)} \tag{4.5}$$

where

$$f(i,j) = \frac{\#(\mathcal{A}_i, \mathcal{A}_j)}{\sum_{k,h} \#(\mathcal{A}_k, \mathcal{A}_h)} \tag{4.6}$$

$$f(i) = \frac{\#\mathcal{A}_i}{\sum_k \#\mathcal{A}_k} \tag{4.7}$$

4.3.4 PAM-like matrices

In Section 3.4.3.1, we have introduced the *PAM* matrices [30, 31, 32] used in molecular biology, which represent one of the universal scoring schemes for that field. Because to our knowledge nothing similar exists in computational linguistics nor in natural language processing, we have decided to pioneer this fascinating approach.

In order to build the matrix *PAM1*, *Dayhoff et al.* [30, 31, 32] built hypothetical phylogenetic trees with the *Maximum Parsimony* method [46] from 71 protein families, where each pair of sequences showed amino acid diversity lower than 15%. They based the count of the accepted point mutations on the phylogenetic trees, in order to compare observed sequences with inferred ancestral sequences, rather than with each other.

Dayhoff and co-workers [30, 31, 32] constructed a non symmetric matrix M of mutation probabilities, where $M(i, j)$ contained the probability that amino acid $\mathcal{A}_j$ mutates to amino acid $\mathcal{A}_i$ in 1 *PAM* unit, performing the following steps. Firstly, a matrix A of accepted point mutation was calculated ignoring the evolutionary direction, meaning that $A(i, j)$ and $A(j, i)$ were incremented every time character $\mathcal{A}_i$ was replaced by $\mathcal{A}_j$ or vice-versa. Secondly, the relative mutability $m(j)$ of each amino acid $\mathcal{A}_j$ was calculated as the ratio of observed changes to the frequency of occurrence. The matrix M of mutation probabilities was computed as follows, where μ is a proportionality constant:

$$M(i,j) = \frac{\mu * m(j) * A(i,j)}{\sum_i A(i,j)} \qquad \forall i \neq j \tag{4.8}$$

$$M(i,i) = 1 - \mu * m(i) \qquad \forall i \tag{4.9}$$

To generate scoring matrices suitable for longer times, *Dayhoff et al.* [30, 31, 32] produced matrices M^n by multiplying matrix M by itself n times, which gives the probability that any particular amino acid mutates to another one in n *PAM* units. Each *PAMn* matrix was obtained by the following *log-odds ratio*, where $f(i)$ and $f(j)$ were the observed frequencies of amino acid $\mathcal{A}_i$ and $\mathcal{A}_j$, respectively, normalised by the number of all mutations:

$$PAMn(i,j) = 10 * \log_{10} \frac{f(j) * M^n(i,j)}{f(i) * f(j)} = 10 * \log_{10} \frac{M^n(i,j)}{f(i)} \tag{4.10}$$

Due to the lack of large and organised datasets of cognate words and to the small length of words, compared with the length of biological sequences, we have been forced to take some decisions that partially differentiate our method from the one *Margaret Dayhoff et al.* [30, 31, 32] used to create the *PAM* matrices for biological sequence analyses.

Indeed, we have not been able to identify in our dataset a useful group of cognate sets where each pair of words showed character diversity up to 15%. In fact, the group of cognate families extracted from our dataset showing up to 15% of diversity, has been completely inadequate, because it was composed of a few families of nearly identical words, where the only mismatches were due to *indels*. Increasing the diversity threshold up to 25% or 35% has not produced any substantial improvement. To understand the reason for this, let us consider, for example, the cognate words Italian *fiore* and French *fleur*, that are clearly very closely related, and show a diversity of 80% as 4 letters out of 5 represent mismatches. As a consequence, we have opted to use the whole dataset and, due to the small dimension of the cognate families and the short length of the cognate words, we have not built hypothetical phylogenetic trees. Instead, we have compared the cognate words with each other and not with their hypothetical ancestors. We have then followed the *Dayhoff et al.* [30, 31, 32] method, as described previously in this section, to produce a family of *PAM-like* matrices.

The *PAM-like* method has been introduced and discussed in [33, 34].

4.4 A family of string similarity measures

Under the similarity approach, alignments of two identical strings do not have a constant rate, because the score depends on the length of the strings and on the substitution rates of the characters involved. For this reason, instead of applying directly an aligning algorithm to the measurement of the similarity of string pairs, we have proposed a family of parameterised string similarity measures, obtained through different normalisations of a generic similarity rating algorithm.

Table 4.1 reports the family of parameterised string similarity measures proposed and the type of normalisation applied to them.

Similarity measure	*Normalised by*
$Sim_1(S_1, S_2, AL) = \frac{2*AL(S_1,S_2)}{AL(S_1,S_1)+AL(S_2,S_2)}$	*Arithmetic Mean*
$Sim_2(S_1, S_2, AL) = \frac{(len(S_1)+len(S_2))*AL(S_1,S_2)}{len(S_1)*AL(S_1,S_1)+len(S_2)*AL(S_2,S_2)}$	*Weighted Arithmetic Mean*
$Sim_3(S_1, S_2, AL) = \frac{AL(S_1,S_2)}{\sqrt{AL(S_1,S_1)*AL(S_2,S_2)}}$	*Geometric Mean*
$Sim_4(S_1, S_2, AL) = \frac{AL(S_1,S_2)}{\sqrt[len(S_1)+len(S_2)]{AL(S_1,S_1)^{len(S_1)}*AL(S_2,S_2)^{len(S_2)}}}$	*Weighted Geometric Mean*
$Sim_5(S_1, S_2, AL) = \frac{(AL(S_1,S_1)+AL(S_2,S_2))*AL(S_1,S_2)}{2*AL(S_1,S_1)*AL(S_2,S_2)}$	*Harmonic Mean*
$Sim_6(S_1, S_2, AL) = \frac{(len(S_1)*AL(S_2,S_2)+len(S_2)*AL(S_1,S_1))*AL(S_1,S_2)}{(len(S_1)+len(S_2))*AL(S_1,S_1)*AL(S_2,S_2)}$	*Weighted Harmonic Mean*
$Sim_7(S_1, S_2, AL) = \frac{3*AL(S_1,S_2)}{AL(S_1,S_1)+\sqrt{AL(S_1,S_1)*AL(S_2,S_2)}+AL(S_2,S_2)}$	*Heronian Mean*
$Sim_8(S_1, S_2, AL) = \frac{AL(S_1,S_2)}{\sqrt{(AL(S_1,S_1)^2+AL(S_2,S_2)^2)/2}}$	*Root Mean Square*
$Sim_9(S_1, S_2, AL) = \frac{(AL(S_1,S_1)+AL(S_2,S_2))*AL(S_1,S_2)}{AL(S_1,S_1)^2+AL(S_2,S_2)^2}$	*Contraharmonic Mean*

Table 4.1: A family of parameterised string similarity measures

Given two strings S_1 and S_2 and a generic rating algorithm AL, we have defined a set of similarity measures by normalising in various ways the similarity rate between the two strings, using the similarity rates of each string with itself. Our aim has been to eliminate, or at least reduce, the bias due to different string length.

The similarity measure Sim_1 normalises the rate of a scoring algorithm AL, applied to calculate the similarity of S_1 with S_2, by the *Arithmetic mean* [17] of the rates given by the same algorithm applied to calculate the similarity of each string with itself. The similarity measure Sim_2 does the same, but normalises the rate by the *Weighted Arithmetic mean* [17], that considers also the length of the two strings. The similarity measures Sim_3 and Sim_4 employ a normalisation by using the *Geometric mean* [17] and the *Weighted Geometric mean* [17], respectively. Sim_5 normalises by the *Harmonic mean* [17] and Sim_6 by the *Weighted Harmonic mean* [17]. The *Heronian mean* [17] is used to normalise the rate in Sim_7, the *Root mean square* [17] is utilised in Sim_8 and the *Contra-Harmonic mean* [17] is employed in Sim_9.

Following the idea of considering the similarity of each string with itself in calculating string similarity, other string similarity measures may be added to the family.

In this study, we have used these new string similarity measures with the *Needleman-Wunsch* algorithm [102, 51] for global alignment and with the *Smith-Waterman* algorithm [125, 51] for local alignment, but they may be used with any other similarity rating algorithm.

The family of parameterised string similarity measures has been proposed in [33].

4.5 Cognate identification

We have applied the proposed string similarity measuring system to the task of cognate identification. We have employed a training dataset and a test dataset without intersection in their language set.

We have sensibly aligned a training dataset of cognate pairs with the linguistic-inspired substitution matrix, presented in Section 4.2. We have then learnt scoring matrices from the aligned cognate pairs using several techniques, described in Section 4.3.

In order to test our cognate identification system, we have used language pairs built from the combination of the languages forming the test dataset, provided as 200-word *Swadesh lists* [132]. For each language pair and for all the word pairs with the same meaning in two languages, we have evaluated the likelihood that two words were cognates, by calculating a score. To give each alignment a score, we have employed the *Needleman-Wunsch* algorithm [102, 51] for global alignment, the *Smith-Waterman* algorithm [125, 51] for local alignment, both explained in Section 3.4.2, and the family of similarity measures based on them, introduced in Section 4.4. We have utilised the substitution matrices produced with the techniques investigated in Section 4.3.

To assess our learning system in the task of cognate identification, we have intentionally employed an evaluation methodology frequently used by other systems in the field of cognate identification [86, 87, 83, 80], with which we wanted to make our results properly comparable. This methodology addresses the cognate identification problem as a classification task, where the terms *True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN)* [108] are used to compare the system classifications with the correct cognateness judgements. The aim of any classification system is, of course, to maximise

TP and *TN*, as well as to minimise *FP* and *FN*. The outcome of a cognate classification system, combined with the available cognateness information, is summarised in Table 4.2.

Word pairs	Real Cognates	Non Real Cognates
Classified as Cognates	*TP*	*FP*
Classified as non-Cognates	*FN*	*TN*

Table 4.2: Contingency table for cognateness

As *Precision* [90] is the ratio of *TP* to the sum of *TP* and *FP*, in our context, it is the proportion of those pairs classified as cognates that are actually true cognates:

$$Precision = \frac{TP}{TP + FP} \tag{4.11}$$

As *Recall* [90] is the ratio of *TP* to the sum of *TP* and *FN*, in our context, it is the proportion of all cognates in the dataset that have been correctly classified:

$$Recall = \frac{TP}{TP + FN} \tag{4.12}$$

Following this evaluation methodology, we have not identified the word pairs "*Classified as Cognate*" or "*Classified as non-Cognate*" using a score threshold, which may be influenced by the type of application, the method used and the degree of language relatedness [80]. Instead, we have sorted the scores of each language pair and, when more word pairs have showed the same rate, we have considered the alphabetic order as well, to avoid random rankings.

We have then borrowed from the field of *Information Retrieval*, a measure specifically designed to evaluate rankings: the *11-point interpolated average precision* [90]. This measure, for each level of recall $R \in \{0.0, 0.1, 0.2, \ldots, 1.0\}$, calculates the interpolated precision, which is the highest precision found for any recall level $R' \geq R$. It then averages these 11 scores, providing a single value.

Figure 4.1 presents an example of *11-point interpolation procedure*, which generates a non-increasing monotonic function from non monotonic precision values. The precision is displayed in blue stars and the interpolated precision in red squares.

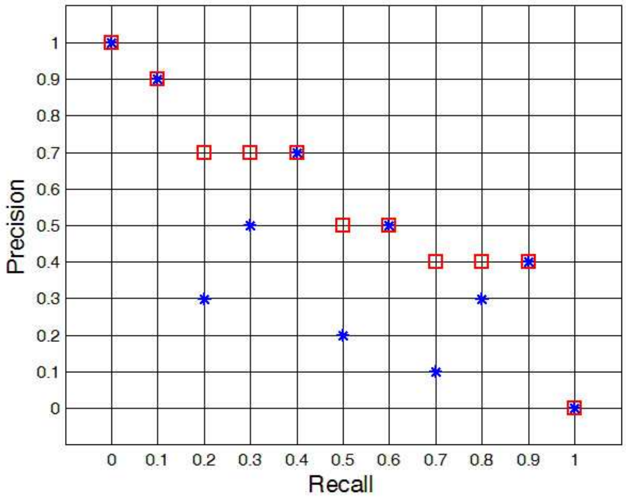


Figure 4.1: An example of *11-point interpolation procedure*

For each word pair in our ordered list, we have calculated the *Precision* and *Recall* [90] achieved. The *Precision* has been computed as the proportion of those word pairs classified as cognates till that point, that are actually true cognates. The *Recall* has been calculated as the proportion of all cognates in the dataset that have been correctly classified as cognates, till that point. We have finally calculated the *11-point interpolated average precision* [90] for each language pair considered.

4.6 Phylogenetic inference

We have developed a model to estimate phylogenies based on the learning system for measuring string similarity, described previously in this chapter. We have intentionally focussed on the tree topology and avoided modelling a dating scheme. We have chosen a training dataset of genetic cognate words without intersection with the test dataset and added an outgroup to root the phylogenetic trees.

Firstly, we have sensibly aligned the training dataset using the linguistic-inspired substitution matrix, introduced in Section 4.2. We have learnt scoring matrices from the aligned cognate pairs, following the techniques explained in Section 4.3. In order to calculate word similarity using these trained scoring matrices, we have employed global and local alignments and the family of string similarity measures based on them, proposed in Section 4.4. We have then utilised these calculated similarity scores between word pairs to compute similarity scores between language pairs. In doing this, we have employed *Swadesh lists* [132] of words with the same meaning in different languages. For each language pair, we have calculated and averaged the word pair similarity rates, producing an entry in a language similarity matrix. The calculation has been done for each substitution matrix, scoring algorithm and similarity measure utilised.

We have then faced the problem of converting these similarity matrices into distance matrices, in order to infer phylogenies utilising *distance-based* methods [46]. Because the similarity measures proposed are defined through different normalisations of a generic rating algorithm, the similarity scores fall in the range [0,1], where 0 means no similarity and 1 means maximum similarity. We have experimented with the three following methods of deriving a distance matrix from a similarity matrix, in the aim of studying possible differences in the resulting phylogenetic trees:

$$\begin{aligned} \mathcal{D}_1 &= 1 - \mathcal{S} \\ \mathcal{D}_2 &= -\ln(\mathcal{S}) \\ \mathcal{D}_3 &= 1/\mathcal{S} - 1 \end{aligned} \tag{4.13}$$

Given a generic n-by-n similarity matrix $\mathcal{S}$, in the first case we have taken the more obvious approach, calculating each entry of an n-by-n distance matrix $\mathcal{D}_1$, by subtracting the corresponding similarity value from 1, which produces rates in the range [0,1].

In the second case, we have calculated each entry of an n-by-n distance matrix $\mathcal{D}_2$, as the negative natural logarithm of the corresponding similarity value, which ranges from 0 to ∞.

In the third case, we have computed each entry of an n-by-n distance matrix $\mathcal{D}_3$, by first taking the reciprocal of the corresponding similarity value and then subtracting 1 from this quantity, which produces rates always equal to or greater than zero.

The last two methods were proposed and investigated by *Feng and Doolittle* [47] for measuring evolutionary times.

Having produced distance matrices from similarity matrices with the conversions described, we have then been able to employ *distance-based* methods, such as the *UPGMA* [126] and *Neighbor-Joining* [119, 130] algorithms, to estimate phylogenetic trees.

Chapter 5

Experimental results

In this chapter we apply the learning system introduced in Chapter 4 to the fields of cognate identification and phylogenetic inference. For the former, we measure word similarities to identify cognate words and for the latter we calculate language similarities, then transformed in language distances, to estimate phylogenies. We utilise orthographic data of Indo-European languages based on the Latin alphabet, in the belief that alphabetic character correspondences represent in some way sound correspondences, as phonetic changes leave traces in the orthography. However, our methodology may easily be adapted to any alphabetic system, including the phonetic alphabet, if data were available.

In cognate identification, when training *PAM-like* matrices, our system advances the state of the art. In fact, it outperforms comparable phonetic and orthographic previous proposals, with results which are statistically significant and remarkably stable, regardless of the variation of the training dataset dimension. When applied to phylogenetic inference of the Indo-European language family, whose higher structure does not have consensus, our system estimates phylogenies compatible with the Indo-European benchmark tree. Indeed, they reproduce correctly all the established major language groups and subgroups present in the dataset.

5.1 Cognate identification

We have applied to the task of cognate identification the learning system for measuring word similarity, introduced in Section 4. Firstly, we have employed a training dataset of genetic cognate words and a test dataset of 200-word *Swadesh lists* [132] with no intersection in their language sets. We have then sensibly aligned the cognate pairs of the training dataset using the *Needleman-Wunsch* algorithm [102, 51] for global alignment, together with the linguistic-inspired substitution matrix introduced in Section 4.2.

From this training dataset of aligned cognate pairs, we have learnt scoring matrices using several techniques described in Section 4.3, such as *Maximum Likelihood*, *Absolute Frequency Ratio*, *Mutual Information* and *PAM-like*. We have then utilised these substitution matrices, together with the *Needleman-Wunsch* algorithm [102, 51] for global alignment, the *Smith-Waterman* algorithm [125, 51] for local alignment and the family of parameterised string similarity measures, proposed in Section 4.4, to rate and order the word pairs of the test dataset. We have evaluated and compared the *11-point interpolated average precision* [90] achieved by our cognate identification system on the test dataset, for each technique employed and for each similarity measure based on global and local alignment, respectively. *PAM-like* matrices have performed very well, achieving the higher accuracy among the tested models, with results that have shown to be remarkably consistent, regardless of the training dataset dimension. Finally, we have assessed our system against comparable phonetic and orthographic methods previously reported in the literature. Our results have outperformed the others with a statistically significant improvement, which has shown to be independent from the training dataset dimension. This suggests that our learning system for measuring string similarity has advanced the state of the art in cognate identification.

5.1.1 Datasets

Any learning system depends heavily on the data with which it is trained and any test is greatly influenced by the complexity of the data it has to analyse. In order to develop our system and make our results properly comparable with others previously reported in the literature, we have intentionally chosen a training and a test dataset which have been utilised several times by other scholars. From these datasets, we have then extracted and exploited training and test orthographic data, which do not present any intersection in their language sets.

The *training dataset*[1] for our cognate identification system has been extracted from the Indo-European corpus provided by *Dyen et al.* [42] and documented in their monograph. This is one of the recommended sources for linguistic studies [106] and has been introduced in Section 2.3. In this dataset, a *Cognate Classification Number* (*CCN*) is utilised to identify different groups of words, with respect to their cognateness.

A *CCN* equal to 0 is used when, for a given meaning in a specific language, there is no word in the dataset or a word is not considered appropriate.

A *CCN* equal to 1 represents words which are believed to be unique in the dataset, i.e. not cognate with others. As a consequence, it is also used to classify borrowings. For example, the English word *flower*, which is considered a loan from Old French, is reported in this category.

A *CCN* in the range [2,99] identifies groups of words that are judged cognate with each other, but not cognate with words from any other group.

A *CCN* in the range [100,199] represents lists of words which are judged doubtfully cognate with each other and not cognate with words from any other list in the dataset.

[1] `http://www.wordgumbo.com/ie/cmp/iedata.txt`

A *CCN* in the range [200,399] classifies groups of words which are judged cognate with each other and either cognate or doubtfully cognate with at least one word from another group.

A *CCN* in the range [400,499] categorises lists of words that are judged doubtfully cognate with each other and doubtfully cognate with at least one word from another list.

From the 84 speech varieties proposed in the monograph, we have considered 6 languages: Italian, Portuguese and Spanish from the Romance family; Dutch, Danish and Swedish from the Germanic family. In doing so, we have aimed to have a balanced training dataset able to learn traces of sound correspondences left in the orthography of most of the language branches of which the test dataset is made, i.e. the Romance, Germanic and Albanian families. Contemporarily, we have avoided overlap between the languages of the training and test datasets.

From this group of 6 languages, we have extracted approximately 650 cognate pairs, by considering only the word pairs reported by *Dyen et al.* [42] as certain cognates with each other, which are classified in the corpus with *CCN* in the range [2,99]. This should ensure that our study does not include doubtful cognates or borrowings, identified by $CCN = 1$, which we wanted to discard from our training.

If more words were provided for the same meaning in the same language, we have considered the first word only, after ensuring that it was always cognate with the group, as explained in Section 6.2. This was achieved by putting in first position the word presenting the smaller averaged *edit distance* [84] with the other members of the group. We have also corrected some orthographic errors.

We have then aligned these cognate pairs using global alignment, together with the linguistic-inspired substitution matrix described in Section 4.2, to produce a meaningful training dataset.

The *test dataset*[2] for our cognate identification system, has consisted of the orthographic form of the 200-word *Swadesh lists* [132] of English, German, French, Latin and Albanian provided by *Kessler* [73] and enhanced with his judgement of their cognateness. We have used the ten language pairs deriving from the combination of these five languages to test our cognate identification system.

It is worth noting that the presence of the Albanian language makes this test dataset very challenging. In fact, Albanian constitutes its own branch in the Indo-European language family and it is not part of any recognised language group [18]. In order to keep the training and the test dataset separate, it has not been possible to train the system for it.

We have discovered in the digital file provided by *Kessler* [73] two inconsistencies related to the cognateness of two French - German word pairs, as the author has confirmed by private correspondence. While the Latin word *folium*, which means *leaf*, is reported to be cognate with the French word *feuille* and the German *Blatt*, the latter two are not reported as cognate with each other. The same happens to the Latin word *collum*, meaning *neck*, with the French *cou* word and German *Hals*. In order to make our results properly comparable with others reported in the literature [86, 87, 83, 80], where the same test dataset has been used, we have not corrected these errors and, for the same reason, we have not distinguished between cognates and loans.

[2] `http://www.artsci.wustl.edu/~bkessler/thesis/comparanda.xml`

5.1.2 NEDIT

In order to evaluate and compare the performance of our cognate identification system when using different learning techniques, we have used as a baseline the *edit distance* with unitary costs [84], normalised by the length of the longer string, called hereinafter *NEDIT*.

Table 5.1 shows the proportion of cognate words per language pair and the *11-point interpolated average precision* [90] for *NEDIT* over the ten language pairs of our test dataset, together with the *average, standard deviation, variance* and *median* [108].

