LEXDIS, a tool for measuring lexical proximity

Collocate lists (alongside with *indicators*) are among the most valuable metalinguistic information items provided by monolingual and bilingual dictionaries to enable the user to distinguish between the various word senses or translations associated with a given lexical item. Most of the time, though, the user's text does not display as head of the relevant syntactic slot (subject, object, etc.) a morphosyntactic word whose lemmatization yields a member of one of the dictionary's collocate lists for that syntactic position. The reader (or in our case the computer program) has to go through the items to be found in the collocate lists associated with the relevant syntactic position, measuring the proximity of the textual element to each of the members of every collocate list. The winner (the selected word sense or translation) should be the bearer of the collocate list one of whose members features the best proximity measure with respect to the textual item.

In a very large number of cases, indicators and collocates are the **only** information made available by dictionaries to enable the reader to distinguish among a range of word senses or translations. It stands to reason that NLP systems that are unable to use that information will equally be bound to stop short of an analysis sufficiently fine-grained to satisfy the requirements of a large number of applications, among which, very obviously, machine or machine-assisted translation.

An example will help to clarify matters. Using **Defidic** (an in-house merge of the Robert-Collins and Oxford-Hachette English-French dictionaries), in

(1) *From his beloved sister Fatmeh, who had given him the gilded **charm** to **wear** round his neck*
(John Le Carré*, The Little Drummer Girl,* Pan Books edition, 1984, p.210)

the analysis of the pair WEAR-CHARM should lead to the selection of the translations **porter** or **mettre** (which both display *jewellery* in their associated collocate lists for the object) rather than **arborer**, **accepter** or **user** whose collocate lists do not provide as good a proximity match with *charm* as does *jewellery*. At the same time, *charm* itself can be disambiguated on the basis of the word sense providing the match, i.e. *ci11034* in our CIDE database, namely:

a small esp gold or silver object worn on a chain as jewellery

Computing a lexical proximity factor also comes in useful in relating **indicators** and **collocates**. Consider 2:

(2) *And now that little bogy had been exorcized with the rest!*
(Angus Wilson, *Hemlock and After*, Penguin ed., pp. 9-10 ; French tr. by Marie Tadié (1954), *La ciguë et après*, Robert Laffont, 10-18, domaine étranger, p. 8 : *Et voilà que ce petit **croquemitaine**-là avait été exorcisé comme les autres!*)

In order to translate *bog(e)y* by **croquemitaine** (in preference to **crotte de nez**, **bogée**, **épouvantail**, **démon**, **spectre** and **bête noire**, the other translations offered by DEFIDIC), we need to explore the collocate list for the object of *exorcize* (a single list), and attempt to find the adequate reading/translation for *bogey* by measuring how well the indicators for *bogey* (namely **in nose** for *crotte de nez*, **in golf** for *bogée*, **frightening** for *épouvantail/démon*, **evil spirit** for *croquemitaine*, **imagined fear** for *spectre*, and **bugbear** for *bête noire*) match against the collocates for the object of *exorcize* (a single, three-item list: **demon**, **memory**, **past**). The task is simple enough for the human user, but we claim that there is no single lexical resource that can be consulted in order to assess the quality of the match, assessment which proves indispensable for the selection of the right translation, in so far as in the case of the *exorcize-bogey* pair the indicators and collocates are the only elements provided by the dictionaries to discriminate between the six candidate translations for *bogey*. LEXDIS, our experimental tool, yields the following proximity ratings (the higher the rating, the closer the lexical items of the pair):

demon	nose	1
demon	spirit	**57**
demon	golf	2
demon	frightening	0
demon	fear	1
demon	bugbear	0
memory	nose	9
memory	spirit	**19**
memory	golf	2
memory	frightening	0
memory	fear	7
memory	bugbear	5
past	nose	4
past	spirit	**10**
past	golf	0
past	frightening	0
past	fear	6
past	bugbear	3

BOG(E)Y is therefore to be translated as[1] :

related through : indicator *spirit*	**86** (57 demon-spirit + 19 past-spirit + 10 memory-spirit)	**croquemitaine** selected as translation
fear	14 (1 demon-fear + 7 memory-fear + 6 past-fear)	spectre
nose	14 (1 demon-nose + 9 memory-nose + 4 past-nose)	crotte de nez
golf	4 (2 demon-golf + 2 memory-golf)	bogée
frightening	0	épouvantail
bugbear	8 (5 memory-bugbear + 3 past-bugbear)	bête noire

