# Ab Initio: Automatic Latin Proto-word Reconstruction

**Alina Maria Ciobanu, Liviu P. Dinu**
Faculty of Mathematics and Computer Science, University of Bucharest
Human Language Technologies Research Center, University of Bucharest
alina.ciobanu@my.fmi.unibuc.ro, ldinu@fmi.unibuc.ro

## Abstract

Proto-word reconstruction is central to the study of language evolution. It consists of recreating the words in an ancient language from its modern daughter languages. In this paper we investigate automatic word form reconstruction for Latin proto-words. Having modern word forms in multiple Romance languages (French, Italian, Spanish, Portuguese and Romanian), we infer the form of their common Latin ancestors. Our approach relies on the regularities that occurred when the Latin words entered the modern languages. We leverage information from all modern languages, building an ensemble system for proto-word reconstruction. We use conditional random fields for sequence labeling, but we conduct preliminary experiments with recurrent neural networks as well. We apply our method on multiple datasets, showing that our method improves on previous results, having also the advantage of requiring less input data, which is essential in historical linguistics, where resources are generally scarce.

## 1 Introduction

Proto-language reconstruction is one of the main concerns of historical linguistics and is central to understanding the evolution of the world's languages. Reconstructing an ancestral language from the modern languages descending from it not only allows an analysis of language change across space and time, but also reveals information about the historical relationships between languages. The traditional process of reconstructing ancient languages is called the comparative method (Anttila, 1989) and relies on cognates (words in different languages descending from a common proto-word). The main idea of the comparative method is to perform a property-based comparison on multiple sister languages in order to infer properties of their common ancestor. For a long period, the comparative reconstruction has been a time-consuming manual process that required a large amount of intensive work. The first step of the process consists in identifying cognate pairs.

Identifying cognates and borrowings by means of computational approaches has attracted considerable attention in recent years (Hall and Klein, 2010; Tsvetkov et al., 2015; Ciobanu and Dinu, 2015). However, few studies went beyond this step, and beyond the comparative method, to automate the process of proto-language reconstruction (Oakes, 2000; Bouchard-Côté et al., 2013; Atkinson, 2013). Reconstructing proto-words is a challenging task. While the main hypothesis in this research problem is that there are regularities and patterns in how words evolved from the ancestor language to its modern daughter languages, there are also words which diverged significantly from their ancestor. Take, for example, the Latin word *umbilicu(lu)s*. It evolved into *buric* (Romanian), *nombril* (French), and *umbigo* (Portuguese).

One of the best approaches to proto-word reconstruction (Bouchard-Côté et al., 2013) relies on an analogy with reconstructing the genealogy of the species from genetic sequences in biology. This approach requires an existing phylogenetic tree and the phonetic transcripts of the words, to infer the ancient word forms based on probability estimates for all the possible sound changes on each branch of the tree.





*Proceedings of the 27th International Conference on Computational Linguistics*, pages 1604–1614
Santa Fe, New Mexico, USA, August 20-26, 2018.

**Contributions of the present work.** In this paper, we investigate proto-word reconstruction starting from modern word forms. The main goal of our study is to achieve state-of-the-art performance in proto-word reconstruction using less resources than in previous studies. We aim at providing a tool for linguists to use in their research on endangered and extinct languages. Our hypothesis is that orthographic changes represent sound correspondences to a large extent, and thus we attempt to reconstruct proto-words from the orthographic form of the modern words.

We address this problem in two steps. Firstly, given cognate pairs in multiple modern languages and their common ancestors, we apply a word production method based on sequence labeling for reconstructing proto-words. We apply the method on each modern language individually. Secondly, we propose several ensemble methods for combining information from multiple word production systems, with the purpose of joining the best productions from all modern languages. Through experiments, we show that our best ensemble system outperforms previous results (Bouchard-Côté et al., 2007). The novelty of our approach is enhancing the sequence alignment system with an additional step of reranking. We also compute the results of an oracle, which shows the potential of this approach. The proposed methods have the advantage of not requiring phonetic transcripts and other data besides the training word pairs (such as a corpus in the target language, as some of the existing methods require); external information regarding language evolution is difficult to obtain for some languages, and this method can be applied on low-resourced and endangered languages. Moreover, as opposed to previous methods, our system is able to reconstruct proto-words even from incomplete cognate sets (cognate pairs in multiple modern languages descending from a common proto-language).