Languages		Cognate proportion	NEDIT
English	German	0.590	0.907
French	Latin	0.560	0.921
English	Latin	0.290	0.703
German	Latin	0.290	0.591
English	French	0.275	0.659
French	German	0.245	0.498
Albanian	Latin	0.195	0.561
Albanian	French	0.165	0.499
Albanian	German	0.125	0.207
Albanian	English	0.100	0.289
Average		0.284	0.584
Standard deviation		0.168	0.231
Variance		0.028	0.054
Median		0.260	0.576

Table 5.1: *11-point interpolated average precision* for *NEDIT*

The *standard deviation* and the *variance* have been reported in order to measure not only how the system performs on average, but also the variability of dispersion of the results produced. A high *standard deviation* indicates that the data are spread over a wide range of values, while a low *standard deviation* signifies that the values tend to be very close to the average. The *median* specifies the central tendency.

5.1.3 Linguistic-inspired substitution matrix

We have tested the performance of our cognate identification system when employing the *Linguistic-Inspired substitution Matrix* (*LIM*) as scoring scheme, for a better evaluation of the benefits added by the learning process.

Table 5.2 displays the results in terms of averaged *11-point interpolated average precision* [90] reached by the family of similarity measures, introduced in Section 4.4, based on the *Needleman-Wunsch* algorithm [102, 51] for global alignment and on the *Smith-Waterman* algorithm [125, 51] for local alignment, respectively. The best outcome is shown in bold and the two algorithms for global and local alignment hereinafter are referred to as *NW* and *SW*, respectively.

Model	Algorithm	Basic	Sim_1	Sim_2	Sim_3	Sim_4	Sim_5	Sim_6	Sim_7	Sim_8	Sim_9
LIM	NW	0.604	0.606	0.603	**0.608**	0.604	**0.608**	0.606	**0.608**	0.603	0.603
LIM	SW	0.536	0.600	0.601	0.606	0.601	0.604	0.600	0.606	0.602	0.601

Table 5.2: Averaged *11-point interpolated average precision* for *LIM*

The results have not shown any significant difference when using global or local alignment, even if the *NW* algorithm [102, 51] and the family of similarity measures, which is based on it, has performed slightly better. When compared with the basic algorithm from which they derive, the similarity measures increase considerably the performance for local alignment, while for global alignment they do not provide a real benefit.

Table 5.3 provides a comparison of the *11-point interpolated average precision* [90] for *NEDIT* with the best results achieved by *LIM*, over the ten language pairs of our test dataset. The *average, standard deviation, variance* and *median* [108] are also reported and the best outcome is displayed in bold.

Languages		Cognate proportion	NEDIT	LIM Sim$_3$(NW)
English	German	0.590	0.907	0.913
French	Latin	0.560	0.921	0.924
English	Latin	0.290	0.703	0.722
German	Latin	0.290	0.591	0.630
English	French	0.275	0.659	0.707
French	German	0.245	0.498	0.580
Albanian	Latin	0.195	0.561	0.550
Albanian	French	0.165	0.499	0.441
Albanian	German	0.125	0.207	0.311
Albanian	English	0.100	0.289	0.300
Average		0.284	0.584	**0.608**
Standard deviation		0.168	0.231	0.219
Variance		0.028	0.054	0.048
Median		0.260	0.576	0.605

Table 5.3: *11-point interpolated average precision* for *NEDIT* and *LIM*

Our cognate identification system, when using the linguistic-inspired matrix as a substitution matrix in the test pairwise aligner, outperforms slightly *NEDIT*, introduced in Section 5.1.2.

5.1.4 Maximum Likelihood matrices

We have generated two *Maximum Likelihood* (*ML*) scoring matrices from the sensibly aligned cognate pairs extracted from the 6-language training dataset considered and based on the two alphabets employed, i.e. the Roman alphabet and its extension with gap. These two models have been named respectively *ML6* and *ML6b*.

Each entry of each matrix has been produced by calculating the number of transformation occurrences of the character $\mathcal{A}_i$ into $\mathcal{A}_j$ divided by the total number of transformations of any character into another, as explained in Section 4.3.1. We have multiplied all the final scores in the matrices by 100 for computational reasons and we have left the final scores with two decimal digits to preserve accuracy.

We have experimented with these substitution matrices on the test datasets with the *Needleman-Wunsch* algorithm [102, 51], the *Smith-Waterman* algorithm [125, 51] and the family of similarity measures based on them, introduced in Section 4.4. For *ML6*, which is the model based on the Roman alphabet, a gap penalty of *−1* has been applied. The results are reported in Table 5.4 and the best outcome is shown in bold.

Model	Algorithm	Basic	Sim_1	Sim_2	Sim_3	Sim_4	Sim_5	Sim_6	Sim_7	Sim_8	Sim_9
ML6	NW	0.531	0.596	0.596	**0.598**	0.596	0.597	0.595	0.596	0.593	0.593
ML6	SW	0.516	0.595	0.591	0.596	0.592	0.596	0.594	0.596	0.593	0.589
ML6b	NW	0.346	0.556	0.520	0.434	0.478	0.393	0.411	0.495	0.529	0.507
ML6b	SW	0.355	0.484	0.485	0.436	0.461	0.407	0.419	0.469	0.498	0.486

Table 5.4: Averaged *11-point interpolated average precision* for *ML6* and *ML6b*

ML6, which uses the Roman alphabet without gap, performs consistently better than *ML6b*, which utilises the extended alphabet and presents a poor performance. This suggests that *ML* is not able to estimate the *indel* rates, whose inference only adds noise into the system. The family of similarity measures introduced in Section 4.4 consistently outperforms the basic algorithms on which it is based. There is no significant difference within the same model in the performance of the two families of similarity measures based on global alignment and local alignment respectively, even if global alignment seems to perform slightly better.

Table 5.5 reports a comparison of the *11-point interpolated average precision* [90] for *NEDIT* with the best results achieved by *LIM* and *ML6*, over the ten language pairs of our test dataset. The *average, standard deviation, variance* and *median* [108] are also displayed and the best outcome is in bold.

Languages		Cognate proportion	NEDIT	LIM Sim$_3$(NW)	ML6 Sim$_3$(NW)
English	German	0.590	0.907	0.913	0.902
French	Latin	0.560	0.921	0.924	0.898
English	Latin	0.290	0.703	0.722	0.719
German	Latin	0.290	0.591	0.630	0.610
English	French	0.275	0.659	0.707	0.693
French	German	0.245	0.498	0.580	0.602
Albanian	Latin	0.195	0.561	0.550	0.555
Albanian	French	0.165	0.499	0.441	0.422
Albanian	German	0.125	0.207	0.311	0.326
Albanian	English	0.100	0.289	0.300	0.251
Average		0.284	0.584	**0.608**	0.598
Standard deviation		0.168	0.231	0.219	0.219
Variance		0.028	0.054	0.048	0.048
Median		0.260	0.576	0.605	0.606

Table 5.5: *11-point interpolated average precision* for *NEDIT*, *LIM* and *ML6*

ML matrices do not produce very good results and their best outcome reaches an average accuracy only slightly better than our baseline *NEDIT* reported in Section 5.1.2, probably because of the small dimension of the training dataset [41].

5.1.5 Absolute Frequency Ratio matrices

We have produced two *Absolute Frequency Ratio* (*AFR*) substitution matrices, one for each of the two alphabets employed, i.e. the Roman alphabet and its extension with gap. The scoring matrices have been trained with the sensibly aligned cognate pairs extracted from the 6-language training dataset considered. These two models have been called respectively *AFR6* and *AFR6b*.

Each entry of each matrix has been produced by calculating the number of transformation occurrences of the character $\mathcal{A}_i$ into $\mathcal{A}_j$ divided by the number of occurrences of $\mathcal{A}_i$ and $\mathcal{A}_j$ respectively, as explained in Section

4.3.2. We have multiplied all the final scores in the matrices by 100 for computational reasons and we have left the final scores with two decimal digits to preserve accuracy.

The *11-point interpolated average precision* [90] has been calculated and averaged over the ten language pairs belonging to the test dataset using *AFR* matrices as substitution matrices for global and local alignment. The family of similarity measures based respectively on the *Needleman-Wunsch* algorithm [102, 51] and on the *Smith-Waterman* algorithm [125, 51] has been employed. A gap penalty of *−1* has been applied to *AFR6*, which is based on the Roman alphabet without gap. The outcome is reported in Table 5.6, where the best result is in bold.

Model	Algorithm	Basic	Sim$_1$	Sim$_2$	Sim$_3$	Sim$_4$	Sim$_5$	Sim$_6$	Sim$_7$	Sim$_8$	Sim$_9$
AFR6	NW	0.533	0.659	0.651	0.664	0.660	**0.669**	0.663	0.661	0.650	0.646
AFR6	SW	0.519	0.654	0.649	0.660	0.656	0.663	0.664	0.656	0.650	0.643
AFR6b	NW	0.457	0.632	0.627	0.634	0.632	0.621	0.637	0.633	0.627	0.623
AFR6b	SW	0.469	0.647	0.646	0.653	0.652	0.641	0.658	0.650	0.645	0.637

Table 5.6: Averaged *11-point interpolated average precision* for *AFR6* and *AFR6b*

AFR6 and *AFR6b* with the family of similarity measures based on *NW* and *SW*, respectively, produce reasonably good results. *AFR6*, which uses the Roman alphabet without gap, performs only slightly better then *AFR6b* and reaches the best result with global alignment. This would suggest that *AFR* matrices are able to estimate the *indel* rates, even if the inference is not precise enough to produce an improvement in the accuracy. The family of similarity measures introduced in Section 4.4 consistently outperforms the basic algorithm on which they are based. There is no significant difference in the performance of the two families of similarity measures based on global alignment and local alignment respectively. However, global alignment seems to perform slightly better for *AFR6*, while local alignment gives a better outcome for *AFR6b*.

Table 5.7 reports a comparison of the *11-point interpolated average precision* [90] for *NEDIT* with the best results achieved by *LIM*, *ML6* and *AFR6*, over the ten language pairs of our test dataset, together with the *average, standard deviation, variance* and *median* [108]. The best outcome is displayed in bold.

Languages		Cognate proportion	NEDIT	LIM Sim_3(NW)	ML6 Sim_3(NW)	AFR6 Sim_5(NW)
English	German	0.590	0.907	0.913	0.902	0.909
French	Latin	0.560	0.921	0.924	0.898	0.924
English	Latin	0.290	0.703	0.722	0.719	0.776
German	Latin	0.290	0.591	0.630	0.610	0.706
English	French	0.275	0.659	0.707	0.693	0.768
French	German	0.245	0.498	0.580	0.602	0.700
Albanian	Latin	0.195	0.561	0.550	0.555	0.584
Albanian	French	0.165	0.499	0.441	0.422	0.557
Albanian	German	0.125	0.207	0.311	0.326	0.486
Albanian	English	0.100	0.289	0.300	0.251	0.280
Average		0.284	0.584	0.608	0.598	**0.669**
Standard deviation		0.168	0.231	0.219	0.219	0.197
Variance		0.028	0.054	0.048	0.048	0.039
Median		0.260	0.576	0.605	0.606	0.703

Table 5.7: *11-point interpolated average precision* for *NEDIT*, *LIM*, *ML6* and *AFR6*

AFR matrices produce considerably better results in terms of averaged *11-point interpolated average precision* [90] than *NEDIT*, *LIM* and *ML* matrices. In addition, the *standard deviation* and *variance* are lower and the *median* is higher.

5.1.6 Pointwise Mutual Information matrices

We have built two *Pointwise Mutual Information* (*PMI*) scoring matrices, based on the Roman alphabet and its extension with gap, from the sensibly aligned cognate pairs belonging to the 6-language training dataset. The two models have been named respectively *PMI6* and *PMI6b*.

Each entry (i, j) of each matrix has been obtained by the *log-odds ratio* of the joint relative frequencies of the two characters $\mathcal{A}_i$ and $\mathcal{A}_j$, over the product of their disjoint relative frequencies, as explained in Section 4.3.3. We have left the final scores with two decimal digits to preserve accuracy.

We have tested these substitution matrices with the *Needleman-Wunsch* algorithm [102, 51], the *Smith-Waterman* algorithm [125, 51] and the family of similarity measures based on them, introduced in Section 4.4. For the model based on the Roman alphabet, a gap penalty of *−1* has been applied in the alignment algorithms. Table 5.8 reports the results of the *11-point interpolated average precision* [90] averaged over the ten language pairs of the test dataset, with the best outcome in bold.

Model	Algorithm	Basic	Sim_1	Sim_2	Sim_3	Sim_4	Sim_5	Sim_6	Sim_7	Sim_8	Sim_9
PMI6	NW	0.606	0.681	0.681	0.683	0.681	0.688	0.683	0.682	0.679	0.678
PMI6	SW	0.581	0.706	0.701	**0.711**	0.705	0.708	0.708	0.707	0.702	0.696
PMI6b	NW	0.393	0.637	0.638	0.602	0.644	0.577	0.633	0.633	0.641	0.635
PMI6b	SW	0.419	0.677	0.672	0.659	0.682	0.613	0.682	0.678	0.672	0.666

Table 5.8: Averaged *11-point interpolated average precision* for *PMI6* and *PMI6b*

The model *PMI6*, which uses the Roman alphabet without gap, produces good results, especially when employing local alignment and outperforms consistently *PMI6b*, which utilises the extended alphabet. This would suggest that *PMI* matrices are not able to estimate the *indel* rates, whose inference adds noise into the system. The similarity measures, proposed in Section 4.4, consistently outperform the basic algorithm on which they are based and Sim_3 with SW produces the best result.

Table 5.9 reports a comparison of the *11-point interpolated average precision* [90] for *NEDIT* with the best results achieved by *LIM*, *ML6*, *AFR6* and *PMI6*, over the ten language pairs of our test dataset, together with the *average, standard deviation, variance* and *median* [108]. The best outcome is displayed in bold.

Languages		Cognate proportion	NEDIT	LIM Sim_3(NW)	ML6 Sim_3(NW)	AFR6 Sim_5(NW)	PMI6 Sim_3(SW)
English	German	0.590	0.907	0.913	0.902	0.909	0.925
French	Latin	0.560	0.921	0.924	0.898	0.924	0.925
English	Latin	0.290	0.703	0.722	0.719	0.776	0.795
German	Latin	0.290	0.591	0.630	0.610	0.706	0.745
English	French	0.275	0.659	0.707	0.693	0.768	0.790
French	German	0.245	0.498	0.580	0.602	0.700	0.757
Albanian	Latin	0.195	0.561	0.550	0.555	0.584	0.676
Albanian	French	0.165	0.499	0.441	0.422	0.557	0.621
Albanian	German	0.125	0.207	0.311	0.326	0.486	0.470
Albanian	English	0.100	0.289	0.300	0.251	0.280	0.404
Average		0.284	0.584	0.608	0.598	0.669	**0.711**
Standard deviation		0.168	0.231	0.219	0.219	0.197	0.173
Variance		0.028	0.054	0.048	0.048	0.039	0.030
Median		0.260	0.576	0.605	0.606	0.703	0.751

Table 5.9: *11-point interpolated average precision* for *NEDIT*, *LIM*, *ML6, AFR6* and *PMI6*

PMI6, when based on *SW*, produces considerably better results than *NEDIT*, *LIM*, *ML* and *AFR* matrices. It is worth noting that not only the *average* and the *median* of the *11-point interpolated average precision* [90] are higher, but also the *standard deviation* and *variance* are much lower. This would suggest that *PMI6* is also more stable in its performance across various language pairs.

Furthermore, *PMI6* with local alignment slightly outperforms comparable phonetic and orthographic models previously proposed in the literature [86, 87, 83, 80], as shown in Section 5.1.9.

Interestingly, out of all the similarity measures proposed, Sim_3 and Sim_5 produce the greater accuracy for every model considered. These measures normalise the rate of the scoring algorithm on which they are based, respectively, by the *Geometric* and *Harmonic mean* [17] of the rates given by the same algorithm applied to calculate the similarity of each string with itself.

5.1.7 PAM-like matrices

We have trained two families of *PAM-like* substitution matrices from the sensibly aligned cognate pairs belonging to the 6-language training dataset employed. One family has been based on the Roman alphabet and one on its extension with gap. The two models have been named *DAY6* and *DAY6b*, respectively.

Each *PAM-like* matrix has been produced following the approach proposed in Section 4.3.4 and inspired by the *PAM* method introduced by *Dayhoff et al.* [30, 31, 32] for biological sequence analysis. We have not scaled the values in the *PAM-like* matrices and we have left the final scores with two decimal digits to preserve accuracy.

Because we have not limited the diversity percentage within the cognate family employed for the training, ten *PAM-like* matrices for each family have shown to be sufficient for modelling the divergence time of the languages present in the test dataset. As the identity matrix can be considered as a *PAM* matrix at 0 evolutionary distance [32], it has been included for completeness in all the results.

We have used the *PAM-like* matrices to align the test dataset with the *Needleman-Wunsch* algorithm [102, 51], the *Smith-Waterman* algorithm [125, 51] and the family of similarity measures based on them, introduced in Section 4.4. A gap penalty of *−1* has been applied in the alignment algorithms when using *DAY6*, which is based on the Roman alphabet without gap. For each alignment algorithm and similarity measure considered, we have calculated the *11-point interpolated average precision* [90] over the ten language pairs of our test dataset, using the two families of *PAM-like* substitution matrices.

Table 5.10 reports *PAM5* generated by *DAY6b* as an example. For readability, only the lower triangular matrix is filled in.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	–
A	3.65																										
B	-5.77	12.69																									
C	-2.89	1.80	3.81																								
D	-3.77	-5.06	0.01	6.24																							
E	2.09	-3.92	-1.44	-2.01	1.92																						
F	-5.80	8.89	3.84	-3.97	-3.71	8.77																					
G	-3.75	-0.73	3.68	0.21	-2.04	1.31	8.92																				
H	-1.14	-0.28	2.07	0.73	-0.26	1.56	0.86	2.94																			
I	1.47	-4.14	-1.24	-1.93	1.66	-3.71	-1.97	0.16	2.32																		
J	-0.36	-2.99	0.53	0.61	0.17	-1.68	2.12	1.24	0.55	1.34																	
K	-4.72	-2.78	5.26	-1.91	-2.71	1.35	5.82	3.59	-2.59	0.50	11.93																
L	-2.75	-4.25	0.60	0.25	-1.55	-2.06	-0.74	0.12	0.55	0.78	-1.21	6.61															
M	-5.14	-2.55	-2.34	-2.80	-3.27	-1.93	-2.22	-1.96	-3.36	0.87	-3.85	-2.50	11.22														
N	-3.32	-4.63	-1.16	-0.33	-1.64	-4.29	-2.32	-0.86	-1.65	-0.78	-0.89	0.35	2.46	6.85													
O	2.35	-5.16	-2.63	-3.54	1.56	-5.05	-3.49	-1.19	1.38	-0.76	-4.39	-2.70	-4.82	-3.05	3.92												
P	-8.88	8.62	3.14	-6.96	-6.43	10.27	-1.14	0.03	-6.08	-4.15	-1.93	-1.08	-4.04	-6.79	-7.96	14.04											
Q	-2.91	4.32	3.87	-1.43	-1.48	5.72	2.41	2.21	-1.42	0.04	3.18	-0.73	-1.23	-1.86	-2.44	5.22	5.03										
R	-6.30	-7.58	-3.30	-3.38	-4.33	-7.21	-4.93	-3.22	-3.92	-2.48	-2.28	1.59	1.46	-1.81	-5.97	-8.90	-4.52	8.04									
S	-2.69	-4.09	0.36	-0.05	-1.12	-3.18	-1.48	0.27	-1.24	1.25	-1.73	-1.75	-2.72	-1.56	-2.50	-6.33	-0.77	-0.02	7.04								
T	-3.67	-4.53	0.56	5.80	-1.93	-2.89	-0.83	1.12	-1.84	0.29	-1.33	0.41	-2.28	-0.81	-3.41	-5.71	-0.66	-3.50	-0.61	6.12							
U	1.82	-2.32	-0.93	-2.24	1.08	-1.64	-1.52	0.43	1.41	0.37	-2.28	0.53	-2.95	-2.04	1.58	-3.50	-0.44	-4.21	-1.66	-2.08	2.33						
V	-2.89	5.98	2.42	-2.77	-1.74	5.92	3.33	0.46	-1.82	-0.43	0.32	-2.19	2.24	-2.39	-2.15	5.76	4.48	-5.01	-2.24	-2.55	0.28	6.31					
W	-0.37	2.99	1.08	-1.98	-0.04	3.36	1.45	0.18	-0.15	-0.28	-0.91	-1.75	0.24	-1.64	0.46	2.37	2.81	-4.20	-1.29	-1.83	0.77	4.48	3.29				
X	-3.93	-2.86	1.91	-0.13	-1.72	-0.46	0.10	0.84	-1.87	1.32	1.08	-1.59	-3.62	-2.18	-3.56	-2.72	1.12	-0.76	7.21	-0.45	-2.17	-1.13	-0.96	7.51			
Y	2.01	-4.55	-1.18	-2.20	1.70	-3.43	-1.41	1.16	2.20	1.24	-2.04	-0.20	-3.13	-2.66	1.07	-6.52	-1.11	-5.37	-1.40	-2.23	2.29	-1.42	-0.05	-2.16	3.39		
Z	-3.26	-3.52	1.04	0.91	-1.50	-1.71	-0.54	0.51	-1.58	1.06	-0.13	-1.16	-2.63	-1.70	-3.02	-4.22	0.14	0.60	6.40	0.60	-1.92	-1.71	-1.22	6.63	-1.78	5.87	
–	0.13	-1.29	0.11	0.16	0.35	-0.92	0.04	0.34	0.37	0.29	-0.08	-0.20	-0.75	0.29	0.10	-2.44	0.01	-1.53	0.53	0.17	0.19	-0.26	0.05	0.55	0.39	0.41	0.18

Table 5.10: *PAM5* generated by *DAY6b*

It is worth noting that this matrix contains positive and negative scores, which indicate conservative and non-conservative substitutions, respectively. The positive scores reproduce linguistic sound changes left in the orthography, which are described in Appendix B. As it was expected, the rates on the main diagonal, which represent identical character substitutions, are all positive and inversely proportional to the character occurrences. Indeed, the less frequent a character is, the lower the probability of finding two of them aligned by chance [56]. For example, the diagonal rates of vowels, which are frequent characters, are lower than the rates of less frequent characters, like the consonants *B, K, M* and *P.*

Tables 5.11 and 5.12 show the averaged *11-point interpolated average precision* [90] obtained by using the *PAM-like* matrices belonging to the *DAY6* model on the test dataset. The *Needleman-Wunsch* [102, 51] algorithm for global alignment and the *Smith-Waterman* algorithm [125, 51] for local alignment have been employed with the family of similarity measures, presented in Section 4.4. The best results are displayed in bold.