QUERY: [demon, n, spirit, n, w:none, m:g]
demon with POS=n is related to spirit with POS=n with weight=57 as follows:
Shared Labels: [my] -> weight: 4
Shared words in definition: [considered, spirit, supernatural, force, activity, believed, peoples, energy] -> weight: 44
Cooccurrence in Roget's thesaurus -> weight: 1
Sharing WordNet path -> weight: 8

Here, the query concerns the two nouns *demon* and *spirit*, no minimal weight is specified (w:none), the mode is global (m:g), which means that we are attempting to match lexemes and not word senses. LEXDIS tells us that a proximity factor of 57 links the two nouns. The strength of the link is assessed on the basis of a shared label (my=*Mythology and Legend*), eight words common to the definitions of the two items, co-occurrence in a slot of Roget's thesaurus (a broad slot, the weight being only 1) and sharing of a node in the hyponym-hypernym paths derived from various relations between WordNet synsets.

Here, the query concerns the two nouns *demon* and *spirit*, no minimal weight is specified (w:none),

1 Using only WordNet thesauric relations instead of the full LEXDIS, we get the following ranking : spirit (12), nose (6) and bugbear (5). As in the experiment reported in Appendix D.5, WordNet results often enable us to select the right reading, although it does not stand out as clearly as with LEXDIS.

the mode is global (m:g), which means that we are attempting to match lexemes and not word senses. LEXDIS tells us that a proximity factor of 57 links the two nouns. The strength of the link is assessed on the basis of a shared label (my=*Mythology and Legend*), eight words common to the definitions of the two items, co-occurrence in a slot of Roget's thesaurus (a broad slot, the weight being only 1) and sharing of a node in the hyponym-hypernym paths derived from various relations between WordNet synsets.

QUERY: [jewellery, n, charm, n, w:t, m:l]
jewellery with POS=n and Idnum=ci38980 is related to charm with POS=n and Idnum=ci11034 with weight=29 as follows:
Shared words in definition: [worn, gold, silver] -> weight: 29

The *w:t* option (top weight only) associated with the *m:l* (word sense level rather than lexeme level) option ensure that LEXDIS gives us only the best match at word sense level.

QUERY: show(charm, n, df:gold)
charm n [jewellery] [] ci11034 [small, gold, silver, object, worn, chain, jewellery]
Def: a small esp gold or silver object worn on a chain as jewellery

charm n [trinket] [] co9323 [small, ornament, silver, gold, fixed, bracelet, necklace]
Def: a charm is a small ornament usually made of silver or gold that is fixed to a bracelet or necklace often with several others

The *show* command is used in order to retrieve entries meeting a certain criterion. Here, the *df:gold* specification asks LEXDIS to select word senses of the noun *charm* whose definitions include the word *gold*. The first of these is the word sense selected on the basis of the best match between the collocate *jewellery* and the textual element *charm* in the example just discussed.

We claim that lexical proximity cannot be reduced to the measure of relatively well-defined relations such as those to be found in a standard thesaurus such as Roget's or in a WordNet type thesaurus. Although these relations are relevant and contribute to the total assessment of lexical proximity, they are too restrictive in the horizontal dimension (synonymy and antonymy providing the bulk of the matches) and tend to wander too far from the pivot along the vertical axis (hyponymy and hypernymy soon provide catch-all categories, or else – as in WordNet– explore a scientific terminology that is of little use for the analysis of lexical relations in the general language). We argue that lexical proximity is best conceived of as the result of an inherently heuristics-based exploration of various lexical associations derivable from (suitably massaged) available lexical resources, dictionaries as well as thesauri. The justification for building such a tool lies wholly in its discriminatory power, i.e. its power to select the right translation or word sense in context.

We have therefore opted for the use of a highly specific type of corpus, namely a corpus based on lexicographical resources, because we believe that good lexicographical resources themselves result from a careful analysis of textual corpora, and incorporate in a nutshell the best lexical information that can be derived from a study of raw textual data. An entry in a dictionary (especially a learner's dictionary, such as the three we have used, namely CIDE, LDOCE and COBUILD) can be looked at as the description of the lexical world of the word being characterized. Definitions and examples tend to capture in as short a piece of text as possible the main elements to be found in the environment of the word, both along the paradigmatic (definition) and syntagmatic (both definition and example) axes. The network of relationships captured in definitions and examples goes far beyond the exploration of the horizontal and vertical thesauric relations embodied in the listing of

hyponyms, hypernyms, synonyms and antonyms. For instance, the relation between an instrument and what it is used for (or the other 'qualia' of Pustejovsky's generative lexicon – see Pustejovsky 1995) will very often be part of the information provided by definitions and examples.