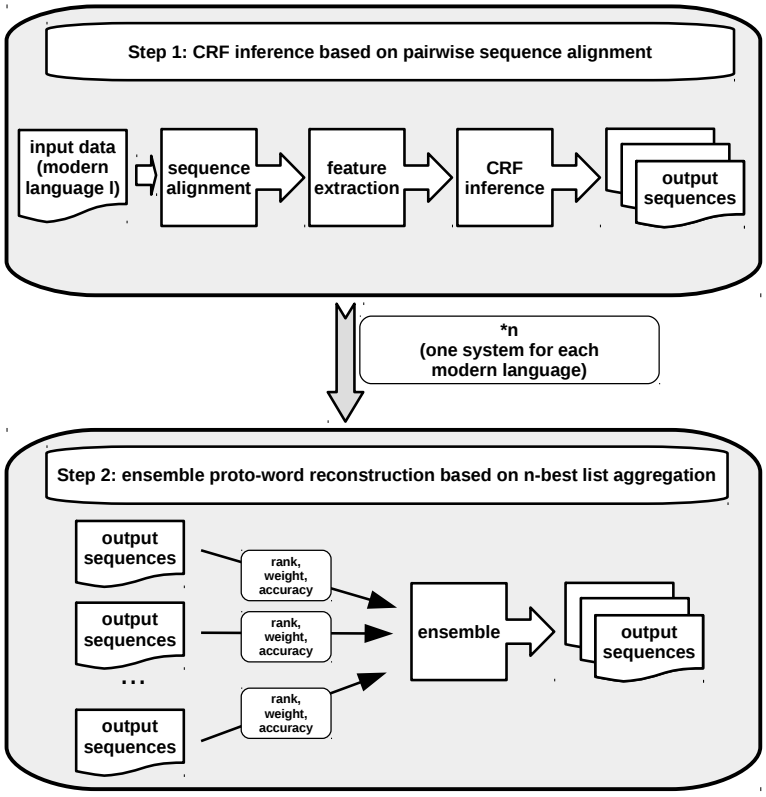


Figure 1: Methodology for proto-word reconstruction.

## 2 Methodology and Algorithm

In this section we introduce a new technique for proto-word reconstruction. Given cognate sets in several Romance languages, we infer the form of their common Latin ancestors. Our goal is to produce a tool that would provide support in historical linguistics, where the produced proto-words would be further analyzed by domain experts. We propose an approach based on conditional random fields. For each

modern language independently, we first apply a sequence labeling method that produces the form of the Latin ancestors. For this first step, we only require *(word, proto-word)* pairs (not cognate sets), as the input data is formed of modern word forms and their ancestors. Then we define several ensemble methods to take advantage of the information provided by all languages, in order to improve performance on Latin proto-word reconstruction. Our methodology is illustrated in Figure 1.

### 2.1 Conditional Random Fields

From the alignment of related words in the training set, the system learns orthographic patterns for the changes in spelling between each modern language and the proto-language. We apply a sequence labeling method to infer the form of the Latin proto-words for the modern words in the test set. The method that we employ is based on sequence labeling, an approach that has been proven useful in generating transliterations (Ganesh et al., 2008; Ammar et al., 2012) and in cognate production (Ciobanu, 2016).

In our case, the words in the modern languages are the sequences, and their characters are the tokens. Our purpose is to obtain, for each input word, a sequence of characters that compose its proto-word. To this end, we use conditional random fields (CRFs) (Lafferty et al., 2001). As features for the CRF system, we use character n-grams from the input words, extracted from a fixed window $w$ around the current token.

### 2.2 Pairwise Sequence Alignment

To align pairs of words, we experimented with two alignment methods that have been proven useful in natural language processing and computational biology: the alignment method based on profile hidden Markov models proposed by Bhargava and Kondrak (2009) and the Needleman-Wunsch global alignment algorithm (Needleman and Wunsch, 1970).

Since the alignment is only a pre-processing step for our task, we evaluate the alignment methods by the downstream results, i.e., the accuracy obtained by the CRF system when using one or the other alignment method. We observed that the results obtained with the two alignment methods were very similar, slightly better for the latter. Thus, we report the results obtained with the Needleman Wunsch alignment algorithm.

