Building a Lexical Database for an Interactive Joke-Generator

R. Manurung[1], D. O'Mara[2], H. Pain[1], G. Ritchie[3], A. Waller[2]
1: School of Informatics, University of Edinburgh, Edinburgh, EH8 9LW, UK.
2: Division of Applied Computing, University of Dundee, Dundee, DD1 4HN, UK.
3: Department of Computing Science, University of Aberdeen, Aberdeen, AB24 3UE, UK.
Email: ruli.manurung@ed.ac.uk; domara@computing.dundee.ac.uk; h.pain@ed.ac.uk; gritchie@csd.abdn.ac.uk; awaller@computing.dundee.ac.uk

Abstract

As part of a project to construct an interactive program which will encourage children to play with language by building jokes, we have developed a large lexical database, closely based on WordNet. As well as the standard WordNet information about part of speech, synonymy, hyponymy, etc, we have added phonetic representations and symbolic links allowing attachment of pictures. All information is represented in a relational database, allowing powerful searches using SQL via a Java API. The lexicon has a facility to label subsets of the lexicon with symbolic names, and we are working to incorporate some educationally relevant word lists as sublexicons. This should also allow us to improve the familiarity ratings which the lexicon assigns to words.

1. Background

Children who have a disability (e.g. cerebral palsy, early brain trauma) which affects their verbal communication often develop their linguistic and interpersonal skills much more slowly than comparable children without these problems. One factor contributing to this slower development may be lack of experience of normal, everyday language use, particularly with the peer group (Donahue & Bryan 1984). A child who is forced to communicate through a voice output communication aid (a speech synthesiser coupled to a suitably engineered text input device) cannot participate fully in the banter, joking and word play which is widespread in the conversation of young children.

The aim of the STANDUP project[1] (System To Augment Non-speakers' Dialogue Using Puns) is to explore a way in which language technology might help to alleviate this situation, by providing a *software language playground* through which a child can play with words and phrases in a way which is exploratory, enjoyable and educational. To be more precise, we are building interactive software which allows children with language difficulties to explore words and phrases by building simple puns through a specialised user interface. The software contains a powerful riddle-generator which the user controls through menus, options, and the selection of words. We are about to evaluate the overall system, by carrying out systematic trials in which young children will be asked to carry out various tasks with the software. Standard literacy tests will be used to see how basic skills and use of the STANDUP system are related.

The feasibility of automating the construction of punning riddles was demonstrated by the JAPE program, which could form simple punning riddles (Binsted et al., 1997). Some of JAPE's better examples were:

What is the difference between leaves and a car?
One you brush and rake, the other you rush and brake.

What do you call a strange market?
A bizarre bazaar.

JAPE was a first research prototype which was limited in certain ways: it was not interactive (and hence had no real user interface), it took a long time to produce jokes, and the quality of the jokes (riddles) was often quite poor. We have used essentially the same ideas as those used in JAPE to build a system which is large-scale, fully engineered, robust, fast enough for interactive use, and with a user interface suitable for use by our target group (children with communication disabilities).

A central part of this endeavour was the creation of a suitable lexicon, since both the joke generator and the user interface would be largely driven by information about words (and simple phrases). This paper is about that aspect of the work – how we defined our lexical requirements, the existing resources available to us, how we combined some of these resources into a lexical database, and the overall facilities provided by the resulting lexicon. Our lexicon is similar to existing lexicons, but it does have some features which may be of interest to other potential users:

- all data is stored in relational database tables, accessible via SQL;
- lexical entries contain a variety of linguistic information – syntax, semantics, phonetics, othography, English gloss;
- a large subset of the lexicon has facilities to attach pictorial images from a standard set;
- the pictorially linkable subset is organised into a simple concept hierarchy;
- various word-frequency information from corpora and educational literature is included.

[1] http://groups.inf.ed.ac.uk/standup

2. Requirements

The requirements for the lexicon module came from two sources: the needs of the riddle-generator, and the requirements of users, in terms of both overall functionality and specific user-interface facilities.

