Extending Levenshtein edit distance with phonological features for cognate identification task

Anonymous EMNLP submission

Abstract

The paper presents an automated task-based evaluation framework for an extension to the Levenshtein edit distance metric. This extension explicitly represents linguistic phonological features of compared characters so the metric can use information about characters' internal structure rather than treat them as elementary atomic units of comparison. The proposed framework allows us to test alternative configurations of phonological features, and automatically select feature arrangements, which on a range of evaluation parameters systematically show greater improvements over the traditional character-based Levenshtein edit distance. Resources for computing Levenshtein edit distance with phonological features for a number of alphabets are made available as open software.

1 Introduction

This paper present a methodology for the development and automated evaluation of a linguistic features set that extends the traditional Levenshtein edit distance metric used for the task of cognate identification.

Cognate identification is important for a range of different applications; in our experimental settings we use it for assisting MT developers in creating cognate lexicon for hybrid MT between closely related languages, some of these languages are under-resourced (e.g., Ukrainian and Russian, Portuguese and Spanish, Dutch and German). For many of such language pairs there are no electronic dictionaries, and there are only small parallel corpora available with limited lexical coverage. Typically these parallel corpora can provide translations for frequently used general words, but miss the 'long tail' of less frequent, often topic-specific or terminological words. However, in closely related languages these words are often cognates, which creates a possibility to rapidly extend bilingual lexicons in semi-automated way using non-parallel, comparable corpora and automated cognate identification techniques.

In this task, cognate candidates are generated from word lists created from large monolingual comparable corpora in both languages The assumption is that the developers have good linguistic intuition of both languages and work through lists of cognate candidates, checking which pairs can be added to the bilingual dictionary. Their productivity depends on whether cognates are presented high up in the list of candidates, ideally at the top, or at least in the top N items, where N lines should at most fit on a single screen.

Levenshtein edit distance (Levenshtein, 1966) is typically used to compare word pairs from different languages and determine if they are cognate candidates. It is computed for every pair of words in the two word lists (their Cartesian product), the search space may be restricted to matching part-of-speech codes, if this annotation is available for the corpus. However, there are several problems with the traditional Levenshtein metric, one of which is that all characters are treated equally. As a result, words that are intuitively close may receive a large distance score, e.g.,

Ukrainian (Uk) "*жовтий*" (zhovtyi)='yellow'
Russian (Ru) "*жёлтый*" (zheltyi) = 'yellow'
(Lev distance = 3),

where, for historical reasons, similar sounds are represented by different characters: [o] by Uk 'о' and Ru 'ё', [y] by Uk 'и' and Ru 'ы'. On the other hand, words that are not cognates and are in-

tuitively far apart, still receive the same distance scores, such as:

Uk "*жовтий*" (zhovtyi) = 'yellow' and

Ru "*жуткий*" (zhutkiy) = 'dismal' (Lev = 3).

Some existing modifications and extensions of the Levenshtein metric introduce weightings for different character mapping, but these weights need to be set or empirically determined for each specific mapping: compared characters do not have internal structure, so there is no way to predict the weights in advance for any possible pair in a principled way.

In this paper we present an automated task-based evaluation framework for an extension of the Levenshtein edit distance metric, which explicitly represents linguistic phonological features of compared characters, so the metric can use information about characters' internal structure rather than treat them as elementary atomic units of comparison. These distinctive features have been proposed for modeling dialectological variation and historical changes in languages (Nerbonne & Heeringa, 1997), and may be also applied to a range of computational linguistic tasks, such as transliteration, morphological alternations and cognate identification. However, there are multiple ways of identifying, representing and structurally arranging these features, so there is a need for a methodology for evaluating alternative feature configurations. In our previously published pilot experiment, a smaller-scale manual evaluation indicated the need to use hierarchical phonological feature structures for consonants rather than flat feature vectors used in dialectology (Anonymous, 2016). An automated evaluation methodology proposed in this paper allows us to design and calibrate feature structures in a systematic way.

In this paper we present a framework for evaluating different arrangements of phonological features using the task of automated cognate identification. Apart from practical applications mentioned above, this task also has standardized evaluation metrics, e.g., cognate in top-N candidates, which can be applied automatically and used in the feature-engineering task for selection of optimal feature arrangements. Our evaluation results show greater improvement for hierarchical feature structures, on a number of automatically computed parameters, compared to our previous pilot experiment, and also rule out unproductive modifications of the metric.

2 Phonological distinctive features and their application for cognate identification

The theory of phonological distinctive features, first proposed by Roman Jacobson (Anderson, 1995: 116), associates each phoneme (an elementary segmental unit of speech which can distinguish meanings) with a unique set of values for categories, which may apply to larger classes of sounds. For example, phoneme [t] has the following values for the categories:

'type': consonant
'voice': unvoiced
'maner of articulation': plosive;
'active articulation organ': front of the tongue
'passive articulation organ': alveolar

Phoneme [d] has the same articulation, but is pronounced with the use of vocal cords, so it differs only in the value of one distinctive feature,

'voice': voiced

with all other categories and values remaining the same.