The alignment algorithm uses, in our case, words as input sequences and a basic substitution matrix, which gives equal scores to all substitutions, disregarding diacritics (e.g., we ensure that *e* and *è* are matched). For example, for the Romanian word *frumos* (meaning *beautiful*) and its Latin ancestor *formosus*, the alignment is as follows:

```
f - r u m o s - -
f o r - m o s u s
```

For each character in the modern word (after the alignment), the associated label is the character which occurs on the same position in its proto-word. In the case of insertions, we add the new character to the previous label, because there is no input character in the source language to which we could associate the inserted character as label. We account for affixes separately: for each input word, we add two more characters B and E, marking the beginning and the end of the word. The characters that are inserted in the target word at the beginning or at the end of the word are associated to these special characters. In order to reduce the number of labels, we replace the label with $\star$ for input tokens that are identical to their labels. Thus, for the previous example, the labels are as follows:

```
B  f  r  u  m  o  s  E
↓  ↓  ↓  ↓  ↓  ↓  ↓  ↓
*  fo *  -  *  *  *  usE
```

### 2.3 Ensembles



Ensembles of classifiers combine the results of multiple classifiers, in order to improve performance. In our case, the classifiers use different input data: we train a classifier for each modern language and

combine their results, in order to obtain Latin proto-words with higher accuracy. Our goal is to improve the performance of the system by taking advantage of the information provided by all languages.

Each classifier produces, for each input word, an n-best list of possible proto-words. To combine the outputs of the classifiers, we propose four fusion methods based on the ranks in the n-best lists and the probability estimates provided by the individual classifiers for each possible production.

Given a cognate set, we combine the n-best lists previously obtained to compute a joint n-best list that leverages information from all modern languages.

**Fusion by rank.** We compute an average rank for each production from the n-best lists, as the average rank from all modern languages. We favor words that occur on the first position in multiple n-best lists. In other words, for a given n-best list, we associate weights in a Borda sense (Borda, 1781) to all the words from the list: we give weight *n* to the word produced with the highest confidence, *n - 1* to the second one, and so on, until weight *1* to the *n*-th produced word. For an n-best list $L_i$ and a word $u$, we denote by $w(u_i)$ the weight of word *u* in $L_i$; if *u* is not a word from list $L_i$, then $w(u_i) = 0$. Given *k* n-best lists (one for each modern language) and a produced word $u$, we define the rank weight of $u$ over the *k* n-best lists as:

$$w_r(u) = (1/k) * \sum_{i=1}^{k} w(u_i). \tag{1}$$

**Fusion by rank and accuracy.** Starting from the previous fusion method, we also take into account the training accuracy for each language. We give more importance to the *vote* of the languages that obtained a better performance, multiplying the weight by the training accuracy. In other words, for each n-best list $L_i$, let $\pi(i)$ be the training accuracy for language $i$ ($COV_{10}$). Given *k* n-best lists (one for each modern language) and a produced word $u$, we define the rank-accuracy weight of $u$ over the *k* n-best lists as:

$$w_{ra}(u) = (1/k) * \sum_{i=1}^{k} w(u_i)\pi(i). \tag{2}$$

**Fusion by weight.** We compute an average confidence score for each production from the n-best lists, using the confidence score reported by the sequence labeling system. We rerank the productions based on the average confidence score. Given *k* n-best lists (one for each modern language), a produced word $u$, and $w(u_i)$ the confidence score of the sequence labeling system in list $L_i$, we define the confidence weight of $u$ over the *k* n-best lists as:

$$w_c(u) = (1/k) * \sum_{i=1}^{k} w(u_i). \tag{3}$$

This is similar to the first fusion method, but uses the sequence labeling system's weights instead of the weights obtained from the ranking in the n-best lists.

**Fusion by weight and accuracy.** Starting from the previous fusion method, we also take into account the training accuracy for each language. We give more importance to the *vote* of the languages that obtained a better performance, multiplying the score by the training accuracy. Given *k* n-best lists (one for each modern language), a produced word $u$, and $w(u_i)$ the confidence score of the sequence labeling system in list $L_i$, we define the confidence-accuracy weight of $u$ over the *k* n-best lists as:

$$w_{ca}(u) = (1/k) * \sum_{i=1}^{k} w(u_i)\pi(i). \tag{4}$$

This is similar to the second fusion method, but uses the sequence labeling system's weights instead of the weights obtained from the ranking in the n-best lists.

The output of each ensemble is a new n-best list in which the words are sorted in descending order of their computed weights, as described in Equations 1-4.

**Oracle.** An oracle classifier is an ensemble method that produces the correct result if any of the comprising classifiers produces the correct result. The purpose of an oracle is to determine the upper limit of an ensemble. The probability of obtaining correct results from an oracle is the following (Kuncheva et al., 2003):

$$P(oracle) = 1 - P(all\_wrong) \tag{5}$$

## 3 Experimental Setup

In this section we describe the experimental setup that we employed to assess the performance and to analyze the proposed method for proto-word reconstruction.