Matrix	NW	Sim$_1$	Sim$_2$	Sim$_3$	Sim$_4$	Sim$_5$	Sim$_6$	Sim$_7$	Sim$_8$	Sim$_9$
PAM0	0.509	0.524	0.524	0.524	0.524	0.524	0.524	0.524	0.524	0.524
PAM1	0.580	0.666	0.660	0.670	0.663	0.674	0.669	0.668	0.662	0.658
PAM2	0.607	0.698	0.691	0.699	0.690	0.705	0.698	0.698	0.691	0.690
PAM3	0.616	0.713	0.708	0.715	0.712	0.721	0.714	0.713	0.711	0.710
PAM4	0.624	0.721	0.713	0.724	0.719	**0.729**	0.722	0.721	0.719	0.713
PAM5	0.623	0.721	0.716	0.723	0.718	0.727	0.722	0.720	0.717	0.714
PAM6	0.619	0.721	0.716	0.725	0.718	0.726	0.722	0.724	0.717	0.713
PAM7	0.617	0.718	0.714	0.723	0.717	0.725	0.720	0.719	0.717	0.715
PAM8	0.616	0.713	0.712	0.719	0.713	0.722	0.717	0.715	0.714	0.711
PAM9	0.613	0.714	0.710	0.718	0.713	0.722	0.715	0.715	0.712	0.709
PAM10	0.609	0.715	0.707	0.716	0.710	0.719	0.712	0.715	0.711	0.708

Table 5.11: Averaged *11-point interpolated average precision* for *DAY6* using *NW*

Matrix	SW	Sim$_1$	Sim$_2$	Sim$_3$	Sim$_4$	Sim$_5$	Sim$_6$	Sim$_7$	Sim$_8$	Sim$_9$
PAM0	0.516	0.585	0.591	0.590	0.591	0.590	0.585	0.590	0.591	0.591
PAM1	0.562	0.670	0.666	0.677	0.666	0.675	0.673	0.671	0.667	0.661
PAM2	0.584	0.698	0.694	0.703	0.699	0.703	0.698	0.696	0.696	0.692
PAM3	0.597	0.718	0.709	0.724	0.716	0.720	0.718	0.720	0.719	0.706
PAM4	0.597	0.727	0.722	0.732	0.723	0.728	0.728	0.729	0.725	0.721
PAM5	0.599	0.733	0.725	0.732	0.729	0.732	0.732	0.733	0.729	0.725
PAM6	0.597	0.730	0.726	**0.735**	0.729	**0.735**	0.732	0.731	0.729	0.726
PAM7	0.596	0.729	0.725	0.732	0.729	0.733	0.731	0.730	0.728	0.724
PAM8	0.587	0.729	0.721	0.730	0.725	0.734	0.731	0.729	0.726	0.720
PAM9	0.581	0.730	0.720	0.729	0.725	0.731	0.731	0.730	0.725	0.718
PAM10	0.577	0.725	0.717	0.725	0.722	0.731	0.725	0.725	0.722	0.719

Table 5.12: Averaged *11-point interpolated average precision* for *DAY6* using *SW*

Tables 5.13 and 5.14 show the outcome in terms of averaged *11-point interpolated average precision* [90] obtained by using the *DAY6b* model on the test dataset. The *Needleman-Wunsch* [102, 51] algorithm and the *Smith-Waterman* algorithm [125, 51] have been employed with the family of similarity measures, introduced in Section 4.4. The best results are displayed in bold.

Matrix	NW	Sim_1	Sim_2	Sim_3	Sim_4	Sim_5	Sim_6	Sim_7	Sim_8	Sim_9
PAM0	0.509	0.524	0.524	0.524	0.524	0.524	0.524	0.524	0.524	0.524
PAM1	0.532	0.681	0.669	0.680	0.678	0.678	0.684	0.682	0.675	0.670
PAM2	0.599	0.716	0.711	0.718	0.717	0.714	0.716	0.716	0.714	0.710
PAM3	0.576	0.728	0.722	0.730	0.727	0.729	0.728	0.729	0.726	0.719
PAM4	0.576	0.735	0.729	0.737	0.733	0.737	0.737	0.735	0.731	0.728
PAM5	0.570	0.733	0.728	**0.743**	0.732	0.741	0.740	0.737	0.730	0.725
PAM6	0.565	0.734	0.727	0.740	0.733	0.739	0.738	0.734	0.728	0.723
PAM7	0.558	0.732	0.723	0.741	0.730	0.738	0.735	0.735	0.727	0.722
PAM8	0.550	0.730	0.721	0.736	0.726	0.731	0.733	0.732	0.725	0.721
PAM9	0.539	0.728	0.718	0.735	0.724	0.730	0.731	0.730	0.724	0.718
PAM10	0.530	0.725	0.717	0.733	0.722	0.724	0.728	0.726	0.722	0.716

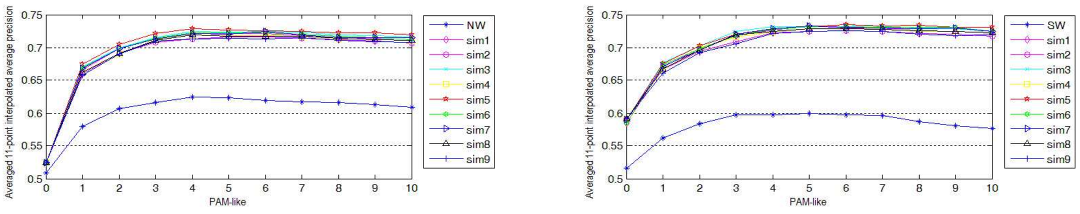
Table 5.13: Averaged *11-point interpolated average precision* for *DAY6b* using *NW*

Matrix	SW	Sim_1	Sim_2	Sim_3	Sim_4	Sim_5	Sim_6	Sim_7	Sim_8	Sim_9
PAM0	0.516	0.585	0.591	0.590	0.591	0.590	0.585	0.590	0.591	0.591
PAM1	0.524	0.683	0.674	0.685	0.681	0.684	0.687	0.685	0.676	0.669
PAM2	0.570	0.707	0.701	0.710	0.705	0.710	0.709	0.709	0.703	0.702
PAM3	0.567	0.727	0.721	0.730	0.725	0.728	0.726	0.728	0.725	0.721
PAM4	0.574	0.737	0.730	0.738	0.735	0.735	0.739	0.738	0.734	0.731
PAM5	0.567	0.736	0.732	**0.749**	0.736	0.743	0.740	0.739	0.734	0.729
PAM6	0.564	0.739	0.731	0.747	0.738	0.747	0.745	0.742	0.734	0.730
PAM7	0.559	0.740	0.725	0.746	0.737	0.745	0.746	0.742	0.730	0.728
PAM8	0.550	0.736	0.725	0.742	0.734	0.742	0.741	0.739	0.729	0.726
PAM9	0.542	0.734	0.724	0.740	0.734	0.738	0.741	0.737	0.730	0.723
PAM10	0.526	0.732	0.722	0.740	0.732	0.734	0.738	0.735	0.728	0.723

Table 5.14: Averaged *11-point interpolated average precision* for *DAY6b* using *SW*

The two models *DAY6* and *DAY6b* achieve excellent results. The family of similarity measures, proposed in Section 4.4, consistently outperforms the basic algorithm on which it is based. The group that performs better employs local alignment, even if the difference when using global alignment is not significant. *DAY6b* outperforms *DAY6* and achieves the best result utilising *PAM5* with Sim_3 based on *SW*. This would suggest that *DAY6b* is also able to learn appropriate gap penalties.

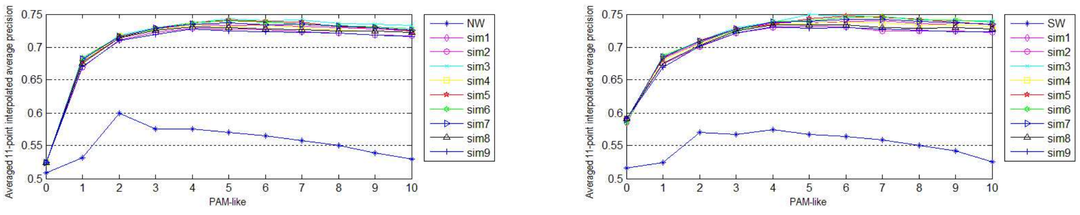
Figure 5.1 shows a graphical representation of the averaged *11-point interpolated average precision* [90] produced by the *PAM-like* matrices of *DAY6*, when using *NW* [102, 51] and *SW* [125, 51], respectively, and the similarity measures that are based on them.



(a) *NW* **(b)** *SW*

Figure 5.1: Averaged *11-point interpolated average precision* for *DAY6* using *NW* and *SW*

Figure 5.2 presents in a graphical format the averaged *11-point interpolated average precision* [90] produced by the *PAM-like* matrices of *DAY6b*, when using *NW* [102, 51] and *SW* [125, 51], respectively, and the similarity measures that are based on them.



(a) *NW* **(b)** *SW*

Figure 5.2: Averaged *11-point interpolated average precision* for *DAY6b* using *NW* and *SW*

Table 5.15 reports a comparison of the *11-point interpolated average precision* [90] for *NEDIT* with the best results achieved by *LIM*, *ML6*, *AFR6*, *PMI6*, *DAY6* and *DAY6b*, over the ten language pairs of our test dataset. The *average, standard deviation, variance* and *median* [108] are also displayed and the best outcome is in bold.

Languages		Cognate	NEDIT	LIM	ML6	AFR6	PMI6	DAY6	DAY6	DAY6b	DAY6b
				Sim$_3$	Sim$_3$	Sim$_5$	Sim$_3$	Sim$_5$	Sim$_5$	Sim$_3$	Sim$_3$
		proportion		NW	NW	NW	SW	NW	SW	NW	SW
English	German	0.590	0.907	0.913	0.902	0.909	0.925	0.932	0.937	0.929	0.934
French	Latin	0.560	0.921	0.924	0.898	0.924	0.925	0.927	0.930	0.921	0.924
English	Latin	0.290	0.703	0.722	0.719	0.776	0.795	0.826	0.833	0.823	0.826
German	Latin	0.290	0.591	0.630	0.610	0.706	0.745	0.741	0.759	0.770	0.772
English	French	0.275	0.659	0.707	0.693	0.768	0.790	0.811	0.815	0.836	0.830
French	German	0.245	0.498	0.580	0.602	0.700	0.757	0.763	0.776	0.796	0.788
Albanian	Latin	0.195	0.561	0.550	0.555	0.584	0.676	0.685	0.683	0.690	0.721
Albanian	French	0.165	0.499	0.441	0.422	0.557	0.621	0.636	0.607	0.607	0.625
Albanian	German	0.125	0.207	0.311	0.326	0.486	0.470	0.508	0.519	0.553	0.552
Albanian	English	0.100	0.289	0.300	0.251	0.280	0.404	0.463	0.487	0.503	0.518
Average		0.284	0.584	0.608	0.598	0.669	0.711	0.729	0.735	0.743	**0.749**
Standard deviation		0.168	0.231	0.219	0.219	0.197	0.173	0.159	0.158	0.149	0.144
Variance		0.028	0.054	0.048	0.048	0.039	0.030	0.025	0.025	0.022	0.021
Median		0.260	0.576	0.605	0.606	0.703	0.751	0.752	0.768	0.783	0.780

Table 5.15: *11-point interpolated average precision* for several models

DAY6 and *DAY6b* produce considerably better results in terms of averaged *11-point interpolated average precision* [90] than *NEDIT*, *LIM*, *ML6*, *AFR6* and *PMI6*. Moreover, the *standard deviation* and the *variance* are smaller, meaning that these models have a high performance also when the languages involved in the test are not closely related. *DAY6b* reaches an accuracy 28% higher than *NEDIT* and significantly outperforms all comparable phonetic and orthographic systems reported in the literature [86, 87, 83, 80], as shown in Section 5.1.9.

5.1.8 Robustness of PAM-like matrices

In order to assess the robustness of the *PAM-like* approach, we have evaluated the influence of the training dataset dimension on the performance of our cognate identification system.

We have extracted from the *Dyen et al.* [42] corpus a training dataset containing all the languages included in the monograph that did not overlap with the test dataset described in Section 5.1.1. In doing so, we have excluded English, German, French and five varieties of Albanian, using a total of 76 Indo-European speech varieties. We have considered only the word pairs reported by *Dyen et al.* [42] as certain cognates with each other, which are coded with *CCN* in the range [2,99]. When more words were provided for the same meaning in the same language, we have considered the first word only, after ensuring that it was always cognate with the group, as explained in Section 6.2. We have reached a total of about 62,000 cognate pairs. We have then globally aligned these word pairs by using the linguistic-inspired substitution matrix, described in Section 4.2. With this 76-language dataset, we have trained two families of *PAM-like* matrices, one based on the Roman alphabet and one on its extension with gap. We have called these learning models *DAY76* and *DAY76b*, respectively. We have then engaged these families of *PAM-like* matrices in the alignment and rating process of the test dataset. We have used standard global and local alignment algorithms [102, 125, 51] and the family of similarity measures, proposed in Section 4.4. We have applied a gap penalty of -1 in the alignment algorithms for *DAY76*, which is based on the Latin alphabet without gap. We have computed the *11-point interpolated average precision* [90] for each of the ten language pairs of our test dataset. We have used each *PAM-like* matrix with each similarity measure, based on both global and local alignment. We have then calculated the *average,* *standard deviation, variance* and *median* [108].

Tables 5.16 and 5.17 report the averaged *11-point interpolated average precision* [90] achieved by the *PAM-like* matrices belonging to the *DAY76* model. The *Needleman-Wunsch* [102, 51] algorithm and the *Smith-Waterman* algorithm [125, 51] have been employed with the family of similarity measures, described in Section 4.4. The best outcome is displayed in bold.

Matrix	NW	Sim$_1$	Sim$_2$	Sim$_3$	Sim$_4$	Sim$_5$	Sim$_6$	Sim$_7$	Sim$_8$	Sim$_9$
PAM0	0.509	0.524	0.524	0.524	0.524	0.524	0.524	0.524	0.524	0.524
PAM1	0.585	0.680	0.675	0.682	0.678	0.684	0.680	0.680	0.679	0.676
PAM2	0.616	0.715	0.713	0.720	0.713	0.719	0.715	0.717	0.715	0.714
PAM3	0.624	0.726	0.720	**0.729**	0.722	0.727	0.727	0.727	0.724	0.721
PAM4	0.615	0.726	0.724	0.728	0.723	0.725	0.724	0.727	0.724	0.726
PAM5	0.606	0.719	0.720	0.718	0.716	0.720	0.718	0.718	0.721	0.721
PAM6	0.595	0.716	0.711	0.717	0.713	0.717	0.714	0.716	0.714	0.713
PAM7	0.583	0.706	0.700	0.708	0.702	0.708	0.703	0.707	0.704	0.703
PAM8	0.571	0.698	0.695	0.700	0.694	0.699	0.697	0.698	0.699	0.697
PAM9	0.561	0.689	0.681	0.693	0.684	0.692	0.691	0.692	0.686	0.683
PAM10	0.552	0.676	0.672	0.678	0.676	0.677	0.676	0.677	0.675	0.672

Table 5.16: Averaged *11-point interpolated average precision* for *DAY76* using *NW*

Matrix	SW	Sim$_1$	Sim$_2$	Sim$_3$	Sim$_4$	Sim$_5$	Sim$_6$	Sim$_7$	Sim$_8$	Sim$_9$
PAM0	0.516	0.585	0.591	0.590	0.591	0.590	0.585	0.590	0.591	0.591
PAM1	0.569	0.682	0.679	0.681	0.682	0.681	0.679	0.680	0.681	0.678
PAM2	0.584	0.720	0.715	0.719	0.716	0.717	0.718	0.721	0.719	0.715
PAM3	0.588	0.735	0.730	0.735	0.733	0.729	0.734	0.734	0.734	0.731
PAM4	0.589	0.736	0.728	0.737	0.732	0.732	0.736	0.737	0.734	0.732
PAM5	0.579	**0.740**	0.731	0.736	0.734	0.734	0.733	0.738	0.736	0.732
PAM6	0.569	0.734	0.728	0.733	0.730	0.732	0.730	0.735	0.731	0.730
PAM7	0.564	0.729	0.725	0.733	0.725	0.731	0.728	0.729	0.729	0.727
PAM8	0.551	0.726	0.719	0.728	0.720	0.725	0.723	0.726	0.725	0.719
PAM9	0.542	0.718	0.708	0.720	0.710	0.716	0.711	0.719	0.717	0.711
PAM10	0.534	0.708	0.702	0.709	0.703	0.704	0.704	0.709	0.706	0.705

Table 5.17: Averaged *11-point interpolated average precision* for *DAY76* using *SW*

Tables 5.18 and 5.19 present the averaged *11-point interpolated average precision* [90] produced by the *PAM-like* matrices belonging to the *DAY76b* model. The *Needleman-Wunsch* [102, 51] algorithm for global alignment and the *Smith-Waterman* algorithm [125, 51] for local alignment have been employed with the family of similarity measures, introduced in Section 4.4. The best results are shown in bold.

Matrix	NW	Sim$_1$	Sim$_2$	Sim$_3$	Sim$_4$	Sim$_5$	Sim$_6$	Sim$_7$	Sim$_8$	Sim$_9$
PAM0	0.509	0.524	0.524	0.524	0.524	0.524	0.524	0.524	0.524	0.524
PAM1	0.525	0.675	0.672	0.673	0.675	0.670	0.676	0.675	0.675	0.672
PAM2	0.583	0.727	0.719	0.730	0.721	0.726	0.728	0.727	0.722	0.718
PAM3	0.554	0.728	0.723	0.733	0.726	0.729	0.731	0.730	0.727	0.724
PAM4	0.546	0.733	0.728	0.741	0.731	0.742	0.734	0.735	0.730	0.727
PAM5	0.529	0.739	0.729	0.740	0.734	**0.743**	0.738	0.740	0.731	0.726
PAM6	0.512	0.735	0.729	0.738	0.731	0.741	0.735	0.736	0.731	0.726
PAM7	0.505	0.731	0.725	0.736	0.727	0.735	0.734	0.733	0.725	0.723
PAM8	0.495	0.731	0.723	0.735	0.727	0.733	0.731	0.731	0.722	0.718
PAM9	0.489	0.724	0.723	0.732	0.726	0.731	0.730	0.727	0.720	0.716
PAM10	0.485	0.725	0.720	0.731	0.725	0.729	0.728	0.726	0.719	0.713

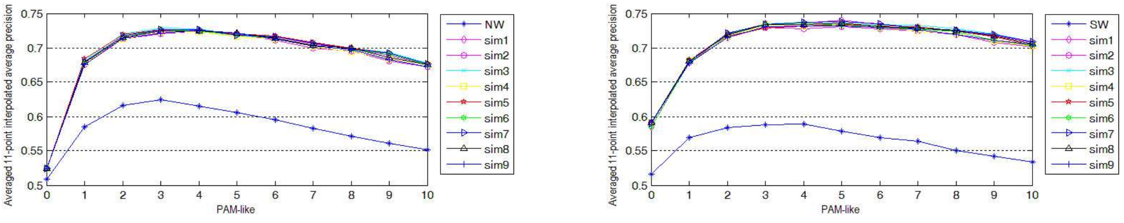
Table 5.18: Averaged *11-point interpolated average precision* for *DAY76b* using *NW*

Matrix	SW	Sim$_1$	Sim$_2$	Sim$_3$	Sim$_4$	Sim$_5$	Sim$_6$	Sim$_7$	Sim$_8$	Sim$_9$
PAM0	0.516	0.585	0.591	0.590	0.591	0.590	0.585	0.590	0.591	0.591
PAM1	0.532	0.683	0.678	0.685	0.681	0.681	0.683	0.683	0.680	0.678
PAM2	0.567	0.723	0.716	0.722	0.721	0.723	0.722	0.722	0.722	0.716
PAM3	0.553	0.735	0.729	0.735	0.733	0.736	0.735	0.736	0.732	0.729
PAM4	0.542	0.737	0.731	0.742	0.734	**0.743**	**0.743**	0.741	0.734	0.733
PAM5	0.529	0.741	0.732	0.742	0.738	0.740	0.740	0.742	0.737	0.733
PAM6	0.512	0.737	0.731	0.740	0.734	0.742	0.738	0.739	0.733	0.728
PAM7	0.503	0.731	0.726	0.738	0.729	0.738	0.738	0.734	0.727	0.723
PAM8	0.494	0.732	0.725	0.738	0.729	0.736	0.733	0.733	0.726	0.721
PAM9	0.488	0.727	0.725	0.734	0.727	0.734	0.731	0.729	0.724	0.720
PAM10	0.484	0.727	0.721	0.731	0.724	0.727	0.730	0.728	0.721	0.716

Table 5.19: Averaged *11-point interpolated average precision* for *DAY76b* using *SW*

The two models *DAY76* and *DAY76b* achieve excellent results. *DAY76b*, which utilises the Latin alphabet extended with gap, performs slightly better than *DAY76* and produces equal top rating results when using either global or local alignment. The similarity measures, proposed in Section 4.4, consistently outperform the basic algorithm on which they are based.

Figure 5.3 shows two graphs of the averaged *11-point interpolated average precision* [90] produced by the *PAM-like* matrices of *DAY76*, when using *NW* [102, 51] and *SW* [125, 51], respectively, and the family of similarity measures.

