Combining Lexical Resources for an Interactive Language Tool

R. Manurung, G. Ritchie, D. O'Mara, H. Pain, A. Waller

Context

Children who are deprived of language play are likely to experience impaired linguistic development and this can in turn inhibit communicative and social skills (Donahue & Bryan 84). A software tool which will permit a child with language and/or speech impairments to explore language in an enjoyable way, by manipulating words and phrases into simple jokes, is being built to address this issue. The automated joke-construction is based on the ideas used in the JAPE program, which could form simple punning riddles broadly similar to those in published joke books (Binsted et al. 97). JAPE was a rather basic prototype, with no real user interface, almost no facilities for guiding the joke-construction, and a response time (hours) wholly unsuitable for an interactive system. Following the philosophy of user-centred design (Vredenburg et al. 02), an extensive consultation was undertaken with domain experts (speech and language therapists) and potential users to ascertain what a joke-building system for children with disabilities should provide (O'Mara et al. 04, Manurung et al. 05).

Requirements

The joke-construction mechanisms underlying the system are principally concerned with three syntactic classes of word – nouns, noun-modifiers (adjectives or nouns used pre-nominally), and compound nouns (two-word sequences of noun-modifier and noun) – adverbs, prepositions, etc can be ignored. These keywords are part of a large-scale lexicon – quantities of data about words, running into thousands or tens of thousands of items. The system uses this data to produce jokes consisting of fixed textual strings, such as `*What do you get when you cross a*….', interspersed with keywords of the three syntactic classes such as *sheep, kangaroo, woolly jumper*. The system (STANDUP: System to Augment Non-speakers' Dialogue Using Puns) consists of a frontend (the user-interface) and a backend (the resources and processes to support the frontend). While the requirements which emerged from the user-consultation focussed on the frontend, some indirectly affected the functionality of the backend, in particular the structure and use of the lexicon. Thus requirements for the lexicon component derived either from the joke-generation mechanisms, from our gathering of user-requirements, or from general practical considerations. Some of the most central and challenging requirements were:

i. lexical items are to be comparable for phonetic similarity (and identity);
ii. lexical items can be spoken by the system, preferably in a way consistent with the phonetic similarity measure;
iii. displayed words should be accompanied by pictorial symbols;
iv. different senses of a particular word (e.g. *match)* should be treated separately and appropriately;
v. word senses should be grouped into subject-areas (topics) to facilitate a user's access to them;
vi. if possible, the topics should be clustered into a hierarchy;
vii. information on synonymy and hyponymy should be available for word-senses;
viii. it must be possible to restrict the available vocabulary to word-sets which are available in the educational or AAC fields, particularly to avoid very obscure, complex or socially inappropriate words;
ix. as much as possible of the data-preparation (e.g. reformatting or editing) should be automated, so that new versions of the lexical data can be prepared at a later date, even if the quantities of data are large.

As there was no single lexical database which would, on its own, support all these functions, methods for preprocessing or combining lexical resources available from a range of sources were considered.

Existing resources

The heart of the system is the *WordNet* electronic lexicon (Miller et al. 1993), which has over 200,000 entries, where each word form has multiple senses, senses are grouped into sets of synonyms and linked to hyponyms and hypernyms (iv. and vii., above). The FreeTTS system[1], which can "speak" any given English text was used for speech output (ii.), and the Unisyn phonetic lexicon[2] provided phonetic representation (i.),. Within AAC, there are a number of pictorial representations for words, two of the most widely used (in the UK) being the Picture Communication Symbols (PCS)[3] and the Widgit Rebus symbols[4], offering over 6000 and 7000 symbols, respectively. The owners of these resources kindly gave permission to use these pictures in the research trials of the STANDUP system. There are a number of word-sets available in the field of literacy teaching such as the Dolch and Fry reading lists (Fry et al. 00), typically containing a few thousand very common words. These supply a ready-made short-list of preferred vocabularies (viii). Various children's dictionaries group words into subject areas, and many AAC devices have a hierarchy of classes to help users to select words (v. and vi).

Complications

Although WordNet is almost ideal for supporting the joke-construction process, it lacks phonetic data, pictorial data, and contains many words unsuitable for target users. The central problem was how to tie together corresponding lexical entries from the different word-sets, given the ambiguity of simple word-forms. Although it