### 3.1 Datasets

We use datasets of Romance languages with Latin ancestors:

**Dataset 1.** The dataset proposed by Bouchard-Côté et al. (2007). It contains 585 complete cognate sets in three Romance languages (Spanish, Portuguese, Italian) and their common Latin ancestors. It is provided in two versions: orthographic and phonetic (IPA transcriptions). This dataset allows us to compare our results with a previous state-of-the-art method for proto-word reconstruction.

**Dataset 2.** The dataset proposed by Reinheimer Ripeanu (2001). It consists of 1,102 cognate sets in five Romance languages (Spanish, Italian, Portuguese, French, Romanian) and their common Latin ancestors. Note that not all of these cognate sets are complete (that is, for some of them there are not cognates provided in all five modern languages).

**Dataset 3.** The dataset proposed by Ciobanu and Dinu (2014b) and previously used for cognate detection (Ciobanu and Dinu, 2014a). It contains 3,218 complete cognate sets in five Romance languages (Spanish, Italian, Portuguese, French, Romanian) and their common Latin ancestors. Below we provide an example of a cognate set from this dataset:

vehicul (**Ro**) | véhicule (**Fr**) | veicolo (**It**) | vehículo (**Es**) | veículo (**Pt**) | vehiculum (**Lat**)

### 3.2 Task Setup

We split each dataset in subsets for train, dev and test (3:1:1 ratio). For inferring the form of the proto-words, we use the CRF implementation provided by the Mallet toolkit (McCallum, 2002). For parameter tuning, we perform a grid search for the number of iterations in $\{1, 5, 10, 25, 50, 100\}$ and for the size of the window $w$ in $\{1, 2, 3, 4, 5\}$.

### 3.3 Evaluation Measures

Following previous work in this area (Bouchard-Côté et al., 2007; Beinborn et al., 2013), we use the evaluation measures listed below to assess the performance of our method:

**Average edit distance.** To assess how close the productions are to the correct form of the proto-words, we report the edit distance between the produced words and the gold standard. We report both the unnormalized and the normalized edit distance. For normalization in the [0,1] interval, we divide the edit distance by the length of the longest string.

**Coverage.** Also known as *top n accuracy*, the coverage is a relaxed metric which computes the percentage of input words for which the n-best output list contains the correct proto-word (the gold standard). We use n $\in \{1, 5, 10\}$. The practical importance of analyzing the top n results is that we offer a filter to narrow down the possible forms of the output words to a low-dimensional list, that linguists can analyze, aiming to identify the correct form of the proto-word. Note that the coverage for n = 1 is the well-known measure accuracy.

**Mean reciprocal rank.** The mean reciprocal rank is an evaluation measure which applies to systems that produce an ordered output list for each input instance. Given an input word, the higher the position of its correct proto-word in the output list, the higher the mean reciprocal rank value:

$$MRR(w_i) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{rank_i}, \tag{6}$$

where m is the number of input instances, and $rank_i$ is the position of $w_i$'s proto-word in the output list. If $w_i$'s correct proto-word is not in the output list, we consider the reciprocal rank 0.

## 4 Results

In this section we analyze and compare the performance of our systems in different scenarios. We also compare our results with previous work.

### 4.1 Results Analysis

In Table 1 we report the results of our individual systems (one for each modern language) and the ensemble results. For individual experiments, Italian obtains the lowest average edit distance on all datasets.
For ensembles, we experimented with the four fusion methods described in the previous section, but report only the best performing one. Out of the four proposed fusion methods, the first two lead to similar results, that are superior to the other two. The best-performing ensemble uses the second fusion method, which assigns scores based on the rank in the n-best lists and the training accuracy for each individual system. We also tried applying the ensembles on language subsets (that is, not to take all modern languages into account at once). We investigated all combinations, and in the majority of cases using all modern languages lead to the highest performance among all ensembles.
In Table 2 we show an example of our systems' output n-best lists. This example illustrates how the ensemble can improve over the individual classifiers, by ranking the correct production higher than all the other systems. For all datasets we obtained performance improvements for proto-word reconstruction when we combined individual results using ensembles. As expected, the highest performance was obtained by the oracle classifiers. The results show the high potential of the ensemble methods for proto-word reconstruction.
The average edit distance of our best-performing ensemble, on Dataset 3, is 1.07, meaning that, on average, the proto-word reconstructions obtained by the system are a little more than one character different from the correct proto-words. Furthermore, the correct proto-word is listed among the 5-best list productions of our system in 70% of the cases (increasing to 74% for 10-best lists). These results are encouraging, having in mind the purpose of our system: to be a tool to be used by linguists (not to substitute the work of the experts).
Overall, we notice that the results are significantly better for Datasets 2 and 3 than for Dataset 1. One possible explanation is the nature of the dataset: while Datasets 2 and 3 are built based on the etymologies of the words (that is, the genetic relationships are taken into account), for Dataset 1 the cognacy decisions have been made based on an edit distance threshold between the words (Bouchard-Côté et al., 2007). To test this assumption, we ran an additional experiment, training the system on a subset of Dataset 3, having the same size as Dataset 1. The performance was close to that reported for Dataset 3 in Table 1, confirming that it is the nature of the dataset rather than its size that influences the results.