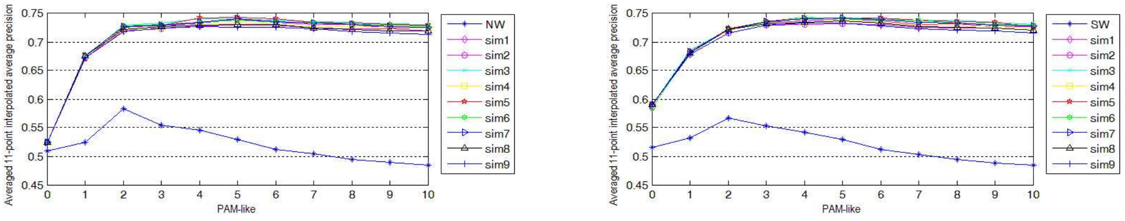


(a) *NW*

(b) *SW*

Figure 5.3: Averaged *11-point interpolated average precision* for *DAY76* using *NW* and *SW*

Figure 5.4 shows in graphical format the averaged *11-point interpolated average precision* [90] produced by the *PAM-like* matrices of *DAY76b*, when using *NW* [102, 51] and *SW* [125, 51], respectively, and the family of similarity measures.



(a) *NW*

(b) *SW*

Figure 5.4: Averaged *11-point interpolated average precision* for *DAY76b* using *NW* and *SW*

Table 5.20 reports a comparison of the best results achieved by *DAY6*, *DAY6b*, *DAY76* and *DAY76b* in terms of *11-point interpolated average precision* [90], over the ten language pairs of our test dataset. The best outcome is in bold and the *average, standard deviation, variance* and *median* [108] are also shown.

Languages		DAY6	DAY6	DAY6b	DAY6b	DAY76	DAY76	DAY76b	DAY76b
		Sim$_5$	Sim$_5$	Sim$_3$	Sim$_3$	Sim$_3$	Sim$_1$	Sim$_5$	Sim$_5$
		NW	SW	NW	SW	NW	SW	NW	SW
English	German	0.932	0.937	0.929	0.934	0.936	0.932	0.933	0.935
French	Latin	0.927	0.930	0.921	0.924	0.928	0.925	0.914	0.918
English	Latin	0.826	0.833	0.823	0.826	0.821	0.845	0.810	0.818
German	Latin	0.741	0.759	0.770	0.772	0.766	0.790	0.777	0.779
English	French	0.811	0.815	0.836	0.830	0.813	0.826	0.823	0.823
French	German	0.763	0.776	0.796	0.788	0.745	0.781	0.763	0.760
Albanian	Latin	0.685	0.683	0.690	0.721	0.676	0.683	0.692	0.698
Albanian	French	0.636	0.607	0.607	0.625	0.642	0.632	0.666	0.663
Albanian	German	0.508	0.519	0.553	0.552	0.498	0.492	0.566	0.554
Albanian	English	0.463	0.487	0.503	0.518	0.465	0.491	0.486	0.485
Average		0.729	0.735	0.743	**0.749**	0.729	0.740	0.743	0.743
Standard deviation		0.159	0.158	0.149	0.144	0.161	0.161	0.143	0.146
Variance		0.025	0.025	0.022	0.021	0.026	0.026	0.020	0.021
Median		0.752	0.768	0.783	0.780	0.756	0.786	0.770	0.770

Table 5.20: *11-point interpolated average precision* for *DAY6*, *DAY6b*, *DAY76* and *DAY76b*

The two models *DAY76* and *DAY76b* trained with the 76-language dataset perform well and produce very similar results when compared with *DAY6* and *DAY6b*. *Sim*$_3$ and *Sim*$_5$ continue to reach the higher accuracy for most of the models considered and *PAM3*, *PAM4* and *PAM5* seem to be able to represent well the divergence of the languages in the test dataset. It is worth noting that the *average, standard deviation, variance* and *median* [108] of the *11-point interpolated average precision* [90] across the models, are remarkably stable. The four models show similar behaviour in relation to the alphabet and the alignment algorithm employed. In fact *DAY6* and *DAY76*, that utilise the Latin alphabet, behave very similarly to each other when using global and local alignment, respectively. The same happens to *DAY6b* and *DAY76b*, that use the Latin alphabet extended with gap.

This is a particularly notable outcome because of the big difference in the training dataset dimension between the two model groups. In fact, *DAY6* and *DAY6b* have been trained with approximately only 650 sensibly aligned cognate pairs, extracted from Italian, Portuguese, Spanish, Dutch, Danish and Swedish. *DAY76* and *DAY76b* have been trained with approximately 62,000 sensibly aligned cognate pairs, extracted from 76 very diverse Indo-European speech varieties that include the 6 languages used to train *DAY6* and *DAY6b*. This corresponds to an increment of the training dataset dimension by a factor of approximately 100, which implies extending the number of Indo-European languages by a factor of approximately 13.

Indeed, this result suggests that when using *PAM-like* matrices with the family of parameterised similarity measures, proposed in Section 4.4, the dimension of the training dataset does not influence the accuracy of our cognate identification system. Interestingly, our learning system needs only a very small amount of training data to reach an outstanding performance. This outcome has been presented in [34].

5.1.9 Comparison

In the task of cognate identification, a phonetic approach is supposed to be more accurate than an orthographic one, because of its insight and understanding of phonetic changes. However, comparative evaluations of some recent orthographic learning methods [86, 87, 83] have shown that they may outperform phonetic systems [76, 80]. This would suggest that phonetic changes can leave enough traces in the word orthography to be successfully utilised by orthographic systems. Our investigation, based on a learning system for measuring string similarities, has confirmed this tendency, producing further improvements in the accuracy of cognate identification.

We have evaluated our models against the most successful phonetic and orthographic comparable studies reported in the literature, i.e. *ALINE* [76] and its variation [80], *PHMM* [87] and *DBN* [83], introduced in Section 2.1. All these methods have utilised the *Kessler* lists [73] as test dataset and the averaged *11-point interpolated average precision* [90] to measure the accuracy of cognate identification. The learning systems have employed the Indo-European corpus by *Dyen et al.* [42], as training dataset. We have intentionally made the same choices in our experimental design in order to build a properly comparable system.

Table 5.21 shows an assessment of all these top phonetic and orthographic approaches, together with our best results achieved by *DAY6b*, described in Section 5.1.7, and *DAY76b*, proposed in Section 5.1.8. *DAY6b* and *DAY76b* both train *PAM-like* matrices based on the Roman alphabet extended with gap. The best outcome is in bold.

Languages		Cognate proportion	NEDIT	ALINE	PHMM	DBN	DAY6b NW	DAY6b SW	DAY76b NW	DAY76b SW
English	German	0.590	0.907	0.912	0.930	0.927	0.929	0.934	0.933	0.935
French	Latin	0.560	0.921	0.862	0.934	0.923	0.921	0.924	0.914	0.918
English	Latin	0.290	0.703	0.732	0.803	0.822	0.823	0.826	0.810	0.818
German	Latin	0.290	0.591	0.705	0.730	0.772	0.770	0.772	0.777	0.779
English	French	0.275	0.659	0.623	0.812	0.802	0.836	0.830	0.823	0.823
French	German	0.245	0.498	0.534	0.734	0.645	0.796	0.788	0.763	0.760
Albanian	Latin	0.195	0.561	0.630	0.680	0.676	0.690	0.721	0.692	0.698
Albanian	French	0.165	0.499	0.610	0.653	0.658	0.607	0.625	0.666	0.663
Albanian	German	0.125	0.207	0.369	0.379	0.420	0.553	0.552	0.566	0.554
Albanian	English	0.100	0.289	0.302	0.382	0.446	0.503	0.518	0.486	0.485
Average		0.284	0.584	0.628	0.704	0.709	0.743	**0.749**	0.743	0.743
Standard deviation		0.168	0.231	0.193	0.194	0.176	0.149	0.144	0.143	0.146
Variance		0.028	0.054	0.037	0.038	0.031	0.022	0.021	0.020	0.021
Median		0.260	0.576	0.627	0.732	0.724	0.783	0.780	0.770	0.770

Table 5.21: *11-point interpolated average precision* for several methods

The results produced by *NEDIT*, the *Levenshtein distance* with unitary costs [84] normalised by the length of the longer string, introduced in Section 5.1.2, are also shown and used as a baseline. The *11-point interpolated average precision* [90] achieved by *ALINE* [76], *PHMM* [87] and *DBN* [83] is reported as in the literature. The variation of *ALINE* [80] is not included, as only the averaged *11-point interpolated average precision* [90], *0.681*, was given in that paper.

PHMM [86, 87] and *DBN* [83] perform better than *ALINE* [76] and its extension [80], reaching very similar averaged *11-point interpolated average precision* [90]. However, *DBN*'s *standard deviation* and *variance* [108] are much lower than those produced by *PHMM*, showing a better data distribution. *PHMM* [87] and *DBN* [83] will be reviewed and discussed in more detail in Chapter 6.

The comparison shows that our learning system based on *PAM-like* matrices, described in Section 4.3.4, consistently outperforms all the other phonetic and orthographic models considered. In fact, *DAY6b* and *DAY76b* have produced an averaged *11-point interpolated average precision* [90] approximately 5% higher than *PHMM* [87] and *DBN* [83], 18% higher than *ALINE* [76] and 28% higher than *NEDIT*. Moreover, not only the average of the *11-point interpolated average precision* [90] is higher, but also the *standard deviation* and *variance* [108] are much lower. This suggests that our learning system is more stable than the compared methods in its performance across various language pairs. This is confirmed by a higher *median* [108], which indicates the central tendency.

It is also interesting to notice that our models accommodate quite well the Albanian language that makes the test dataset challenging. In fact Albanian constitutes its own branch in the Indo-European language family and it is not part of the language branches with which our system has been trained.

It is worth mentioning that in this section we have limited the comparison only to the most successful of our models proposed in Section 4 and tested in Section 5.1. In doing so, we have not reported here the other well behaving methods that have also outperformed previous comparable results. For example, *PMI6* has achieved an accuracy, in terms of averaged *11-point interpolated average precision* [90], of *0.711* with *SW*; *DAY6* has reached an accuracy of *0.729* with *NW* and of *0.735* with *SW*; *DAY76* has attained an accuracy of *0.729* with *NW* and of *0.740* with *SW*.

This suggests that our learning system is generally very successful in cognate identification and outperforms comparable phonetic and orthographic studies previously reported in the literature [76, 86, 87, 83, 80]. These results have been proposed in [33, 34].

5.1.10 Statistical significance of PAM-like matrices

In order to understand if our results represent a statistically significant improvement or have been achieved by chance, we have run some paired two-sample *Student's t-tests* [108]. A *Student's t-test* determines whether two samples having a comparable average are likely to have come from the same population or from two different populations. We have assumed that the two samples are normally distributed, but we have not supposed that the *variances* are equal, because the sample size of the two compared groups is the same. This assures that the *Student's t-test* is highly robust to the presence of unequal *variances* [91]. Each sample has consisted of the ten *11-point interpolated average precision* [90] scores between language pairs produced by one of the systems reported in Table 5.21. We have conducted paired tests, which calculate the difference between arithmetic means of paired samples, because the samples to compare were not independent. For each test, our experimental hypothesis has been that our sample contained higher *11-point interpolated average precision* [90] scores than the sample with which we wanted to compare. As a consequence, the null hypothesis we have tested for rejection has been that our sample did not contain *11-point interpolated precision* [90] scores higher than the sample with which we wanted to compare. Because the null hypothesis states a predicted direction of outcome, we have run one-tailed *t-tests*, meaning that our interest is only in one tail of the *Student*'s distribution.

Table 5.22 shows the *p-values* [108] and the consequent statistical significance of the *t-tests* that we have run to compare the best results obtained by *DAY6b* and *DAY76b* using local alignment, with the other systems reported in Table 5.21.

Student's t-test			
Sample1	**Sample2**	**p-value**	**Statistical significance**
Main comparisons			
DAY6b	DBN	0.030	Good evidence
DAY76b	DBN	0.028	Good evidence
Secondary comparisons			
DAY6b	NEDIT	0.0004	Strong evidence
DAY6b	ALINE	0.001	Strong evidence
DAY6b	PHMM	0.025	Good evidence
DAY76b	NEDIT	0.0004	Strong evidence
DAY76b	ALINE	0.0004	Strong evidence
DAY76b	PHMM	0.029	Good evidence

Table 5.22: Statistical significance of *DAY6b* and *DAY76b* using *SW*

All the *t-tests* have rejected the null hypothesis with strong or good evidence and have confirmed the experimental hypothesis. This validates the statistical significance of our results in the task of cognate identification that outperform those achieved by comparable systems previously reported in the literature [76, 86, 87, 83, 80].

It is worth noting that the statistical significance has remained stable with the enlargement of the training dataset dimension. In fact, we have run a *t-test* between the best results of *DAY6b* and *DAY76b* when using local alignment to check any possible statistical difference between the two. The *p-value* found, which is *0.199*, has given no evidence of any statistical difference between *DAY6b* and *DAY76b* samples. This would suggest that the dimension of the training dataset for the learning system does not influence its statistical significance.

We can therefore state that the *PAM-like* method proposed, significantly outperforms all the comparable phonetic and orthographic systems reported in the literature to date [76, 86, 87, 83, 80]. These results have been described in [34].

5.1.11 Conclusion

The learning system proposed in Chapter 4 has achieved very good results in cognate identification, when training *PAM-like* substitution matrices. The best outcome has been produced when the learnt parameters were associated with the characters of the Roman alphabet extended with gap, suggesting that the system is able to learn appropriate gap penalties.

The methodology has proved to reach outstanding results with a 6-language dataset of sensibly aligned cognate words and been able to maintain a remarkably stable performance with a 76-language training dataset. In fact, it has shown no sensitivity to the training dataset dimension, when it was increased by a factor of approximately 100.

All the models based on *PAM-like* matrices have outperformed consistently comparable phonetic and orthographic systems reported in the literature [76, 86, 87, 83, 80] and the best results have shown to be statistically significant. This is a particularly interesting outcome because, not only does it advance the state of the art in cognate identification, but also reinforces the hypothesis that orthographic learning models can outperform systems specifically designed for the task of phonetic alignment.

5.2 Phylogenetic inference

We have applied the learning system for measuring word similarity, presented in Chapter 4, to the task of linguistic phylogenetic inference.

Firstly, we have prepared a training dataset and a test dataset for the Indo-European language family with no intersection in the meanings included. We have then sensibly aligned the cognate pairs of the training dataset using the *Needleman-Wunsch* algorithm [102, 51] for global alignment and the linguistic-inspired substitution matrix, presented in Section 4.2. From this training dataset, we have learnt a family of substitution matrices using the *PAM-like* approach based on the Roman alphabet extended with gap, that has proved in Section 5.1 to be more successful in cognate identification than the basic Latin alphabet. We have then utilised these *PAM-like* substitution matrices for measuring the lexical similarity within the word pairs of the test dataset. We have employed the family of parameterised string similarity measures proposed in Section 4.4 based on the *Smith-Waterman* algorithm [125, 51] for local alignment, that has shown in Sections 5.1.7 and 5.1.8 to perform slightly better than global alignment. From these word pair similarities, we have calculated language pair similarities and then we have transformed them into language pair distances. Finally, we have utilised these distances to estimate phylogenetic trees of languages using standard *distance-based* methods.

Our results are compatible with the Indo-European benchmark tree, have reproduced correctly all the established major language groups and subgroups present in the test dataset, and have also included some of the supported higher-level structures. This would suggest that our methodology successfully satisfies the "*Compatible resolution*" and the "*No missing subgroups*" criteria [106], which are utilised in linguistic evaluation of phylogenetic estimation.

5.2.1 Datasets

We have employed the Comparative Indo-European corpus by *Dyen et al.* [42] described in Section 2.3, considering the 84 languages documented in the monograph. In the absence of two large homogeneous linguistic datasets to be used as training and test dataset without intersection, we have split the *Dyen et al.* corpus [42] into two groups of meanings, identified by odd and even ordinal numbers. Firstly, we have created a training dataset from the odd meanings and prepared a test dataset from the even meanings, called *test-even*. Secondly, we have done the opposite, using the even meanings as training dataset, and the odd meanings as test dataset, named *test-odd*.

For the training datasets, we have used only the word pairs reported by *Dyen et al.* [42] as certain cognates with each other, which are classified with a *Cognate Class Number* (*CCN*) in the range [2,99]. If more words were provided for the same meaning in the same language, we have considered the first word only, after ensuring that it was always cognate with the group, as explained in Sections 5.1.1 and 6.2. We have then aligned the two training groups of word pairs using the linguistic-inspired substitution matrix proposed in Section 4.2, obtaining two separate training datasets, called respectively *training-odd* and *training-even*.

For the test datasets, we have considered all the word pairs reported by *Dyen et al.* [42] as certain or uncertain cognates, but we have excluded those words classified with $CCN = 0$ as not acceptable, or with $CCN = 1$ as not cognate with any other. This should ensure that our study does not include borrowings, which we wanted to discard from our analysis, as far as the cognateness judgements of *Dyen et al.* [42] are correct. For example, the English word *flower* is classified with $CCN = 1$, since it is considered a loan from Old French. We have also corrected some orthographic errors.

Because there is no clarity yet about a feasible outgroup for the Indo-European family [49], in order to root our phylogenetic tree, we have made a difficult decision considering several options. By using Hittite or Tocharian we would have inserted a bias in the results given that the root of the tree is controversial for the Indo-European family. By choosing any other non Indo-European language, in principle we would have made a questionable choice, because there is no consensus about any phylogenetic connection between Indo-European languages and any other language. We have finally decided to include the Turkish language as an outgroup using the *Swadesh list* [132] provided by *Kessler* [73], excluding the words reported as loans. In fact, even if Turkish belongs to the Altaic language family [85], which is not related to the Indo-European language family, the *Kessler lists* [73] show a weak connection between them, that motivated our choice. For example, the Turkish word *baba* is reported cognate with the Albanian *babë*, meaning father, which, in the *Dyen et al.* [42] corpus, is judged cognate with words belonging to several of the other Indo-European language branches (e.g. Romance, Iranian and Indo-Aryan).

We have added to the Turkish list provided by *Kessler* [73] the 9 words in which this list differs from the 200-word *Swadesh list* by *Dyen et al.* [42], checking multiple sources to ensure reliability. We have extended *test-odd* and *test-even* respectively with the odd and even meanings from this Turkish list, reaching a total of 85 languages. Having two training datasets and two test datasets has avoided any data overlap, thereby ensuring that independent analyses have been conducted and their results subsequently averaged, as explained in Section 5.2.2.

We would have liked to have included in our study also Hittite, Tocharian A and Tocharian B provided by *Gray and Atkinson* [52], but examining the data we have found them inappropriate for our analysis. On several occasions, the same meaning for the same language (i.e. Hittite, Tocharian A, Tocharian B) has been classified more than once with different *CCN*, which is not the case for the rest of the original dataset by *Dyen et al.* [42]. This would have biased the learning procedure towards Hittite, Tocharian A and Tocharian B, which would have given more contributions to the *PAM-like* matrices than the other languages.

5.2.2 Experimental design

We have designed our experiments with the aim of estimating phylogenetic trees that may reflect lexical similarity between languages. *PAM-like* substitution matrices may be seen as an indicator of the relative evolutionary interval since the languages diverged. Given that languages evolve at changing rates, there is no simple connection between evolutionary *PAM-like* distance and evolutionary time. However, for an analysis of a specific language family across multiple speech varieties, the corresponding *PAM-like* matrices provide a relative evolutionary distance between the languages and allows accurate phylogenetic inference [46].

We have employed the two training datasets, *training-odd* and *training-even* described in Section 5.2.1, to learn two families of *PAM-like* matrices based on the Roman alphabet extended with gap. In fact, it has been proved in Sections 5.1.7 and 5.1.8 that learning gap penalties increases the effectiveness of the system. We have called these two matrix families *DAY84b-odd* and *DAY84b-even*, respectively.

We have tested the performance of the *DAY84b-odd* and *DAY84b-even* families in the task of cognate identification on the English, German, Latin, French and Albanian lists provided by *Kessler* [73], to choose the matrix and similarity measure for the estimation of phylogenies. The *Kessler lists* do not cover all the Indo-European branches that are present in the *test-even* and *test-odd* datasets. As a consequence, we could not have been sure that the *PAM-like* matrix and the similarity measure that achieved the better result were also adequate for the other branches of the Indo-European family. For this reason, we have chosen a set of *PAM-like* matrices and similarity measures that have shown a very good performance.

PAM3, *PAM4*, *PAM5* and *PAM6*, from both the families *DAY84b-odd* and *DAY84b-even*, have achieved a very high accuracy when used with Sim_1, Sim_3, Sim_5 and Sim_6 introduced in Section 4.4, based on the *Smith-Waterman* algorithm [125, 51] for local alignment. We have used these *PAM-like* matrices and these similarity measures based on *SW* to calculate the language similarity between each of the 85 speech varieties in *test-even* and *test-odd*, respectively.

The similarity between two languages has been defined as the average similarity between the 200 word pairs belonging to the language pair and having the same meaning. We have not considered those word pairs having one word missing or classified as unacceptable or as a borrowing. We have supported polymorphism and, if one or both languages presented more than one word for a meaning, we have considered the maximum similarity between the different pairs in the average calculation.

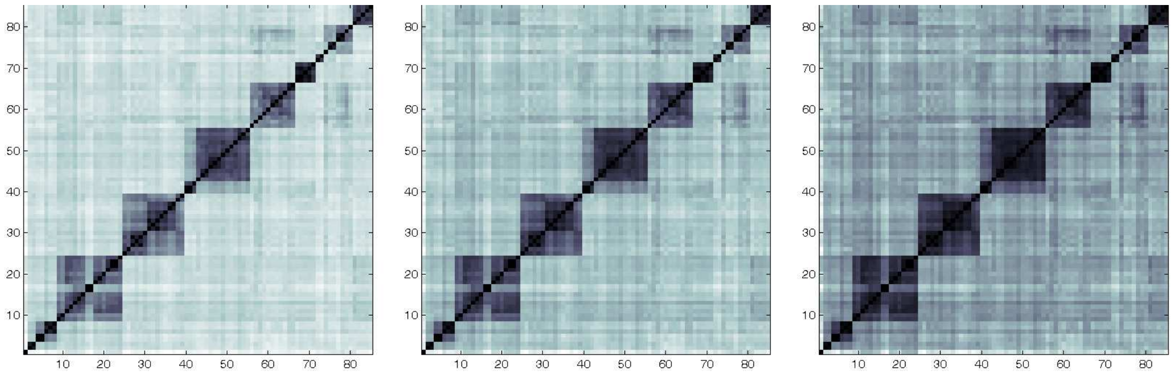
In this way, we have obtained two 85-by-85 similarity matrices and we have calculated their average scores, reaching a single 85-by-85 similarity matrix, for each of the four *PAM-like* matrix pairs (*odd, even*) and for each of the four similarity measures employed, for a total of 16 matrices.

