Lexical similarity metrics for vocabulary learning modeling in Computer-Assisted Language Learning (CALL)

Abstract: This paper discusses a technique for measuring lexical similarity in terms of its effect on the perceptual ability of learners in recognizing L2 words with the help of L1. This technique can be used in many modules of an ITS CALL implementation, in particular in the initialization of the learner model based on his/her native language and in the diagnose of errors due to interference from L1. The rationale for such an implementation is discussed and a brief description of the technique is given.

Keywords: natural language processing, cross-linguistic influence, interference (language), learner errors (language), learner model.

Introduction

The very particular nature of second language teaching comes from the fact that the language itself is the learning goal, the main instructional resource and the key aspect defining learners' background knowledge. This contrasts neatly with other teaching areas, indicating the need for an adequate understanding of second language learning and demanding implementation techniques capable of capturing its richness. Hence, the Computer-Assisted Language Learning (CALL) field demands very specific instructional tools and strategies as well as accurate techniques for learner modeling. For instance, it is well known that the first language (L1) can create a basis for learning the vocabulary of the second language (L2), since the already acquired L1 lexicon can help the learner to infer the meanings of words in L2, most of all if both languages have lexical similarities. In order to model (qualify and quantify) this cross-linguistic influence, techniques to compare the lexical distance between L1 and L2 are required. This comparison can be done in terms of how similar is the form of semantically related words in L1 and L2, so that the ITS can know in advance which lexical units from L2 will be more easily learned due to transfers from L1 and which ones are likely to produce interferences. The ITS can use the results of this comparison to initialize the learner model or, by means of a similar technique, to continuously assess the learning process by measuring how distant the learner's answers are from the right answers.

The lexical distance can be relevant to a greater or a lesser extent depending on the adopted instructional strategy. If teachers decide to organize the lexical units based on their frequency of use, teaching first the most used words, they can rely on objective metrics that refer only to L2, and which can be established in terms of some ranking of most frequent words (for example, in everyday vocabulary, or in some particular area of interest such as business, tourism, etc.). If, on the other hand, the lexical units are to be organized in terms of their easiness for the learners, this is indeed a relative criterion that will depend on the L1(s) of the target audience(s). In this case, the easiness of each lexical item will strongly depend on its resemblance with the corresponding word in L1, and the use of some metrics for quantifying this similarity would be desirable.

In this paper we present a work in progress, in which we are applying a technique for measuring lexical similarity in terms of its effect on the perceptual ability of learners in recognizing L2 words with the help of L1.

1. Lexical distance as a predictor of transference likelihood in ITS CALL systems

In L2 learning, it is possible and even inevitable that the learner's L1 lexicon will influence the easiness she/he will have assimilating L2 vocabulary. If the involved languages are closely related, many L2 words will probably be more easily learned since they look similar to their counterparts in L1, usually because they share a same origin (cognate words). This is, for instance, the case of words such as the Spanish "corazón" and the Portuguese "coração". Such lexical similarities may occur even between not so closely related languages, such as English and French (e.g. "liberty" – "liberté") or German and French (e.g. "blau" – "bleu"). Lexical similarities may even be found in totally distant languages due to borrowings (e.g. Japanese "arigato" and Portuguese "obrigado") or to accidental coincidences (Greek "oikia" and Tupi "oca").

Regardless of the origin of these similarities, from the didactic point of view this is an aspect that impacts the entire language learning process and therefore needs to be carefully accounted for by the ITS. This implies evaluating the level of similarity, classifying its dimensions and assessing its potential effects (beneficial or detrimental): similarity it is not always a facilitating feature, since in the case of false friends it tends to induce cross-linguistic interferences rather than correct inferences (transferences).

The level of lexical similarity can be used in many modules of an ITS CALL. For example, to determine the learner's background knowledge, and then to initialize parts of the learner model. Also knowing how distant a learner's answer is from the correct answer to a question is something that can be used to quantify and qualify the learning results and, in case of discrepancies, be a clue to diagnose causes of error (interference from L1, overgeneralization, etc.). Measuring word-level dissimilarities regarding right answers or similarities to common errors is a valuable tool in educational applications.