It should be clear by now that if lexical distance is computed on the basis of data gathered from a textual corpus (as is the case in Church and Hanks 1989 and in the considerable body of derived and associated research), the syntagmatic axis is privileged. Such a bias is suitable for the design of lexicographer's workbenches and similar tools, but is not so helpful when the task is to measure the proximity of textual data and lexicographical metalinguistic information such as indicator and collocate lists, as in the example outlined above.
Nor is an emphasis on the paradigmatic axis more suitable for such a task, as the indicators and collocates are not to be read as thesauric heads any more than they are to be read as standing for individual lexical items. More subtly, they give an idea of the lexical 'world' in which the item described 'belongs' or 'feels at home', to use metaphors that reflect how difficult it is to pinpoint the relationship between a collocate in a dictionary collocate list and the textual items it is supposed to 'stand for'.
Among the resources tapped by LEXDIS, some are clearly paradigm-oriented and some are clearly syntagm-oriented. The syntagmatic axis is reflected in *dictionary examples*, *collocate lists* and *environments*; the paradigmatic axis comes to the fore in *indicators*, *guidewords*, *WordNet Synsets and Relations* and *Roget's Categories*. But LEXDIS also makes use of *dictionary definitions*, and these, whenever they are well designed, offer in themselves a balance between syntagmatic and paradigmatic information, whether they espouse the traditional Aristotelian format of genus+differentiae or conform to the context-setting environment of the COBUILD definition type.
Appendix B lists the lexical items that reach a proximity threshold of 24 when matched with the noun 'doctor'.
LEXDIS does not relate *word forms* (the morpho-syntactic words making up a textual corpus), but either *lemmas* (global mode) or *word senses* (local mode). Whatever the ontological status of the latter (see Kilgarriff 1999), it stands to reason that they are essential to the monolingual organisation of the semantic space covered by a lemma, just as the division of this semantic space into areas covered by the various translations of the source item in the target language is at the very heart of bilingual lexicography. We need not look for justifications for such practice: they are all over the place.
We do need the two LEXDIS modes. The problem is that words do not carry labels indicating the word sense they illustrate, neither in the user's text (the task – Word Sense Discrimination – is precisely to provide such labels), nor in the lexicographical resources (where it would be reasonable to expect them, e.g. we can conceive of collocate lists where it would be clear that the *watch* that is offered as a collocate for the object of *wear* is not the lemma *watch* but a word sense or set of closely related word senses associated with that lemma). Consequently, when we attempt to assess the proximity of a textual element to a collocate, we are working with two lemmas, and the global mode is the one we have to use.
However, LEXDIS is also happy to work in local mode, trying to constrain the hypotheses as to what area of the semantic space is covered by a lexical item be found in a piece of text. Consider:
(3) *He was wearing an expensive red tie and a gold watch.*
from which a parser should enable us to derive the triplet: *t(wear, tie, watch),*i.e. *t(ArgBearer, FirstArg, SecondArg).*
Wear, *watch* and *tie* are all three polysemic, as are all reasonably frequent and reasonably heavy lexical items. By using LEXDIS on the triplet, we are able to reduce the number of word senses that ought to be taken into consideration to account for a 'standard' reading of (3) (see Appendix C).

LEXDIS is designed to measure the proximity of any two English lexical items belonging to one or more of the following parts of speech : noun, verb, adjective, adverb. It does not need to be fed any

body of textual material such as a corpus. It calls on the following lexical resources, available to us mainly through research contracts:

semdic : dictionary *clauses* (i.e. Prolog clauses) derived from CIDE, COBUILD, LDOCE and the WordNet Synsets and Synset Glosses (the dictionary clauses feature the lexical items in both definitions and examples as *word bags*, to the exclusion of a specially designed list of stopwords, both tool words and words specific to lexicographic practice, such as *especially*)
Here is one of the dictionary clauses for the entry CAT (derived from CIDE):

```
mono(lem('cat'), ori('ci'), idnum('ci10278'), pos('n'), lab([]), gw([]),
deflex(['small', 'four-legged', 'furry', 'animal', 'tail', 'claws', 'pet', 'catching', 'mice', 'member', 'biologically', 'similar',
'animals', 'lion']),
exlex(['pet', 'stray', 'feed', 'holiday']),
def(' a small four-legged furry animal with a tail and claws usually kept as a pet or for catching mice or any member of
the group of biologically similar animals such as the lion')).
```

