Induction of Cross-Language Affix and Letter Sequence Correspondence

Ari Rappoport
Institute of Computer Science
The Hebrew University
www.cs.huji.ac.il/~arir

Tsahi Levent-Levi
Institute of Computer Science
The Hebrew University

Abstract

We introduce the problem of explicit modeling of form relationships between words in different languages, focusing here on languages having an alphabetic writing system and affixal morphology. We present an algorithm that learns the cross-language correspondence between affixes and letter sequences. The algorithm does not assume prior knowledge of affixes in any of the languages, using only a simple single letter correspondence as seed. Results are given for the English-Spanish language pair.

1 Introduction

Studying various relationships between languages is a central task in computational linguistics, with many application areas. In this paper we introduce the problem of induction of form relationships between words in different languages. More specifically, we focus on languages having an alphabetic writing system and affixal morphology, and we construct a model for the cross-language correspondence between letter sequences and between affixes. Since the writing system is alphabetic, letter sequences are highly informative regarding sound sequences as well.

Concretely, the model is designed to answer the following question: what are the affixes and letter sequences in one language that correspond frequently to similar entities in another language? Such a model has obvious applications to the construction of learning materials in language education and to statistical machine translation.

The input to our algorithm consists of word pairs from two languages, a sizeable fraction of which is assumed to be related graphemically and affixally. The algorithm has three main stages. First, an alignment between the word pairs is computed by an EM algorithm that uses an edit distance metric based on an increasingly refined individual letter correspondence cost function. Second, affix pair candidates are discovered and ranked, based on a language independent abstract model of affixal morphology. Third, letter sequences that correspond productively in the two languages are discovered and ranked by EM iterations that use a cost function based on the discovered affixes and on compatibility of alignments.

The affix learning part of the algorithm is totally unsupervised, in that we do not assume knowledge of affixes in any of the single languages involved. The letter sequence learning part utilizes a simple initial correspondence between individual letters, and the rest of its operation is unsupervised.

We believe that this is the first paper that explicitly addresses cross-language morphology, and the first that presents a comprehensive inter-language word form correspondence model that combines morphology and letter sequences.

Section 2 motivates the problem and defines it in detail. In Section 3 we discuss relevant previous work. The algorithm is presented in Section 4, and results for English-Spanish in Section 5.

2 Problem Motivation and Definition

We would like to discover characteristics of word form correspondence between languages. In this section we discuss what exactly this means and why it is useful.

Word form. Word forms have at least three different aspects: sound, writing system, and internal structure, corresponding to the linguistics fields of phonology, orthography and morphology. When the writing system is phonetically based, the written form of a word is highly informative of how the word sounds. Individual writing units are referred to as graphemes.

Morphology studies the internal structure of words when viewed as comprised of semantics carrying components. Morphological units can be classified into two general classes, stems (or roots) and bound morphemes, which combine to create words using various kinds of operators. The linear affixing operator combines stems and bound morphemes (affixes) using linear ordering with possible fusion effects, usually at the seams.

Word form correspondence. In this paper we study cross-language word form correspondence. We should first ask why there should be any relationship at all between word forms in different languages. There are at least two factors that create such relationships. First, languages may share a common ancestor. Second, languages may borrow words, writing systems and even morphological operators from each other. Note that usage of proper names can be viewed as a kind of borrowing. In both cases form relationships are accompanied by semantic relatedness. Words that possess a degree of similarity of form and meaning are usually termed *cognates*.

Our goal in examining word forms in different languages is to identify correspondence phenomena that could be useful for certain applications. These would usually be correspondence similarities that are common to many word pairs.

Problem statement for the present paper. For reasons of paper length, we focus here on languages having the following two characteristics. First, we assume an alphabetic writing system. This implies that grapheme correspondences will be highly informative of sound correspondences as well. From now on we will use the term 'letter' instead of 'grapheme'. Second, we assume linear affixal morphology (prefixing and suffixing), which is an extremely frequent morphological operator in many languages.

We address the two fundamental word form entities in languages that obey those assumptions: affixes and letter sequences. Our goal is to discover frequent cross-language pairs of those entities and quantify the correspondence. Pairing of letter sequences is expected to be mostly due to regular sound transformations and spelling conventions. Pairing of affixes could be due to morphological principles – predictable relationships between the affixing operators (their form and meaning) – or, again, due to sound transformations and spelling.