Finally, we have transformed these similarity matrices into distance matrices in three different ways, as described in Section 4.6. We have calculated the weighted average of each group of distance matrices to reach a consensus [92, 46] and we have called the three resulting distance matrices $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$. We have then applied to them the *UPGMA* [126] and *Neighbor-Joining* [119, 130] algorithms to estimate phylogenies.

5.2.3 Results

In order to investigate the structure of these three pairwise distance matrices obtained from similarity matrices, as described in Section 5.2.2, and to picture the information stored in them as images, we have scaled the image data to the full range of a chosen colormap [93].

Figure 5.5 shows the graphic representation of the 85-by-85 pairwise distance matrices $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$, with the outgroup occupying first position in the Cartesian planes.



(a) $\mathcal{D}_1$ (b) $\mathcal{D}_2$ (c) $\mathcal{D}_3$

Figure 5.5: Graphic representation of the distance matrices $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$

It is worth noting that this visual representation highlights clearly the subsets of languages that are more closely related to each other, represented by the darker tones in the central clusters.

All three matrices display the major Indo-European branches, with the addition of the outgroup in first position. They follow the order of the *Dyen et al.* dataset [42] classification that, from bottom-left to top-right, shows Celtic, Italic, Germanic, Balto-Slavic, Indo-Aryan, Greek, Armenian, Iranian and Albanian. The first matrix $\mathcal{D}_1$ presents a clearer and neater distinction between the central clusters and the rest of the data.

5.2.3.1 UPGMA

We have applied the *distance-based* method *UPGMA* [126] to these three distance matrices $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$, as defined in Section 5.2.2, to study correspondences and differences in the estimated phylogenetic trees.

Interestingly, the topologies of the consensus trees produced by using *UPGMA* with $\mathcal{D}_1$ and $\mathcal{D}_2$, have shown an identical canonical form. In addition, the canonical form of the consensus tree estimated by using $\mathcal{D}_3$ has given a variation only within the Indo-Aryan subgroup.

Applied to $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$, the algorithm has produced three trees rooted on the Turkish language, that is the outgroup we have added to the *Dyen at al.* corpus [42]. The confidence of the three consensus trees has been 100% for 77% of the branches and the uncertainty has derived only from the internal Albanian and Indo-Aryan subgroups. Due to the definitions of the three distances, the three trees have presented different, but proportional, branch lengths. This diversity has not influenced the grouping between languages, but has reflected how their relatedness has been calculated.

The trees estimated are compatible with the Indo-European benchmark tree [106] and have reproduced all the established major groups and subgroups present in the dataset. The position of the French Creole speech varieties, which are not even considered as Indo-European languages [85], is justified by the nature of creolisation, which would require network models of evolution [106]. The tree topologies have also shown some of the higher-level supported structures, such as Balto-Slavic grouping with Indo-Iranian and Celtic departing early. Italic has grouped with Celtic, but after forming a clade with Albanian.

Figures 5.6, 5.7 and 5.8 display the topologies of the consensus trees reached using *UPGMA* [126] with the distance matrices $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$, respectively.

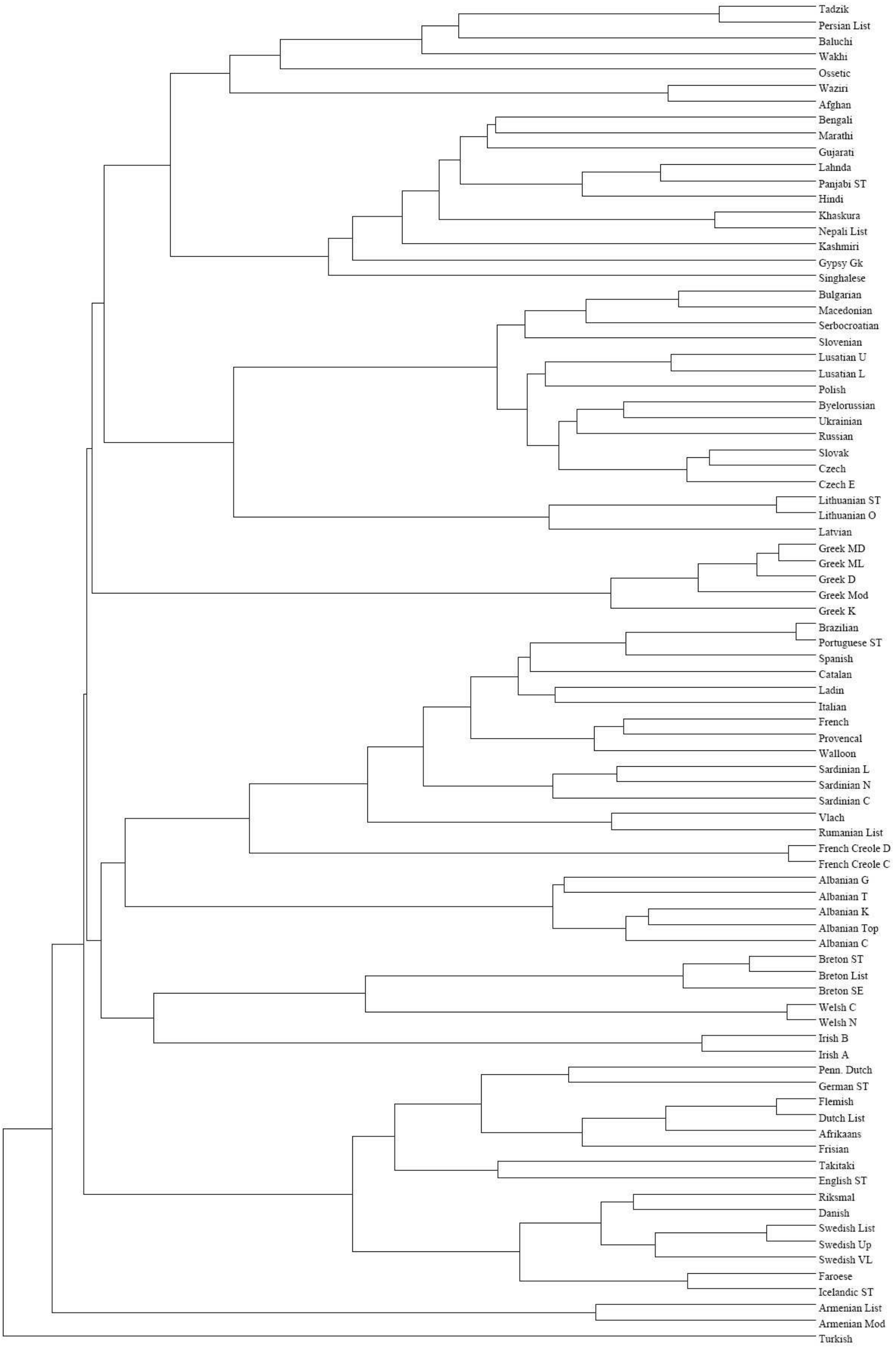


Figure 5.6: Indo-European phylogenetic tree produced using *UPGMA* with $\mathcal{D}_1$

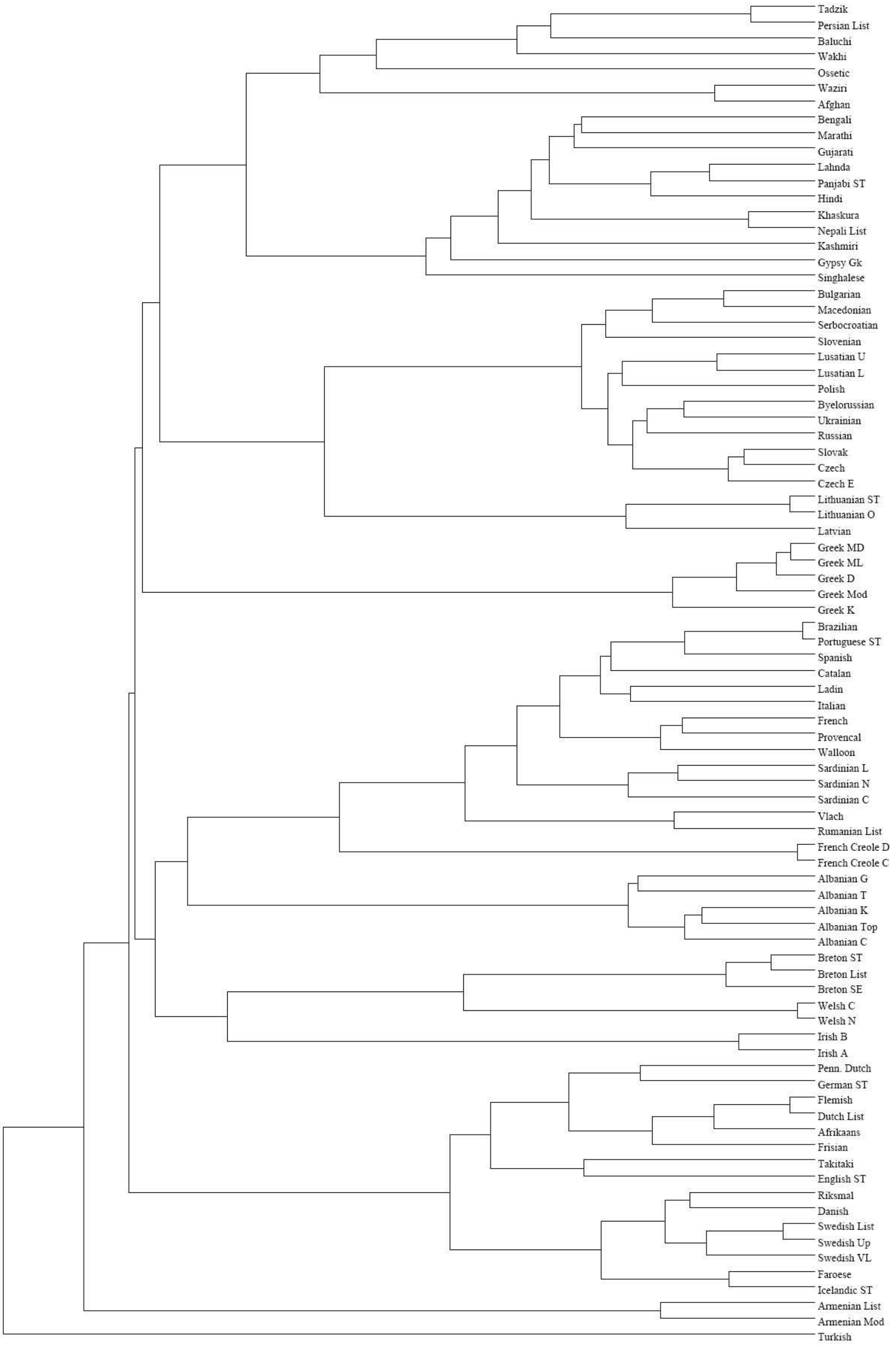


Figure 5.7: Indo-European phylogenetic tree produced using *UPGMA* with $\mathcal{D}_2$

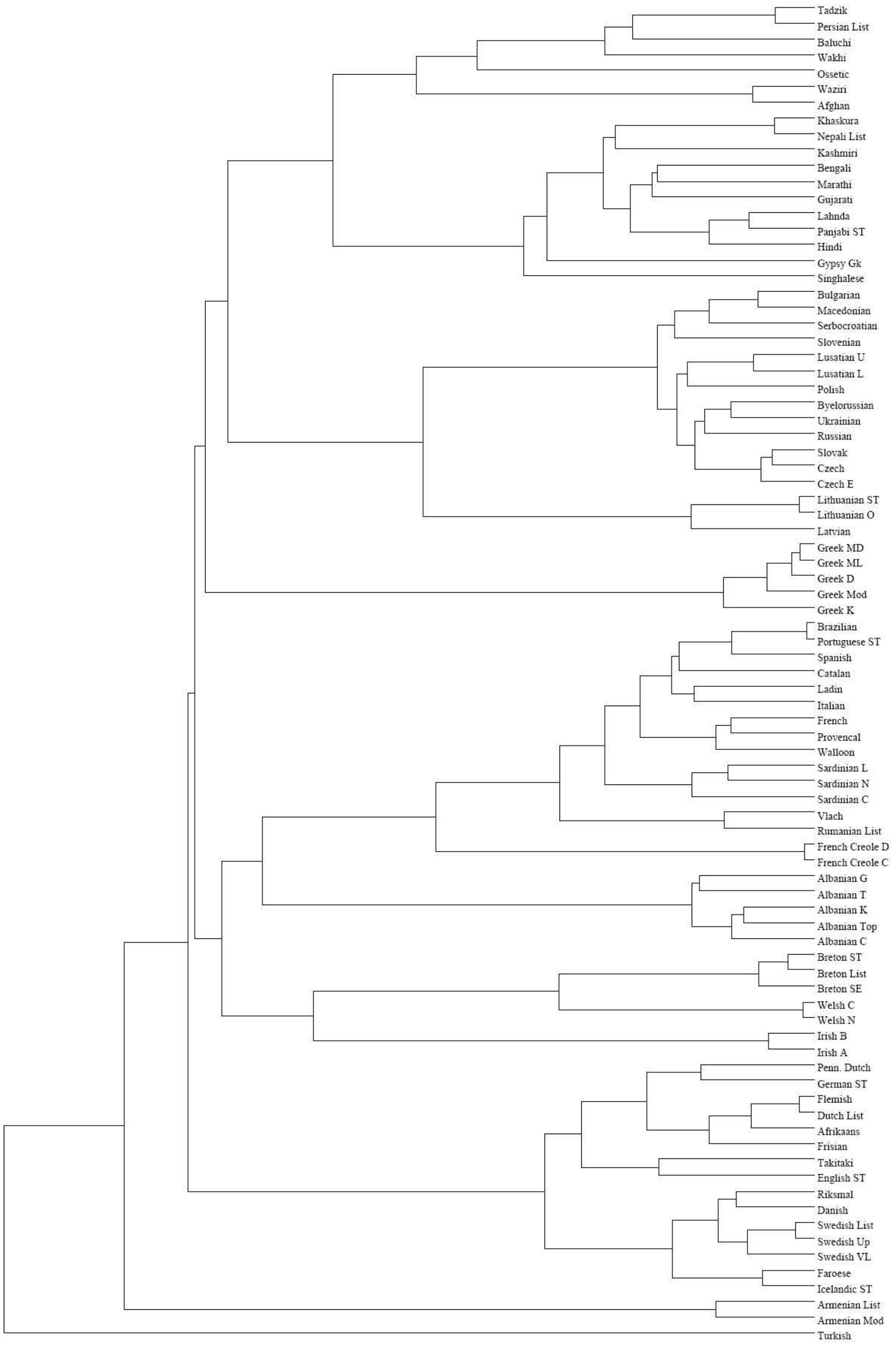


Figure 5.8: Indo-European phylogenetic tree produced using *UPGMA* with $\mathcal{D}_3$

UPGMA [126] has worked extremely well here because the *PAM-like* matrices utilised to calculate the language similarities assume a constant rate of evolution. This is the prerequisite for *UPGMA* to infer phylogenies accurately and in this case it is also a reasonable assumption, because the languages considered belong to the same family and are closely related [46].

The usage of different distance definitions has not altered substantially the canonical form of the tree topologies, that have shown to be fairly consistent with each other.

5.2.3.2 Neighbor-Joining

We have applied the *distance-based* method *Neighbor-Joining* (*NJ*) [119, 130] to the three distance matrices $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$, defined in Section 5.2.2, to analyse correspondences and differences in the inferred phylogenetic trees.

The topologies of the consensus trees reached by the distance matrices have shown different canonical forms. The three estimated trees have reproduced all the established major Indo-European groups present in the dataset, but with some important differences in the subgroups.

The tree estimated using $\mathcal{D}_1$ is compatible with the benchmark tree [106]. The position of the French Creole speech varieties is not precise, as in the case of *UPGMA* [126]. However, these languages are not even classified as Indo-European because of their creolisation [85], whose study would involve more elaborate models of evolution [106]. The trees built using $\mathcal{D}_2$ and $\mathcal{D}_3$ have revealed some subgrouping problems. For example, they have classified accurately the French Creole speech varieties, joining them to the Gallo-Romance branch, but have failed in grouping correctly the East Slavic branch. For this reason, they are not reported.

Figure 5.9 presents the unrooted tree calculated by using *Neighbor-Joining* [119, 130] with the distance matrix $\mathcal{D}_1$.

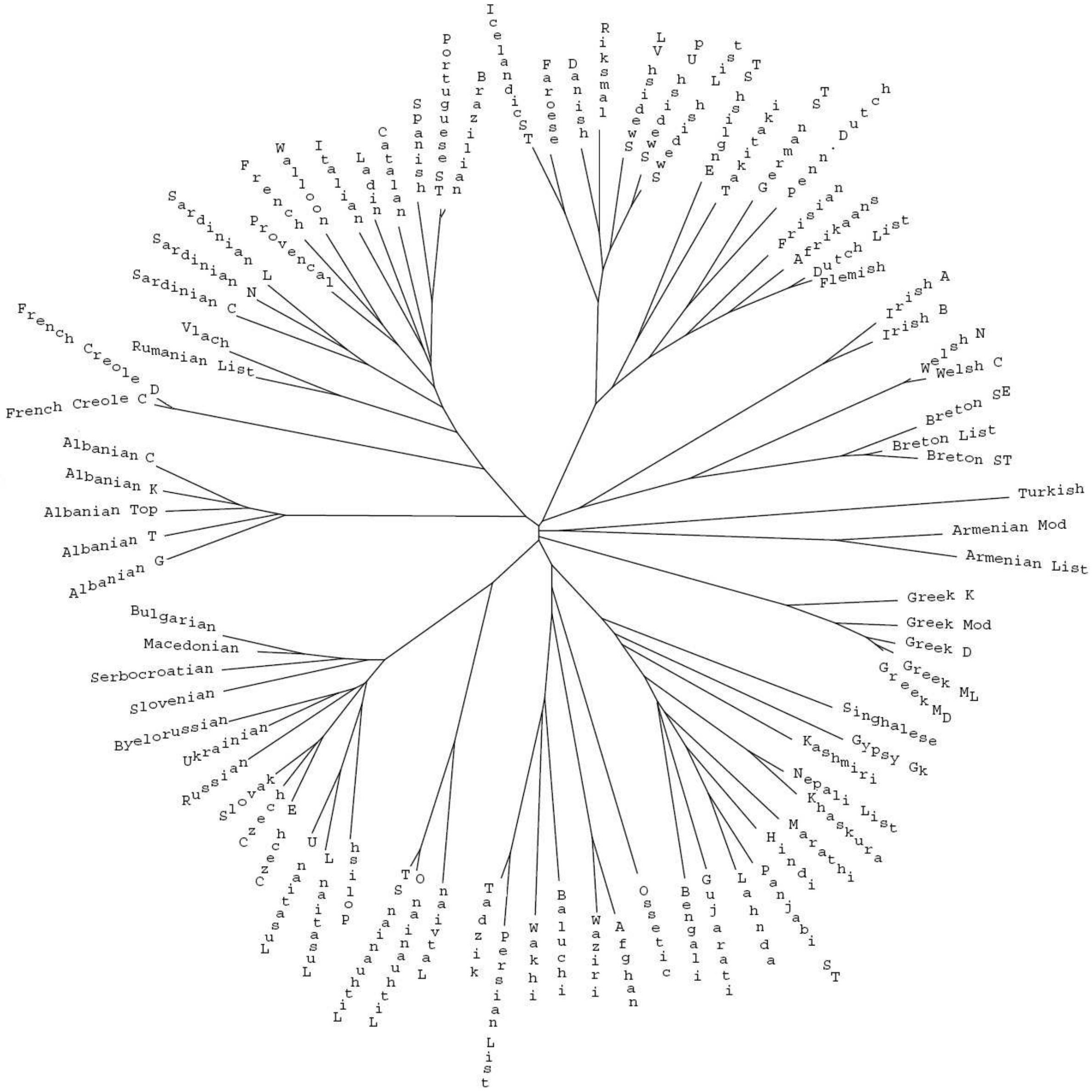


Figure 5.9: Indo-European unrooted phylogeny produced using *NJ* with $\mathcal{D}_1$

The confidence of this consensus tree has been 100% for 55% of the branches and the uncertainty has spread across the tree with the exclusion of the Armanian, Greek, Italic and Baltic groups.

Neighbor-Joining [119, 130], when applied to $\mathcal{D}_1$, has estimated a phylogenetic tree compatible with the Indo-European benchmark tree and has correctly reproduced all the established major language groups and subgroups present in the test dataset. *NJ* has not been successful when applied to $\mathscr{D}_2$ and $\mathscr{D}_3$, which have been produced by transforming similarity matrices into distance matrices using the definitions $\mathcal{D}_2$ and $\mathcal{D}_3$, introduced in Section 4.6. This would suggest that these transformations may add noise to the data to be interpreted by *NJ*, when the similarities are produced using *PAM-like* substitution matrices. Deriving distances from similarities using $\mathcal{D}_1$, as defined in Section 4.6, seems to be the more sensible choice.

5.2.4 Discussion

Serva and Petroni [123], *Petroni and Serva* [110], *Blanchard et al.* [13], *Bakker et al.* [8] and *Downey et al.* [40] used *distance-based* methods to infer phylogenies, as reported in Section 2.2. In all these cases, the language distance was calculated by averaging the distance of word pairs having the same meaning in compared languages. In order to compute word distance, the first four scholar groups utilised the *Levenshtein distance* [84], choosing different normalisation. The fifth group employed *ALINE* [76], normalised as well. *Serva and Petroni* [123], *Petroni and Serva* [110] and *Blanchard et al.* [13] considered only 50 languages from the *Dyen at al.* dataset [42], reducing enormously the complexity of the phylogeny. *Bakker et al.* [8] developed their own dataset and *Downey et al.* [40] applied their method to the Sumbanese language family. Because of these differences, a specific comparison of our results with theirs is not possible. However, it has been shown in [34] that our cognate identification system produces an average accuracy approximately 28% higher than the *Levenshtein distance* [84] normalised by the length of the longer word, and 18% higher than *ALINE* [76], as reported in the literature. This would suggest that our methodology may infer phylogenies more accurately than the other methods reported.

5.2.5 Conclusion

We have applied the string similarity measuring system, trained with *PAM-like* matrices and proposed in chapter 4, to the task of phylogenetic inference in order to test its effectiveness against recognised aspects of the Indo-European language family.

