Lars Borin:
Linguistics isn't always the answer:
Word comparison in computational linguistics.

pp. 140–151 in:

THE 11TH NORDIC CONFERENCE ON COMPUTATIONAL LINGUISTICS.

Copenhagen, 28–29 January 1998.

NODALIDA '98
PROCEEDINGS.

Center for Sprogteknologi
and
Department of General and Applied Linguistics (IAAS)
University of Copenhagen
Njalsgade 80
DK-2300 Copenhagen S, Denmark

Linguistics isn't always the answer: Word comparison in computational linguistics[1]

Lars Borin
Department of Linguistics
Uppsala University
Lars.Borin@ling.uu.se

Abstract

String similarity metrics are important tools in computational linguistics, extensively used e.g. for comparing words in a variety of problem domains. This paper examines the sometimes made assumption that the performance of such word comparison methods would benefit from the use of linguistic, *viz*. phonological and morphological, knowledge. One linguistically naive method and one incorporating a moderate amount of linguistic sophistication were compared on a bilingual and a monolingual word comparison task for a range of languages. The results show the performance, measured as recall and precision, of the linguistically naive method to be superior in all cases.

1. Introduction

A good method for string comparison, or *string similarity metric*, is an important and useful item in the computational linguist's toolkit. Its special case, word (or rather: word form) comparison is extensively used in dealing with the following tasks, among others:

• Spelling checking and correction, where it is used to find lexical entries similar to putative misspellings (Kukich 1992; Oflazer 1996);

• Historical linguistics and dialectology, for the reconstruction of earlier language stages and for the subgrouping of related languages or dialects, through the comparison of suspected cognates (Guy 1994; Kessler 1995; Covington 1996);

• Information retrieval, in particular for finding proper names and trademarks (Kukich 1992; Siegfried 1992; Lambert 1997);

• Multilingual corpus linguistics, for sentence and word alignment and for the establishment of translation equivalences on the word level (Simard *et al*. 1992; Melamed 1995; McEnery and Oakes 1996; Tiedemann 1997);

• Extraction of lexical information from monolingual text corpora, where word form comparison can be used for finding morphologically related word forms.

[1]The research reported in this paper was carried out within the project *Creating and annotating a parallel corpus for the recognition of translation equivalents* in the research program *Translation and interpreting: A meeting between languages and cultures*, funded by the The Bank of Sweden Tercentenary Foundation. I wish to thank Kamal Khaledzadegan for his help with the transliteration of the Arabic and Persian material, Bo Utas and Carina Jahani for introducing me to the intricacies of the Arabic script, and the anonymous reviewers for helping me to clarify some of the more muddled points of my exposition.

Even though word comparison is normally used in conjunction with other methods specific to each task (such as dictionary lookup for spelling checking, other alignment methods, including incremental translation dictionary construction, for finding translation equivalents etc.), it is ubiquitous enough that it is important to know which of several available methods is the most effective for a particular task. Strangely enough, however, evaluations of the relative efficacy of these methods are scarce in the literature, as a rule. Some notable exceptions are Lambert (1997), who compares edit distance with bigram and trigram Dice scores for assessing the confusability of drug names, McEnery and Oakes (1996), who compare edit distance, truncation (i.e., initial substring matching), and bigram Dice scores for finding translation equivalents in bilingual parallel corpora, and Tiedemann (1997), who compares a variety of substring matching methods, also for finding translation equivalents in a bilingual parallel corpus. Lambert reports recall, fallout (or false positive rate), and accuracy for the methods, McEnery and Oakes give precision scores and estimated recall, while Tiedemann only calulates precision.

From a linguistic point of view, an important dividing line is that between word comparison methods which use some linguistically, *viz*. phonologically and morphologically, motivated comparison metric, and those which do more linguistically 'naive' character string comparisons. The methods compared in the works referred to above all belong in the latter category. In the literature, it is sometimes assumed—perfectly reasonably, in my view—that the performance of certain word comparison tasks would benefit from the use of a more linguistically motivated comparison method (e.g., Brasington *et al*. 1988; Borin 1991; Kessler 1995; Covington 1996). By this is usually understood a non-language-specific method[2], i.e. the idea is that the *general* performance—across a range of problems and languages—of such a method could be enhanced by the introduction of some linguistic sophistication.