mt : data base of RC/OH collocates - the pivotal property is co-presence within the same collocate field (cf. the hypothesis put forward in Montemagni et al. 1996). The co-occurrence lists are assigned as early as possible in the alphabetical ranking of the lexical items and it is therefore the 'smaller' word that should be explored. An mt line looks like the following:

```
mt(digestion, [ [growth,1], [machine,1], [mind,1], [movement,1], [reaction,1], [recovery,1], [stomach,4]]).
```

this means that the word 'digestion' co-occurs 1 time with 'growth' in a collocate list ... and 4 times with 'stomach'; the sharing of 'digestion' with a word preceding 'digestion' should be looked for under that word

roget : database of Roget's Thesaurus Categories (three levels)
Connectedness is established through the sharing of Roget's categories; three levels of delicacy in thesaurus organisation are catered for. A *r* line looks like the following:

```
r( 'antiquarian', [ ['n','122','4','4'],  ['n','492','4','2'] ])
```

which means that the word *antiquarian* is a noun that belongs to the two category triples *122/ 4/ 4* and *492 /4 /2* where the broadest category comes first (492), followed by sub-category(4) and sub-sub-category(2).

s : Synset WordNet database in Prolog format downloadable from the WordNet website. The WordNet 's' predicate yields the Synset to which a Word-Pos pair belongs, as in

```
s(105441468,1,'suppressor',n,2,0).
```

paths : database of paths derived from various WordNet data bases. The 'path' predicate' is built on the basis of a recursive exploration of the following Synset-to-Synset relations : cs, ent, hyp, ins, mm, mp, ms. A path line looks like this

```
path(105441468,[105436752,108459252,108457976,108456993,107938773,100031264,100002137,100001740]).
```

i.e. a synset identifier followed by a hierarchical list of hypernym synsets of various types (hypo-hyper stricto sensu, part-whole, etc.)

indic: data base of RC(Robert/Collins)/OH(Oxford/Hachette) indicators (in these two bilingual E-F/ F-E dictionaries, only the E->F direction is explored). Here are the indicators for the lexeme CAT:

```
ind(lemma('cat'),pos(n),indic(['catalytic', 'converter'])).
ind(lemma('cat'),pos(n),indic(['domestic'])).
ind(lemma('cat'),pos(n),indic(['feline', 'species'])).
ind(lemma('cat'),pos(n),indic(['female'])).
ind(lemma('cat'),pos(n),indic(['guy'])).
ind(lemma('cat'),pos(n),indic(['man', 'woman'])).
ind(lemma('cat'),pos(n),indic(['man'])).
ind(lemma('cat'),pos(n),indic(['woman'])).
```

coll : data base of RC/OH collocates - the pivotal element is the collocate bearer
Connectedness is established through collocate sharing in RC/OH collocate data base. A sample line:

coll(lemma('abandonment'), pos(n), coll(['property', 'right'])).
Here, the two items are related if they POSSESS common elements in their collocate lists, whereas in metameet it is the co-presence within a collocate list (associated with whatever item) that is significant.

envir : data base of environments derived from RC/OH 'extended' lemmas i.e. including phrases and examples
e(hdwd('dative'), envir(['case','ending'])).
e(hdwd('cat'), envir(['big', 'cats', 'burglar', 'cat-basket', 'cat-lick', 'cat-onine-tails', 'catbird', 'seat', 'catfood', 'catgut', 'cathouse', 'catmint', 'bag', 'dogs', 'mice', 'play', 'cats-cradle', 'cats-eye', 'cats-paw', 'cats-whisker', 'catsuit', 'door', 'family', 'fight', 'dog', 'flap', 'give', 'grin', 'hardly', 'room', 'swing', 'hot', 'bricks', 'tin', 'roof', 'jump', 'kill', 'laugh', 'lead', 'life', 'let', 'look', 'brought', 'dragged', 'king', 'pigeons', 'cat-and-mouse', 'game', 'mouse', 'rain', 'see', 'jumps', 'skin', 'take', 'catnap', 'think', 'meow', 'pajamas', 'whiskers', 'thinks', 'wait'])).

pesi : data base recording the lexical weight of lemmas
We keep track of the lexicographical space occupied by lexical items for weighting purposes. We can thus decrease the factor of computed proximity in the case of 'heavy' lexical items. Lexical weight is computed by LEXDIS itself; it is the weight that LEXDIS assigns to the link between a *Word,Pos* pair and itself. By leaving Word and Pos as variables and executing the query *[Word,Pos,Word,Pos,w:none,m:g]*, we obtain the weights of all the lexical items making up **semdic**.
w(cat, n, 512).
w(dog, n, 522).