Our results, using the *UPGMA* [126] and *Neighbor-Joining* [119, 130] algorithms, have reproduced correctly all the established major language groups and subgroups present in the dataset and have shown to be compatible with the Indo-European benchmark tree. In doing so, our outcome has successfully met both the required linguistic evaluation criteria for phylogenetic estimation, i.e. the "*Compatible resolution*" and the "*No missing subgroups*".

UPGMA has estimated phylogenetic trees that also include some of the supported higher-level structures and has performed particularly well. This is because it shares with the *PAM-like* method the assumption of a constant rate of evolution, which is a reasonable statement for our investigation, where the languages considered are closely related, belonging to the same family. These results have been presented in [35].

Chapter 6

Related work

In Section 5.1 we have presented the results achieved by our string similarity measuring system in the task of cognate identification and we have compared them with those produced by other successful models reported in the literature. In this chapter, we review and discuss some of these methods, which share with our proposal an orthographic learning approach. As explained in Section 5.1.9, whenever possible, we have intentionally used in our experiments the same training dataset, test dataset and evaluation methodology utilised by these previous investigations, to make our system properly comparable.

6.1 Review

Mackay [86] followed the orthographic approach and developed a suite of *Pair Hidden Markov Models* (*PHMMs*) to measure word similarity in the task of cognate identification. His system was based on a model originally presented by *Durbin et al.* [41] for biological sequence analysis. *PHMMs* are particularly suitable for pairwise alignment, because they allow the examination of a string pair as a single entity, instead of two separate streams of characters, producing an alignment. The training procedure, performed by the *Baum-Welch* algorithm [10], had to determine three sets of parameters: a 26-by-26 symmetric matrix representing the substitution probabilities for each character of the Roman alphabet; the insertion and deletion probabilities for each character of the Roman alphabet; the transition probabilities between the model states corresponding to the edit operations of insertion, deletion and substitution. The training dataset consisted of about 120,000 word pairs extracted from the Comparative Indo-European corpus by *Dyen et al.* [42], described in Section 2.3. The author considered the 95 speech varieties present in the digital file and added to the training data the reverse of each word pair, to avoid possible bias due to the ordering of the words. At this point, to reduce the large dimension of the training dataset, he discarded all the word pairs containing at least one word less then 4 characters long. A development dataset, consisting of two language pairs (Italian and Serbo-Croatian as an example of distant relatedness, Polish and Russian as an example of close relatedness) was used to determine several parameters of the model, including the transition probabilities. The test dataset was extracted from the 200-word *Swadesh lists* prepared by *Kessler* [72] for Albanian, English, French, German and Latin, and assembled by pairing the words having the same meaning in these 5 languages, for a total of 10 language pairs. The suite of *PHMMs* corresponded to several alignment algorithms utilised to calculate word pair similarity, including the *Viterbi* algorithm [41], the *forward* algorithm [41], a *log-odds* version of the *Viterbi* algorithm [41] and a variation of it [86], which employed a forward approach. The model that achieved the higher averaged *11-point interpolated average precision* [90] on the task of cognate identification utilised the *log-odds* version of the *Viterbi* algorithm [41], with uniform gap and transition probabilities.

Mackay and Kondrak [87] compared four of the *PHMMs* proposed by *Mackay* [86] with other methods in the task of cognate identification.

They employed the same test dataset that *Mackay* used, which is composed of the Albanian, English, French, German and Latin lists, provided by *Kessler* [73]. The four *PHMMs* corresponded to the four scoring algorithms employed to compute similarity scores over word pairs, mentioned previously. The authors tested the *PHMMs* against the *Levenshtein distance with Learned Weights* (*LLW*) method, formerly proposed by *Mann and Yarowsky* [88] in the task of lexicon translation. *LLW* learnt the costs for edit operations from the same orthographic training dataset using a stochastic transducer. The authors also compared the results achieved by the *PHMMs* with those reached by *ALINE* [76], introduced in Section 2.1.2, as the *Kessler lists* [73] provide word phonetic transcriptions. They used as a baseline the *Longest Common Subsequence Ratio* (*LCSR*) [96], described in Section 2.1.1. The authors showed that all the four *PHMMs* outperformed *LCSR*, *LLW* and *ALINE*, in terms of averaged *11-point interpolated average precision* [90] in the task of cognate identification. The one that performed better and showed a significant improvement compared with the others, as mentioned before, employed a *log-odds* variation of the *Viterbi* algorithm [41] with uniform gap and transition probabilities. In this thesis, we have referred to it with *PHMM* only.

The *Mackay*'s approach was employed by *Wieling et al.* [140] and by *Wieling et al.* [141] in the field of dialectology. It was also utilised by *Nabende* [99] in the task of transliteration.

Kondrak and Sherif [83], working on orthographic data, developed four different models of a *Dynamic Bayesian Network* (*DBN*) for the task of cognate identification. They based their system on a method previously proposed by *Filali and Bilmes* [48] in the field of pronunciation classification. They used the *Graphical Modelling ToolKit* (*GMTK*) [12] for the implementation. The four *DBN* models were a *Memoryless and Context-Independent model*, a *Memory model*, a *Context-Dependent model* and a *Length model* [48]. The training dataset consisted of about 180,000 word pairs extracted from the Comparative Indo-European corpus by *Dyen et al.* [42]. They used each word pair twice, inverting the source-target direction, to enforce the symmetry of the scoring. In order to determine several parameters of their system, the authors built up a development dataset composed of three language pairs: Italian-Croatian, Spanish-Romanian and Polish-Russian representing respectively distant, medium and close relatedness. *Kondrak and Sherif* used the same test dataset that *Mackay* [86] and *Mackay and Kondrak* [87] utilised, which is extracted from the *Kessler lists* [73]. They tested their *DBNs* in the task of cognate identification and compared them with other phonetic and orthographic systems, including *ALINE* [76], *LLW* [88], and *PHMM* [86, 87]. *NEDIT*, introduced in Section 5.1.2, was used as a baseline. Only the *Context-Dependent model* achieved very good results and outperformed in terms of averaged *11-point interpolated average precision* [90] the other systems including *PHMM*, but not significantly. In this thesis, we have called it *DBN* only.

6.2 Discussion

The results of the studies reported in the previous section suggest that orthographic learning models can outperform static systems specifically designed for the task of phonetic alignment and cognate identification, like *ALINE* [76] and its variation [80], if enough training data were available. Nevertheless, *PHMM* [86, 87] and *DBN* [83] share the same philosophy, as a *Hidden Markov Model* may be considered the simplest type of *Dynamic Bayesian Network* [98]. They are both very powerful and effective statistical models, used especially in pattern recognition and bioinformatics, but whose structure design is more of an art [41]. We have identified the following weaknesses that both models appear to present.

PHMM [86, 87] and *DBN* [83] both need a large training dataset, which has to be processed twice for symmetry, thus creating a time-consuming learning process. This issue could be related to the quality of the data that have been collected, which maybe resulted in being only partially meaningful.

It is well known that the Comparative Indo-European corpus by *Dyen et al.* [42] utilises a peculiar coding. The data are grouped by meaning and cognateness, reported as certain or doubtful, and each group is identified by a *Cognate Class Number* (*CCN*). An explanation of this classification has been reported in Section 5.1.1. Learning from words that are classified with $CCN = 0$, $CCN = 1$, *CCN* in the range [100,199] and *CCN* in the range [400,499], is likely to add noise to the system, because the words are not cognate with each other or their cognateness is doubtful.

Moreover, there are some potential problems for the other categories. Indeed, the data are grouped in cognate sets by the highest degree of cognateness, which also determines how these sets are related to each other. This signifies that, if a language presents more than one word for a meaning, these words are not necessarily cognate with each other, and it is not indicated which word or words are actually cognate with the rest of the group and which are not.

Table 6.1 shows a simplified example of this coding, where one of the cognate groups for the meaning *to dig*, identified by $CCN = 3$, contains 13 words and reports 2 words for Catalan, 2 for Italian and 2 for Provençal.

Language	Word1	Word2
Spanish	cavar	
Catalan	cavar	penetrar
Italian	vangare	scavare
Sardinian C	skavai	
Ladin	chaver	
Provençal	cava	fura
Brazilian	cavar	
Portuguese ST	cavar	
Sardinian N	iskavare	
Sardinian L	iscavare	

Table 6.1: Example of cognate set for the meaning *to dig*

The words Catalan *penetrar*, Italian *vangare* and Provençal *fura* are not actually part of the cognate group. By using all the 13 words reported in the set, as the compared methods did, the word pairs produced for that group would be $\binom{13}{2} = 78$, but only $\binom{10}{2} = 45$ would be correct. As a consequence, the system would align and learn parameters from 33 wrong word pairs, which represent more than 40% of the word pairs in this group.

It should be clearer now why learning processes, even if guided by very powerful models like *PHMM* [86, 87] and *DBN* [83], cannot reach the highest accuracy, if the data are particularly untidy. Indeed the systems learnt, together with correct information, a high percentage of noise as well. Probably it is for this reason that *Mackay* [86], *Mackay and Kondrak* [87] and *Kondrak and Sherif* [83] had to rely on a very large number of pairs to try to neutralise the errors contained in the data.

To avoid these problems in the learning process, we have included in the training dataset only those groups containing words judged certain cognate with each other, which are classified in the corpus with *CCN* in the range [2,99]. Moreover, we have ensured that the first word for each language was always cognate with the group, as it is shown in Table 6.2.

Language	Word1	Word2
Spanish	cavar	
Catalan	cavar	penetrar
Italian	scavare	vangare
Sardinian C	skavai	
Ladin	chaver	
Provençal	cava	fura
Brazilian	cavar	
Portuguese ST	cavar	
Sardinian N	iskavare	
Sardinian L	iscavare	

Table 6.2: Example of rearranged cognate set for the meaning *to dig*

This may be achieved by an automatic procedure that puts in first position the word presenting the smaller averaged *edit distance* [84] with the other members of the group. We have then created our training dataset of word pairs considering only the first word listed for each language: for example, from the group shown in Table 6.2 we have produced $\binom{10}{2} = 45$ correct cognate pairs. This suggests that the word pairs we used for the training process were really cognate pairs, as far as the cognateness judgements of *Dyen et al.* [42] are correct.

Another issue we have found in the proposals reporting *PHMM* [86, 87] and *DBN* [83] may be related to the ranking order. In these studies, it is not specified if they imposed an alphabetic order on the word pairs that received the same score. If they ordered the word pairs only by rates, this would have created random results in the case of word pairs presenting the same rate. On the other hand, ordering the word pairs by rates and alphabetically, would have created, correctly, a reproducible semi-random order. This may explain why the results of *ALINE* [76] are all slightly different in [87] and [83] and why the *PHMMs* results are not the same in [86] and [87].

Chapter 7

Conclusion

The main objective of this thesis has been the investigation of data driven models for the study of language evolution. We have explored the most important and promising tasks in computational historical linguistics, namely cognate identification and phylogenetic inference.

We have stated the cognate identification problem as an approximate string matching problem and, in order to solve it, we have chosen the similarity approach, which is the standard in bioinformatics and in many fields of natural language processing. This approach aims to find the maximum similarity between two strings, which may be achieved by discovering their optimal global or local alignments, whose detection is crucially influenced by the scoring scheme employed.

We have designed a new orthographic learning system for measuring string similarity [33], which consists of three main components, each including an original aspect. The first component allows a meaningful pairwise global alignment of the training dataset, aided by a novel linguistic-inspired substitution matrix. This matrix, based on the Roman alphabet, tries to encode well-known systematic sound changes left in the written orthography. The second component generates scoring matrices using several techniques, including *Maximum Likelihood*, *Absolute*

Frequency Ratio, *Pointwise Mutual Information* and *PAM-like* [33]. The latter has been inspired by the *Point Accepted Mutation* (*PAM*) method, widely used for amino acid sequence analysis [30, 31, 32]. The third component performs pairwise alignments in order to measure the similarity between words and benefits from the generated substitution matrices and from a novel family of parameterised string similarity measures. Each of these measures derives from the normalisation of a generic scoring algorithm, achieved by using the similarity of each string with itself in different ways, in the aim of minimising the bias due to different string length.

We have applied this learning system for measuring string similarity to the tasks of cognate identification, using standard Indo-European linguistic datasets. Whenever possible, we have intentionally used the same training dataset, test dataset and evaluation methodology utilised by previous successful investigations, with which we wanted to make our method properly comparable. Our system, trained with *PAM-like* matrices [33], has achieved an excellent accuracy in cognate identification. It has shown its superior performance and higher consistency across different language pairs, when evaluated against the best comparable phonetic and orthographic studies previously reported in the literature [76, 86, 87, 83, 80]. We have assessed the robustness of our learning system [34] by increasing the training dataset dimension by a factor of approximately one hundred. The outcome has been impressively stable, showing no relevant difference in the performance. The results have also proved to be statistically significant [34], when compared with earlier proposals and with each other.

We have also employed our learning system in the task of phylogenetic inference of the Indo-European language family, whose higher structure remains very controversial. In order to estimate phylogenies, we have transformed language similarities into language distances and we have experimented with *distance-based* methods. Our learning system has been successful in detecting accurate language similarity [35]. Indeed, it has inferred phylogenies that are compatible with the Indo-European benchmark tree and has reproduced all the established major groups and subgroups present in the dataset. It has also included some of the supported higher-level structures and has satisfied the linguistic criteria [106] for the evaluation of phylogenetic estimation.

7.1 Outcome

The outcome of this thesis is particularly promising for several reasons.

Firstly, it does advance the state of the art in cognate identification, with theoretical and applicative original contributions, which allow the achievement of an outstanding performance.

Secondly, it strengthens the hypothesis that orthographic learning systems may detect traces of sound changes left in the orthography and perform better than static models, specifically designed for the task of phonetic alignment. This idea is encouraging, considering that accurate phonetic transcriptions are difficult to produce and frequently performed manually, with the consequent loss of time and the possible lack of accuracy and uniformity.

Finally, the methodology proposed seems to overcome one of the limits of learning systems, which is the need for a large training dataset.

If a small group of sensibly aligned cognate pairs is able to train properly our learning system, not only may it help to discover relationships between languages when there is no consensus, but may also be particularly useful in the study of those languages that do not benefit from large cognate corpora. This may be the case with extinct languages and their relationships in the field of computational historical linguistics. Furthermore, our proposal may be beneficial when applied to less studied and less documented speech varieties in several fields of natural language processing, including machine translation, parallel bilingual corpora processing and lexicography.

7.2 Future work

Our future plans include additional investigation of substitution matrices and alignment techniques, with the aim of increasing further the cognate identification accuracy and, as a consequence, the capacity of inferring phylogenies. We are particularly interested in the following tasks:

- Development of a technique able to learn *BLOSUM-like* substitution matrices [62] and comparison of their performance in the tasks of cognate identification and phylogenetic inference, against the *PAM-like* matrices proposed in this study [33, 34, 35].
- Creation of an improved linguistic-inspired substitution matrix to be used in the alignment of the training datasets. This matrix should be *bigram-based* as opposed to *character-based.* Its impact would have to be tested on the performance of the string similarity measuring system in the task of cognate identification.
- Development of *PAM-like* and *BLOSUM-like* substitution matrices *bigram-based*, as opposed to *character-based.*
- Employment of a different linguistic training dataset to be used instead of the *Dyen et al.* corpus [42] and evaluation of the data impact on the tasks of cognate identification and phylogenetic inference. A good candidate may be the *Ringe et al.* dataset [115], which is also recommended for historical linguistic studies.

Bibliography

[1] Adamson, G. W., and Boreham, J. The use of an association measure based on character structure to identify semantically related pairs of words and document titles. *Information Storage & Retrieval 10*, 7-8 (1974), 253–260.

[2] Anttila, R. *An Introduction to Historical and Comparative Linguistics.* Macmillan Publishing Co., Inc., New York, U.S.A., 1972.

[3] Atkinson, Q. D., and Gray, R. D. Curious parallels and curious connections - phylogenetic thinking in biology and historical linguistics. *Systematic Biology 54*, 4 (2005), 513–526.

[4] Atkinson, Q. D., and Gray, R. D. Are accurate dates an intractable problem for historical linguistics? In *Mapping our Ancestry: Phylogenetic Methods in Anthropology and Prehistory*, C. Lipo, M. OBrien, S. Shennan, and M. Collard, Eds. Aldine Press, Chicago, Illinois, 2006, pp. 269–296.

[5] Atkinson, Q. D., and Gray, R. D. How old is the indo-european language family? Illumination or more moths to the flame? In *Phylogenetic methods and the prehistory of languages*, P. Forster and C. Renfrew, Eds. McDonald Institute for Archaeological Research, Cambridge, U.K., 2006, ch. 8, pp. 91–109.

[6] Atkinson, Q. D., Meade, A., Venditti, C., Greenhill, S. J., and Pagel, M. Languages evolve in punctuational bursts. *Science 319* (February 2008), 588.

[7] Atkinson, Q. D., Nicholls, G., Welch, D., and Gray, R. D. From words to dates: water into wine, mathemagic or phylogenetic inference? *Transactions of the Philological Society 103*, 2 (2005), 193–219.

[8] Bakker, D., Mller, A., Velupillai, V., Wichmann, S., Brown, C. H., Brown, P., Egorov, D., Mailhammer, R., and Grant, A. Adding typology to lexicostatistics: a combined approach to language classification. *Linguistic Typology 13* (2009), 167–179.

[9] Barbançon, F., Warnow, T., and Evans, S. N. An experimental study comparing linguistic phylogenetic reconstruction. In *Proceedings of the Conference on Language and Genes* (University of California, Santa Barbara, California, U.S.A., September 2006), pp. 45–55.

[10] Baum, L. E., Petrie, T., Soules, G., and Weiss, N. A maximization technique occurring in the statistical analysis of probabilistic function of markov chains. *The Annals of Mathematical Statistics 41*, 1 (1970), 164–171.

[11] Bellman, R. E. *Dynamic Programming.* Prinston University Press, 1957.

[12] Bilmes, J., and Zweig, G. The graphical models toolkit: An open source system for speech and time-series processing. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing* (2002), pp. 3916–3919.

[13] Blanchard, P., Petroni, F., Serva, M., and Volchenkov, D. Geometric representations of language taxonomies. *Computer Speech and Language*. (In press).

[14] Bodlaender, H. L., Fellows, M. R., and Warnow, T. J. Two strikes against perfect phylogeny. In *Automata, Languages and Programming. Lecture Notes in Computer Science*, W. Kuich, Ed., vol. 623. Springer Verlag, Berlin, Germany, 1992, pp. 273–283.

[15] Brew, C., and McKelvie, D. Word-pair extraction for lexicography. In *Proceedings of the 2nd International Conference on New Methods in Language Processing (NEMLP)* (Ankara, Turkey, 1996), pp. 45–55.

[16] Brown, C. H., Holman, E. W., Wichmann, S., and Vilupillai, V. Automated classification of the world's languages: A description of the method and preliminary results. *STUF Language Typology and Universals 61*, 4 (2008), 285–308.

[17] Bullen, P. S. *Handbook of Means and Their Inequalities*, 2nd edition ed., vol. 560 of *Mathematics and Its Applications.* Kluwer Academic Publishers, 2003.

[18] Campbell, L. *Historical Linguistics: An Introduction*, 2nd edition ed. The MIT Press, 2004.

[19] Cavalli-Sforza, L. L. *Genes, Peoples and Languages.* University of California Press, 2001.

[20] Chor, B., and Tuller, T. Finding a maximum likelihood tree is hard. *Journal of the ACM (JACM) 53*, 5 (2006), 722–744.

[21] Church, K. W. Char_align: a program for aligning parallel texts at the character level. In *Proceedings of the 31st Annual Meeting of the*

Association for Computational Linguistics (ACL-1993) (Columbus, Ohio, U.S.A., 1993), pp. 1–8.

[22] Covington, M. A. An algorithm to align words for historical comparison. *Computational Linguistics 22*, 4 (December 1996), 481–496.

[23] Covington, M. A. Alignment of multiple languages for historical comparison. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL 1998)* (1998), pp. 275–280.

[24] Crowley, T. *An Introduction to Historical Linguistics.* Oxford University Press, 1998.

[25] Cysouw, M., and Jung, H. Cognate identification and alignment using practical orthographies. In *Proceedings of the 9th Meeting of the ACL Special Interest Group in Computational Morphology and Phonology* (Prague, Czech Republic, 2007), pp. 109–116.

[26] Czekanowski, J. *Zarys metod statystycznych w zastosowaniu do antropologii [An outline of statistical methods applied in anthropology].* Towarzystwo Naukowe Warszawskie, 1913.

[27] Damerau, F. J. A technique for computer detection and correction of spelling errors. *Communications of the ACM 7*, 3 (March 1964), 171–176.

[28] Darwin, C. R. *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life.* John Murray, London, U.K., 1859.

[29] Darwin, C. R. *The Descent of Man, and Selection in Relation to Sex*. John Murray, London, U.K., 1871.

[30] Dayhoff, M. O., and Eck, R. V. A model of evolutionary change in proteins. *Atlas of Protein Sequence and Structure 1967-1968 3* (1968), 33–41.

[31] Dayhoff, M. O., Eck, R. V., and Park, C. M. A model of evolutionary change in proteins. *Atlas of Protein Sequence and Structure 5* (1972), 89–99.

[32] Dayhoff, M. O., Schwartz, R. M., and Orcutt, B. C. A model of evolutionary change in proteins. *Atlas of Protein Sequence and Structure 5*, 3 (1978), 345–352.

[33] Delmestri, A., and Cristianini, N. String similarity measures and PAM-like matrices for cognate identification. *Bucharest Working Papers in Linguistics XII*, 2 (2010), 71–82.

[34] Delmestri, A., and Cristianini, N. Robustness and statistical significance of PAM-like matrices for cognate identification. *Journal of Communication and Computer 7*, 12 (2010), 21–31.

[35] Delmestri, A., and Cristianini, N. Linguistic phylogenetic inference by pam-like matrices. *Submitted* ().

[36] Deza, E., and Deza, M. M. *Dictionary of distances.* Elsevier, 2006.

[37] Diamond, J. M. Express train to Polynesia. *Nature 336*, 6197 (1988), 307–308.

[38] Diamond, J. M., and Bellwood, P. Farmers and their languages: The first expansions. *Science 300*, 5619 (April 2003), 597–603.