To my knowledge this assumption has not earlier been explicitly tested by experiment. The purpose of the research reported here is to do this, by applying both a linguistically naive and a linguistically more sophisticated string comparison method to the same two word comparison tasks and the same material for a range of languages, and comparing the performance of the two methods on these tasks in terms of precision and recall.

As a representative of the class of linguistically naive methods was chosen a metric that hence-forth will be referred to as *LCS*, which is defined as the length of the longest common subsequence (hence the name *LCS*) of the two strings, i.e. the maximum number of exact character matches between the two strings, possibly with non-matching characters in-between, divided by the length of the longer string, thus yielding a real value in the range 0–1. The algorithm for calculating LCS is a special case of a well-known string alignment method with a time complexity $O(nm) \approx O(n^2)$, i.e. roughly quadratic[3] (Sankoff and Kruskal 1983).

[2] Which should exclude spelling checking and information retrieval algorithms incorporating explicitly language-specific pronunciation information. Such language-specific methods are useful and needed, to be sure, but their existence for some languages does not exclude the search for general methods.

[3] n and m are the lengths of the two strings. The calculation is simplified by the assumption that, on the average, $n=m$.

LCS was chosen under the assumption that the results obtained by it will be largely valid for other linguistically naive string comparison metrics described in the literature. These are methods such as

- common substrings, e.g., initial (Simard et al. 1992; McEnery and Oakes 1996; Tiedemann 1997), final (Tiedemann 1997), or in any position (Zhang and Kim 1990);
- edit distance, when it is understood as the special case of Levenshtein distance where all insertions, deletions and substitutions are given the weight 1 (Sankoff and Kruskal 1983);
- *n*-gram comparisons, often expressed as Dice scores[4] (Lambert 1997; McEnery and Oakes 1996).

For the linguistically more sophisticated method, a method described by Covington (1996; henceforth called *COG*, for *COGnate alignment*) was chosen. This is a depth-first algorithm for ranking sound alignments in word pairs, as the first step in the reconstruction of proto-forms by the comparative method used in historical linguistics. The algorithm has exponential time complexity, producing (in the worst case) approximately 3^{n-1} alignments. It works by performing a depth-first enumeration, or search, of all possible alignments of the segments of the two strings with each other[5] or with Ø. Each kind of alignment incurs a 'cost', according to the phonological nature of the segments aligned. The cost assignment scheme is shown in figure 1.

C(onsonant) with identical C	0
V(owel) with identical V	5
short V with long V, or V with S(emi-V)	10
V with different V	30
C with different C	60
completely dissimilar segments	100
segment–Ø (a 'skip') after segment–Ø	40
segment–Ø otherwise	50

Figure 1: COG cost assignment scheme

Additionally, the configuration $\begin{matrix} - & a & Ø & - \\ - & Ø & b & - \end{matrix}$ i.e., alternating skips, is not allowed.

Thus, Covington's algorithm uses fairly coarse linguistically relevant features of the word pairs to guide the alignment process, namely the phonological trichotomy V–C–S, i.e., syllabicity, according greater importance to consonants than to vowels or semivowels in determining the preferred alignment, and consequently also in determining the similarity score. There is also some linguistically motivated context sensitivity in the cost assignment scheme, since contiguous skips are preferred (and consequently also contiguous segment–segment alignments), reflecting the morphological fact that morphs tend to be contiguous. Hence, the COG method utilizes both

[4] The Dice score for an *n*-gram comparison of two strings is calculated as: 2C/(A+B), where C is the number of unique *n*-grams common to the two strings, and A and B the total number of unique *n*-grams in each string.

[5] Although the search space is minimized by the heuristic of only allowing a search to continue as long as its accumulated cost is less than the lowest total cost found so far.

phonological and morphological knowledge, admittedly of a fairly coarse and rudimentary kind, in determining the similarity score.

2. Material

The material used for the experiments consisted of a small sample from a multilingual parallel corpus under construction at our department. The corpus is made up of articles from *Invandrartidningen*, a Swedish periodical appearing in 42 issues yearly, in eight parallel language versions: Arabic, English, Finnish, Persian, Polish, Serbocroatian, Spanish and simple Swedish[6]. We also had access to the Swedish original manuscript version, which is not published as such, but on the basis of which the other language versions are produced.