The input to the algorithm consists of a set of ordered pairs of words, one from each language. We do not assume that all input word pairs exhibit the correspondence relationships of interest, but obviously the quality of results will depend on the fraction of the pair set that does exhibit them. A particular word may participate in more than a single pair. As explained above, the relationships of interest to us in this paper usually imply semantic affinity between the words; hence, a suitable pair set can be generated by selecting word pairs that are possible translations of each other. Practical ways to obtain such pairs are using a bilingual dictionary or a word aligned parallel corpus. We had used the former, which implies that we addressed only derivational, not inflectional, morphology. Using a dictionary provides a kind of semantic supervision that allows us to focus on the desired form relationships.

We also assume that the algorithm is provided with a prototypical individual letter mapping as seed. Such a mapping is trivial to obtain in virtually all practical situations, either because both languages utilize the same alphabet or by using a manually prepared, coarse alphabet mapping (e.g., anybody even shallowly familiar with Cyrillic or Semitic scripts can prepare such a mapping in just a few minutes.)

We do not assume knowledge of affixes in any of the languages. Our algorithm is thus fully unsupervised in terms of morphology and very weakly seeded in term of orthography.

Motivating applications. There are two main applications that motivate our research. In second language education, a major challenge for adult learners is the high memory load due to the huge number of lexical items in a language. Item memorization is known to be greatly assisted by tying items with existing knowledge (Matlin02). When learning a second language lexicon, it is beneficial to consciously note similarities between new and known words. Discovering and explaining such similarities automatically would help teachers in preparing reliable study materials, and learners in remembering words.

Recognition of familiar components also helps learners when encountering previously unseen words. For example, suppose an English speaker who learns Spanish and sees the word 'parcial-

mente'. A word form correspondence model would tell her that 'mente' is an affix strongly corresponding to the English 'ly', and that the letter pair 'ci' often corresponds to the English 'ti'. The model thus enables guessing or recalling the English word 'partially'.

Our model could also warn the learner of cognates that are possibly false, by recognizing similar words that are not paired in the dictionary.

A second application area is machine translation. Both cognate identification (Kondrak et al 03) and morphological information in one of the languages (Niessen00) have been proven useful in statistical machine translation.

3 Previous Work

Cross-language models for phonology and orthography have been developed for back-transliteration in cross-lingual information retrieval (CLIR), mostly from Japanese and Chinese to English. (Knight98) uses a series of weighted finite state transducers, each focusing on a particular mapping. (Lin02) uses minimal edit distance with a 'confusion matrix' that models phonetic similarity. (Li04, Bilac04) generalize using the sequence alignment algorithm presented in (Brill00) for spelling correction. (Bilac04) explicitly separates the phonemic and graphemic models. None of that work addresses morphology and in all of it grapheme and phoneme correspondence is only a transient tool which is not studied on its own. (Mueller05) explicitly models phonological similarities between related languages, but does not address morphology and orthography.

Cognate identification has been studied in computational historical linguistics. (Covington96, Kondrak03a) use a fixed, manually determined single entity mapping. (Kondrak03b) generalizes to letter sequences based on the algorithm in (Melamed97). The results are good for the historical linguistics application. However, morphology is not addressed, and the sequence correspondence model is less powerful than that employed in the back-transliteration and spelling correction literature. In addition, all effects that occur at word endings, including suffixes, are completely ignored. (Mackay05) presents good results for cognate identification using a word similarity measure based on pair hidden Markov models. Again, morphology was not modeled explicitly.

A nice application for cross-language morphology is (Schulz04), which acquires a Spanish medical lexicon from a Portuguese seed lexicon using a manually prepared table of 842 Spanish affixes.

Unsupervised learning of affixal morphology in a single language is a heavily researched problem. (Medina00) studies several methods, including the squares method we use in Section 4. (Goldsmith01) presents an impressive system that searches for 'signatures', which can be viewed as generalized squares. (Creutz04) presents a very general method that excels at dealing with highly inflected languages. (Wicentowsky04) deals with inflectional and irregular morphology by using semantic similarity between stem and stem+affix, also addressing stem-affix fusion effects. None of these papers deals with cross-language morphology.