The similarity level has two main parallel dimensions: orthographic and phonetic. Each of them may vary from a level of "no similarity" to a level of "absolute match". For instance, the English and French words "direction" share the same spelling, but somewhat distinct pronunciations (and slightly different meanings), whereas English "house" and German "Haus" present "partial orthographic match" but have similar pronunciations (and meanings). Therefore, in order to correctly evaluate the proximity between lexical units in L1 and L2, or between learner's answer and the right answer, the CALL system needs to distinguish and compare these dimensions while applying quantitative metrics of similarity.

In our ITS CALL application we employed a multidimensional similarity measure based on perceptual criteria, involving correspondences such as letter-by-letter match, same initials, equivalent consonant order and phonetic distance. The calculation of the similarity use weights to balance the influence of orthographic and phonetic features in the overall similarity and can be used in combination with AI algorithms, such as those discussed in [1], in order to classify or cluster errors in terms of their most likely causes. Our ITS CALL application is applied in a web-based language course. As the (L2) learning object of the course we chose the international language Esperanto for two reasons: (i) it has a compact lexicon; (ii) its lexicon is based on international roots. But we believe that to some extent the achieved results will be also valid for any other languages. In the next sections we discuss these implementation in detail.

3. Methods for calculation of lexical similarity

According to [1], the manipulation of symbolic data, such as words and sentences, has usually been outside the focus of the research on neural networks and related learning algorithms, which have mainly dealt with numeric data. This was due to the fact that sensory data from real world information processing are generally numeric by definition. When it comes to numerical data, the average and the similarity are easily computed in terms of arithmetical mean and inverse distance, respectively. Although, for non-numerical data, like letter strings, both measures tend to be more complicated to compute, both calculations for letter strings can also be based on a distance measure, just like their numerical counterparts, by means of techniques such as the Levenshtein or the Feature distance. Consequently, the average of a set of strings can be obtained as a string with the smallest distance from all strings in the set, whereas the similarity can be defined as the inverse or negative distance between the strings [1]. And with those two measures and substituting reference vectors by reference strings one can construct self-organizing maps of letter strings.

As pointed out by [1], a letter string cannot be represented by a numerical vector, since a coding in which numerical differences between the codes reflect dissimilarities among corresponding letters is hard to achieve, and even more difficult when one tries to compare strings of different lengths, or when one string is derived from another by insertion or deletion of letters, something that is very common in the case of cognate words in different languages.

Hence, distance measures suited for letter strings are required. One such measure is the *Levenshtein distance*, defined as the minimum number of basic transformations – insertion, deletion and substitution of letter – to transform one string into another [2]:

$$LD(s_1, s_2) = \min (n_{ins} + n_{del} + n_{subst})$$

Derived from it is the *weighted Levenshtein distance* [3], also known as edit distance [4], where different costs are assigned to each edit operation.

The *Feature distance* [4] is given by the number of features in which two strings differ. In Feature distance, N-grams (substrings of N consecutive letters) are the usual choice for features, and if one string is longer than the other, the unmatched N-grams are also counted as differences [1]:

$$FD(s_1, s_2) = \max (N_1 + N_2) - m(s_1 + s_2)$$

Where N_1 and N_2 denote the number of N-grams in strings s_1 and s_2 and $m(s_1 + s_2)$ is the number of matching N-grams [1].

The *Levenshtein distance* leads, according to [1], to slightly better classification accuracy than the *Feature distance*, but the latter allows for much faster searching. It is worth noting that these general-purpose methods are not aimed at specific applications. Thus, in some cases, betterments have been proposed to make these calculations more suited to real world problems. In [5], for instance, the authors applied Levenshtein Distance to measure language distances so as to produce phylogenetic trees of language families based on the similarities of their basic vocabularies. However, so as to account for the fact that one letter change has more relevance in short words than in long ones, the authors developed a normalized version of LD.

