Cognate identification and alignment using practical orthographies

Michael Cysouw
Max Planck Institute for Evolutionary Anthropology, Leipzig
cysouw@eva.mpg.de

Hagen Jung
Max Planck Institute for Evolutionary Anthropology, Leipzig
jung@eva.mpg.de

Abstract

We use an iterative process of multi-gram alignment between associated words in different languages in an attempt to identify cognates. To maximise the amount of data, we use practical orthographies instead of consistently coded phonetic transcriptions. First results indicate that using practical orthographies can be useful, the more so when dealing with large amounts of data.

1 Introduction

The comparison of lexemes across languages is a powerful method to investigate the historical relations between languages. A central prerequisite for any interpretation of historical relatedness is to establish lexical cognates, i.e. lexemes in different languages that are of shared descend (in contrast to similarity by chance). If a pair of lexemes in two different languages stem from the same origin, this can be due to the fact that both languages derive from a common ancestor language, but it can also be caused by influence from one language on another (or influence on both language from a third language). To decide whether cognates are indicative of a common ancestor language ("vertical transmission") or due to language influence ("horizontal transmission") is a difficult problem with no shortcuts. We do not think that one kind of cognacy is more interesting that another. Both loans (be it from a substrate or a superstrate) and lexemes derived from a shared ancestor are indicative of the history of a language, and both should be acknowledged in the unravelling of linguistic (pre)history.

In this paper, we approach the identification of cognate lexemes on the basis of large parallel lexica between languages. This approach is an explicit attempt to reverse the "Swadesh-style" wordlist method. In the Swadesh-style approach, first meanings are selected that are assumed to be less prone to borrowing, then cognates are identified in those lists, and these cognates are then interpreted as indicative of shared descend. In contrast, we propose to first identify (possible) cognates among all available information, then divide these cognates into strata, and then interpret these strata in historical terms. (Because of limitations of space, we will only deal with the first step, the identification of cognates, in this paper.) This is of course exactly the route of the traditional historical-comparative approach to language comparison. However, we think that much can be gained by applying computational approaches to this approach.

A major problem arises when dealing with large quantities of lexical material from many different languages. In most cases it will be difficult (or very costly and time consuming in the least) to use coherent and consistent phonetic transcriptions of all available information. Even if we would have dictionaries with phonetic transcriptions for all languages that we are interested in, this would not necessarily help, as the details of phonetic transcription are normally not consistent across different authors. In this paper, we will therefore attempt to deal with unprocessed material in practical orthographies. This will of course pose problems for history-ridden orthographies like in English or French. However, we beleve that for most of the world's languages the practical

Proceedings of Ninth Meeting of the ACL Special Interest Group in Computational Morphology and Phonology, pages 109–116,
Prague, June 2007.

orthographies are not as inconsistent as those (because they are much younger) and might very well be useful for linguistic purposes.

In this paper, we will first discuss the data used in this investigation. Then we will describe the algorithm that we used to infer alignments between word pairs. Finally, we will discuss a few of the results using this algorithm on large wordlists in practical orthography.

2 Resources

In this study we used parallel wordlists that we extracted from the Intercontinental Dictionary Series (IDS) database, currently under development at the Max Planck Institute for Evolutionary Anthropology in Leipzig (see http://www.eva.mpg.de/lingua/files/ids.html for more information). The IDS wordlists contain more than thousand entries of basic words from each language, and many entries contain alternative wordforms. At this time, there are only a few basic transcription languages (English, French and Portuguese) and some Caucasian languages available. We choose some of them for the purpose of the present study and preprocessed the data. To compare languages, we chose only word pairs that were available and non-compound in both languages. For all words that occurred several times in the whole collection of a language, we accepted only one randomly choosen wordform and left out all others. We also deleted content in brackets or in between other special characters. If, after these preparation, a wordform is still longer than twelve UTF-8 characters, we disregard these for reasons of computational efficiency. After this, we are still left with a large number of about 900 word pairs for each pair of languages.