4 The Algorithm

Overview. Letter sequences and affixes are different entities exhibiting different correspondence phenomena, hence are addressed at separate stages. The result of addressing one will assist us in addressing the other.

The fundamental tool that we use to discover correspondence effects is alignment of the two words in a pair. Stage 1 of the algorithm creates an alignment using the given coarse individual letter mapping, which is simultaneously improved to a much more accurate one.

Stage 2 discovers affix pairs using a general language independent affixal morphology model.

In stage 3 we utilize the improved individual letter relation from stage 1 and the affix pairs discovered in stage 2 to create a general letter sequence mapping, again using word alignments. In the following we describe in detail each of these stages.

Initial alignment. The main goal of stage 1 is to align the letters of each word pair. This is done by a standard minimal edit distance algorithm, efficiently implemented using dynamic programming (Gusfield97, Ristad98). We use the standard edit distance operations of replace, insert and delete. The letter mapping given as input defines a cost matrix where replacement of corresponding letters has a low (0) cost and of all others a high (1) cost. The cost of insert and delete is arbitrarily set to be the same as that of replacing non-identical letters. We use a hash table rather than a matrix, to prepare for later stages of the algorithm.

When the correspondence between the languages is very high, this initial alignment can

already provide acceptable results for the next stage. However, in order to increase the accuracy of the alignment we now refine the letter cost matrix by employing an EM algorithm that iteratively updates the cost matrix using the current alignment and computes an improved alignment based on the updated cost matrix (Brill00, Lin02, Li04, Bilac04). The cost of mapping a letter K to a letter L is updated to be proportional to the count of this mapping in all of the current alignments divided by the total number of mappings of the letter K.

Affix pairs. The computed letter alignment assists us in addressing affixes. Recall that we possess no knowledge of affixes; hence, we need to discover not only pairing of affixes, but the participating affixes as well. Our algorithm discovers affixes and their pairing simultaneously. It is inspired by the squares algorithm for affix learning in a single language (Medina00)[1].

The squares method assumes that affixes generally combine with very many stems, and that stems are generally utilized more than once. These assumptions are due to a functional view of affixal morphology as a process whose goal is to create a large number of word forms using fewer parameters. A stem that combines with an affix is quite likely to also appear alone, so the empty affix is allowed.

We first review the method as it is used in a single language. Given a word W=AB (where A and B are non-empty letter sequences), our task is to measure how likely it is for B to be a suffix (prefix learning is similar.) We refer to AB as a *segmentation* of W, using a hyphen to show segmentations of concrete words. Define a *square* to be four words (including W) of the forms W=AB, U=AD, V=CB, and Y=CD (one of the letter sequences C, D is allowed to be empty.)

Such a square might attest that B, D are suffixes and that A, C are stems. However, we must be careful: it might also attest that B, D are stems and A, C are prefixes. A square attests for a segmentation, not for a particular labeling of its components.

As an example, if W is 'talking', a possible square is {talk-ing, hold-ing, talk-s, hold-s} where A=talk, B=ing, C=hold, and D=s. Another possible square is {talk-ing, danc-ing, talk-ed, danc-ed}, where A=talk, B=ing, C=danc, and D=ed. This demonstrates a drawback of the method, namely its sensitivity to spelling; C with the empty suffix is written 'dance', not 'danc'. The four words {talking, dancing, talk, dance} do not form a square.

We now count the number of squares in which B appears. If this number is relatively large (which needs to be precisely defined), we have a strong evidence that B is a suffix or a stem. We can distinguish between these two cases using the number of *witnesses* – actual words in which B appears.

We generalize the squares method to the discovery of cross-language affix pairs, as follows. We now use W to denote not a single word but a word pair W1:W2. B does not denote a suffix candidate but a suffix pair candidate, B1:B2, and similarly for D. A and C denote stem pair candidates A1:A2 and C1:C2, respectively.

We now define a key concept. Given a word pair W=W1:W2 aligned under an alignment T, two segmentations W1=A1B1 and W2=A2B2 are said to be *compatible* if no alignment line of T connects a subset of A1 to a subset of B2 or a subset of A2 to a subset of B1. This definition is also applicable to alignments between letter sequences.

We now impose our key requirement: for all of the words involved in the cross-lingual square, their segmentations into two parts must be compatible under the alignment computed at stage 1.