Regarding the use of the lexical similarity as a parameter to determine language proximity, the authors argue that the grammatical differences would be too hard to compute, and also point out that an automated method avoids the subjectivity that is inherent when these comparisons are made by humans. Subjectivity arises because scholars tend to see similarity in remote kinship linking cognate words even when the current word forms look very different one from another, such as the Spanish word "*leche*" and the Greek "*gala*" [5]. It is worth noting that in our course we are interested in measuring effective similarity rather than in linguistic kinship, since from a didactic viewpoint, similarity, even if accidental, is what matters for learning easiness. Thus, L2 word recognition is, in such a learning context, a shortcut to vocabulary learning.

An instructional application requires similarity measures that encompass the main features that facilitate the recognition (and memorization) of a given L2 word on the basis of its alikeness with the corresponding word in L1. This measuring could involve some sort of letter-by-letter comparison, as discussed above. However, from a semiotic standpoint, the recognition of an L2 word due to its similarity to a semantically correlated L1 word is a kind of inference that is essentially based on diagrammatic (iconic) features, although both words are symbols (arbitrary signs) rather than icons. Then, in this case the similarity points from an L2 symbol (word) to a corresponding L1 symbol, contrary to ordinary icons, whose similarity (such as the picture of a car) links to physical features of an actual object. So as to emphasize the particular nature of this phenomenon we have coined the expression "intersymbolic iconicity or similarity".

As in the case discussed in [5], this requires objective criteria, based on effective similitude, rather than subjective ones, founded on remote etymological kinship. Thus, the calculation of a letter-by-letter similarity is a good starting point. Nevertheless, the evaluation of a level of similarity is not limited to an orthographic correspondence. It implies assigning more weight to key features such as correspondence of initials or coincidence in the positions of consonants, considering that the consonants in general, and initials in particular, form a diagrammatic image of any given word. This fact has a lot of support in the area of perceptual psychology, since a written or printed word is a visual stimulus in the first place [6].

According to [6], for instance, for the vast majority of people, the left hemisphere is more important than the right hemisphere for language processing, what makes the word recognition slightly easier after fixation of the leftmost than the rightmost letter of a word (in languages that are read from left to right the leftmost letter is the initial), simply because information in the right visual half-field is projected directly onto the left cerebral hemisphere whereas information in the left visual half-field requires inter-hemispheric transfer to reach the left cerebral hemisphere. Another reason for the strong word-beginning advantage in words that are read from left to right is related to the fact that fixation on the leftmost letter makes the whole word fall in the right visual half-field, which has direct connections to the dominant left hemisphere.

Word processing accuracy and speed depend on two factors: (i) perceptibility of the individual letters as a function of the fixation location and (ii) the extent to which the most visible letters isolate the target word from its competitors [6].

These word recognition factors are also applicable as a common sense technique to create word abbreviations: *tks* (*thanks*), *pg* (*page*), *cmd* (*command*) or *ctrl* (*control*). For this reason, the matching of initials and consonants is more likely to enable word recognition than matching a comparable number (i.e. same LD) of other letters without the initial or with vowels included (resp. *tak*, *ae*, *oma*, *coto*). Hence, in our technique we assign more value to the diagrammatic role of consonants than to other matchings and emphasize the function of consonants and initials, as indicated in the next section.

But these similarities can be realized also in a more phonetic level, even when the spelling rules are not equivalent (as in the case of English "physics", Czech "fyzika", Polish "fizyka", Italian "fisica", Afrikaans, "fisika" and French "physique"). According to [6], it is now clear that reading and word recognition are not simply based on orthographic information but involve the activation of phonological codes. This has been shown, for example, by [7] and [8]. In our technique the overall similarity score combines orthographic and phonetic features. It includes a *grapheme → phoneme* conversion (normalization) prior to calculating phonetic similarity of words, since a more straightforward mechanism for computing the phonetic similarities would depend on a support for the international phonetic alphabet (IPA) in the simulation tool at hand, what is not always true.