3 Alignment

An alignment of two words w_a and w_b is a bijective and maintained ordered one-to-one correspondence from all subsequences s_a of the word w_a with $w_a = concat(s_{a_1}, s_{a_2}, \ldots, s_{a_k})$ to all subsequences s_b of the word w_b with $w_b = concat(s_{b_1}, s_{b_2}, \ldots, s_{b_k})$. It is possible that one of the associated subsequences is the empty word ϵ. In general one may construct a distance measure from such a linked sequence of two given words by assigning a cost for each single link of the alignment. There are many such alignment/cost functions described in the literature, and they are often used to calculate a distance measure between two sequences of characters (Inkpen et al., 2005). A measurement regularly used for linguistic sequences is the Levenshtein distance, or a modifications of it. Other distance measures detect, for example, the longest common subsequences or the longest increasing subsequences.

It is our special interest to use multi-character mappings for calculating a distance between two words. Therefore, we adapt and extend the Levenshtein measurement. First, we allow for mapping of any arbitrary string length (not just strings of one character as in Levenshtein) and, second, we assign a continuous cost between 0 and 1 for every mapping.

Our algorithm consist basically of two steps. In the first step, all possible subsequence pairs between associated words are considered, and a cost function is extracted for every multi-gram pair from their co-occurrences in the whole wordlist. In a second step, this cost function is used to infer an alignment between whole words. On the basis of this alignment a new cost function is established for all multi-gram pairs. This second step can be iterated until the cost function stabilizes.

3.1 Cost of an multi-gram pair

For every pair of subsequences s_{a_i} and s_{b_j} we count the number of co-occurrences. The subsequences s_{a_i} and s_{b_j} co-occur when they are found in two associated words w_a and w_b from a language wordlist of two languages L_a and L_b. We then use a simple Dice coefficient as a cost function between all possible subsequences. For computational reasons, it is necessary to limit the size of the multi-grams considered. We decided to limit the multi-gram size to a number of maximally four UTF-8 characters. Still, in the first step of our algorithm, there is a very large set of such subsequence pairs because all possible combinations are considered. When an alignment is inferred in the iterative process, only the aligned subsequences are counted as co-occurrences, so the number of possible combinations is considerably lower. Further, to prevent low frequent co-occurrences to have a disproportional impact, we added an attestation threshold of 2% of the wordlist size for two subsequences to be accepted for the alignment process.

3.2 Alignment of words

An alignment of two words is a complete ordered linking of subsequences. We annotate it in the following way (vertical dashes delimit the subsequences; note that subsequences may be empty):

$(\quad | \, w \, | \, ool)(\text{щерс} \, | \, \text{ть} \, | \quad)$

There is a huge amount of possible combinations of aligned subsequences. On the basis of the cost function, a distance is established for every word pair alignment. The summation of all multi-gram mapping costs represents the distance of the alignment. Because we are dealing with multi-grams of variable length, alternative alignments of the same word pair will consist of a different number of subsequences. So, simple summation would lead to distances out of the range from 0 to 1. To counteract this, we normalized the word distance. We weighted each subsequence relative to the number of characters in the subsequence. For example, the mapping of w and ть in the example above would be multiplied by $\frac{3}{10}$, because w and ть have together 3 characters and the complete words have in total 10 characters.

To make use of efficient divide and conquer solving strategies and to get meaningful linguistic statements with the base of the calculated best alignments, we decided to look for a special subset of best alignments. As (Kondrak, 2002) pointed out, there are some situations in which the consideration of local alignment gets the required results. If only a part of a word aligning sequence is of high similarity then sometimes a linguistic justification of the whole word similarity is given. Those alignments contain the lowest cost multi-gram pairs, but are not necessarily of best similarity in total.

To illustrate the difference between local and global alignment, consider an example that shows different results, depending whether the total sum of multi-gram similarities is taken or the best local one. Look at the two words 'abc' and '$\alpha\beta\gamma$' and a part of its multi-gram cost function in Table 1. The summation of the costs would prefer alignment A_2, as can be seen in Table 2. But we prefer A_1, because it contains the subsequence pair $(ab \mid \alpha\beta)$ with the lowest cost.

multi-gram 1	multi-gram 2	cost
ab	$\alpha\beta$	0.1
bc	$\beta\gamma$	0.3
a	α	0.4
c	γ	0.8
$\vdots$	$\vdots$	$\vdots$

Table 1: Costs for constructed subsequence pairs (ordered by cost)