[1] http://freetts.sourceforge.net
[2] http://www.cstr.ed.ac.uk/projects/unisyn
[3] The Picture Communication Symbols (c) 1981-2005 by Mayer-Johnson LLC. All Rights Reserved Worldwide.
[4] http://www.widgit.com

was essential that the software could distinguish different senses of a word (here, WordNet's "synsets"), each of the lexical resources listed information not against a word-sense (semantic unit) but against a textual word-form. For example, it was essential to ensure that pictorial information for *match* (a sports event) was connected to the corresponding sense, and not to another sense of *match* (small stick for initiating fire). Similarly, phonetic information for the verb *record* should not be attached to the noun *record*, as these differ in stress pattern. When creating sets of words in a given topic, it is senses which should be clustered, not words: the incendiary *match* should not be grouped in a "sports" topic set. This meant that it was not feasible to simply use an existing topic-structured children's dictionary without further work. A similar problem arose when adopting educationally-based word sets to define limited vocabulary for the system, since these are also sets of word-forms rather than sets of word-senses. Also, compound nouns in WordNet were represented as whole items (e.g. *school_bus*), preventing matches of other words (e.g. *school, bus*) against the component parts.

Remedies

Solutions (of varying degrees of adequacy) to these problems were developed as pre-processing steps to build the lexical database from the separate resources:

1. The WordNet data were organised into a relational (SQL-accessible) database, with tables systematically relating word-forms, word-senses, phonetic representation, the subparts of compound nouns, etc.

2. The Unisyn phonetic resource is transcribed using 'metaphonemes', thus allowing users to choose from various accents; a version using an Edinburgh accent was constructed. This gave a table where an entry contains a word-form, a unique ID, a part of speech (POS), and a phonetic sequence. By comparing word-forms and POS data, nearly 100,000 non-compound-noun WordNet entries (senses) were unambiguously allocated a phonetic representation. WordNet noun entries with word-forms of "X_Y" or "X-Y" were taken to be compound nouns (e.g. "blind_alley", "self-service") and their parts stored separately, with phonetic representations for the parts being sought using the Unisyn data (with POS for X, Y inferred from their positions). Over 32,000 WordNet compound nouns were unambiguously allocated phonetic sequences in this manner. Phonetic similarity was computed using a normalised minimum edit distance cost between the phonetic representations, and all pairs reaching a threshold of similarity (currently 0.75) were stored in a table, along with the actual score.

3. Rebus symbols are linked to 'conceptcodes', provided by Widgit Software. We linked (by hand) WordNet senses to these codes. Hence, if Rebus symbols are legitimately available, they can be attached to word senses (cf. the Concept Coding Framework[5]).

4. No principled way to create automatically a set of topics (subject-matter clusters of words) or a topic hierarchy could be found. WordNet has a hierarchy, but it is more of a philosophical ontology, distinguishing (e.g.) "animate" from "inanimate", rather than a classification of a child's everyday world into recognisable categories. Adopting one of the existing word-class systems from AAC tools was the simplest solution, although it was suspected that these taxonomies might be more suitable for indexing communicative actions than for classifying words in jokes. The topic hierarchy supplied by Widgit Software was used, as it is defined over conceptcodes, which we were already linking to WordNet senses (see 3. above). As the PCS and Rebus word sets are much smaller in size than WordNet, some words were omitted from the topic sets, but this was acceptable, as topics are just one way for the user to access the lexical and joke resources.

5. We plan to pre-process a number of limited vocabulary lists from the educational literature into the database format, to serve as a library of possible "filters" on available words. The disambiguation problem arises again, and once again can only be solved by human intervention.

Result

The outcome is a lexical relational database, accessible from a Java program, with around 130,000 word-senses, all with phonetic data. About 7500 entries have codes allowing the attachment (subject to licensing) of pictorial images. Most of the construction has been automated, to ease the building of revised versions. The database is at the centre of the STANDUP interactive joke-generation system, which allows users, through an interface (customisable from standard mouse-keyboard interaction to single-switch scanning), to browse through available types of jokes, possible words and phrases, a hierarchy of topics, and to request the generation of a joke to meet certain criteria. The system is being tested with users, and this evaluation will be reported elsewhere. Although this is a specialised application, we hope that the lexical resource will be of wider use.

References

Binsted, K., Pain, H. & Ritchie, G. (1997). Children's evaluation of computer-generated punning riddles. *Pragmatics & Cognition*, 5, 2, 309-358.

Donahue, M., & Bryan, T. (1984). Communicative skills and peer relations of learning disabled adolescents. *Topics in Language Disorders*, 4, 10-21.

Fry, E.B.; Kress J.E., Fountoukidis, D.L. (2000). The reading teacher's book of lists. 4th edition. Jossey-Bass, U.S.

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 HCI 2005 (CD), Vol.5, G. Salvendy (Ed). Lawrence Erlbaum, NJ, U.S.

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

[5] http://www.conceptcoding.org

O'Mara, D, Waller, A, Ritchie, G, Pain H, Manurung, H.M. (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, 6-10 October 2004).
Vredenburg, K., Isensee, S., Righi, C. (2002) User-Centered Design: An integrated approach. Prentice Hall. NJ.