2.1 Joke generator requirements

Experience with the JAPE program, and some planned improvements, led us to stipulate that the lexicon should:

i. allow lexical items to be compared for phonetic similarity and identity;
ii. associate part-of-speech (POS) with each lexical item;
iii. include simple common noun compounds (e.g. *door stop*), and idiomatic phrases consisting of a noun and pre-modifier (e.g. *red herring*)
iv. distinguish different senses of a word /phrase;
v. include information about synonymy ;
vi. include hyponymy/hypernymy information;
vii. include meronymy information if feasible.

2.2 User requirements

We followed a user-centred design methodology. This led us to consult two interested groups: potential users, and suitable experts (teachers, speech and language therapists). After drafting some initial design ideas, we presented these to our informants in deliberately low-tech manner, involving sketches and paper mock-ups of user-interface screens (Manurung et al., 2005; O'Mara et al., 2004). This led to a number of requirements for the system as a whole; it did not make sense to ask our informants directly about the needs of individual modules within the system, such as the lexicon. We thus developed a specification for the system, including a suitable user-interface, and tested the latter part with users via a mockup (with no real joke-generator or lexicon). The specification for the entire STANDUP system, particularly the user-interface, had consequences for the functionality of the lexicon, as follows:

i. speech output should be available;
ii. when displaying a lexical item, a pictorial symbol should, if possible, accompany it, preferably from a standard symbol-library used in augmentative and alternative communication (AAC);
iii. word-senses should be grouped into subject-areas (topics) to facilitate access by the user;
iv. the topics should be clustered into a hierarchy;
v. it is desirable to allow restricting the available vocabulary to word-sets available in the educational or AAC fields;
vi. it must be possible to avoid words deemed unsuitable for the target users (e.g. swear words, sexual terminology).

2.3 Practical considerations

General considerations of practicality, maintainability, etc. meant data-preparation (e.g. reformatting or editing) should be automated where feasible, so that new versions of the lexical resource can be prepared, even if the quantities of data are large.

3. Existing Resources

3.1 WordNet

No single lexical database supported all these functions. The JAPE program used WordNet (Fellbaum,1998), which fulfils most of the joke-generator's requirements: it has a large number of entries (around 200,000), each word form is associated with multiple senses, senses are grouped into sets of synonyms and linked to hypernyms, and it is annotated with word-sense frequency information (SemCor) derived from a large corpus (Miller et al., 1993). Its use by JAPE demonstrated that it provided the broad functionality needed for creating riddles. It is also freely available. However, it lacks phonetic data -- JAPE used phonetic identity (not similarity), computed using various resources, including a homophone list and the British English Example Pronunciation dictionary. WordNet also lacks pictorial data, and contains many words which are unsuitable for our target users (mostly as a result of being highly obscure, non-British, or archaic, rather than being socially unacceptable).

3.2 The disambiguation problem

There are a variety of lexicons around, mostly based on conventional dictionaries owned by publishers. All of these provide fewer of the required facilities (for joke generation) than does WordNet. Moreover, they tend to have two major limitations: they are not freely available for incorporation into our software (particularly as we hope to make our system available at little or no cost), and useful information (e.g. pictures, frequency data) is usually attached to word forms (word strings as spelled in normal text) rather than to word senses (distinct meanings). The latter was a serious deficiency for us. If a word had two radically different senses (for example, *match* meaning "a sporting event", or *match* meaning "a small stick for creating fire"), it would not be appropriate to use the picture for one sense when displaying the other sense. Also, one word-sense might be very common but the other very obscure; for example, *bus* as means of transport, or as "the topology of a network whose components are connected by a busbar". In such a case, our joke generator needs to be able to make puns which depend only on the familiar meaning, as the user is unlikely to know of very arcane senses. Hence, any frequency rating which is attached only to the word form (e.g. *bus*), as, for example, in the COBUILD dictionary, could be misleading. We considered various ways in which statistical or text-matching methods could be used to associate the attached information (from publishers' dictionaries) with separate WordNet senses, but could not find or devise one which seemed sufficiently reliable. The SemCor frequencies within WordNet, on the other hand, are

attached to senses ("synsets"), which made them immediately usable.

