Lang Resources & Evaluation (2008) 42:319–324
DOI 10.1007/s10579-008-9069-5

Adding phonetic similarity data to a lexical database

**Ruli Manurung · Graeme Ritchie · Helen Pain · Annalu Waller ·
Rolf Black · Dave O'Mara**

Published online: 19 July 2008

Abstract As part of a project to construct an interactive program which would encourage children to play with language by building jokes, we developed a lexical database, starting from WordNet. To the existing information about part of speech, synonymy, hyponymy, etc., we have added phonetic representations and phonetic similarity ratings for pairs of words/phrases.

Keywords Lexical database · Phonetic representation · WordNet

R. Manurung · H. Pain
University of Edinburgh, Edinburgh, Scotland, UK

H. Pain
e-mail: h.pain@ed.ac.uk

Present Address:
R. Manurung (✉)
Faculty of Computer Science, University of Indonesia, Depok 16424, Indonesia
e-mail: maruli@cs.ui.ac.id

G. Ritchie
University of Aberdeen, Aberdeen, Scotland, UK
e-mail: g.ritchie@abdn.ac.uk

A. Waller · R. Black · D. O'Mara
University of Dundee, Dundee, Scotland, UK

A. Waller
e-mail: awaller@computing.dundee.ac.uk

R. Black
e-mail: RolfBlack@computing.dundee.ac.uk

D. O'Mara
e-mail: domara@computing.dundee.ac.uk

1 Background

The STANDUP project (System To Augment Non-speakers' Dialogue Using Puns) (Manurung et al., forthcoming) set out to provide a 'language playground' for children with *complex communication needs* (CCN). The interactive software created allows children with CCN to explore words and phrases by having the computer build simple punning riddles, via a specialised user interface. The aim was to improve these children's linguistic, communicative and interpersonal skills, which usually develop much more slowly than those of comparable children without CCN. The feasibility of computer-generated riddles had been demonstrated by the JAPE program (Binsted and Ritchie 1997; Binsted et al. 1997). Central to this project was the lexicon, since both the joke generator and the user interface were largely driven by lexical information. In particular, the joke-generation task required a notion of *phonetic similarity*, so that puns could be made using words which were similar, not simply identical. The joke generation mechanisms, being closely based on those of JAPE, very much indicated the desirability of using WordNet (Miller et al. 1990; Fellbaum 1998), which provides certain key facilities: it has a large number of entries (around 200,000), each word form (which may be a two-word compound noun) is associated with multiple senses, each with a part-of-speech symbol, senses are grouped into sets of synonyms and linked to hypernyms and meronyms. However, it lacks phonetic data (JAPE used phonetic identity, not similarity, estimated in various ways). We have added a phonetic representation, and a similarity metric on this representation.

Other facets of the project are dealt with elsewhere: requirements (O'Mara et al. 2004; Manurung et al. 2005), overall design (Ritchie et al. 2006; Manurung et al. forthcoming), evaluation (Black et al., 2007).

2 Phonetic representations

The Unisyn[1] pronunciation dictionary supplies, for a large set of English words, strings over a phonetic alphabet (encoded in standard ASCII text characters). We used Unisyn with its 'Edinburgh' pronunciation, as our users were from central Scotland and phonetic similarity can vary between regional accents. A program matched the orthographic forms of Unisyn and WordNet entries to find phonetic strings for WordNet entries. Occasionally, there was not a unique Unisyn entry matching a given WordNet entry, usually as a result of ambiguity about stress placement. However, most of these were disambiguated by matching part-of-speech (POS) data. A few hundred entries remained ambiguous (e.g. '*lead*' as rhyming with '*seed*' or with '*bed*') and were manually disambiguated. This gave a table of nearly 100,000 entries, where an entry contains: word-form, unique ID, POS, phonetic sequence. Additionally, over 32,000 noun word-forms in WordNet of the forms X_Y or X − Y (e.g. '*blind_alley*', '*self-service*') were treated as compound nouns, and

[1] http://www.cstr.ed.ac.uk/projects/unisyn

Table 1 Vowel features, Level 2

Symbol	H	F	R	Symbol	H	F	R	Symbol	H	F	R
@	MC	C	U	eir	MC	F	U	or	M	B	R
@@r	MC	C	U	er	MO	F	U	ou	MC	B	R
@r	MC	C	U	i	MCC	F	U	our	MC	B	R
a	O	F	U	ii	C	F	U	ow	OC	B	U
ae	O	F	N	ii;	C	F	U	owr	OC	B	U
aer	O	F	N	ir	C	F	U	uh	MO	B	U
ai	O	F	N	ir;	C	F	U	ur	C	C	R
ar	O	F	U	oi	MO	B	R	ur;	C	C	R
e	MO	F	U	oir	MO	B	R	uu	C	C	R
ei	MC	F	U	oo	M	B	R	uu;	C	C	R

phonetic representations for the parts were unambiguously allocated using Unisyn (with POS for X, Y inferred from their positions).