In historical development of languages and in morphological variation within a language the sound changes more often apply only to values of certain distinctive features within characters, but not to the whole category-value system, e.g.: Ge "Tag" = Nl "dag" ('day); Ge: "machen" = Nl "maken" ('make'). Therefore, in languages where the writing system is at least partially motivated by pronunciation, it would be useful to represent the phonological distinctive features, in order to differentiate between varying degrees of closeness for different classes of characters, e.g., vowels, sonants and consonants, or sounds with identical or similar articulation. Greater closeness between characters in terms of their phonological features has important linguistic and technical applications, such as modeling dialectal variation, historical change, morphological and derivational changes in words, such as stem alternations in inflected forms.

Phonological distinctive features have been integrated into the Levenshtein distance metric in the following way (c.f. Anonymous, 2016: 123):

(1) Substitution cost for two characters is calculated as 1 [minus] F-measure between Precision and Recall of their feature overlap; this allows calculating the cost for character with different numbers of features, and the metric remains symmetric. Intuitively, to substitute [t] with [d] in

“Tag” → “dag” we need to re-write only one feature out of 5, so the cost is 0.2 rather than 1.

(2) The order of matching the distinctive features was found to be important. Sections 3 and 4 describe an experiment on comparing two different arrangements of features: as flat feature vectors and as feature hierarchies, where matching lower level features is a pre-condition for attempting to match lower level features. Hierarchical organization achieved better performance compared to traditional flat feature vectors. Intuitively this means that not all feature categories should be treated equally, some of them have higher priority and license comparison of lower level features

(3) Insertion and deletion costs have been calibrated for the range between 0.2 and 1 using the proposed evaluation framework, described in this paper in Section 3. Optimal performance on cognate identification was achieved for cost of insertion = deletion = 0.8.

For the task of cognate identification, the introduction of these features distinguishes different types of character substitutions and gives more accurate prediction of the degree of closeness between compared characters and words, e.g., for the word pairs discussed above, where Levenshtein distance =3 in both:

Ukrainian (Uk) “жовтий” (zhovtyi) = ‘yellow’
ж 'type:consonant', 'voice:ff-voiced',
'maner:ff-fricative', 'active:ff-fronttongue',
'passive:ff-palatal'
о 'type:vowel', 'backness:back',
'height:mid', 'roundedness:rounded',
'palate:nonpalatalizing'
в 'type:consonant', 'voice:fl-voiced',
'maner:fl-fricative', 'active:fl-labial',
'passive:fl-bilabial'
т 'type:consonant', 'voice:pf-unvoiced',
'maner:pf-plosive', 'active:pf-fronttongue',
'passive:pf-alveolar']
и 'type:vowel', 'backness:front',
'height:closemid', 'roundedness:unrounded',
'palate:nonpalatalizing']
й 'type:consonant', 'voice:am-sonorant',
'maner:am-approximant','active:am-midtongue',
'passive:am-palatal'

Russian (Ru) “жёлтый” (zheltyi) = ‘yellow’ (non-matching characters) -- highlighted matches with corresponding Uk characters.

ё **'type:vowel', 'backness:back',**
'height:mid', 'roundedness:rounded',
'palate:palatalizing'
л **'type:consonant',** 'voice:lf-sonorant', '
maner:lf-lateral', 'active:lf-fronttongue',
'passive:lf-alveolar'
ы **'type:vowel',** 'backness:central',
'height:closemid', 'roundedness:unrounded',
'palate:nonpalatalizing'

Calculation of the GLev metric for (Ru) “жёлтый” (zheltyi) = ‘yellow’:

0.0	1.0	2.0	3.0	4.0	5.0	6.0
1.0	**0.0**	1.0	2.0	3.0	4.0	5.0
2.0	1.0	**0.2**	1.2	2.2	3.2	4.2
3.0	2.0	1.2	**1.0**	2.0	3.0	4.0
4.0	3.0	2.2	2.0	**1.0**	2.0	3.0
5.0	4.0	3.2	3.0	2.0	**1.2**	2.2
6.0	5.0	4.2	4.0	3.0	2.2	**1.2**

cf.: Metric calculated with Ru “жуткий” (zhutkiy) ‘dismal’, while Lev=2 for both, GLev = 2.0 > 1.2

0.0	1.0	2.0	3.0	4.0	5.0	6.0
1.0	**0.0**	1.0	2.0	3.0	4.0	5.0
2.0	1.0	**0.2**	1.2	2.2	3.2	4.2
3.0	2.0	1.2	**1.0**	1.2	2.2	3.2
4.0	3.0	2.2	2.0	**1.8**	2.2	3.0
5.0	4.0	3.2	3.0	2.8	**2.0**	3.0
6.0	5.0	4.2	4.0	3.8	3.0	**2.0**

3 Set-up of the experiment and results

The proposed methodology is an automated performance-based evaluation for testing different settings of phonological categories and values for the extended Levenshtein metric. The experiment is set up in the following way:

(1) A small Ukrainian – Russian and Russian-Ukrainian dictionary was used to compile a gold-standard translation glossary of 11k Ukrainian words, each having one or more Russian translation equivalents.