For example, consider the English-Spanish word pair affirmation : afirmacion. The segmentation affirma-tion : afirma-cion is attested by the square

- A1B1 : A2B2 = affirma-tion : afirma-cion
- A1D1 : A2D2 = affirma-tively : afirma-tivamente
- C1B1 : C2B2 = coopera-tion : coopera-cion
- C1D1 : C2D2 = coopera-tively : coopera-tivamente

assuming that the appropriate parts are aligned. Note that 'tively' is comprised of two smaller affixes, but the squares method legitimately considers it an affix by itself. Note also that since all of A1, A2, C1 and C2 end with the same letter, that letter can be moved to the beginning of B1, B2, D1, D2 to produce a different square (affirm-ation : afirm-acion, etc.) from the same four word pairs.

Since we have no initial reason to favor a particular affix candidate over another, and since the total computational cost is not prohibitive, we

[1] (Medina00) attributes the algorithm to Joseph Greenberg.

now simply count the number of attesting squares for *all* possible compatible segmentations of all word pairs, and sort the list according to the number of witnesses. To reduce noise, we remove affix candidates for which the absolute number of witnesses or squares is small (e.g., ten.)

Letter sequences. The third and last stage of the algorithm discovers letter sequences that correspond frequently. This is again done by an edit distance algorithm, generalizing that of stage 1 so that sequences longer than a single letter can be replaced, inserted or deleted. In order to reduce noise, prior to that we remove word pairs whose stems are very different. Those are identified by comparing their edit distance costs, which should hence be normalized according to length (of the longer stem in a pair.) Note that accuracy is increased by considering only stems: affix pairs might be very different, thus might increase edit distance cost even when the stems do exhibit good sequence pairing effects.

When generalizing the edit distance algorithm, we need to specify which letter sequences will be considered, because it does not make sense to consider all possible mappings of all subsets to all possible subsets – the number of different such pairs will be too large to show any meaningful statistics.

The letter sequences considered were obtained by 'fattening' the lines in alignments yielding minimal edit distances, using an EM algorithm as done in (Brill00, Bilac04, Li04). The details of the algorithm can be found in these papers. The most important step, line fattening, is done as follows. We examine all alignment lines, each connecting two letter sequences (initially, of length 1.) We unite those sequences with adjacent sequences in all ways that are compatible with the alignment, and add the new sequences to the cost function to be used in the next EM iteration.

If we kept letter sequence pairs that are not frequent in the cost function, they would distort the counts of more frequent letter sequences with which they partially overlap. We thus need to retain only some of the sequence pairs discovered. We have experimented with several ways to do that, all yielding quite similar results. For the results presented in this paper, we used the idea that sequences that clearly map to specific sequences are more important to our model than sequences that 'fuzzily' map to many sequences. To quantify this approach, for each language-1 sequence we sorted the corresponding language-2 sequences according to count, and removed pairs in which the language-2 item was responsible for only a small percentage of the total (we used a threshold of 0.05). We further removed sequence pairs whose absolute counts are low.

Discussion. We deal with affixes before sequences because, as we have seen, identification of affixes helps us in identifying sequences, while the opposite order actually hurts us: sequences sometimes contain letters from both stem and affix, which invalidates squares that are otherwise valid.

It may be asked why the squares stage is needed at all – perhaps affixes would be discovered anyway as sequences in stage 3. Our assumption was that affixes are best discovered using properties resulting from their very nature. We have experimented with the option of removing stage 2 and discovering affixes as letter sequences in stage 3, and verified that it gives markedly lower quality results. Even the very frequent pair -ly:-mente was not signaled out, because its count was lowered by those of the pairs -ly:-ente, -ly:nte, -y:-te, etc.

5 Results

We have run the algorithm on several language pairs using affixal morphology and the Latin alphabet: English vs. Spanish, Portuguese and Italian, and Spanish vs. Portuguese. All of them are related both historically and through borrowing (obviously at varying degrees), so we expect relatively many correspondence phenomena. Testing results for one of these pairs, English – Spanish, are presented in this section.

The input word pair set was created from a bilingual dictionary (Freelang04) by taking all translations of single English words to single Spanish words, generating about 13,000 word pairs.