### 4.2 Additional Experiments: Neural Word Production

We performed additional experiments using a recurrent neural network (RNN) system with an encoder-decoder architecture instead of a CRF. RNNs have been proven useful in many applications, and are suitable for sequence labeling problems. However, they require large amounts of training data. In historical linguistics in general, and in the problem of proto-word reconstruction in particular, the resources are often scarce. Thus, we were interested to find out if RNNs can outperform the CRF system. We experimented with an encoder-decoder system with two long short-term memory (LSTM) layers, with stochastic gradient descent (SGD) optimization and a global attention mechanism (Luong et al.,

| Language | EDIT | $COV_1$ | $COV_5$ | $COV_{10}$ | MRR |
|---|---|---|---|---|---|
| **Italian** | 2.45 (0.29) | 0.14 | 0.32 | 0.35 | 0.22 |
| **Spanish** | 2.51 (0.29) | 0.15 | 0.21 | 0.23 | 0.18 |
| **Portuguese** | 2.61 (0.30) | 0.16 | 0.24 | 0.31 | 0.21 |
| **Ensemble** | **2.31 (0.27)** | **0.22** | **0.32** | **0.42** | **0.27** |
| **Oracle** | 1.76 (0.20) | 0.28 | 0.41 | 0.47 | 0.34 |

(a) Dataset 1 orthographic (Bouchard-Côté et al. (2007))

| Language | EDIT | $COV_1$ | $COV_5$ | $COV_{10}$ | MRR |
|---|---|---|---|---|---|
| **Italian** | 2.52 (0.29) | 0.16 | 0.28 | 0.32 | 0.21 |
| **Spanish** | 2.61 (0.30) | 0.12 | 0.22 | 0.25 | 0.17 |
| **Portuguese** | 2.95 (0.34) | 0.07 | 0.17 | 0.22 | 0.13 |
| **Ensemble** | **2.28 (0.26)** | **0.14** | **0.32** | **0.36** | **0.21** |
| **Oracle** | 1.93 (0.22) | 0.23 | 0.36 | 0.39 | 0.30 |

(b) Dataset 1 phonetic (Bouchard-Côté et al. (2007))

| Language | EDIT | $COV_1$ | $COV_5$ | $COV_{10}$ | MRR |
|---|---|---|---|---|---|
| **Italian** | 1.57 (0.24) | 0.25 | **0.52** | **0.55** | 0.37 |
| **Spanish** | 1.78 (0.27) | 0.22 | 0.35 | 0.39 | 0.28 |
| **Portuguese** | 1.76 (0.28) | 0.19 | 0.34 | 0.39 | 0.26 |
| **Romanian** | 2.12 (0.32) | 0.18 | 0.31 | 0.36 | 0.25 |
| **French** | 2.31 (0.35) | 0.13 | 0.24 | 0.30 | 0.18 |
| **Ensemble** | **1.55 (0.23)** | **0.29** | 0.49 | **0.55** | **0.38** |
| **Oracle** | 0.95 (0.14) | 0.43 | 0.60 | 0.66 | 0.51 |

(c) Dataset 2 (Reinheimer Ripeanu (2001))

| Language | EDIT | $COV_1$ | $COV_5$ | $COV_{10}$ | MRR |
|---|---|---|---|---|---|
| **Italian** | 1.12 (0.14) | 0.46 | 0.62 | 0.66 | 0.54 |
| **Spanish** | 1.31 (0.16) | 0.42 | 0.59 | 0.61 | 0.49 |
| **Portuguese** | 1.30 (0.16) | 0.41 | 0.58 | 0.61 | 0.49 |
| **Romanian** | 1.36 (0.16) | 0.43 | 0.61 | 0.64 | 0.51 |
| **French** | 1.52 (0.18) | 0.43 | 0.57 | 0.61 | 0.50 |
| **Ensemble** | **1.07 (0.13)** | **0.50** | **0.70** | **0.74** | **0.59** |
| **Oracle** | 0.65 (0.08) | 0.66 | 0.77 | 0.79 | 0.71 |

(d) Dataset 3 (Ciobanu and Dinu (2014a))

Table 1: Proto-word reconstruction for Latin proto-words. The first column indicates the modern language (or ensemble) that we used for training. For ensembles we report the results for the best performing ensemble. We report the average edit distance between the produced form and the correct form of the proto-word (**EDIT**) un-normalized (and in parantheses the normalized version), the coverage (**COV** for n $\in \{1, 5, 10\}$) and the mean reciprocal rank (**MRR**).