All language versions except Finnish and simple Swedish were used in the experiments. Six short bilingual parallel texts were prepared, with Swedish as L1, and Arabic, English, Persian, Polish, Serbocroatian and Spanish as L2. The parallel texts were manually aligned at the sentence level, and the Arabic, Persian and Slavic texts were transliterated into a Latin-1 coding devised specially for the purposes of the experiments. Figure 2 shows some statistical data about the texts used, and in figure 3, some of the parallel sentences from the material are shown.

language	***sentences***	***words***	***words/sentence***	***=/sentence***	***corr/sentence***
Arabic	28	291	10.39	0	1.32
English	28	336	12	1.18	1.43
Persian	26	329	12.65	0	1.42
Polish	28	292	10.43	1.18	1.04
Serbocr.	28	315	11.25	0.89	1.18
Spanish	28	287	10.25	1.04	1.25
Swedish	28	238	8.5	—	—
average	28	298	10.78	0.71 / 1.07	1.27

Figure 2: Text statistics. The last two columns show the average number of exact string matches (mostly proper names) per sentence, and the average number of other correspondences per sentence (see section 3), respectively.

3. The experiments

The text material used for the experiments was deliberately kept small, primarily so that it would be possible to calculate the recall of the two methods—i.e., the ratio of the number of found correspondences to the total number of correspondences—a parameter which is important to know in order to compare the methods fairly. Where string similarity metrics are used on large text corpora (e.g., McEnery and Oakes 1996; Tiedemann 1997), it is generally not feasible to calculate recall, other than as an estimate (as is done by McEnery and Oakes), mainly because

[6] The following language name abbreviations are used in this paper:
Ar = Arabic; En = English; Pe = Persian; Pl = Polish; Sc = Serbocroatian; Sp = Spanish; Sw = Swedish.

the total number of valid correspondences must be determined by a human judge, or judges[7], and the sheer size of most corpora normally precludes this.

The performance of the two word similarity metrics were evaluted for two kinds of computational linguistic problem, namely those of

(1) bilingual word comparison, or the problem of finding word correspondences[8], i.e., putative translation equivalents, in parallel texts in two languages (for the six language pairs mentioned in the previous section);

(2) monolingual word comparison, or the problem of finding morphologically related words in a text in one language (for five of the languages; Arabic and Persian were not included in this experiment).

These two tasks, of course, are instances of the last two types mentioned in the introduction, i.e., the establishment of translation equivalences on the word level and extraction of lexical information from monolingual text corpora, respectively.

[7] Actually, if there existed a method by which recall could be automatically determined in these cases, we would of course use this method instead of the one that we are evaluating for the task at hand. Note also that calculating recall by sampling a smaller part, or smaller parts, of a larger corpus basically reduces to the procedure described here, although using several samples would presumably make the results more reliable.

[8] Often called 'cognates' in the corpus linguistics literature. I prefer the term 'correspondence', mainly because 'cognate' has a well-established use as a term in historical linguistics, which in practice is disjoint with that now being introduced in corpus linguistics; most of the items picked out by methods for finding 'cognates' in bilingual corpora (e.g. by Simard *et al*. 1992, or Melamed 1995) are actually loanwords (e.g. the English-French word pair *fraternity ~ fraternité*), i.e. virtually the opposite of cognates in the historical linguist's sense (a real English-French cognate pair, related to the previous one, would be *brotherly ~ fraternel*).

	(1)	(2)
AR	ª**l**ªntqª**l** ª**ly** **rnkby**	tqdm ªlxßr v ªl°mr
EN	Move to Rinkeby	**Reds** and **Greens** gain ground
PE	bh **rynkhby** nql mkªn knyd	srxhª v sbzhª py¢ myrvnd
PL	Przeniešþ do Rinkeby	Czerwoni i Zieloni coraz popularniejsi
SC	Preseliti nadleštvo u Rinkeby	Crveni i zeleni na usponu
SP	Trasladarse a Rinkeby	**Rojos** y verdes adelante
SW	Flytta till **Rinkeby**	Framåt för de **gröna** och **röda**

	(3)
AR	lys bªmkªn 83 fy ªlm½t ªltfkyr bªltcvyt lcªl° °zb **ªldymqrª&yt** ªljdydt
EN	Eighty-three per cent could not imagine voting for ***New*** **Democracy**
PE	83 drcd ªz mrdm nmytvªnnd tcvr r¼y dªdn bh °zb **dmkrªsy** ***nvyn*** rª dr sr bprvrªnnd
PL	83% ankietowanych nie wyobra$a sobie aby mogli poprzeþ w wyborach ***nowâ*** **demokracjê**
SC	Oko 83 odsto ne bi moglo da glasa za ***Novu*** **demokratiju**
SP	El 83 por ciento no puede pensarse votar por ***nueva*** **democracia**
SW	83 procent kan inte tänka sig att rösta på ***ny*** **demokrati**

Figure 3: Three parallel sentences from the material (in adapted orthography). Valid correspondences are boldfaced. Cognates excluded by the minimum-length criterion are boldfaced and italicized.