Individual letter mapping. The cost matrix after EM convergence (25 iterations) exhibits the following phenomena (e:s (c) denotes that the final cost of replacing the English letter e by the Spanish letter s is c): (1) English letters mostly map to identical Spanish letters, apart from letters that Spanish does not make use of like k and w; (2) some English vowels map frequently to some Spanish vowels: y maps almost exclusively to i (0.01), e:a (0.47) is highly productive, e:o (0.98), i:e (0.97), e:o (0.98); (3) some English consonants map to different Spanish ones: t:c

(0.89) (due to an affix, -tion:-cion); m:n (0.44) is highly frequent; b:v(0.80); x:j (0.78), x:s(0.94); w always maps to v; j:y (0.11); (4) h usually disappears, h:NULL (0.13); and (5) inserted Spanish letters include the vowels o, e, a and i, at that order, where o overwhelms the others. The English o maps exclusively to the Spanish o and not to other vowels.

Affixes. Table 1 shows some of the conspicuous affix pairs discovered by the algorithm. We show both the number of witnesses and of squares.

The table shows many interesting correspondence phenomena. However, discussing those at depth from a linguistic point of view is out of the scope of this paper. Some notes: (1) some of the most frequent affix pairs are not that close orthographically: -ity:-idad, -ness:- (nouns), -ate:-ar (verbs), -ly-:-mente (adverbs), -al:-o (adjectives), so will not necessarily be found using ordinary edit distance methods; (2) some affixes are ranked high both with and without a letter that they favor when attaching to a stem: -ation:-acion, -ate:-ar; (3) some English suffixes map strongly to several Spanish ones: -er:-o, -er:-ador.

Recall that the table cannot include inflectional affixes, since our input was taken from a bilingual dictionary, not from a text corpus.

Letter sequences. Table 2 shows some nice pairings, stemming from all three expected phenomena: st-:est- (due to phonology), ph:f, th:t, ll:l (due to orthography), and tion:cion, tia:cia (due to morphology: affixes located in the middle of words.)

Such affix and letter sequence pairing results can clearly be useful for English speakers learning Spanish (and vice versa), for remembering words by associating them to known ones, for avoidance of spelling mistakes, and for analyzing previously unseen words.

Evaluation. An unsupervised learning model can be evaluated on the strength of the phenomena that it discovers, on its predictive power for unseen data, or by comparing its data analysis results with results obtained using other means. We have performed all three evaluations.

For evaluating the discovered phenomena, a repository of known phenomena is needed. The only such repository of which we are aware are language learning texts. Unfortunately, the phenomena these present are limited to the few most conspicuous pairs (e.g., -ly:-mente, -ity:-idad, ph:f), all of which are easily discovered by our model. The next best thing are studies that present data of a single language. We took the affix information given in a recent, highly detailed, corpus based English grammar (Biber99), and compared it manually to ours. Of the 35 most productive affixes, our model finds 27. Careful study of the word pair list showed that the remaining 8 (-ment, -ship, -age, -ful, -less, -en, dis-, mis-) indeed do not map to Spanish ones frequently. Note that some of those are indeed extremely frequent inside English yet do not correspond significantly with any Spanish affix.

As a second test, we took a comprehensive English-Spanish dictionary (Collins), selected 10 pages at random (out of 680), studied them, and listed the prominent word form phenomena (85). All but one (the verbal suffix in seduce:seducir) were found by our model.

The numbers reported above for the two tests are recall numbers. To evaluate affix precision, we have manually graded the top 100 affix pairs (as sorted at the end of stage 2 of the algorithm.) 8 of those were clearly not affixes; however, 3 of the 8 (-t:-te, –t:-to, -ve:-vo) were important phonological phenomena that should indeed appear in our final model. Of the remaining 92, 15 were valid but ‘duplicates’ in the sense of being substrings of other affixes (e.g., -ly:-mente, -ly:-emente.) In the next 50 pairs, only 6 were clearly not affixes. Note that by their very definition, we should not expect the number of frequent derivational affixes to be very large, so there is not much point in looking further down the list. Nonetheless, inspection of the rest of the list reveals that it is not dominated by noise but by duplicates, with many specialized, less frequent affixes (e.g., -graphy:-grafia) being discovered.

Regarding letter sequences, precision was very high: of the 38 different pairs discovered, only one (hr:r) was not regular, and there were 11 duplicates. Recall was impressive, but harder to verify due to the lack of standards. We found only one (not very frequent) pair that was not discovered (-sp:-esp).