Index	Alignment	Distance
A_2	$(a \mid bc)(\alpha \mid \beta\gamma)$	$0.4 + 0.3 = 0.7$
A_1	$(ab \mid c)(\alpha\beta \mid \gamma)$	$0.1 + 0.8 = 0.9$
$\vdots$	$\vdots$	$\vdots$

Table 2: Alignments with distance

With these assumptions, we composed a fast and easy method to find the best alignment. We prefer alignments where some links are very good, but the rest might not be. We assume that words are more related to each other, if there are such highly rated pairs. This approach can also be found in other string based comparing methods like, for example, the Longest Common Increasing Subsequence method, which calculates the longest equal multi-gram and neglects the rest of the word. We first order all possible multi-gram mappings by their costs and pick the subsequence pair with the lowest cost. Starting from this mapping seed, we look for mappings for the rest of the word pair, both before and after the initial mapped subsequence. For both these suffixes and prefixes, we again search for the subsequence with the lowest cost. This process is re-applied until the whole words are mapped. If there is more than one optimal linking subsequence pair, then all possible alignments are considered. In this way, we do not restrict, in contrast to Kondrak, which position for the multi-gram mapping will be preferred for the local alignment. The algorithm runs in $O(n^6)$. It takes $O(n^4)$ time for all combinations of different multi-gram pairs within $O(n)$ steps in $O(n)$ iterations.

4 Experimental Evaluation

As mentioned above, we applied our model to some test data from the IDS database. For later analyses, we also constructed some random wordlists. With these we are able to say something about how significant our results are. To make these random wordlists we remap each word w_a from L_a to an arbitrarily chosen word w_b from collection L_b. This new mapped word was adjusted to the size of the originally associated word from L_b. The adjustment works by stretching or shrinking the new word to the required length by doubling the word several times and cutting of the overlaying head or tail afterwards. In this way, we controlled for word length and multi-gram frequencies. This randomization process was performed five times from L_a to L_b, and five the times from L_b to L_a, and the results were averaged over all these ten cases.

For the calculation process, we stored all lists in SQL tables. We first built a preprocessed working table with the lexemes from the languages to be compared, and afterwards we constructed the resulting tables that hold all the results:

- compare table: the word pairs, their alignments and alignment goodness;
- subsequence table: the subsequence pairs found and their co-occurrence coefficients;
- random compare table: pseudo random word pairs like the compare table;
- random subsequence table: the subsequence pairs found from random compare table.

Table 3 consists of the best alignments for word pairs of English and French after 30 iterations, and Table 4 shows the best alignments for the comparison of English and Hunzib (a Caucasian language). First note that our algorithm works independently of the orthography used. We do not assume that the same UTF-8 characters in the two languages are identical. The fact that ⟨c⟩ is mapped between English *clan* and French *clan* is a result of the statistical distribution of these characters in the two languages. This orthography-independence means that we can apply our algorithm without modifications to cyrillic scripts as shown with the English-Hunzib comparison. Second, we payed close attention to the fact that the word similarity values are comparable among different language comparisons. This means that it is highly significant that the highest word similarities between English and French are much higher than those between English and Hunzib (actually, the alignments between English and Hunzib are nonsensical, but more about that later). Further, our algorithm finds vowel-consonant multi-grams in some cases (e.g. see Table 5). As far as we can see, there are not linguistically meaningful and should be considered an artifact of our current approach. We hope to fine-tune the algorithm in the future to prevent this behavior.

Our method finds alignments, but also the subsequences in the alignments are of interest. The best mapped multi-grams between English and French are illustrated in Table 5. Strangely, the highest ranked ones are a few vowel+consonant bigrams, that occur not very often. Since the Dice coefficient depends on the size of the investigated collection, we assumed a minimum frequency of co-occurrences in each calculation step of 2% of the collection size (which is 20 cases in the English-French comparison). The high-ranked bigrams are all just above this threshold. Therefore, we might argue that all the bigrams from the top of the list are a side-effect of the collection size itself.

Following these bigrams are many one-to-one matches of all alphabetic characters except ⟨j,k,q,w,x,y,z⟩. These mappings are found without assuming any similarity based on the UTF-8 encoding of the characters. What we actually find here is a mapping for the orthography of the stratum of the French loan words in English. As can be seen in the histogram in Figure 1, the mapping between multi-grams falls off dramatically after these links.