Phonetic similarity is a research area in itself (cf. Kondrak 2003), but we needed a definition which was relatively simple and computationally tractable. Specifically, we required a measure which mapped any two strings in the Unisyn phonetic alphabet to a similarity value in the range [0, 1].

We considered a normalised minimum edit distance based on the Levenshtein technique (Jurafsky and Martin 2000), but this measures only sequence-similarity, with no allowance for degrees of similarity (as opposed to identity) between individual phonetic segments. As the Unisyn alphabet is closely related to the International Phonetic Alphabet, we started from Ladefoged and Halle (1988)'s feature decomposition of the IPA symbols, which seemed to reflect a traditional consensus on the relative dominance of phonetic properties.

Our framework involves various *attributes*, such as Voicing, Height, etc., each of which has a set of allowable values. The attributes are in three *Levels*. Level 1 has one attribute, VC, distinguishing vowels from consonants. Level 2 has six attributes: Height, Frontness, Rounding for vowels (Table 1), and Voicing, Place, Manner for consonants (Table 2).

Level 3 has various attributes, as follows. Each valid triple of Level 2 values defines a narrow class of Unisyn symbols, but such classes may not be singletons; for example, the Level 2 marking {Height: MC, Frontness: C, Rounding: U} characterises the class {@, @r, @@r} (variants on schwa). There is a unique Level 3 attribute for each such class (i.e. for each possible combination of Level 2 values); e.g. MC−C−U. This attribute then has one possible value for each Unisyn symbol in that class, so that each Level 3 attribute-value pair (e.g. MC−C−U: @r) corresponds to a unique Unisyn symbol. Hence a Unisyn symbol has exactly one Level 1 feature, three Level 2 features, and one Level 3 feature, with most of the interesting distinctions at Level 2.

Each attribute has an associated *cost*, a value in [0, 1], which was our intuitive guess at how much the similarity of two phonetic symbols was affected by differing values for this attribute. Two symbols are costed at the highest level at which they do

Table 2 Consonant features, Level 2

Symbol	V	P	M	Symbol	V	P	M	Symbol	V	P	M
?	U	G	SP	l	U	L	F	sh	U	PA	F
b	V	BL	SP	l!	V	A	FL	t	U	A	SP
ch	U	PA	A	m	V	BL	N	t^	V	PA	FL
d	V	A	SP	m!	V	LD	N	th	U	D	F
dh	V	D	F	n	V	A	N	v	V	LD	F
f	U	LD	F	n!	V	P	N	w	V	LV	A
g	V	V	SP	ng	V	V	N	x	V	V	F
h	U	G	F	p	U	BL	SP	y	V	P	A
hw	U	LV	A	r	V	PA	F	z	V	A	F
jh	V	PA	A	s	U	A	F	zh	V	PA	F
k	U	V	SP								

not have identical feature values. That is, if Unisyn symbols S_1 and S_2 have identical Level 1 features, then their Level 2 features are considered, and so on. At the level used for costing (i.e. the highest at which a difference exists) the cost is the total of the costs associated with those attributes for which the symbols have different values. Costs are allocated following certain postulates. Not only are vowels and consonants phonetically quite dissimilar, vowel–consonant substitutions tend to disrupt syllable structure, whereas vowel–vowel or consonant–consonant substitutions tend to preserve structure. So VC costs 1.0 (maximum dissimilarity). Substituting a consonant for a consonant will cost slightly more than a vowel for a vowel, given comparable levels of dissimilarity. Intuitively, consonants indicate the structure of the word. All Level 2 consonant attributes are costed at 0.28, vowel attributes at 0.15. Symbols which differ only at Level 3 should have extremely small costs for substitution, as they are almost identical. Level 3 attributes are costed at 0.15.

This mechanism assigned every pair of phonetic symbols a real value in [0, 1], indicating the *cost* of substituting one for the other (hence an identity pair costs 0.0). For example, f and v match at Level 1 (both being consonants) but their Level 2 features are $\texttt{Voicing} = \texttt{U}, \texttt{Place} = \texttt{LD}, \texttt{Manner} = \texttt{F}$ and $\texttt{Voicing} = \texttt{V}, \texttt{Place} = \texttt{LD}, \texttt{Manner} = \texttt{F}$ respectively, a difference of one feature. The cost of this difference is therefore the value allocated to Level 2 consonant features; in our implementation this is 0.28. In contrast, k has Level 2 features $\texttt{Voicing} = \texttt{U}, \texttt{Place} = \texttt{V}, \texttt{Manner} = \texttt{SP}$, and so differs from v in three Level two consonant features; this would total to 0.84 in our cost system. We do not claim that this system is perfect—it sufficed for our purposes (punning) and was much better than treating all symbols as equally different.

We then implemented a modified Levenshtein algorithm (normalised for length), in which substitution costs varied according to the similarity value for the pair involved. This allowed the assignment of a value in [0, 1] to any pair of phonetic strings, with 1.0 representing identity and 0.0 being extreme dissimilarity. Queries could then be defined which selected only those pairs of lexical entries which exceeded a specified value. Some concessions had to be made for efficiency reasons.