2015). We used the RNN implementation provided by TensorFlow (Luong et al., 2017). We experimented with Word2Vec character embeddings as features, and also with embeddings extracted from the aligned words (similar to the features used for the CRF system).

The results of the RNN system are lower than those of the CRF system (for example, $COV_{10}$ is about

| Language | Word | 5-best productions |
|---|---|---|
| **French** | voisin | vosinum, vosnum, vosine, vosinus, voiinum |
| **Italian** | vicino | vicinum, **vicinus**, vicenum, vicenus, vicnum |
| **Portuguese** | vizinho | vizinus, vizinum, **vicinus**, vizinium, vizinhum |
| **Spanish** | vecino | vecinum, vecinus, vicinum, vecenum, **vicinus** |
| **Romanian** | vecin | vicenus, vicenum, **vicinus**, vicinum, vecenus |
| **Ensemble** | all words | **vicinus**, vicinum, vicenus, vicenum, vecinus |

Table 2: Proto-word reconstruction example for the Latin proto-word **vicinus** (meaning **neighbor**). The correct productions are highlighted in bold.

0.10 for Dataset 1, while the CRF system obtains about 0.15 on the same dataset). The results are not included here, since they do not provide better performance. We leave for future work experimenting with additional features and setups, in order to compensate for the lack of training data.

### 4.3 Error Analysis

In this subsection we perform a brief analysis of the errors of our ensemble systems. The purpose of this step is to understand where the systems are not able to learn the correct production rules, in order to improve them in the future.

Looking at the incorrect productions that have one character different from the correct proto-word, we notice that sometimes the final vowel is mistaken. Most commonly, *um* instead of *us*: *serenum* instead of *serenus*, *cantum* instead of *cantus*, *novum* instead of *novus*. Another one-character mistake is, sometimes, failing to double a consonant. For example, *ll* or *ss*: *colapsus* instead of *collapsus*, *intervalum* instead of *intervallum*, *disociatio* instead of *dissociatio*, *esentia* instead of *essentia*.

For productions that have two characters different from the correct proto-word , we notice the following patterns in the incorrect productions: sometimes the character *f* is mistakenly obtained instead of *ph*: *asfaltus* instead of *asphaltus*, *eufonia* instead of *euphonia*, *diafragma* instead of *diaphragma*. Another interesting pattern is obtaining the suffix *a* instead of *us*: *citrina* instead of *citrinus*, *alba* instead of *albus*. When this occurs for adjectives, the productions are not incorrect words in Latin; we obtain the feminine form instead of the masculine.

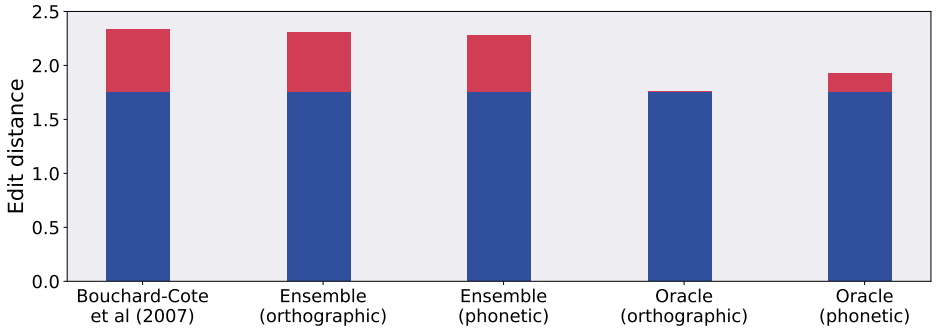


Figure 2: Comparison between previous work (Bouchard-Côté et al., 2007) and our ensembles and oracles on Dataset 1.