<table>
<tr><th>English</th><th>French</th><th>Alignment</th><th>similarity</th></tr>
<tr><td>tribe,clan</td><td>tribu,clan</td><td>(| c | | l | | an |) (| c | | l | | an |)</td><td>0.955872</td></tr>
<tr><td>long</td><td>long</td><td>(| l | | on | | g |) (| l | | on | | g |)</td><td>0.925542</td></tr>
<tr><td>lion</td><td>lion</td><td>(| l | | i | | on |) (| l | | i | | on |)</td><td>0.916239</td></tr>
<tr><td>canoe</td><td>canoe,pirogue</td><td>(| c | | an | | o | | e |) (| c | | an | | o | | e |)</td><td>0.911236</td></tr>
<tr><td>famine</td><td>famine,disette</td><td>(| f | | a | | m | | in | | e |) (| f | | a | | m | | in | | e |)</td><td>0.910465</td></tr>
<tr><td>innocent</td><td>innocent</td><td>(| in | | n | | o | | c | | e | | n | | t |) (| in | | n | | o | | c | | e | | n | | t |)</td><td>0.908913</td></tr>
<tr><td>prison,jail</td><td>prison</td><td>(| p | | r | | i | | s | | on |) (| p | | r | | i | | s | | on |)</td><td>0.9089</td></tr>
<tr><td>poncho</td><td>poncho</td><td>(| p | | on | | c | | h | | o |) (| p | | on | | c | | h | | o |)</td><td>0.907496</td></tr>
<tr><td>sure,certain</td><td>sûr,certain</td><td>(| c | | e | | r | | t | | a | | in |) (| c | | e | | r | | t | | a | | in |)</td><td>0.905022</td></tr>
<tr><td>tapioca,manioc</td><td>manioc</td><td>(| m | | an | | i | | o | | c |) (| m | | an | | i | | o | | c |)</td><td>0.904811</td></tr>
<tr><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td></tr>
</table>

Table 3: English-French best rated alignments after 30 iterations

<table>
<tr><th>English</th><th>Hunzib</th><th>Alignment</th><th>similarity</th></tr>
<tr><td>jewel</td><td>жавгьар,йакъут</td><td>(| j | | e | | w | | e | | l |) (| ж | | а | в | гь | | а | | р |)</td><td>0.507094</td></tr>
<tr><td>see</td><td>нацIа</td><td>(| s | | e | | e |) (| н | | а | цI | а |)</td><td>0.489442</td></tr>
<tr><td>grease,fat</td><td>маъа</td><td>(g | r | | e | a | s | | e |) (| м | | а | | ъ | | а |)</td><td>0.464667</td></tr>
<tr><td>heaven</td><td>гIалжан</td><td>(| h | | e | | a | | v | | e | | n |) (г | I | | а | | л | | ж | | а | | н |)</td><td>0.445626</td></tr>
<tr><td>ocean</td><td>акеан</td><td>(| o | | c | | e | a | n |) (а | к | | е | | а | | н |)</td><td>0.419629</td></tr>
<tr><td>pocket</td><td>киса,жиби</td><td>(p | o | | c | | k | | e | t) (| к | | и | | с | | а |)</td><td>0.410143</td></tr>
<tr><td>sweep</td><td>лъалIа</td><td>(| s | w | e | | e | p) (л | ъ | | а | лI | а |)</td><td>0.395264</td></tr>
<tr><td>measure</td><td>маса</td><td>(| m | | e | a | s | ur | e |) (| м | | а | | с | | а |)</td><td>0.393806</td></tr>
<tr><td>flower</td><td>гьакI</td><td>(flo | w | | e | | r |) (| гь | | а | | к | I)</td><td>0.391867</td></tr>
<tr><td>rebuke,scold</td><td>акьа</td><td>(r | e | | b | | u | k | e |) (| а | | к | | ь | | а |)</td><td>0.387163</td></tr>
<tr><td>⋮</td><td>⋮</td><td>⋮</td><td>...</td></tr>
</table>