4. Calculation of intersymbolic similarity

The calculations involved in measuring word similarity in our application attempt to capture the features that matter when a learner first encounters a new L2 lexical unit. As discussed in the previous section, the main features are:

Orthographic (in order of importance):

-Initials

-Consonants (in the order they appear)

-Vowels (in the order they appear)

Phonetic:

-Phonemes (in the order they appear)

A phoneme match implies equal pronunciation even if written with different graphemes such as "c" and "k"; phonemes are considered similar in cases such as "s" and "z", "r" and "l", etc., but the similarity will depend on the languages involved, and thus a previous mapping of phonetic correspondence between L1 and L2 is necessary.

The orthographic criteria are modulated by the phonetic ones, in such a way that, if the orthographic rules of L1 use one letter to represent the same phoneme that in L2 is represented by two or three letters (e.g. Czech "š", English "sh" and German "sch"), the phonetic matching should cause the system to treat the consonantal cluster in L2 as a surrogate for the one letter initial in L1, and vice-versa. This solution tends to be more accurate in representing the similarity perceived by learners than a letter-by-letter comparison, which, by the way, could incur distortion of the similarity measure due to the risk of comparing the final letter(s) of the consonantal cluster in L2 word with the second letter/phoneme in L1. Therefore, the first step in the method deals with the segmentation of the strings in order to establish the L1–L2 grapheme/phoneme pairs. The second step evaluates distances between paired segments. The third step calculates the total intersymbolic distance, assigning weights to the parameters in the equation so that the final result is contained between 0 (match) and 1 (no match).

The equation for intersymbolic similarity is:

$$IS = \alpha(\gamma_1 I + \gamma_2 C + \gamma_3 V) + \beta P \qquad (1)$$

Where: IS: intersymbolic similarity (maximum =1, minimum = 0)

I: initials

C: consonants

V: vowels

P: phonemes (can be decomposed as the orthographical part: $\gamma_4 I + \gamma_5 C + \gamma_6 V$)

α: weight of the orthographical similarity (adjusted according to the context)

β: weight of the phonetic similarity (adjusted according to the context)

γ_n: weights of factors of similarity (e.g. γ_1=0.4; γ_2=0.4; γ_3 =0.2)

$\alpha + \beta = 1$ and $\gamma_1 + \gamma_2 + \gamma_3 = 1$ and $\gamma_4 + \gamma_5 + \gamma_6 = 1$

Note 1: Weights of the equation are adjusted so that the maximum similarity is 1 (for totally matching words) and the minimum is 0 (for totally different words).

Note 2: Weights of the orthographic features can be adjusted to assign more relevance to initials and consonants while preserving some of the effect of the vowels (e.g. γ_1=0.4; γ_2=0.4; γ_3 =0.2). The phonetic factors can be adjusted differently, if necessary.

Note 3: While initials are compared one-to-one, the comparisons of the consonant or vowel sequences consider letter groupings such as "cntrl" or "oo". The values assigned to each individual letter will depend on the length of the corresponding sequence in the original (L2) word. If the reference consonant sequence is, as in the example below, formed by "tmp", and the maximum similarity is valued as 1, each matching letter will be assigned the value of 0.33. Therefore, if the L1 word has the sequence "tm", the total score for consonant similarity will be 0.66. It goes without saying that the order of the letters is important. An alternative sequence such as "mt" would be valued 0 since it does not retain a diagrammatic representation of the L2 word morphology, and then would not have the same effect in facilitating word recognition. Here we think of the isolated role of these middle letters in the overall process of word recognition, in spite of the fact that the swap of middle letters does not impede the recognition of the word as a whole if the first and the last letters of the word are correct [9].