For both sets of word comparison experiments, the texts were normalized: All texts were lower-cased, and most vowel accent marks were removed. Additionally, all word tokens shorter than four characters were removed from the texts. Consequently, the comparisons were made on word token pairs where both tokens were at least four characters long. This minimum length criterion, borrowed from Simard *et al*. (1992), excludes mainly function words, which tend not to correspond to each other in the sense understood here, although it turned out that it also excluded some cognates, such as the Swedish word *ny* 'new' and its correspondences in the other languages, except Arabic (which, not being Indo-European, understandably does not share this cognate).

The bilingual word comparison experiments were carried out as follows. Each word in each Swedish sentence was compared with each word in the corresponding sentence in the target language, and the two similarity values (LCS and smallest COG) were calculated. For various thresholds, the string pairs falling above and below the threshold (excluding exact string matches) were compared with a precompiled list of valid correspondences (see below), whereby three important values were obtained:

P(recision), i.e. the number of valid correspondences found, divided by the total number of correspondences found;

R(ecall), i.e. the number of valid correspondences found, divided by the total number of valid correspondences for the text pair in question;

Effectiveness, or *F* (= 2·P·R/(P+R)), a frequently used combination value of P and R (see van Rijsbergen 1979, ch. 7).

The kinds of correspondences recognized as *a priori* valid in the parallel texts were (easily recognizable) *loanwords* (including proper nouns) and *cognates*, not necessarily belonging to the same part of speech in the two languages, but containing the same number of lexical morphemes[9], for example:

	L1 (Swedish) word form		***L2 correspondence***	
Sw-Ar	europa	*Europe* N	[a]l[a]vrvby	
Sw-Sp	kritiserar	*criticize* V	critica	*criticism* N
Sw-Sc	populärare	*more popular* A	popularnost	*popularity* N
Sw-En	skepsis	*scepticism* N	scepticism	
	slakt	*slaughter* N	slaughtered	

Only the words in the last correspondence pair in this list (*slakt — slaughtered*) are cognates. All the others are loanwords (including proper names).

The monolingual word comparison experiments were carried out in a similar fashion: For each text, a list of its word tokens was prepared. Each item in this list was compared with each of the items following it, and the two similarity values (MEL and smallest COG) were calculated for each word pair thus compared. P, R and F were calculated in the same way as for the parallel texts, but checked against a different list of valid correspondences than in the bilingual experiments, of course. The criteria used in compiling this list were that the correspondences should be morphologically related words, by the morphological mechanisms of (not necessarily productive) *inflection*, *derivation*, or *both derivation and inflection*, but excluding *compounding*, for example:

			corresponds to	
En	democracy		democratic	
	democracy		democrats	
Pl	nowy	*new* A	wznowieniem	*renewal* N
	g£osowa£oby	*would vote* V	g£osy	*votes, voices* N
Sw	minskar	*diminish* V	minskning	*diminishing* N
	företagandet	*the business activity* N	företagens	*the businesses'* N

4. Results

Each of the experiments yielded a range of precision/recall values, which (together with the resulting F values) were plotted against the corresponding threshold values[10], resulting in a number of diagrams like the one shown in figure 4, where the performance of both methods on monolingual Serbocroatian data is displayed.

[9] In practice, this criterion excludes all words containing more than one lexical morpheme, since, out of the languages investigated here, only Swedish and English make extensive use of compounding and only in Swedish are compounds written as one word (hence the low word count for Swedish in the table in figure 2).

[10] In order to make the values obtained by the two methods comparable, COG thresholds were mapped into the LCS threshold range using the following functions: (500–COG) / 500 (bilingual case) and (230–COG) / 200 (monolingual case)

Sensible LCS threshold values tended to fall in the range 0.45–0.8, in steps of 0.05, while the values for COG varied with the type of problem. Values of 50–250 in steps of 25 turned out to be a suitable range for for the bilingual experiments, with F values peaking for thresholds in the range 150–200, while the monolingual experiments needed less range and finer granularity: 50–150 with step size 10 and the maximal F values occurring in the threshold interval 90–110.