Table 4: English-Hunzib best rated alignments after 30 iterations

E	F	freq	dice
ar	ar	21	1
in	in	26	1
on	on	22	1
an	an	22	1
m	m	80	0.92786
n	n	188	0.92161
c	c	120	0.91815
p	p	78	0.91798
r	r	277	0.91665
f	f	35	0.90647
l	l	132	0.90534
v	v	26	0.90346
t	t	165	0.8719
b	b	44	0.86301
s	s	126	0.85915
d	d	66	0.82913
o	o	192	0.82325
e	e	417	0.81479
a	a	229	0.81367
g	g	34	0.79683
h	h	53	0.7856
i	i	183	0.75961
u	u	94	0.69546
⋮	⋮	⋮	⋮

Table 5: Best English (E) and French (F) multi-gram mappings after 30 iterations.

The character-independence of our method is illustrated by the character mapping between English and Russian in Table 6. Shown in the table are only the highest ranked orthographic mappings. Again we see an almost complete alphabetic linkage, probably caused by the French loanwords shared by both English and Russian.

With this approach, we are also able to find some vestiges of sound changes, as illustrated by the character mapping between Spanish and Portuguese in Table 7. Shown here are only the highest ranked *non-identical* multi-grams. The dice coefficients of the pairs ⟨h⟩ − ⟨ll⟩, ⟨f⟩ − ⟨h⟩ show the results of sound changes that were dramatically enough to be represented in the orthography. The pairs ⟨ç⟩ − ⟨z⟩ and ⟨n⟩ − ⟨ñ⟩ show difference in orthographic convention (though the best pair should have been ⟨nh⟩ − ⟨ñ⟩).

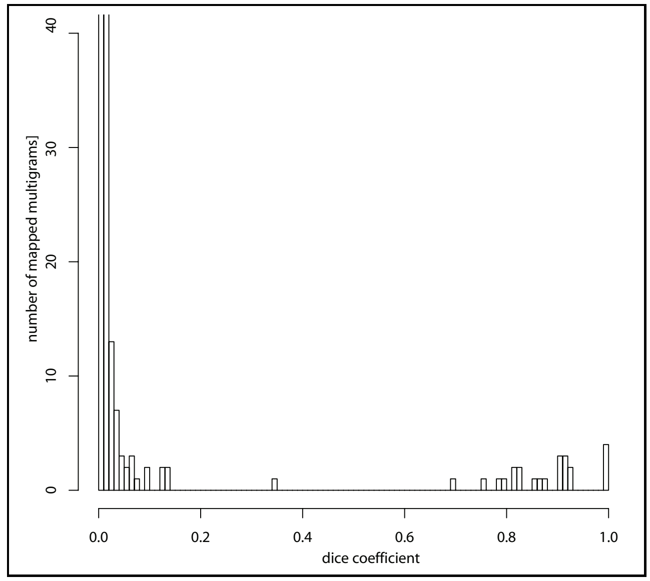

Figure 1: Histogram of dice-coefficients for English-French multi-gram mappings.

E	R	freq	dice
r	р	184	0.88874745
n	н	115	0.8461936
l	л	104	0.79646295
s	с	114	0.7927922
t	т	165	0.7701921
m	м	47	0.7699933
o	о	184	0.7510106
k	ть	21	0.74458015
p	п	50	0.7388723
i	и	102	0.7034591
a	а	221	0.6866478
u	у	40	0.6449104
c	к	77	0.6251676
e	е	219	0.59066784
b	б	32	0.525643
w	в	46	0.46787763
d	д	42	0.381996
⋮	⋮	⋮	⋮

Table 6: Best English (E) and Russian (R) multi-gram mappings after 30 iterations.