To evaluate the model on its data analysis capability, we took out 100 word pairs at random, trained the model without them, analyzed them using the final cost function, and compared with prominent phenomena noted manually (again, we had to grade manually due to the lack of a gold standard.) The model identified those prominent phenomena (including a total lack thereof) in 91 of the pairs. Notable failures included the pairs superscribe : sobrescribir and coded : codificado, where none of the prefixes and suffixes were

identified. Some successful examples are listed below (affixes are denoted by [], sequences by <>, and insert by _: or :_):

installation : instalacion. <ll:l>, [ation:acion]
volution : circonvolucion. _:c, _:i, _:r, _:c, _:o, _:n, [tion:cion]
intelligibility : inteligibilidad. [in:in], <ll:l>, [ity:idad]
sapper : zapador. <s:z>, <pp:p>, [er:ador]
harpist : arpista. <h:_>, [ist:ista]
pathologist : patologo. <th:t>, [ist:o]
elongate : prolongar. [te:r]
industrialize: industrializar. [in:in], <ial>, [e:ar]
demographic : demografico. <ph:f>, [ic:ico]
gynecological :ginecologico. <yn:in>, [ical:ico]
peeled : pelado. [ed:ado]

The third and final evaluation method is to compare the model's results with results obtained using other means. We are not aware of any data bank in which cross-language affix or letter sequence correspondences are explicitly tagged, so we had used a relatively simple algorithm as a baseline: We invoked the squares method for each language independently, ending up with affix candidates. For every word pair E:S, if E contains an affix candidate C and S contains an affix candidate D, we increment the count of the candidate affix pair C:D. Finally, we sort the candidates according to their count.

Baseline recall is obviously as good as in our algorithm (it produces a superset), but precision is so bad so as to render the baseline method useless: out of the first 100, only 19 were affixes, the rest being made up of noise and badly segmented 'duplicates'.

In summary, the results are good, but gold standards are needed for a more consistent evaluation of different cross-language word form algorithms. Results for the other language pairs were overall good as well.

6 Discussion

We have introduced the problem of cross-language modeling of word forms, presented an algorithm for addressing affixal morphology and letter sequences, and described good results on English-Spanish dictionary word pairs.

Natural directions for future work on the model include: (1) test the algorithm on more language pairs, including languages utilizing non-Latin alphabets; (2) modify the input model to assume that single language affixes are known; (3) address additional morphological operators, such as templatic morphology; (4) address phonology directly instead of indirectly; (5) use pairs acquired from a parallel corpus rather than a dictionary, to address inflectional morphology and to see how the algorithm performs with more noisy data; (6) extend the algorithm to other types of writing systems; (7) examine more sophisticated affix discovery algorithms, such as (Goldsmith01); and (8) improve the evaluation methodology.

There are many possible applications of the model: (1) for statistical machine translation; (2) for computational historical linguistics; (3) for CLIR back-transliteration; (4) for constructing learning materials and word memorization methods in second language education; and (5) for improving word form learning algorithms inside a single language.

The length and diversity of the lists above provide an indication of the benefit and importance of cross-language word form modeling in computational linguistics and its application areas.

References

Biber Douglas, 1999. Longman Grammar of Spoken and Written English. (Pages 320, 399, 530, 539.)

Bilac Slaven, Tanaka Hozumi, 2004. A Hybrid Back-Transliteration System for Japanese. COLING 2004.

Brill Eric, Moore Robert, 2000. An Improved Error Model for Noisy Channel Spelling Correction. ACL 2000.

Covington Michael A, 1996. An Algorithm to Align Words for Historical Comparison. Comput. Ling., 22(4):481—496.

Creutz Mathias, Lugas Krista, 2004. Induction of a Simple Morphology for Highly-Inflecting Languages. ACL 2004 Workshop on Comput. Phonology and Morphology.

Freelang, 2004. http: // www.freelang.net / dictionary / spanish.html.

Goldsmith John. 2001. Unsupervised Learning of the Morphology of a Natural Language. Comput. Ling. 153-189 (also see an unpublished 2004 document at http://humanities.uchicago.edu/faculty/goldsmith.)