4. The STANDUP Lexicon

4.1 Overview

Using data from WordNet and other sources, we have built a relational database, with tables containing fields for word-forms, word-senses, phonetic representations, the subparts of compound nouns, etc. There are also familiarity scores and codes to link to pictorial images. The database also contains various pre-cached tables of useful linguistic relations, such as phonetic similarity and rhyming. Access to the database from our main program (in Java) was handled by connecting to a Postgres server, which could respond to queries in SQL.

4.2 Phonetic forms

From the Unisyn text-to-speech dictionary[2], we constructed 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 WordNet entries (senses) were unambiguously allocated a phonetic representation. 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 phonetic representations for the parts were unambiguously allocated using Unisyn (with POS for X, Y inferred from their positions).

4.3 Phonetic similarity

Phonetic similarity ($0 < s \leq 1$, 1 being identity) was computed between pairs involving all the word forms used as lexical head words, using a normalised minimum edit distance (Jurafsky & Martin 2000, Chapter 5) between the Unisyn phonetic representations, and pairs reaching a threshold ($s \geq 0.75$) were stored in the database, along with the actual score. SQL queries could then be defined which selected only those entries which exceeded some threshold (which had to be greater than this baseline).

4.4 Other phonetic relations

Various relationships computable from the basic phonetic forms were pre-computed and stored for faster access: homophones, e.g. *board* and *bored*; rhymes (defined – roughly -- as having phonetic forms which ended identically from the last stressed syllable onwards), e.g. *pub* and *rub*; word forms which were prefixes of other words (phonetically), e.g. *axe* and *access*; and spoonerism sequences (quadruples of lexemes whose phonetic forms <A,B,C,D> can be segmented into x,y,z,w such that A = xz, B = yw, C= yz, D = xw, with some syllabic constraints), e.g. *burn*, *ache*, *urn*, *bake*.

4.5 Frequency/familiarity ratings

As noted already, each lexeme has a SemCor frequency value, taken directly from WordNet. We have also included SemCor ratings in some of the other tables which we have pre-computed to assist the joke-generator. However, this rating has certain weaknesses for our purposes. It is based on a sense-annotated version of the Brown corpus (Francis & Kučera 1982), which contains texts published in the USA in 1961. This means that the pattern of frequencies is not highly reliable as a guide to familiarity for young British children in 2006. For example, some common words, such as *baker*, *onion*, and *sleepy* score 0 (i.e. do not appear in the corpus), others (*milk*, *nail)*, have very low scores (i.e. appear very rarely in the corpus), whereas some more obscure terms, such as *stock*, *business*, *performance, vocational* and *polynomial,* are highly rated (frequent). We are therefore treating SemCor scores as a provisional familiarity rating, until we can devise and implement something better.

4.6 Pictures

In order to have pictures associated with lexemes, there were two problems to solve: finding a suitable set of electronic pictorial images, and ensuring that these images were attached to appropriate senses. The Rebus set of symbols (small picture images), owned by Widgit Software Ltd[3], are used in a number of proprietary programs in the general area of special needs and AAC. They are intended to depict the meanings of individual words, and can be used (in the Widgit software) for tasks such as elucidating the meanings of individual words within a text, or constructing picture arrays for communication devices. Widgit granted us permission to use the Rebus symbol set (which contains over 10,000 items) in the STANDUP interactive software. However, we still faced the disambiguation problem: the symbols were linked not to word senses but to word forms. In view of the demand from our users for picture support, we decided to invest the effort in disambiguating the Rebus symbol set by hand. As a result, approximately 7500 lexemes in our database have symbolic codes which allow the direct attachment of Rebus pictures.