P	S	freq	dice
⋮	⋮	⋮	⋮
ç	z	20	0.6316202
h	ll	20	0.4552776
f	h	34	0.43381172
n	ñ	24	0.37720457
ã	n	33	0.31106696
h	h	23	0.23646937
v	b	32	0.2165933
t	h	29	0.2127131
z	c	24	0.15424858
o	e	305	0.12838262
⋮	⋮	⋮	⋮

Table 7: Spanish (S) and Portuguese (P) multi-gram mappings after 30 iterations. Only the highest ranking non-identical mappings are shown

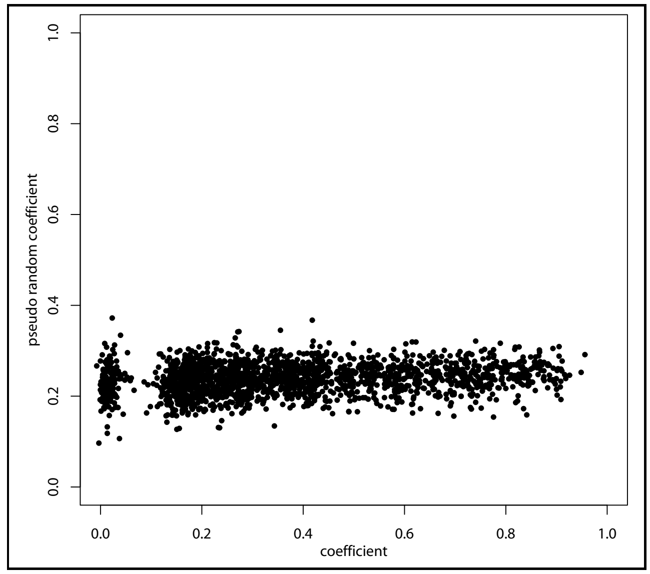

Figure 2: English-French similarities for word alignments plotted against the similarities with random language entries.

A promising indicator for cognate identification is the comparison of word alignment similarities with the similarities between randomly associated word pairs. We generated pseudo random word pairs as described above. Therefore we caluclate for each word from one language one coeffiecent value for the linkage with the assocciated word and a second avarage value for the linkage with some random words. In Figure 2 we plot these two values for all words of English and all words of French (after 30 iterations) against each other. Each dot represents a word. The x-axis shows the similarity coefficient between the real words and the y-axis shows the similarity coefficient from the comparison with the pseudo random words. As can be seen, many of the actual similarities are more to the right of the $y = x$ line indicating more than chance frequency similarity.

In contrast, in comparing English with Hunzib in Figure 3 there is only a slight tendency of stretching of the scatterplot. So one could conclude that English and Hunzib have probably no cognates at all, although there are some strongly related word pairs. However, some slight stretching will always be seen, because of the usage of an algorithm with iterations. Such a process will always strengthen some random tendencies.

The iterative process is illustrated in Figure 4. Shown here are the alignment similarities for all word pairs between French and Portuguese. After the first round of alignment, there is only a slight stretch in the scatterplot. Already after the second iteration, the plot is stretched strongly. In the further iterations the situation changes only slightly. Apparently, two rounds of alignment and reassignment of the cost function suffice for convergence.

5 Conclusion

The big advantage of using original orthographies in the study of linguistic relationships is that much more information is readily available. Because of the wealth of available data, we can use computational approaches for the comparison of wordlists. In principle, the kind of approach that we have sketched out in this paper can just as well be used for the comparison of complete dictionaries. The comparison of real wordlists with randomly shuffled wordlists indicated that even on purely statistical grounds it might be possible to separate meaningful alignments from random alignments.

The most promising result of our investigation is

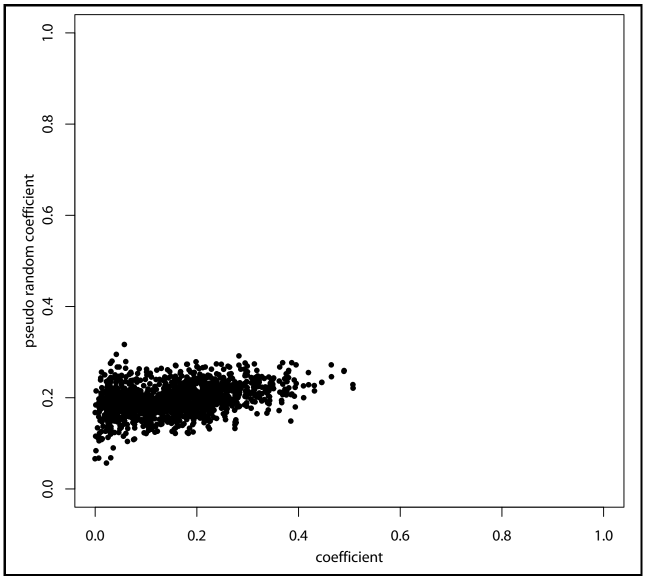

Figure 3: English-Hunzib similarities for word alignments plotted against the similarities with random language entries.