### 4.4 Comparison with Previous Work



Dataset 1 allows us a fair comparison with the state-of-the-art method proposed by Bouchard-Côté et al. (2007), as illustrated in Figure 2. On the same dataset, the authors report 2.34 average edit distance

between the produced words and the gold standard, when recreating Latin proto-words using the phonetic transcriptions. On the same dataset, we obtain better results on both the orthographic and the phonetic version of the dataset. The best of our systems (ensemble with fusion method based on the rank in the n-best lists and the training accuracy) obtains 2.28 average edit distance on the phonetic version of the dataset and 2.31 average edit distance on the orthographic version. The oracle obtains 1.76 average edit distance on the orthographic version of the dataset and 1.93 on the phonetic version.
The improvement of our systems is significant because we are able to obtain these results with less data than in previous experiments – which in historical linguistics is essential, as resources are most of the time scarce: the system is able to perform well even when not having the phonetic transcripts of the words.

## 5 Related Work

Researchers have been continuously interested in language derivation (Pagel et al., 2013). One of the main concerns in historical linguistics is language change across space and time (Rama and Borin, 2014). The first attempts to address this problem focused on regular sound correspondences to construct modern forms of the words, given a proto-language, or vice-versa. Traditionally, proto-language reconstruction has been investigated with comparative linguistics instruments (Campbell, 1998) and required a manual process and intensive work from linguistics and domain experts. Modern approaches impose the use and development of quantitative and computational methods in this field (McMahon et al., 2005; Heggarty, 2012; Atkinson, 2013), or even cross-disciplinary methods (such as those borrowed from biology).
Some of the early studies on partially automating proto-language reconstruction belong to Covington (1998) – investigating multiple alignment for historical comparison, and Kondrak (2002) – proposing, among others, methods for cognate alignment and identification.
More recent approaches addressed the complete automation of the reconstruction process. Oakes (2000) proposed two systems, Jakarta and Prague, that combined cover the steps of the comparative method for proto-language reconstruction. Another probabilistic approach belongs to Hall and Klein (2010), who obtained an average edit distance of 3.8 on the dataset of Romance languages proposed by Bouchard-Côté et al. (2009). Bouchard-Côté et al. (2013) used probabilistic models to trace language change in the Austronesian languages, based on a given phylogenetic tree.

## 6 Conclusions

In this paper, we proposed an automatic method for proto-word reconstruction. We applied our method on multiple datasets of Romance languages, in order to reconstruct Latin proto-words.
We used an approach based on sequence labeling and sequence alignment, combining the results of individual systems using ensembles. We obtained fairly good results (our best performance is 70% top 5 accuracy), and improved over previous work in this area. Our method has the advantage of requiring less input data than previous methods, and also accepting incomplete data, which is essential in historical linguistics, where resources are scarce.
We conclude that leveraging information from multiple modern languages, in ensemble systems, improves the performance on this task, producing n-best list of proto-words to be further analyzed by linguists and to aid in the process of comparative reconstruction for endangered or extinct languages.
As future work, we intend to refine the fusion methods for the ensemble classifiers – as the oracle results showed the high potential of the approach – and to evaluate our method on other datasets that cover more languages (for example, the Swadesh lists or the Austronesian basic vocabulary database (Greenhill et al., 2008)). We also intend to study rhotacism (Campbell, 1998), to investigate further ways of improving the performance of our CRF systems, and to enhance the RNN system even with little data available.

## Acknowledgments

We thank the anonymous reviewers for their helpful and constructive comments. The contribution of the authors to this paper is equal. Research supported by UEFISCDI, project number 53BG/2016.

## References


Waleed Ammar, Chris Dyer, and Noah A Smith. 2012. Transliteration by Sequence Labeling with Lattice Encodings and Reranking. In *Proceedings of the 4th Named Entity Workshop*, pages 66–70.

Raimo Anttila. 1989. *Historical and Comparative Linguistics*. Benjamins.

Quentin D. Atkinson. 2013. The Descent of Words. *Proceedings of the National Academy of Sciences*, 110(11):4159–4160.

Lisa Beinborn, Torsten Zesch, and Iryna Gurevych. 2013. Cognate Production using Character-based Machine Translation. In *Proceedings of IJCNLP 2013*, pages 883–891.

Aditya Bhargava and Grzegorz Kondrak. 2009. Multiple Word Alignment with Profile Hidden Markov Models. In *Proceedings of NAACL-HLT 2009, Companion Volume: Student Research Workshop and Doctoral Consortium*, pages 43–48.

Jean Charles Borda. 1781. Memoire sur les elections au scrutin. *Memoires de lAcademie Royale des Sciences de Paris pour l'Anne 1781*, pages 657–665.

Alexandre Bouchard-Côté, Percy Liang, Thomas Griffiths, and Dan Klein. 2007. A Probabilistic Approach to Diachronic Phonology. In *Proceedings of EMNLP 2007*, volume 12, pages 887–896.