4.7 Labelled word sets

The software allows for any arbitrary set of lexemes to be grouped together and given a mnemonic name, thereby allowing subsets of the overall lexicon to be manipulated separately. We have made use of this to impose prohibitions on particular words. For our educational application, it was important to be able to exclude certain words from appearing in computer-generated jokes: swear words,

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

[3] http://www.widgit.com

racially offensive terms, etc. We therefore incorporated an explicit list of words to be excluded from use by the joke generator. This was done by looking in an electronic version of the Shorter OED for entries which had "coarse slang" or "racially offensive" in the relevant fields, then (by hand) creating a STANDUP-style sublexicon containing only the corresponding STANDUP lexical entries.

We are also looking into having a set of *preferred word sets*, based on various vocabularies from the educational literature, for two reasons. Firstly, when evaluating the full STANDUP system with users, it is useful to categorise the lexemes used within jokes according to their level of accessibility to children. Secondly, to increase the likelihood that the joke-generator produces jokes comprehensible to young children, words in these word-sets should be preferred in searches for possible words/phrases. We are currently planning how to integrate this with SemCor data to give an improved measure of familiarity.

Once again, disambiguation by hand is required to create these lexeme sets, as published word lists contain only word forms, not specific senses. Fortunately, the sets are typically fairly small – two or three thousand words.

4.8 Topic hierarchy

As noted earlier, we wanted users to be able to access information via topics (subclasses of subject matter). WordNet's hypernym hierarchy is unsuitable for this purpose, being a philosophical ontology rather than a classification of a child's everyday world into recognisable categories. However, the Rebus pictorial symbols are linked to "conceptcode" IDs defined by Widgit, and the conceptcodes are clustered into topics. Once the WordNet senses were linked to Widgit conceptcodes, this automatically connected them both to the pictures and the Widgit topic sets. The hand-disambiguation between word-senses and pictorial images mentiond earlier was carried out using these concept-codes, thereby linking this subset of WordNet senses to the Widgit topic hierarchy.

5. Distribution

Distribution arrangements, for the full STANDUP system or for the lexicon module, are not decided, but we intend to make the software as freely available as possible; details will be posted on the STANDUP website (see footnote 1). Some of the annotations may be lodged with the Concept Coding Framework[4]. Although Widgit have given permission for their Rebus pictorial images to be used in the full STANDUP system, no such arrangement has been made for the lexicon on its own. However, a few thousand of the commoner senses in the lexicon do contain connections from WordNet senses to Widgit symbol identifiers, which means that a researcher who had legitimate access to the Rebus images could attach them.

6. Conclusions

The development of the STANDUP lexicon is still in progress at present (February 2006). We have a lexical database, accessible from a Java API, which systematically links phonetic, topic and pictorial information to a large subset of the WordNet senses. It has around 130,000 word-senses, all with phonetic information, and around 7500 are linked to "conceptcodes" which allow the attachment (subject to licensing) of pictorial symbols. This is at the centre of the STANDUP interactive joke-generation system, which allows users to browse through available types of riddles, possible words and phrases, a hierarchy of topics, and to request the generation of a riddle to meet certain criteria. Although this is a specialised application, we hope that the lexical resource will be of wider use.

Acknowledgements

The work reported here was supported by grants GR/S15402/01 and GR/R83217/01 from the UK Engineering and Physical Sciences Research Council. We are grateful for the help of Widgit Software Ltd.

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.

Fellbaum, C. (1998). WordNet: An Electronic Lexical Database. MIT Press, Cambridge, Mass.

Francis, W. N. and Kučera, H. 1982. *Frequency Analysis of English Usage: Lexicon and Grammar*. Boston: Houghton Mifflin.

Jurafsky, D. & J. H. Martin (2000) Speech and Language Processing. Prentice Hall, New Jersey, USA.

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, USA.

Miller G., Leacock, C., Randee, T., & Bunker, R. (1993). A Semantic Concordance. *Proc. 3rd DARPA Workshop on Human Language Technology*, Princeton, USA.

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* (CD) Natal, Brazil, 6-10 October 2004.

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