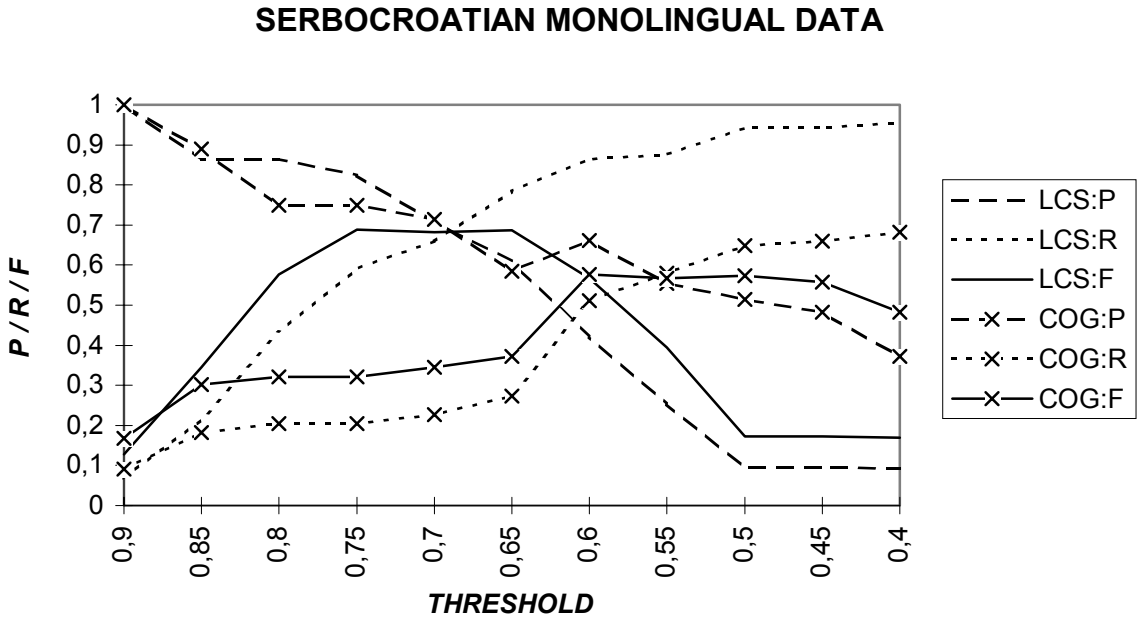

Figure 4: Precision, recall and F for the monolingual Serbocroatian experiment.

For LCS, too, the results were systematically different for the two kinds of problem. We find that maximal F values were obtained for thresholds between 0.6 and 0.5 in the bilingual case, but in the threshold interval 0.8–0.7 in the monolingual experiments. Figure 5 shows the maximal F values obtained in all the experiments.