Note 4: In the comparisons, it may be necessary to normalize consonants and clusters to a same notation: for instance, "š", "ŝ" and "sch" to "sh". Depending on the required transformations in the normalization, different similarity values can be assigned:

-Total match = 1: Exactly the same letter(s)

-Equivalent = 0.9: Letters have closely the same function (e.g. "š" and "ŝ");

-Similar = 0.8: One letter corresponds to a consonant cluster (e.g. "š" and "sch").

Note 5: Depending on the context of the implementation, developers may neglect the phonetic similarity. In our case, however, given the multimedia nature of a Web-based course, the phonetic similarity can provide an effective basis for L2-word recognition.

Note 6: Although the final letter of a word can also play a role in its diagrammatic recognition, in our technique we decided not to emphasize final letters because in our target language the final letter is not part of the word root, but a syntactical marker. This does not preclude other developers to adapt the technique to other languages.

The algorithm for word comparison (implemented in Matlab) has the following steps:

- Identification of L1 (in order to identify the orthographic and phonetic rules)
- Segregation of initials, consonants and vowels
- Conversion of consonant clusters (normalization)
- Comparison of initials, consonants and vowels
- Calculation of the final similarity score

Obs.: All these steps were implemented as a function that can be called by other algorithms, such as AI applications for classification or clustering of data (SOM).

Example: The intersymbolic similarities of the Italian word "tempo" respectively to speakers of Portuguese, Spanish, English, German and Finnish are:

L1 (tempo)→L2 (tempo): Initials: t=t; Consonants: tmp=tmp; Vowels: eo=eo
IS = 0.6*(0.4*1+0.4*1+0.2*1)+0.4*1 = **1**

L1 (tempo)→L2 (tiempo): Initials: t=t; Consonants: tmp=tmp; Vowels: eo≈ieo
IS = 0.6*(0.4*1+0.4*1+0.2*0.66)+0.4*0.9 = **0.92**

L1 (tempo)→L2 (time): Initials: t=t; Consonants: tmp≈tm; Vowels: eo≠ie
IS = 0.6*(0.4*1+0.4*0.66+0.2*0)+0.4*0.4 = **0.48**

L1 (tempo)→L2 (Zeit): Initials: t≈Z(ts); Consonants: tmp≈Zt; Vowels: eo≈ei
IS = 0.6*(0.4*0.5+0.4*0.16+0.2*0.33)+0.4*0.2 = **0.28**

L1 (tempo)→L2 (aika): Initials: t≠a; Consonants: tmp≠k; Vowels: eo≠aia
IS = 0.6*(0.4*0+0.4*0+0.2*0)+0.4*0 = **0**

5. Experimental results

In order to evaluate the proposed technique we took the word "physics" and some of its synonyms in other languages, such as mentioned in Section 3, and compared the scores of similarity with the results produced by one of the existing distance measures, in this case the Levenshtein Distance. In LD, i = insertion, s = substitution, x = no change, one insertion counts 1, whereas one substitution counts 2 (since it means one deletion + one insertion) as follows:

Original word: “physics”		transformations	
	to Czech “fizyka”	(sisssss)	LD=13
	to Polish “fyzika”	(sixsxss)	LD=9
	to Afrikaans “fisika”	(sisxxss)	LD=9
	to Italian “fisica”	(sisxxxs)	LD=7
	to French “physique”	(xxxxxssi)	LD=5

The results for intersymbolic similarity are:

$$IS_1 = 0.6*(0.4*0.8 + 0.4*0.65 + 0.2*0.8) + 0.4*0.8 = 0.764$$
$$IS_2 = 0.6*(0.4*0.8 + 0.4*0.65 + 0.2*0.9) + 0.4*0.8 = 0.776$$
$$IS_3 = 0.6*(0.4*0.8 + 0.4*0.72 + 0.2*0.8) + 0.4*0.8 = 0.781$$
$$IS_4 = 0.6*(0.4*0.8 + 0.4*0.80 + 0.2*0.8) + 0.4*0.8 = 0.800$$
$$IS_5 = 0.6*(0.4*1.0 + 0.4*0.90 + 0.2*0.9) + 0.4*0.8 = 0.884$$

In comparison with LD, which produced totally different distances, ranging from 5 to 13, we can see that the intersymbolic similarity technique produced similar scores for the five L2 words, arguably because the technique can capture the fact that all the L2 words are more or less recognizable based on the knowledge of the original word.