Gusfield, Dan, 1997. Algorithms on Strings, Trees, and Sequences. Cambridge University Press.

Knight Kevin, Graehl Jonathan, 1998. Machine Transliteration. Comput. Ling. 24(4):599—612.

Kondrak Grzegorz, 2003a. Phonetic Alignment and Similarity. Comput. & the Humanities 37:273—291.

Kondrak Grzegorz, 2003b. Identifying Complex Sound Correspondences in Bilingual Wordlists.. Comput. Ling. & Intelligent Text Processing (CICLing 2003).

Kondrak Grzegorz, Marcu Daniel, Knight Kevin, 2003. Cognates Can Improve Statistical Transla-

tion Models. Human Language Technology (HLT) 2003.
Li Haizhou et al, 2004. A Joint Source-Channel Model for Machine Transliteration. ACL 2004.
Lin Wei-Hao, Chen Hsin-Hsi, 2002. Backward Machine Transliteration by Learning Phonetic Similarity. CoNLL 2002.
Mackay Wesley, Kondrak Grzegorz, 2005. Computing Word Similarity and Identifying Cognates with Pair Hidden Markov Models. CoNLL 2005.
Matlin Margaret W., 2002. Cognition, 6th ed. John Wiley & Sons.
Medina Urrea Alfonso, 2000. Automatic Discovery of Affixes by Means of a Corpus: A Catalog of Spanish Affixes. J. of Quantitative Linguistics 7(3):97 – 114.
Melamed Dan, 1997. Automatic Discovery of Non-Compositional Compounds in Parallel Data. EMNLP 1997.
Mueller Karin, 2005. Revealing Phonological Similarities between Related Languages from Automatically Generated Parallel Corpora. ACL '05 Workshop on Building and Using Parallel Texts.
Niessen Sonja, Ney Hermann, 2000. Improving SMT Quality with Morph-syntactic analysis. COLING 2000.
Ristad Eric Sven, Yianilos Peter, 1998. Learning String Edit Distance. IEEE PAMI, 20(5):522—532.
Schulz Stefan, et al 2004. Cognate Mapping. COLING 2004.
Wicentowsky Richard, 2004. Multilingual Noise-Robust Supervised Morphological Analysis using the WordFrame Model. ACL 2004 Workshop on Comput. Phonology and Morphology.

Eng.	Span.	Wit.	Squ.	Example
-tion	-	623	309	reformation:reforma
-e	-ar	461	1182	convene:convocar
-tion	-cion	434	3770	vibration:vibracion
co-	co-	363	95	coexistence:coexistencia
-ness	-	352	128	persuasiveness:persuasiva
-ation	-acion	333	4854	formulation:formulacion
in-	in-	332	1294	inapt:inepto
re-	re-	312	194	recreative:recreativo
-ed	-ado	289	102	abridged:abreviado
-ic	-ico	274	3192	strategic:estrategico
-ly	-mente	269	207	aggressively:agresivamente
-y	-ia	251	2086	agronomy:agronomia
-ble	-ble	238	153	incredible:increible
-al	-al	233	440	genital:genital
-ity	-idad	208	687	stability:estabilidad
-te	-r	206	3603	tabulate:tabular
-er	-o	203	166	biographer:biografo
-al	-o	186	2728	practical:practico
de-	de-	174	68	deformation:deformacion
-ate	-ar	170	3593	manipulate:manipular
-ous	-o	154	59	analogous:analogo
con-	con-	153	53	conceivable:concebible
-ism	-ismo	147	2173	tourism:turismo
un-	In-	134	164	undistinguishable:indistinto
-er	-ador	134	95	progammer:programador
-nt	-nte	120	514	tolerant:tolerante
-ical	-ico	111	3185	lyrical:lirico
-ist	-ista	111	1691	tourist:turista
-ize	-izar	90	974	privatize:privatizar
-ce	-cia	87	445	belligerence:beligerancia
-tive	-tivo	70	249	superlative:superlativo

Table 1: Some affix pairs discovered.

Eng.	Span.	Example
ph	f	aphoristic:aforistico
th	t	lithography:litografia
ll	l	collaboration:colaboracion
tion	cion	unconditional:incondicional
st-	est-	stylist:estilista
tia	cia	unnegotiable:innegociable

Table 2: Some letter sequence pairs discovered.