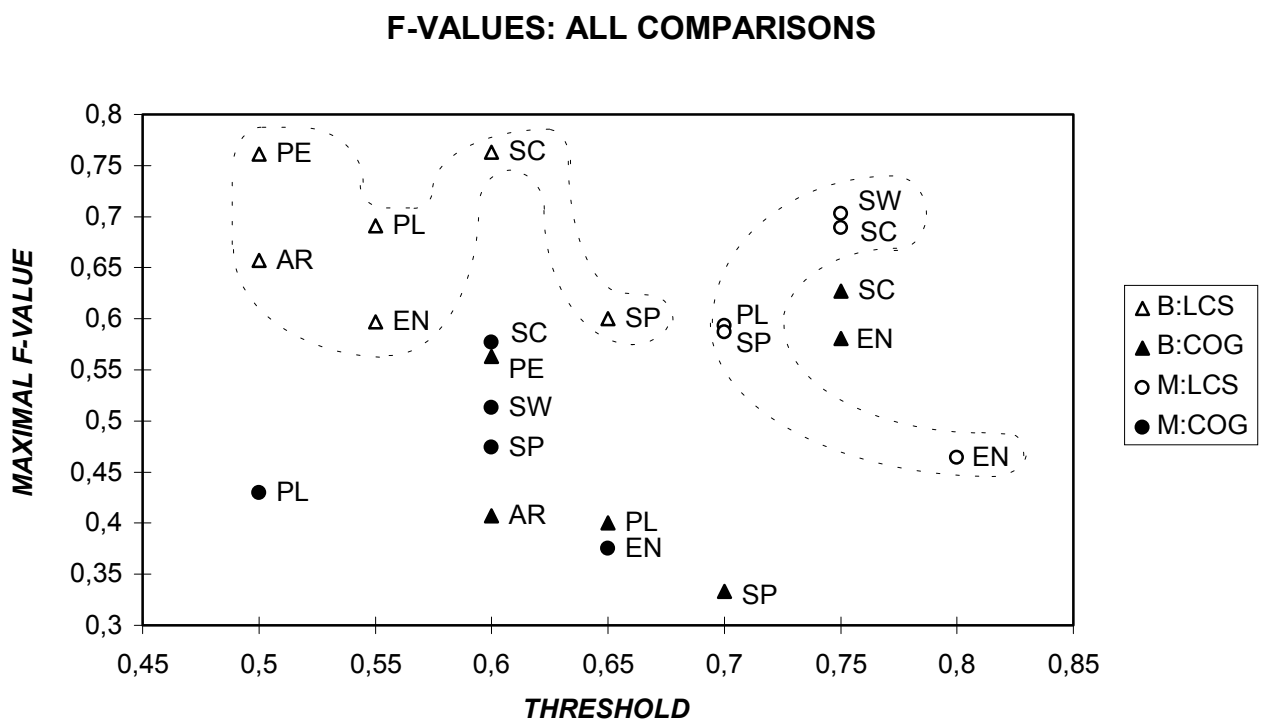

Figure 5: Maximal F values obtained in the monolingual (circles) and bilingual (triangles) experiments, for both LCS (unfilled) and COG (filled). The 'LCS regions' are marked with dashed lines.

5. Discussion and conclusions

The picture which emerges from the results of the experiments leads to a clear, but from a linguistic point of view somewhat depressing conclusion: LCS performs better overall. This is surprising; the expectation was that COG—with its greater linguistic sophistication—would have shown better performance (or been on a par with LCS), at least for the parallel texts, which are, after all, its 'natural domain' as an algorithm for comparing related words in different languages.

It is true that the material examined is very small, and thus no very reliable statistical correlations can be established on the basis of it. On the other hand, however, for all the language pairs and languages, LCS performs consistently better than COG.

Some conceivable reasons for this could be:

(1) The original formulation of COG uses a coarse phonetic transcription instead of the conventional orthography used here. Thus, it could be that the performance of the algorithm is degraded by orthographic 'noise'. This cannot apply for the monolingual experiments, however.

(2) COG was originally devised for a different type of problem—namely that of aligning word forms for historical reconstruction—than the two to which it was applied here. This *should* not

be a problem, since the mechanisms and processes that must be reckoned with are, by and large, the same for all three problem domains (see, e.g., Lass 1977).

(3) There is not enough (or not the right kind of) linguistic knowledge in COG. It has sometimes been suggested, (e.g. Borin 1991; Kessler 1995; Covington 1996) that a distance metric for phonological segments could be used as a cost scheme for linguistic string comparisons. Preliminary experiments were carried out on parts of the text material using a modified version of the COG algorithm, where consonants are compared according to their place and manner of articulation, and vowels according to (binary) phonological feature values. The results were far from encouraging, however; recall increased to some extent, but at the cost of much lower precision scores, with a resulting overall worse performance, i.e. the method found too many invalid correspondences. More experiments with various cost assignment schemes are needed, however, before this idea can be dismissed as not workable.

On the positive side, LCS is the less resource-demanding of the two algorithms, with quadratic complexity, instead of the exponential complexity of COG. We also know that bigram comparisons are about as good as LCS, at least for some word comparison tasks. McEnery and Oakes (1996) find that bigram Dice scores perform slightly better than edit distance (i.e., a method comparable to LCS) for finding translation equivalents in parallel corpora, while Lambert (1997) arrives at the opposite result: edit distance outperforms bigram and trigram comparison on the task of determining the confusability of (American) drug names. Given that the time complexity of bigram comparison is linear in the sum of the lengths of the strings, and given the results presented here, bigram comparison should be the method of choice for many word comparison applications. However, LCS (or a similar method, such as edit distance) should be chosen if there is a need to keep a statistical record of character substitutions, e.g. for adapting the word comparison method to a specific language or language pair and a specific problem domain.