Conversely, we can have an opposite situation in which two words produce smaller Levenshtein Distance, but score worse on intersymbolic similarity, such as the case of the English word “glamour” and the French “amour”, whose LD=2 scores better than the synonyms in the example above, but whose IS=0.52 indicates less actual similarity.

In order to further test the proposed technique, we selected three words from the basic lexicon of our L1 (Esperanto) and calculated their respective levels of similarity to corresponding words in 16 other (L2) languages, from different families, as shown in Table 1. For languages that do not use Latin script, we used a phonetic transcription of the words in question. The results are presented in the form of total similarity scores.

Table 1: Similarity levels for different words and languages

Language	Word 1	IS	Word 2	IS	Word 3	IS
Esperanto	floro	-	ĉokolado	-	cirko	-
English	flower	0.91	chocolate	0,79	circus	0,84
French	fleur	0.90	chocolat	0,81	cirque	0,88
Spanish	flor	1.00	chocolate	0,81	circo	0,94
Portuguese	flor	1.00	chocolate	0,81	circo	0,94
Italian	fiore	0.90	cioccolata	0,81	circo	0,94
Romanian	floare	0.91	ciocolată	0,81	cirk	0,86
German	Blume	0.11	Schokolade	0,88	Zirkus	0,88
Dutch	bloem	0,29	chocolade	0,88	circus	0,84
Afrikaans	blom	0,31	sjokolade	0,88	sirkus	0,84
Polish	kwiat	0,12	czekolada	0,83	cyrk	0,89
Indonesian	bunga	0,00	cokelat	0,48	sirkus	0,84
Russian	Цветок (tsvetok)	0,10	Шоколад (shokolad)	0,88	Цирк (cyrk)	0,89
Hindi	फल (fool)	0,65	चॉकलेट (chākleṭ)	0,57	सर्कस (sarkas)	0,65
Arabic	زَهْرَة (zahra)	0,00	شوكولاتة (shūkulāta)	0,60	سيرك (zirk)	0,63
Japanese	花 (hana)	0,00	チョコレート (chokorēto)	0,79	サーカス(sākasu)	0,31
Chinese	花 (huā)	0,00	巧克力 (qiǎo kē lì)	0,42	馬戲 (mǎ xì)	0,00

The difference of writing systems, as illustrated in the lower rows of Table 1, can be an additional difficulty in the learning process. In a Web-based context, however, one can assume that many of the learners from those cultural regions will likely be already used with the Latin script. For other contexts one could, for instance, represent the different scripts as a reduction factor in the calculation of word similarity (equation 1).

6. Discussion of the results and conclusions

We believe that the technique provides similarity values that capture the crucial features that make a word more easily recognizable by learners whenever their L1s contain a lexical unit that favors such iconic inference. In terms of effective word recognition, we conjecture that the higher the level of similarity between L1 and L2 words, the higher the probability of correct recognition (and easier memorization). Furthermore, we can assume that there is a threshold below which the recognition will no longer be possible (at least based on intersymbolic iconicity). The identification of the specific thresholds for speakers of each L1 is something that could be done in tests involving a significant number of individuals of each linguistic group. This was not in the scope of this paper. However, a field study with a reasonable number of individuals is being designed so that we can investigate how this threshold relates to the linguistic knowledge of each subject, such as the lexicon of L1 or other known languages (what is especially relevant in the cases of native speakers of languages with little lexical similarity with the target L2, if those speakers have some basic skills of another L2 more closely related to the target language).