(2) Translation equivalents in the glossary have been crosschecked with monolingual Russian and Ukrainian word lists compiled from PoS-tagged corpora for these languages. Only those entries in the gold standard were retained which exist in both words lists, so the cognate identification programme can in principle find them.

(3) An additional requirement has been introduced that in both word lists the cognates should be tagged with the same part-of-speech. This reduced the search space for cognates and reduced computing time needed to calculate distances for the entries in the Russian word list. Frequency information was not used, since the monolingual word lists from both languages came from diverse corpora of different sizes. Still calculation time is a limiting factor in evaluation, so 809 entries from the gold standard were randomly chosen for computting different types of Levenshtein edit distance.

(4) For each Ukrainian entry in the gold standard, the top-N list has been calculated from the Russian word list for each of available translation equivalents.

(5) Importantly, as the evolution methodology is automatic, all translation equivalents available in the gold standard are treated equally: in this stage no distinction is made between cognates and non-cognate equivalents. This removes the need for the manual filtering of the gold standard and also naturally covers 'near-cognates' or words with cognate morphemes where only parts of words match. Since the baseline and the modified Levenshtein metric are evaluated on the same gold standard, performance figures are relative and show the difference in finding translation equivalents for any degree of 'cognateness'.

(6) Different methods are compared by the following parameters: Median top-N number for the metric; In top-1, top-5, top-10 and top-25.

(7) The following settings were compared:

(a) Baseline Levenshtein edit distance;

(b) Levenshtein distance extended with phonological features with flat feature vectors

(c) Levenshtein distance extended with hierarchical phonological features (where manner and active place of articulation are treated as top-level features which need to be matched in order for other features to match

(d) Variants of the (b) and (c) metric with different insertion / deletion values – between 0.2 and 0.8.

The results of the evaluation experiment are presented in Table 1, where:

BaseL Lev = baseline Levenshtein metric

Phon Lev H = Phonological extension to Levenshtein metric with feature hierarchy

Phon Lev V = Phonological extension to Levenshtein metric with flat feature vectors

Experiment	Med topN	Top 1	Top 5	Top 10	Top 25
BaseL Lev	50	206	328	**360**	382
Phon Lev H	**47.5**	**240**	**334**	359	**385**
Improv BaseL	+5%	+16.5%	+2%	0%	+1%
Phon Lev V	87.5	**215**	289	319	349
DiffBase L	-75%	+4.4%	-10%	-11%	-9%
PhonLevi=0.2	125.5	**216**	**291**	315	342
PhonLevi=0.4	54.5	**230**	307	334	367
PhonLevi=0.6	**48**	**235**	328	354	**385**
PhonLevi=0.8	**40**	**240**	**337**	359	**391**
Improv BaseL	+20%	+16.5%	+3%	0%	2%
Improv PhLevH	+15%	0%	+1%	0%	2%

Table 1: Automated evaluation of metric settings.

PhonLevi=0.X = Phonological extension to Levenshtein metric with modified insertion / deletion cost: i0.2 = the cost of insertion deletion is set to 0.2, i0.8 = is set to 0.8 (it is set to 1 in the Phon Lev metrics.

4 Discussion of the results, conclusion

It can be seen from Table 1 that: (1) Hierarchical phonological Levenshtein metric outperforms the baseline on the Top 1 and Top 2 measures, the median rank improvements is +5%

(2) Flat phonological feature vector metric on all measures performs worse than the baseline. This can be interpreted as the need to take into account the order of matching higher-level features. Match of low-level features is not meaningful if higher-level features are not matched.

(3) The metric with insertion / deletion cost set to 0.8 outperforms both the baseline and the Hierarchical phonological Levenshtein metric, especially on the Median Top N, Top 1 and Top5 measures. This can be interpreted as the need to scale down insertion cost moderately, since the average substitution cost is down.

The results show that phonological extension to the Levenshtein edit distance metric on the task of cognate identification outperforms the character-based baseline. The proposed framework also allows accurate calibration of the feature arrangement and other parameter settings to the metric. Future work will include systematic evaluation of different possible feature hierarchies and costs, and metrics application to other tasks, such as transliteration.

References

Anonymous, (2016)

Anderson, S. R. (1985). Phonology in the twentieth century: Theories of rules and theories of representations. University of Chicago Press.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady 10 (8): 707–710.

Nerbonne, J., & Heeringa, W. (1997). Measuring dialect distance phonetically. In Proceedings of the Third Meeting of the ACL Special Interest Group in Computational Phonology (SIGPHON-97)