6. Future work

Our future work within this problem area will concentrate on the following issues:

(i) Test whether the results hold for larger text materials and more languages;

(ii) Investigate other string comparison methods, e.g. non-symbolic methods such as neural networks, or other automatic learning methods such as that described by Ristad and Yanilos (1996), which have been systematically left out of the preceding discussion;

(iii) Investigate whether the language or language pair and the kind of correspondence (e.g. loan-word, cognate, inflectionally or derivationally related) shows a correlation with the performance of the comparison methods. The results presented in the previous section do show differences for the two kinds of problem: LCS thresholds were higher, and COG was more sensitive, in the monolingual experiments, but this was not investigated at this time;

(iv) Pursue further the idea of devising, on linguistic grounds, a more effective cost assignment scheme for the word comparison problems considered here (see the preceding section). This could involve, e.g., additional preprocessing of the text in order to make the orthography more phonological.

References

Borin, Lars. 1991. The automatic induction of morphological regularities. Reports from Uppsala University, Department of Linguistics (RUUL) 21.

Brasington, Ron, Steve Jones and Colin Biggs. 1988. The automatic induction of morphological rules. *Literary and Linguistic Computing* 3(2):71-78.

Covington, Michael A. 1996. An algorithm to align words for historical comparison. *Computational Linguistics*, 22(4):481-496.

Guy, Jacques B. M. 1994. An algorithm for identifying cognates in bilingual wordlists and its applicability to machine translation. *Journal of Quantitative Linguistics*, 1:35-42.

Kessler, Brett. 1995. Computational dialectology in Irish Gaelic. http://xxx.lanl.gov/{ ps | e-print | format } /cmp-lg/9503002.

Kukich, Karen. 1992. Techniques for automatically correcting words in text. *ACM Computing Surveys*, 24:377-439.

Lambert, Bruce L. 1997. Predicting look-alike and sound-alike medication errors. *American Journal of Health-System Pharmacy*, Vol. 54, No. 10, pp. 1161-1171.

Lass, Roger. 1977. Internal reconstruction and generative phonology. *Transactions of the Philological Society 1975*, 1-26. Oxford: Basil Blackwell.

McEnery, T. and M. Oakes 1996. Sentence and word alignment in the CRATER project. *Using Corpora for Language Research. Studies in Honour of Geoffrey Leech* ed. by J. Thomas and M. Short. London: Longman.

Melamed, I. Dan. 1995. Automatic evaluation and uniform filter cascades for inducing N-best translation lexicons. *Proceedings of the 3rd Workshop on Very Large Corpora*. Boston, MA.

Oflazer, Kemal. 1996. Error-tolerant finite-state recognition with applications to morphological analysis and spelling correction. *Computational Linguistics*, 22(1): 73-89.

van Rijsbergen, C. J. 1979. *Information Retrieval*. 2nd ed. London: Butterworths. [References here are to the electronically available version: http://www.dcs.gla.ac.uk/Keith/Preface.html]

Ristad, Eric Sven and Peter N. Yanilos. 1996. Learning string edit distance. Dept. of Computer Science, Princeton University, Research Report CS-TR-532-96.
[= http://xxx.lanl.gov/{ ps | e-print | format } /cmp-lg/9610005]

Sankoff, David and Joseph B. Kruskal (eds.)1983. *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. Reading, MA: Addison-Wesley.

Siegfried, Susan. 1992. Synoname™: A personal name-matching program for use in the humanities. *Literary and Linguistic Computing*, 7(1): 64-67.

Simard, M., G. Foster and P. Isabelle. 1992. Using cognates to align sentences in parallel corpora. *Proceedings of the 4th International Conference on Theoretical and Methodological Issues in Machine Translation*, Montreal, 67-81. [References here are to the electronically available version: http://www-rali.iro.umontreal.ca/Publications/sfiTMI92.ps]

Tiedemann, Jörg. 1997. Automatical lexicon extraction from aligned bilingual corpora. Diploma thesis in Computer Science. Otto-von-Guericke-Universität Magdeburg.

Zhang, Byong-Tak and Yung-Taek Kim. 1990. Morphological analysis and synthesis by automated discovery and acquisition of linguistic rules. *Proceedings of the 13th International Conference of the Association for Computational Linguistics*, Helsinki, Vol. 2, 431-436.