Still related to the iconic link to L1 vocabulary, a pertinent question is how the word recognition process could be affected by other similar derivative words, such as, for example, the case of the word "episkopo", that has weak similarity with its English translation, "bishop", but a very high similarity with the corresponding adjective in English, "episcopal". A full-fledged implementation should be capable of considering such indirect similarities in the calculations, for instance, by measuring the distance not only to the direct counterpart, but the average distance to all correlated word, and maybe assigning different weights to similarities with less used words (such as in the case of "episcopal", that is less frequent than "bishop").

The purpose of this technique is to offer a practical word-level similarity metric to compare symbols from different languages so that this measure can be used as an input to initialize the learner model or to evaluate word-level errors in the context of CALL applications. It is not aimed to replace formalisms such as HPSG [10], neither to create new computational treatments of lexical rules, such as those discussed in [11, 12, 13].

In what refers to the performance of the described technique, we need to point out that calculation speed was not a primary concern, since we are more interested in the accuracy in capturing intersymbolic similarity. Furthermore, in the particular context of our ITS CALL, such lexical (dis)similarities can be used to initialize the learner models *a priori*, and then the processing load of the technique can be less relevant because it is used offline. And even in the case of the error module, responsible for comparing learner answers with the right answers, much of the calculation can be done offline, if one uses the technique to create a list of common cross-linguistic errors for every learner L1 profile, leaving to the online processing the more simple task of finding the applicable error case from among a limited list of preprocessed error types.

As discussed in [1], once the similarity (and then the distance) values are known, it is possible to apply some kind of classification or clustering algorithm, such as self-organized maps, to classify new strings. In our application we are developing a SOM, which will be used to classify word-level errors in terms of their similarities with common error types, including interferences caused by influence from L1, in which case we expect to see such errors clustered around the position that represents the corresponding L1-word.

References

[1] Fischer, I., Zell, A.: Processing Symbolic Data with Self-Organizing Maps. *Workshop SOAVE*, (2000).

[2] Levenshtein, L.I.: Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics-Doklady, 10 (8) (1966).

[3] Kohonen, T.: Self-Organization and Associative Memory, *Springer Series in Information Sciences*, vol. 8, Springer Berlin Heidelberg (1988).

[4] Wagner, R.A., Fischer, M.J.: The string-to-string correction problem, *Journal of the ACM* 21 168-173 (1974).

[5] Petroni, F., Serva, M.: Language distance and tree reconstruction. Journal of Statistical Mechanics: Theory and Experiment. IOP Publishing Ltd and SISSA (2008).

[6] Brysbaert, M., Nazir, T.: Visual constraints in written word recognition: evidence from the optimal viewing-position effect, *Journal of Research in Reading*, v.28, i.3, pp. 216-228 (2005).

[7] Drieghe, D., Brysbaert, M. Strategic effects in associative priming with words, homophones, and pseudohomophones, *Journal of Experimental Psychology: Learning, Memory and Cognition*, 28, 951-961.

[8] Harm, M.W., Seidenberg, M.S., Computing the meaning of words in reading: Cooperative division of labor between visual and phonological processes. *Psychological Review*, 111, 662-720 (2004)

[9] Grainger, J., Whitney, C.: Does the huamn mnid raed wrods as a whole? *Trends in Cognitive Science*, 8, 58-59.

[10] Pollard, C., Sag I. A.: *Information-based Syntax and Semantics*. Volume 1: Fundamentals. Stanford: CSLI Publications (1987).

[11] Meurers, W.D.: Towards a semantics for lexical rules as used in HPSG. University of Tübingen (1995).

[12] Meurers, W.D., Minnen, G.: A computational treatment of lexical rules in HPSG as covariation in lexical entries. Association for Computational Linguists (1997).

[13] Dickinson, M., Meurers, W.D.: Detecting errors in part-of-speech annotation. In Proceedings of EACL, pages 107–114, Budapest, Hungary (2003).