Alexandre Bouchard-Côté, Thomas L. Griffiths, and Dan Klein. 2009. Improved Reconstruction of Protolanguage Word Forms. In *Proceedings of NAACL 2007*, volume 7, pages 65–73.

Alexandre Bouchard-Côté, David Hall, Thomas L. Griffiths, and Dan Klein. 2013. Automated Reconstruction of Ancient Languages Using Probabilistic Models of Sound Change. *Proceedings of the National Academy of Sciences*, 110(11):4224–4229.

Lyle Campbell. 1998. *Historical Linguistics. An Introduction*. MIT Press.

Alina Maria Ciobanu and Liviu P. Dinu. 2014a. Automatic Detection of Cognates Using Orthographic Alignment. In *Proceedings of ACL 2014, Volume 2: Short Papers*, pages 99–105.

Alina Maria Ciobanu and Liviu P. Dinu. 2014b. Building a Dataset of Multilingual Cognates for the Romanian Lexicon. In *Proceedings of LREC 2014*, pages 1038–1043.

Alina Maria Ciobanu and Liviu P. Dinu. 2015. Automatic Discrimination between Cognates and Borrowings. In *Proceedings of ACL 2015, Volume 2: Short Papers*, pages 431–437.

Alina Maria Ciobanu. 2016. Sequence Labeling for Cognate Production. In *Proceedings of KES 2016*, pages 1391–1399.

Michael A. Covington. 1998. Alignment of Multiple Languages for Historical Comparison. In *Proceedings of ACL 1998, Volume 1*, pages 275–279.

Surya Ganesh, Sree Harsha, Prasad Pingali, and Vasudeva Verma. 2008. Statistical Transliteration for Cross Language Information Retrieval Using HMM Alignment Model and CRF. In *Proceedings of CLIA 2008*, pages 42–47.

Simon J Greenhill, Robert Blust, and Russel D. Gray. 2008. The Austronesian Basic Vocabulary Database: From Bioinformatics to Lexomics. *Evolutionary Bioinformatics*, 4:271–283.

David Hall and Dan Klein. 2010. Finding Cognate Groups Using Phylogenies. In *Proceedings of ACL 2010*, pages 1030–1039.

Paul Heggarty. 2012. Beyond Lexicostatistics: How to Get More out of "Word List" Comparisons. In *Quantitative Approaches to Linguistic Diversity: Commemorating the Centenary of the Birth of Morris Swadesh*, pages 113–137. Benjamins.

Grzegorz Kondrak. 2002. *Algorithms for Language Reconstruction*. Ph.D. thesis, University of Toronto.

Ludmila I. Kuncheva, Christopher J. Whitaker, Catherine A. Shipp, and Robert P. W. Duin. 2003. Limits on the Majority Vote Accuracy in Classifier Fusion. *Pattern Analysis and Applications*, 6(1):22–31.





John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of ICML 2001*, pages 282–289.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of EMNLP 2015*, pages 1412–1421.

Minh-Thang Luong, Eugene Brevdo, and Rui Zhao. 2017. Neural Machine Translation (seq2seq) Tutorial. *https://github.com/tensorflow/nmt*.

Andrew Kachites McCallum. 2002. MALLET: A Machine Learning for Language Toolkit. http://mallet.cs.umass.edu.

April McMahon, Paul Heggarty, Robert McMahon, and Natalia Slaska. 2005. Swadesh Sublists and the Benefits of Borrowing: an Andean Case Study. *Transactions of the Philological Society*, 103(2):147–170.

Saul B. Needleman and Christian D. Wunsch. 1970. A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins. *Journal of Molecular Biology*, 48(3):443–453.

Michael P. Oakes. 2000. Computer Estimation of Vocabulary in a Protolanguage from Word Lists in Four Daughter Languages. *Journal of Quantitative Linguistics*, 7:233–243.

Mark Pagel, Quentin D Atkinson, Andreea S Calude, and Andrew Meade. 2013. Ultraconserved Words Point to Deep Language Ancestry across Eurasia. *Proceedings of the National Academy of Sciences*, 110(21):8471–8476.

Taraka Rama and Lars Borin. 2014. Comparative Evaluation of String Similarity Measures for Automatic Language Classification. In George K. Mikros and Jn Macutek, editors, *Sequences in Language and Text*. De Gruyter Mouton.

Sanda Reinheimer Ripeanu. 2001. *Lingvistica Romanica: Lexic, Morfologie, Fonetica*. Ed. All. Bucuresti.

Yulia Tsvetkov, Waleed Ammar, and Chris Dyer. 2015. Constraint-Based Models of Lexical Borrowing. In *Proceedings of NAACL-HLT 2015*, pages 598–608.