that we were able to find cognates even without any knowledge about the orthographic conventions used in the languages that were compared. In the comparison English-French and English-Russian there appear to be many French loanwords among the well-aligned wordpairs. If this impresion holds, we are in fact only able to infer the stratum of French influence in European languages. An interesting next step would then be to redo the analyses after removing this stratum from the data and look for deeper strata in the lexicon. As shown by the Spanish-Portuguese comparison, sound changes can be picked up by our approach as long as the changes have left a trace in the orthography.

References

Diana Inkpen, Oana Frunza, and Grzegorz Kondrak. 2005. Automatic identification of cognates and false friends in french and english. In *RANLP-2005, Bulgaria*, pages 251–257, September.

Grzegorz Kondrak. 2002. *Algorithms for language reconstruction*. Ph.D. thesis, University of Toronto.

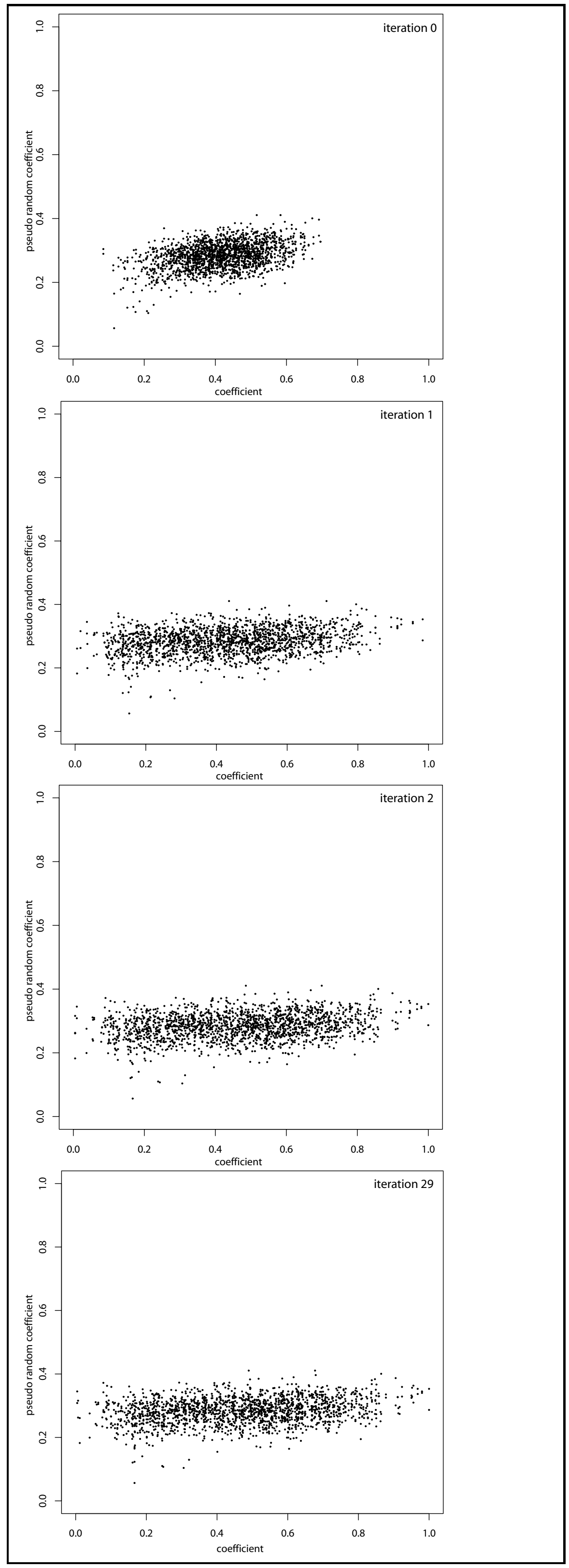

Figure 4: Plots of four iterations after 1, 2, 3 and 30 rounds of the French-Portuguese comparison. The coefficients are plotted against coefficients that were build with randomized language entries.