It was not practicable to compute similarity in real-time when searching for matching items, so a table was pre-computed of these ratings. It was impractical to store every possible word pair explicitly, so only pairs reaching a baseline threshold (in practice, 0.75) were stored in the table, with their actual similarity ratings. Hence, searches for similar strings would find only pairs rated above this baseline.

For example, '*phonetic*' and '*fanatic*' score 0.9655, whereas '*phonetic*' and '*pathetic*' score 0.75; '*backing*' and '*baking*' score 0.9524, '*backing*' and '*liking*', 0.7519.

3 The database

The full database consists of: a core lexicon of around 130,000 lexemes, each having a concept-ID (indicating the sense within WordNet), a part of speech (using WordNet's categories), a familiarity score (in [0, 1]), and a link to the word-form used; a set of about 79,000 wordforms (strings), each with an orthographic and a phonetic form; about 32,000 compound nouns, each with a head and a modifier; a phonetic similarity table; a table linking about 85,000 concept-IDs to their WordNet glosses (chunks of explanatory English text); a WordNet-derived table of about 65,000 hypernym links; a WordNet-derived table of about 75,000 meronym links; and tables which connect (some) word senses to a library of graphic images. Some relationships, computable from the basic phonetic forms, were pre-computed and stored for faster access by our joke-generator: near-homophones, rhymes (forms which were identical from the last stressed syllable to the end of the string), overlaps (a pair where one was phonetically a prefix or suffix of the other), spoonerisms (quadruples of words whose phonetic forms $\langle A,B,C,D\rangle$ can be segmented into x,y,z,w such that $A = xz$, $B = yw$, $C = yz$, $D = xw$, with some syllabic constraints).

The database is implemented using the free PostgreSQL software (http://www.postgresql.org/). The data, a Java API for database access, and software tools for customisation are downloadable from the project's website, http://www.csd.abdn.ac.uk/research/standup.

4 Future directions

The overriding aim in building this lexicon was to have a workable resource to support our own application—we were not carrying out empirical studies of lexical phenomena. The methods outlined above for phonetic similarity are documented as practical approaches which seemed to be effective in this particular case, and might be of relevance in other situations. There has been no formal evaluation of the lexicon or the construction methods—their efficacy was judged only indirectly, by the success of the system (STANDUP) in which the lexicon was embedded (Black et al. 2007). It would be very interesting to test the psychological adequacy of the phonetic similarity metric.

Acknowledgments This work, and the preparation of this paper, was supported by grants GR/S15402/01, GR/S15419/01 and EP/E011764/1 from the Engineering and Physical Sciences Research Council.

References

Binsted, K., Pain, H., & Ritchie, G. (1997). Children's evaluation of computer-generated punning riddles. *Pragmatics and Cognition, 5*(2), 305–354.

Binsted, K., & Ritchie, G. (1997). Computational rules for generating punning riddles. *Humor: International Journal of Humor Research, 10*(1), 25–76.

Black, R., Waller, A., Ritchie, G., Pain, H., & Manurung, R. (2007). Evaluation of joke-creation software with children with complex communication needs. *Communication Matters, 21*(1), 23–27.

Fellbaum, C. (1998). *WordNet: An electronic lexical database*. Cambridge, MA: MIT Press.

Jurafsky, D., & Martin, J. H. (2000). *Speech and language processing: An introduction to natural language processing, computational linguistics and speech recognition*. New Jersey: Prentice-Hall.

Kondrak, G. (2003). Phonetic alignment and similarity. *Computers and the Humanities, 37*(3), 273–291.

Ladefoged, P., & Halle, M. (1988). Some major features of the international phonetic alphabet. *Language, 64*(3), 577–582.

Manurung, R., O'Mara, D., Pain, H., Ritchie, G., & Waller, A. (2005). Facilitating user feedback in the design of a novel joke generation system for people with severe communication impairment. In G. Salvendy (Ed.), *Proceedings of HCI 2005 (CD)* (Vol. 5). New Jersey, USA: Lawrence Erlbaum.

Manurung, R., Ritchie, G., Pain, H., Waller, A., O'Mara, D., & Black, R. (forthcoming). The construction of a pun generator for language skills development. To appear in *Applied Artificial Intelligence*.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. (1990). Five papers on WordNet. *International Journal of Lexicography, 3*(4). Revised March 1993.

O'Mara, D., Waller, A., Ritchie, G., Pain, H., & Manurung, H. (2004). The role of assisted communicators as domain experts in early software design. In: *Proceedings of the 11th Biennial Conference of the International Society for Augmentative and Alternative Communication*. Natal, Brazil.

Ritchie, G., Manurung, R., Pain, H., Waller, A., & O'Mara, D. (2006). The STANDUP interactive riddle builder. *IEEE Intelligent Systems, 21*(2), 67–69.