[39] Dice, L. R. Measures of the amount of ecologic association between species. *Ecology 26*, 3 (July 1945), 297–302.

[40] Downey, S. S., Hallmark, B., Cox, M. P., Norquest, P., and Lansing, S. J. Computational feature-sensitive reconstruction of language relationships: Developing the ALINE distance for comparative historical linguistic reconstruction. *Journal of Quantitative Linguistics 15*, 4 (2008), 340–369.

[41] Durbin, R., Sean, E. R., Krogh, A., and Mitchison, G. *Biological Sequence Analysis.* Cambridge University Press, Cambridge, U.K., 1998.

[42] Dyen, I., Kruskal, J. B., and Black, P. An Indoeuropean classification: A lexicostatistical experiment. *Transactions of the American Philosophical Society 82*, 5 (October 1992).

[43] Ellison, M. T., and Kirby, S. 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* (Sydney, Australia, 2006), pp. 273–280.

[44] Evans, S. N., Ringe, D. A., and Warnow, T. Inference of divergence times as a statistical inverse problem. In *Phylogenetic methods and the prehistory of languages*, P. Forster and C. Renfrew, Eds. McDonald Institute for Archaeological Research, Cambridge, U.K., 2006, ch. 10, pp. 119–140.

[45] Fano, R. M. *Transmission of Information: A Statistical Theory of Communications.* Massachusetts Institute of Technology (MIT) Press, Cambridge, Massachusetts, U.S.A., 1961.

[46] Felsenstein, J. *Inferring Phylogenies.* Sinauer Associates Inc. Publishers, Sunderland, Massachusetts, U.S.A., 2004.

[47] Feng, D. F., and Doolittle, R. F. Converting amino acid alignment scores into measures of evolutionary time: a simulation study of various relationships. *Journal of Molecular Evolution 44*, 4 (April 1997), 361–370.

[48] Filali, K., and Bilmes, J. A dynamic bayesian framework to model context and memory in edit distance learning: An application to pronunciation classification. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)* (Ann-Arbor, Michigan, U.S.A., 2005), pp. 338–345.

[49] Forster, P., and Toth, A. Toward a phylogenetic chronology of ancient gaulish, celtic, and indo-european. *Proceedings of the National Academy of Sciences of the U.S.A. (PNAS) 100*, 15 (July 2003), 9079–9084.

[50] Foulds, L. R., and Graham, R. L. The steiner problem in phylogeny is np-complete. *Advances in Applied Mathematics 3*, 1 (March 1982), 43–49.

[51] Gotoh, O. An improved algorithm for matching biological sequences. *Journal of Molecular Biology 162*, 3 (December 1982), 705–708.

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

[53] Gray, R. D., and Jordan, F. M. Language trees support the express-train sequence of Austronesian expansion. *Nature 405* (June 2000), 1052–1055.

[54] Greenberg, J. H. *Essays in Linguistics.* University of Chicago Press, Chicago, Illinois, U.S.A., 1957.

[55] Greenhill, S. J., Atkinson, Q. D., Meade, A., and Gray, R. D. The shape and tempo of language evolution. In *Proceedings of the Royal Society, Series B: Biological Science* (2010), vol. 277, pp. 2443–2450.

[56] Gusfield, D. *Algorithms on Strings, Trees and Sequences.* Cambridge University Press, New York, U.S.A., 1997.

[57] Guy, J. B. M. An algorithm for identifying cognates in bilingual word-lists and its applicability to machine translation. *Journal of Quantitative Linguistics 1*, 1 (1994), 35–42.

[58] Hall, P. A. V., and Dowling, G. R. Approximate string matching. *ACM Computing Surveys (CSUR) 12*, 4 (December 1980), 381–402.

[59] Hamming, R. W. Error detecting and error correcting codes. *The Bell System Technical Journal 29*, 2 (April 1950), 147–160.

[60] Harris, B. Bi-Text, a new concept in translation theory. *Language Monthly 54* (March 1988), 8–10.

[61] Hastings, W. K. Monte Carlo sampling methods using Markov Chains and their applications. *Biometrika 57* (1970), 97–109.

[62] Henikoff, S., and Henikoff, J. G. Amino acid substitution matrices from protein blocks. In *Proceedings of the National*

Academy of Sciences of the United States of America (1992), vol. 89, pp. 10915–10919.

[63] Holden, C. J., and Gray, R. D. Rapid radiation, borrowing and dialect continua in the Bantu languages. In *Phylogenetic methods and the prehistory of languages*, P. Forster and C. Renfrew, Eds. McDonald Institute for Archaeological Research, Cambridge, U.K., 2006, pp. 19–31.

[64] Holman, E. W., Wichmann, S., Brown, C. H., Velupillai, V., Mller, A., and Bakker, D. Explorations in automated language classification. *Folia Linguistica 42*, 2 (2008), 331–354.

[65] Huelsenbeck, J. P., Larget, B., Miller, R. E., and Ronquist, F. Potential applications and pitfalls of Bayesian inference of phylogeny. *Systematic Biology 51*, 5 (2002), 673–688.

[66] Huson, D. H., and Steel, M. Phylogenetic trees based on gene content. *Bioinformatics 20*, 13 (2004), 2044–2049.

[67] Inkpen, D., Frunza, O., and Kondrak, G. Automatic identification of cognates and false friends in French and English. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2005)* (Borovets, Bulgaria, September 2005), pp. 251–257.

[68] Jain, A. K., and Dubes, R. C. *Algorithms for clustering data.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[69] Jelinek, F. *Statistical methods for speech recognition.* The MIT Press, Boston, Massachusetts, U.S.A., 1997.

[70] Kannan, S., and Warnow, T. A fast algorithm for the computation and enumeration of perfect phylogenies when the

number of character states is fixed. In *Proceedings of the 6th annual ACM-SIAM symposium on Discrete algorithms* (San Francisco, California, U.S.A., 1995), pp. 595–603.

[71] Ke, Y., Su, B., Song, X., Lu, D., Chen, L., Li, H., Qi, C., Marzuki, S., Deka, R., Underhill, P., Xiao, C., Shriver, M., Lell, J., Wallace, D., Wells, R., Seielstad, M., Oefner, P., Zhu, D., Jin, J., Huang, W., Chakraborty, R., Chen, Z., and Jin, L. African origin of modern humans in East Asia: A tale of 12,000 Y chromosomes. *Science 292* (May 2001), 1151–1153.

[72] Kessler, B. Computational dialectology in Irish Gaelic. In *Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics (EACL)* (San Francisco, California, U.S.A., 1995), pp. 60–66.

[73] Kessler, B. *The Significance of Word Lists.* CSLI Publications, Stanford, California, U.S.A., 2001.

[74] Kimura, M. *The Neutral Theory of Molecular Evolution.* Cambridge University Press, 1983.

[75] Koehn, P., and Knight, K. Knowledge sources for word-level translation models. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing* (2001), vol. 4, pp. 27–35.

[76] Kondrak, G. A new algorithm for the alignment of phonetic sequences. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics (ANLP-NAACL 2000)* (Seattle, Washington, U.S.A., 2000), vol. 4,

Morgan Kaufmann Publishers Inc., San Francisco, California, U.S.A., pp. 288–295.

[77] Kondrak, G. *Algorithms for Language Reconstruction.* PhD thesis, University of Toronto, Canada, 2002.

[78] Kondrak, G. Cognates and word alignment in Bitexts. In *Proceedings of the 10th Machine Translation Summit (MT Summit X)* (Phuket, Thailand, 2005), pp. 305–312.

[79] Kondrak, G. N-gram similarity and distance. In *Proceedings of the 12th International Conference on String Processing and Information Retrieval (SPIRE 2005)* (Buenos Aires, Argentina, November 2005), pp. 115–126.

[80] Kondrak, G. Identification of cognates and recurrent sound correspondences in word lists. *Traitement automatique des langues 50*, 2 (October 2009), 201–235.

[81] Kondrak, G., and Dorr, B. J. Identification of confusable drug names: A new approach and evaluation methodology. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)* (Geneva, Switzerland, August 2004), pp. 952–958.

[82] Kondrak, G., Marcu, D., and Knight, K. Cognates can improve statistical translation models. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2003)* (Edmonton, Alberta, Canada, May 2003), vol. Companion volume, pp. 46–48.

[83] Kondrak, G., and Sherif, T. Evaluation of several phonetic similarity algorithms on the task of cognate identification. In

Proceedings of the COLING-ACL 2006 Workshop on Linguistic Distances (Sydney, Australia, July 2006), pp. 43–50.

[84] Levenshtein, V. I. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady 10*, 8 (1966), 707–710.

[85] Lewis, M. P., Ed. *Ethnologue: Languages of the World*, 16th ed. SIL International, Dallas, Texas, 2009.

[86] Mackay, W. Word similarity using Pair Hidden Markov Models. Master's thesis, University of Alberta, Canada, Fall 2004.

[87] Mackay, W., and Kondrak, G. Computing word similarity and identifying cognates with Pair Hidden Markov Models. In *Proceedings of the 9th Conference on Computational Natural Language Learning (CoNLL 2005)* (Ann Arbor, Michigan, U.S.A., June 2005), pp. 40–47.

[88] Mann, G. S., and Yarowsky, D. Multipath translation lexicon induction via bridge languages. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL 2001)* (Pittsburgh, Pennsylvania, U.S.A., 2001), pp. 151–158.

[89] Manning, C. D., Raghavan, P., and Schütze, H. *Introduction to Information Retrieval.* Cambridge University Press, 2008.

[90] Manning, C. D., and Schütze, H. *Foundations of Statistical Natural Language Processing.* The Massachusetts Institute of Technology (MIT) Press, Cambridge, Massachusetts, U.S.A., 1999.

[91] Markowski, C. A., and Markowski, E. P. Conditions for the effectiveness of a preliminary test of variance. *The American Statistician 44*, 4 (November 1990), 322–326.

[92] Matlab. Analyzing the origin of the Human Immunodeficiency Virus. `http://www.mathworks.com/computational-biology/demos.html?file=/products/demos/shipping/bioinfo/hivdemo.html`.

[93] Matlab. Imagesc. `http://www.mathworks.com/help/techdoc/ref/imagesc.html`.

[94] McEnery, A. M., and Oakes, M. P. Sentence and word alignment in the CRATER project. In *Using Corpora for Language Research*, J. Thomas and M. Short, Eds. Longman, 1996, ch. 13, pp. 211–231.

[95] Melamed, D. I. Automatic evaluation and uniform filter cascades for inducing n-best translation lexicons. In *Proceedings of the 3rd Workshop on Very Large Corpora (WVLC3)* (Boston, Massachusetts, U.S.A., 1995).

[96] Melamed, D. I. Bitext maps and alignment via pattern recognition. *Computational Linguistics 25*, 1 (March 1999), 107–130.

[97] Mulloni, A., and Pekar, V. Automatic detection of orthographic cues for cognate recognition. In *Proceedings in the 5th international conference on Language Resources and Evaluation (LREC 2006)* (Genoa, Italy, 2006), pp. 2387–2390.

[98] Murphy, K. P. *Dynamic Bayesian networks: representation, inference and learning.* University of California, Berkeley, 2002.

[99] Nabende, P. Transliteration system using pair hmm with weighted fsts. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (ACL-IJCNLP 2009)* (Stroudsburg, PA, USA, August 2009), NEWS '09, Association for Computational Linguistics, pp. 100–103.

[100] Nakhleh, L., Warnow, T., Ringe, D. A., and Evans, S. N. A comparison of phylogenetic reconstruction methods on an Indo-European dataset. *Transactions of the Philological Society 103*, 2 (2005), 171–192.

[101] Navarro, G. A guided tour to approximate string matching. *ACM Computing Surveys 33*, 1 (March 2001), 31–88.

[102] Needleman, S. B., and Wunsch, C. D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology 48*, 3 (March 1970), 443–453.

[103] Nerbonne, J., and Heeringa, W. Measuring dialect distance phonetically. In *Proceedings of the 3rd Meeting of the ACL Special Interest Group in Computational Phonology (SIGPHON-97)* (Madrid, Spain, 1997), pp. 11–18.

[104] Nicholls, G. K., and Gray, R. D. Quantifying uncertainty in a stochastic model of vocabulary evolution. In *Phylogenetic methods and the prehistory of languages*, P. Forster and C. Renfrew, Eds. McDonald Institute for Archaeological Research, Cambridge, U.K., 2006, pp. 161–171.

[105] Nicholls, G. K., and Gray, R. D. Dated ancestral trees from binary trait data and their application to the diversification of languages. *Journal of the Royal Statistical Society, Series B: Statistical Methodology 70*, 3 (July 2008), 545–566.

[106] Nichols, J., and Warnow, T. Tutorial on computational linguistic phylogeny. *Language and Linguistics Compass 2*, 5 (2008), 760–820.

[107] Oakes, M. P. Computer estimation of vocabulary in protolanguage from word lists in four daughter languages. *Journal of Quantitative Linguistics 7*, 3 (December 2000), 233–243.

[108] Ott, L. R., and Longnecker, M. *An Introduction to Statistical Methods and Data Analysis*, 5th ed. Duxbury Press, Pacific Grove, California, U.S.A., 2001.

[109] Pagel, M., Atkinson, Q. D., and Meade, A. Frequency of word-use predicts rates of lexical evolution throughout Indo-European history. *Nature 449* (October 2007), 717–721.

[110] Petroni, F., and Serva, M. Language distance and tree reconstruction. *Journal of Statistical Mechanics: Theory and Experiment P08012* (August 2008), 1–15.

[111] Pirkola, A., Toivonen, J., Keskustalo, H., Visala, K., and Järvelin, K. Fuzzy translation of cross-lingual spelling variants. In *Proceedings of the 26th Annual International ACM SIGIR'03 Conference on Research and Development in Information Retrieval* (Toronto, Canada, 2003), pp. 345–352.

[112] Rexová, K., Bastin, Y., and Frynta, D. Cladistic analysis of Bantu languages: a new tree based on combined lexical and grammatical data. *Naturwissenschaften 93* (2006), 189–194.

[113] Rexová, K., Frynta, D., and Zrzavý, J. Cladistic analysis of languages: Indo-European classification based on lexicostatistical data. *Cladistics 19* (2003), 120–127.

[114] Ringe, D. A. On calculating the factor of chance in language comparison. *Transactions of the American Philosophical Society 82*, 1 (1992).

[115] Ringe, D. A., Warnow, T., and Taylor, A. Indo-European and computational cladistics. *Transactions of the Philological Society 100*, 1 (March 2002), 59–129.

[116] Ristad, E. S., and Yianilos, P. N. Learning string-edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 5 (May 1998), 522–532.

[117] Ruhlen, M. *The Origin of Language.* John Wiley & Sons Inc, 1994.

[118] Ryder, R. J., and Nicholls, G. K. Missing data in a stochastic Dollo model for cognate data, and its application to the dating of Proto-Indo-European. *Journal of the Royal Statistical Society, Series C: Applied Statistics*. (In press).

[119] Saitou, N., and Nei, M. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution 4*, 4 (1987), 406–425.

[120] Sankoff, D., and Kruskal, J. B. *Time Warps, String Edits, and Macromolecules. The Theory and Practice of Sequence Comparison.* The David Hume Series. CSLI Publications, U.S.A., 1999.

[121] Saussure de, F. *Course in General Linguistics.* Open Court, Illinois, U.S.A., 1983.

[122] Schulz, S., Markó, K., Sbrissia, E., Nohama, P., and Hahn, U. Cognate mapping - a heuristic strategy for the semi-supervised acquisition of a Spanish lexicon from a Portuguese seed lexicon. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)* (Geneva, Switzerland, 2004), pp. 813–819.

[123] Serva, M., and Petroni, F. Indo-European languages tree by levenshtein distance. *EPL (Europhysics Letters) 81*, 6 (March 2008), 68005–p1:p5.

[124] Simard, M., Foster, G. F., and Isabelle, P. Using cognates to align sentences in bilingual corpora. In *Proceedings of the 4th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-92)* (Montreal, Canada, 1992), pp. 1071–1082.

[125] Smith, T. F., and Waterman, M. S. Identification of common molecular subsequences. *Journal of Molecular Biology 147*, 1 (March 1981), 195–197.

[126] Sokal, R. R., and Michener, C. D. A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin 38* (1958), 1409–1438.

[127] Somers, H. L. Using cognates to align sentences in bilingual corpora. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL'98)* (1998), pp. 1227–1231.

[128] Somers, H. L. Aligning phonetic segments for children's articulation assessment. *Computational Linguistics 25*, 2 (1999), 267–275.

[129] Sørensen, T. J. A method of establishing groups of equal amplitude in plant sociology based on similarity of species, and its application to analyses of the vegetation on danish commons. *Kongelige Danske Videnskabernes Selskab 5*, 4 (1948), 1–34.

[130] STUDIER, J. A., AND KEPPLER, K. J. A note on the neighbor-joining algorithm of Saitou and Neil. *Journal of Molecular Biology and Evolution 5*, 6 (1988), 729–731.

[131] SWADESH, M. Salish internal relationships. *International Journal of American Linguistics 16*, 4 (October 1950), 157–167.

[132] SWADESH, M. Lexico-statistic dating of prehistoric ethnic contacts. *Proceedings of the Americal Philosophical Society 96*, 4 (August 1952), 452–463.

[133] SWADESH, M. Towards greater accuracy in lexicostatistics dating. *International Journal of American Linguistics 21*, 2 (April 1955), 121–137.

[134] TIEDEMANN, J. Automatic construction of weighted string similarity measures. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-99)* (College Park, Maryland, U.S.A., 1999), pp. 213–219.

[135] TURCHI, M., AND CRISTIANINI, N. A statistical analysis of language evolution. In *The Evolution of Language: Proceedings of the 6th Internationl Conference (EVOLANG6)* (Rome, Italy, 2006), pp. 348–355.

[136] WAGNER, H. *Linguistic atlas and survey of Irish dialects.* Dublin Institute for Advanced Studies, 1958-1969. 4 volumes.

[137] WAGNER, R. A., AND FISHER, M. J. The string to string correction problem. *Journal of the Association for Computing Machinery 21*, 1 (January 1974), 168–173.

[138] Wang, W. S.-Y., and Minett, J. W. Vertical and horizontal transmission in language evolution. *Transactions of the Philological Society 103*, 2 (August 2005), 121–146.

[139] Wichmann, S., and Saunders, A. How to use typological databases in historical linguistic research. *Diachronica 24*, 2 (2007), 373–404.

[140] Wieling, M., Leinonen, T., and Nerbonne, J. Inducing sound segment differences using Pair Hidden Markov Models. In *Computing and Historical Phonology: 9th Meeting of ACL Special Interest Group for Computational Morphology and Phonology Workshop* (Prague, Czech Republic, 2007), pp. 48–56.

[141] Wieling, M., Prokić, J., and Nerbonne, J. Evaluating the pairwise alignment of pronunciations. In *Proceedings of the 12th Workshop of the European Chapter of the Association for Computational Linguistics (EACL 2009) on Language Technology and Resources for Cultural Heritage, Social Sciences, Humanities, and Education* (Athens, Greece, 2009), pp. 26–34.

Appendix A

Swadesh lists

Swadesh [132] prepared a list of 200 universal and non-cultural words that he considered the *intimate* part of any vocabulary. After more research, he proposed a new list [133], which he recognised as being even more general and stable. It contained 100 words only, collected mainly from the previous list, but with the addition of seven new meanings: *breast, claw, full, horn, knee, moon* and *round.*

Dyen et al. [42] assembled the Comparative Indo-European corpus using 200-word *Swadesh lists. Kessler* [73] based the lists he prepared on a previous work of *Ringe* [114], who proposed a variant of the 200-word *Swadesh list.* He included the seven words added in the 100-word *Swadesh list* previously listed, plus the meanings *knife* and *now*, but excluded *to fear, to float, how, leg, to live, rope, to turn, when* and *where.*

We have extracted the training dataset for our cognate identification system from the Comparative Indo-European corpus by *Dyen et al.* [42], in so using 200-word *Swadesh lists*, whereas we have employed the *Kessler lists* [73] for test purposes, avoiding any overlap in their language sets.

We have also utilised the Comparative Indo-European dataset by *Dyen et al.* [42] with its 200-word *Swadesh lists* for the experiments of our phylogenetic tree builder.

Table A.1 summarises the lists mentioned and their composition.

#	Meaning	Swadesh 200	Swadesh 100	Kessler
1.	all	X	X	X
2.	and	X	–	X
3.	animal	X	–	X
4.	ashes	X	X	X
5.	at	X	–	X
6.	back (person's)	X	–	X
7.	bad	X	–	X
8.	bark (of tree)	X	X	X
9.	because	X	–	X
10.	belly	X	X	X
11.	big	X	X	X
12.	bird	X	X	X
13.	to bite	X	X	X
14.	black	X	X	X
15.	blood	X	X	X
16.	to blow (of wind)	X	–	X
17.	bone	X	X	X
18.	to breathe	X	–	X
19.	to burn (intrans.)	X	X	X
20.	child	X	–	X
21.	cloud	X	X	X
22.	cold (of weather)	X	X	X
23.	to come	X	X	X
24.	to count	X	–	X
25.	to cut	X	–	X
26.	day	X	–	X
27.	to die	X	X	X
28.	to dig	X	–	X
29.	dirty	X	–	X
30.	dog	X	X	X
31.	to drink	X	X	X

continued on next page

#	Meaning	Swadesh 200	Swadesh 100	Kessler
32.	dry	X	X	X
33.	dull (as a knife)	X	–	X
34.	dust	X	–	X
35.	ear	X	X	X
36.	earth	X	X	X
37.	to eat	X	X	X
38.	egg	X	X	X
39.	eye	X	X	X
40.	to fall (to drop)	X	–	X
41.	far	X	–	X
42.	fat (grease)	X	X	X
43.	father	X	–	X
44.	to fear	X	–	–
45.	feather	X	X	X
46.	few	X	–	X
47.	to fight	X	–	X
48.	fire	X	X	X
49.	fish	X	X	X
50.	five	X	–	X
51.	to float	X	–	–
52.	to flow	X	–	X
53.	flower	X	–	X
54.	to fly	X	X	X
55.	fog	X	–	X
56.	foot	X	X	X
57.	four	X	–	X
58.	to freeze	X	–	X
59.	fruit (berry)	X	–	X
60.	to give	X	X	X
61.	good	X	X	X
62.	grass	X	–	X
63.	green	X	X	X
64.	guts	X	–	X

continued on next page

#	Meaning	Swadesh 200	Swadesh 100	Kessler
65.	hair	X	X	X
66.	hand	X	X	X
67.	he	X	–	X
68.	head	X	X	X
69.	to hear	X	X	X
70.	heart	X	X	X
71.	heavy	X	–	X
72.	here	X	–	X
73.	to hit	X	–	X
74.	to hold (in hand)	X	–	X
75.	how	X	–	–
76.	to hunt	X	–	X
77.	husband	X	–	X
78.	I	X	X	X
79.	ice	X	–	X
80.	if	X	–	X
81.	in	X	–	X
82.	to kill	X	X	X
83.	to know	X	X	X
84.	lake	X	–	X
85.	to laugh	X	–	X
86.	leaf	X	X	X
87.	left (hand)	X	–	X
88.	leg	X	–	–
89.	to lie (on side)	X	X	X
90.	to live	X	–	–
91.	liver	X	X	X
92.	long	X	X	X
93.	louse	X	X	X
94.	man (male human)	X	X	X
95.	many	X	X	X
96.	meat (flesh)	X	X	X
97.	mother	X	–	X

continued on next page

#	Meaning	Swadesh 200	Swadesh 100	Kessler
98.	mountain	X	X	X
99.	mouth	X	X	X
100.	name	X	X	X
101.	narrow	X	–	X
102.	near	X	–	X
103.	neck	X	X	X
104.	new	X	X	X
105.	night	X	X	X
106.	nose	X	X	X
107.	not	X	X	X
108.	old	X	–	X
109.	one	X	X	X
110.	other	X	–	X
111.	person (human)	X	X	X
112.	to play	X	–	X
113.	to pull	X	–	X
114.	to push	X	–	X
115.	to rain	X	X	X
116.	red	X	X	X
117.	right (hand)	X	–	X
118.	right (correct, true)	X	–	X
119.	river	X	–	X
120.	road (path)	X	X	X
121.	root	X	X	X
122.	rope	X	–	–
123.	rotten	X	–	X
124.	rub	X	–	X
125.	salt	X	–	X
126.	sand	X	X	X
127.	to say	X	X	X
128.	scratch	X	–	X
129.	sea (ocean)	X	–	X
130.	to see	X	X	X

continued on next page

#	Meaning	Swadesh 200	Swadesh 100	Kessler
131.	seed	X	X	X
132.	to sew	X	–	X
133.	sharp (as a knife)	X	–	X
134.	short	X	–	X
135.	to sing	X	–	X
136.	to sit	X	X	X
137.	skin (person's)	X	X	X
138.	sky	X	–	X
139.	to sleep	X	X	X
140.	small	X	X	X
141.	to smell (trans.)	X	–	X
142.	smoke (of fire)	X	X	X
143.	smooth	X	–	X
144.	snake	X	–	X
145.	snow	X	–	X
146.	some	X	–	X
147.	to spit	X	–	X
148.	to split	X	–	X
149.	to squeeze	X	–	X
150.	to stab (to stick)	X	–	X
151.	to stand	X	X	X
152.	star	X	X	X
153.	stick (of wood)	X	–	X
154.	stone	X	X	X
155.	straight	X	–	X
156.	to suck	X	–	X
157.	sun	X	X	X
158.	to swell	X	–	X
159.	to swim	X	X	X
160.	tail	X	X	X
161.	that	X	X	X
162.	there	X	–	X
163.	they	X	–	X

continued on next page

#	Meaning	Swadesh 200	Swadesh 100	Kessler
164.	thick	X	–	X
165.	thin	X	–	X
166.	to think	X	–	X
167.	this	X	X	X
168.	thou (you sing.)	X	X	X
169.	three	X	–	X
170.	to throw	X	–	X
171.	to tie	X	–	X
172.	tongue	X	X	X
173.	tooth (front)	X	X	X
174.	tree	X	X	X
175.	to turn (intrans.)	X	–	–
176.	two	X	X	X
177.	to vomit	X	–	X
178.	to walk (to go)	X	X	X
179.	warm (hot)	X	X	X
180.	to wash	X	–	X
181.	water	X	X	X
182.	we	X	X	X
183.	wet	X	–	X
184.	what	X	X	X
185.	when	X	–	–
186.	where	X	–	–
187.	white	X	X	X
188.	who	X	X	X
189.	wide	X	–	X
190.	wife	X	–	X
191.	wind	X	–	X
192.	wing	X	–	X
193.	wipe	X	–	X
194.	with	X	–	X
195.	woman	X	X	X
196.	woods	X	–	X

continued on next page

#	Meaning	Swadesh 200	Swadesh 100	Kessler
197.	worm	X	–	X
198.	ye (you plural)	X	–	X
199.	year	X	–	X
200.	yellow	X	X	X
201.	breast	–	X	X
202.	claw (nail)	–	X	X
203.	full	–	X	X
204.	horn	–	X	X
205.	knee	–	X	X
206.	moon	–	X	X
207.	round	–	X	X
208.	knife	–	–	X
209.	now	–	–	X

Table A.1: Several variations of the *Swadesh list*

Appendix B

A linguistic-inspired substitution matrix

"*L'étymologie est une science où les voyelles ne font rien, et les consonnes fort peu de chose*".

Voltaire

We have produced a symmetric 26-by-26 linguistic-inspired substitution matrix based on knowledge of phonetic changes left in the orthography of the Indo-European language family, using the Latin alphabet without diacritics. We have used this matrix to align the training datasets of our cognate identification system and phylogenetic tree builder.

As introduced in Section 4.2, we have given a value of *2* to all the elements of the main diagonal, because it is likely that a character preserves itself. We have assigned a value of *0* to all the character transformations considered "*possible*", a value of *−3* to all the character transformations considered "*impossible*" and a gap penalty of *−1* for insertion and deletion, in order to have no overlaps between two *indels* and an "*impossible*" match. However, the terms "*possible*" and "*impossible*" do not have to be interpreted in a strict way, as they want only to represent traces of sound changes that are likely or unlikely to be found.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	2																									
B	-3	2																								
C	-3	-3	2																							
D	-3	0_3	-3	2																						
E	0_1	-3	-3	-3	2																					
F	-3	0_5	0_9	0_9	-3	2																				
G	-3	0_9	0_2	0_5	-3	-3	2																			
H	-3	-3	0_2	0_5	-3	0_5	-3	2																		
I	0_1	-3	-3	-3	0_1	-3	-3	0_{10}	2																	
J	0_1	-3	-3	0_5	0_1	-3	0_5	0_{10}	0_1	2																
K	-3	-3	0_9	-3	-3	-3	0_2	0_2	0_9	0_9	2															
L	-3	-3	0_3	0_3	-3	0_3	0_9	0_9	0_8	0_8	-3	2														
M	-3	0_3	-3	0_3	-3	-3	-3	-3	-3	0_9	-3	-3	2													
N	-3	-3	0_3	0_3	-3	-3	0_3	-3	-3	-3	0_3	0_4	0_3	2												
O	0_1	-3	-3	-3	0_1	-3	-3	-3	0_1	0_1	-3	-3	-3	-3	2											
P	-3	0_2	0_3	-3	-3	0_2	-3	-3	-3	-3	0_9	0_3	0_3	-3	-3	2										
Q	-3	-3	0_{10}	-3	-3	0_9	0_5	0_2	-3	-3	0_3	-3	-3	-3	-3	0_3	2									
R	-3	-3	-3	0_6	-3	-3	-3	0_9	-3	-3	0_3	0_6	0_3	0_6	-3	-3	-3	2								
S	-3	0_3	0_7	0_9	-3	-3	0_9	0_5	-3	0_5	0_9	0_3	-3	-3	-3	-3	-3	0_6	2							
T	-3	0_3	0_3	0_2	-3	-3	-3	0_5	-3	-3	0_9	0_3	0_3	0_3	-3	0_3	-3	0_3	0_2	2						
U	0_1	-3	-3	-3	0_1	-3	-3	0_9	0_1	0_1	-3	0_8	-3	-3	0_1	-3	-3	-3	-3	-3	2					
V	-3	0_5	0_9	-3	-3	0_{10}	0_9	0_5	-3	-3	0_9	-3	0_9	-3	-3	0_5	0_9	-3	-3	-3	0_{10}	2				
W	0_1	0_5	-3	-3	0_1	-3	-3	-3	0_1	0_1	-3	-3	-3	-3	0_1	-3	0_9	-3	-3	-3	0_1	0_{10}	2			
X	-3	-3	0_3	-3	-3	-3	0_9	0_5	-3	0_5	0_9	-3	-3	-3	-3	-3	-3	-3	0_5	-3	-3	-3	-3	2		
Y	0_1	-3	-3	-3	0_1	-3	-3	0_{10}	0_1	0_1	-3	0_8	-3	-3	0_1	-3	-3	-3	-3	-3	0_1	-3	-3	-3	2	
Z	-3	-3	0_9	0_5	-3	-3	0_7	-3	-3	0_9	-3	-3	-3	0_9	-3	-3	-3	0_6	0_9	0_2	-3	-3	-3	0_9	-3	2

Table B.1: A linguistic-inspired substitution matrix

Table B.1 shows the linguistic-inspired substitution matrix and, for readability, only the lower triangular matrix is filled in. In order to explain the traces that systematic sound changes left in written words, we have identified and listed several linguistic motivations [2, 24, 18], which we have found useful from an orthographic point of view. The list does not mean to be complete and may contain a few errors.

In the matrix, each character transformation considered "*possible*" is displayed having in subscript the identification number of its motivation, even if frequently more justifications may apply to the same character pair. Examples have been provided in brackets, arrows have been used when one orthographic form derives from another one, and commas when daughter languages testify a sound change from a non-documented common ancestor. Transliteration has been utilised to report words belonging to languages not using the Roman alphabet.

1. *Vowel change*: a change in the way vowels or semi-vowels are pronounced or written. For example:
    - A → E (Latin *basium* → Spanish *beso*; 'kiss')
    - A – E (Latin *mater*, Greek *meter*; 'mother')
    - A → I (Latin *caelum* → Italian *cielo*; 'sky')
    - A – I (Dutch *nacht*, German *Nacht*, English *night*; 'night')
    - A – J (Flemish *aerde*, Danish *jord*; 'earth')
    - A – O (Latin *mater*, Lithuanian *mote*, English *mother*; 'mother')
    - A – U (Sanskrit *matar*, German *Mutter*; 'mother')
    - E → I (Latin *fenestra* → Italian *finestra*; 'window')
    - E – I (Old Slavonic *gnezdo*, Sanskrit *nidah*; 'nest')
    - E – J (Serbo-Croatian *pepeo*, Slovenian *pepju*; 'ashes')
    - E – O (Lithuanian *vemti*, Sardinian *vomitare*; 'to vomit')
    - E – U (Slovak *kedy*, Albanian *kur*; 'when')
    - E – Y (Breton *nez*, Welsh *nyth*; 'nest')
    - I → J (Latin *iniustus* → Portuguese *injusto*; 'unfair')
    - I – O (Ukrainian *rik*, Polish *rok*; 'year')
    - I – U (Irish *ní*, Albanian *nuk*; 'not')
    - I → Y (Latin *abbatia* → French *abbaye*; 'abbey')
    - J → I (Latin *januarius* → Romanian *ianuarie*; 'January')
    - J – O (Slovenian *osjba*, Belarusian *asoba*; 'person')

- U → O (Latin *abundare* → French *abonder*; ‘to abound’)
- U – Y (Sanskrit *mus*, Greek *mys*; ‘mouse’)
- U – W (Breton *ui*, Welsh *wy*; ‘egg’)
- Y → I (Latin *gyrus*, Spanish, Italian *giro*; ‘turn’)

2. *Consonant shift*: a change in the way consonants are pronounced and written. If several consonant sounds move stepwise along a phonetic scale, the consonant shift is called a *consonant chain shift*. There are several famous examples of consonant chain shifts, including *Grimm's law* and *Verner's law* [18]. For example:

   - B – P (Latin *labium*, English *lip*, Swedish *läpp*; ‘lip’)
   - C – H (Latin *canis*, Welsh *ci*, Gothic *hunds*); ‘hound’)
   - D – T (Sanskrit *dvau*, Latin *duo*, Dutch *twee*, English *two*; ‘two’)
   - G – C (Latin *gelu*, English *cold*; ‘cold’)
   - G – K (Latin *gelu*, German *kalt*, Icelandic *kaldr*; ‘cold’)
   - K – H (Greek *kyon*, Old Norse *hundr*, English *hound*; ‘hound’)
   - P – F (Sanskrit *pat*, Greek *pos*, English *foot*, German *Fuß*; ‘foot’)
   - Q – H (Latin *quod*, Gothic *hva*, Danish *hvad*; ‘what’)
   - T – S (English *eat*, German *essen*; ‘eat’)
   - T – Z (English *two*, German *zwei*; ‘two’)

3. *Assimilation*: the change of a sound that becomes more similar to another one present in the word. For example:

   - BD → DD (Latin *abdomen* → Italian *addome*; ‘abdomen’)
   - BM → MM (Latin *submergere* → Italian *sommergere*; ‘to flood’)
   - BS → SS (Latin *obsequium* → Italian *ossequio*; ‘homage’)

- BT → TT (Latin *subtilis* → *Italian sottile*; ‘thin’)
- CL → LL (Latin *clavis* → Spanish *llave*; ‘key’)
- CT → TT (Latin *octo* → Italian *otto*; ‘eight’)
- DM → MM (Latin *admittere* → Italian *ammettere*; ‘to admit’)
- DN → NN (Latin *adnectere* → Italian *annettere*; ‘annex’)
- DR → RR (Latin *quadratus* → French *carré*; ‘square’)
- FL → LL (Latin *flamma* → Spanish *llama*; ‘flame’)
- GD → DD (Latin *frigidus/frigdus* → Italian *freddo*; ‘cold’)
- K – Q (Lithuanian *penke*, Latin *quinque*; ‘five’)
- LD – LL (English *cold*, Swedish *kall*; ‘cold’)
- LN – LL (Lithuanian *kalnelis*, Latin *collis*; ‘hill’)
- LS → SS (Latin *pulsare* → French *pousser*; ‘to push’)
- LT → CH (Latin *cultellus* → Spanish *cuchillo*; ‘knife’)
- MN → NN (Latin *somnus* → Italian *sonno*; ‘sleep’)
- MN → MM (Latin *somnus* → French *sommeil*; ‘sleep’)
- MR → RR (Latin *cumrumpere* → French *corrompre*; ‘to corrupt’)
- NK – KK (German *trinken*, Faroese *drekka*; ‘to drink’)
- NK – CK (English *drink*, Swedish *dricka*; ‘to drink’)
- NL → LL (Latin *inludere* → Italian *illudere*; ‘to deceive’)
- NR → RR (Latin *ponere/ponre* → Italian *porre*; ‘to put’)
- P → C (Early Latin *pequere* → Classical Latin *coquere*; ‘to cook’)
- P – Q (Greek *pente*, Latin *quinque*; ‘five’)
- PL → LL (Latin *pluvia* → Spanish *lluvia*; ‘rain’)
- PN – MN (Sanskrit *svapnah*, Latin *somnus*; ‘sleep’)

- PT → TT (Latin *septem* → Italian *sette*; ‘seven’)
- RK – RR (Swedish *torka*, Old Norse *thurr*, Danish *torre*; ‘to dry’)
- TL → LL (Latin *spatula/spatla* → Italian *spalla*; ‘shoulder’)
- TR → RR (Latin *petra* → French *pierre*; ‘stone’)
- X → SS (Latin *saxum* → Italian *sasso*; ‘stone’)
- XC → CC (Latin *excedere* → Italian *eccedere*; ‘to exceed’)

4. *Dissimilation*: the change of a sound that becomes less similar to another present in the word. For example:

    - N → L (Latin *venenum* → Italian *veleno*; ‘poison’)
    - N → R (Latin *hominem* → Spanish *hombre*; ‘man’)
    - Q → C (Latin *quinque* → Italian *cinque*, French *cinq*; ‘five’)
    - R → D (Latin *rarus* → Italian *rado*; ‘rare’)
    - R → L (Latin *arbor* → Spanish *arból*; ‘tree’)

5. *Lenition*: the change of a consonant sound that becomes weaker or a semi-vowel. For example:

    - B – F (Icelandic *blóm*, German *Blume*, Latin *flos*; ‘flower’)
    - B → V (Latin *fabula* → Italian *favola*; ‘tale’)
    - C → G (Latin *amicus* → Portuguese *amigo*; ‘friend’)
    - D → G (Latin *diurnus* → Italian *giorno*; ‘day’)
    - D → H (Latin *cadere* → Portuguese *cahir*; ‘to fall’)
    - D → Z (Latin *dies* → Romanian *zi*; ‘day’)
    - F → H (Latin *ficatum* → Spanish *hígado*; ‘liver’)
    - G → J (Latin *gamba* → French *jambe*; ‘leg’)
    - P → B (Latin *scopa* → Spanish *escoba*; ‘broom’)

- P → V (Latin *aprilis* → French *avril*; 'April')
- Q → G (Latin *aqua* → Catalan *aigua*; 'water')
- S – H (Latin *septem*, Avestan *hapta*, Old Persian *haft*; 'seven')
- S → J (Latin *sapo* → Spanish *jabón*; 'soap')
- T → D (Latin *natare* → Spanish *nadar*; 'to swim')
- T → H (Latin *fructus* → Provençal *frucho*; 'fruit')
- X → H (Old Slavonic *xoditŭ* → Slovenian *hodit*; 'to walk')
- X → J (Latin *fixum* → Spanish *fijo*; 'fixed')
- X → S (Latin *extremus* → Italian *estremo*; 'extreme')

6. *Rhotacism*: the change into *R* of another consonant, which is a form of lenition. For example:

   - D → R (Latin *cadere* → Catalan *caurer*; 'to fall')
   - L → R (Latin *caelum* → Romanian *cer*; 'sky')
   - N → R (Latin *fenestra* → Romanian *fereastră*; 'window')
   - Z – R (Avestan *mazja*, Old Irish *mor*, German *mehr*; 'more')

7. *Fortition*: the change of a consonant sound from a weak to a strong sound. For example:

   - J → G (Latin *januarius* → Italian *gennaio*; 'January')
   - S → C (Latin *basium* → Italian *bacio*; 'kiss')
   - V → B (Latin *servire* → Romanian *serbi*; 'to serve')
   - Z → G (Latin *zelosus* → Italian *geloso*; 'jealous')

8. *L-vocalisation*: the replacement of an *L* by a vowel or semi-vowel. For example:

   - L → I (Latin *florem* → Italian *fiore*; 'flower')

- L → U (Latin *caldus* → French *chaud*; ‘hot’)

9. Other examples of changes in the way consonants are pronounced or written, including *palatalisation* and *coalescence*:

   - CL → CH (Latin *clamare* → Portoguese *chamar*; ‘to call’)
   - CT → PT (Latin *coctum* → Romanian *copt*; ‘cooked’)
   - C → Z (Avestan *panca* → Waziri *pinze*; ‘five’)
   - DV → B (Old Latin *dvis* → Latin *bis*; ‘twice’)
   - FL → CH (Latin *flamma* → Portuguese *chama*; ‘flame’)
   - G – B (Greek *gune*, Welsh *benyw*, Irish *bean*; ‘woman’)
   - K – C (Greek *hekaton*, Latin *centum*, Old Irish *cet*; ‘hundred’)
   - K – P (Avestan *yakar*, Greek *hepar*; ‘liver’)
   - K – S (Breton *kant*, Sanskrit *satam*; ‘hundred’)
   - K – T (Lithuanian *penke*, Greek *pente*; ‘five’)
   - K – X (Russian *kto*, Ukrainian *xto*; ‘who’)
   - LL → GL (Latin *allium* → Italian *aglio*; ‘garlic’)
   - PL → CH (Latin *pluvia* → Portuguese *chuva*; ‘rain’)
   - S → G (Latin *ros* → Italian *rugiada*; ‘dew’)
   - T → Z (Latin *sapientia* → Italian *sapienza*; ‘wisdom’)
   - V → G (Latin *pluvia* → Italian *pioggia*; ‘rain’)

10. *Homophony*: the representation of the same sound by different characters in different languages or in different historical times. For example:

    - F – V (German *Fuß*, Dutch *voet*; ‘foot’)
    - I → H (Latin *Hispania* → Portuguese *Espanha*; ‘Spain’)

- Q → C (Latin *antiquus* → Italian *antico*; 'ancient')
- V → U (Latin *avis* → *avicella* → *aucellus*; 'bird')
- V – W (Swedish *vinna*, German gewinnen); 'to win')

It is worth noting that *indels* can also have several linguistic motivation, as sounds can be lost or introduced. Possible types of sound loss and introduction include:

- *Aphaeresis*: the loss of initial sounds.
  - Latin *ecclesia* → Italian *chiesa*; 'church'.
  - Latin *episcopus* → Italian *vescovo*; 'bishop'.
  - Latin *instrumentum* → Italian *strumento*; 'tool'.
- *Syncope*: the loss of medial sounds.
  - Latin *insula* → Italian *isola*; 'island'.
  - Latin *regalis* → Portuguese, Spanish *real*; 'regal'.
  - Latin *tabula* → Spanish *tabla*; 'table'.
- *Apocope*: the loss of final sounds.
  - Latin *libertatem* → Italian *libertade/libertà*; 'freedom'.
  - Latin *lupus* → French *loup*; 'wolf'.
  - Latin *panis* → Spanish *pan*; 'bread'.
- *Prothesis*: the insertion of an initial sound.
  - Latin *laurus* → Italian *alloro*; 'laurel'.
  - Latin *strata* → Portuguese *estrada*; 'road'.
  - Latin *vulturius* → Italian *avvoltoio*; 'vulture'.

- *Epenthesis*: the insertion of a medial sound.
  - Latin *hominem* → *homne* → *homre* → Spanish *hombre*; ‘man’.
  - Old English *thunor* → English *thunder*; ‘thunder’.
  - Latin *tremulare* → French *trembler*; ‘to tremble’.
- *Metathesis*: the contemporary loss and introduction of two sounds that switch place.
  - Latin *crocodilus* → Italian *coccodrillo*; ‘crocodile’.
  - Latin *parabola* → Spanish *palabra*; ‘word’.