LEXDIS is written in SWI-Prolog (see Wielemaker 2003) and runs on standard PCs under the various operating systems for which there exists a SWI-Prolog interpreter/compiler (Windows, Linux, Mac-OS). Appendix A gives the protocol of a short LEXDIS session.

LEXDIS should be used in conjunction with a parser that is able to retrieve the syntactic relations that are targeted by the collocate lists of standard monolingual or bilingual dictionaries. Besides, such a parser should preferably be lexicon-centred, so that multi-word units can be recognized as such. Multi-word units have their own argument structure, in which some of the material can be lexically described, i.e. described at the level of individual lexical entries rather than broad grammatical categories. For example, in *go through the motions*, we identify the various components of the multi-word unit by providing descriptions that go as deep as the specification of individual lexical entries (*go*, *through*, *the*, *motions*), either as lemmas (*go*) or as textual forms (*motions*).

Thus the parser (namely VERBA – see Michiels 2009) will work with entries such as the following two for *go through* (two among a dozen – they are all listed in Appendix D.4):

A.
```
% they finally went through the marriage ceremony for the sake of their children
% they went through the scene over and over again
verb([v(goes,go,went,gone,going,go_through_3_perform_rehearse)],v_mwu_prep,
arglist:[subject:[type:np, canon:0,gappable:yes, oblig:yes,
constraints:[sem:[hum]]],
     athematic:[type:prep,canon:2,gappable:no,
                  oblig:yes,constraints:[lex:through]],
     arg_prep:[type:np,canon:3,gappable:yes,oblig:yes,
                    constraints:[c_str:[head:[lex:Lex]]]]],
ft:[pc:[coll(arg_prep,Lex,[marriage, initiation, scene, lesson, programme,
               ceremony, formality, procedure])]]).
```
B.
```
% go through the motions
verb([v(goes,go,went,gone,going,go_through_the_motions)],v_mwu_prep,
arglist:[subject:[type:np, canon:0,gappable:yes, oblig:yes,
constraints:[sem:[hum]]],
```

```
     athematic:[type:prep,canon:2,gappable:no,
                   oblig:yes,constraints:[lex:through]],
          arg_prep:[type:np,canon:3,gappable:no,oblig:yes,
constraints:[c_str:[det:[lex:the]],
        c_str:[head:[txt:motions]]]]],
     ft:[]).
```

In the first (A) we have one constraint specified down to the level of lexis, namely the lexeme value of the preposition (*through*), whereas in the second (B), we also reach the lexical level in specifying the constraints on the argument of the preposition: namely, that the determiner should be *the* and the textual form of the head noun phrase should be *motions*. In (A), the constraints on the argument of the preposition target a variable Lex which will be passed on to the *coll* procedure. This procedure will call on LEXDIS to measure the proximity of this lexeme with respect to each of the members of the collocate list associated with the argument of the preposition, namely marriage, initiation, scene, lesson, programme, ceremony, formality and procedure.

It should be stressed again that here as in innumerable similar cases, the only way to keep the various word senses apart is to exploit the collocate lists. And it is very uncommon for the lemma of the head of the textual argument to happen to be one of the collocates, however 'typical' the collocates are supposed to be[2]. We do need such a tool as LEXDIS to be able to make full use of the information on the environment of the argument bearer captured by the collocate lists.
Obviously, the parser must be able to keep track of the disruptions to the canonical order of the arguments brought about by various types of 'transformations'. In
I know the rituals inspectors are expected to go through.

whose parse is given in appendix D, VERBA must be able to retrieve the syntactic link between *through* and *rituals* despite the disruption brought about by relativization. Similarly, the subject raising due to *expect* should not prevent the parser from assigning '*inspectors*' as subject of '*go through*'. Otherwise, the constraints specified in the lexical entry could not be met (constraints requiring that the subject of *go through* should bear a +HUM semantic feature, and that the head of the prepositional object should be available for LEXDIS to compute its proximity with respect to the eight collocates).

Similarly, in our second entry, we need to recognize the lexical chain *the motions* in the appropriate syntactic slot, but we also need to be able to keep track of the subject, which can be embedded as far down as in the possessive adjective of a deverbal noun, as in:
She knew his refusal to go through the motions.
where the *his* of *his refusal* yields the subject of *go through* (third person personal pronoun, singular masculine).
Of course, in
I know the rituals inspectors are expected to go through.
the head of the textual argument of the preposition will be matched, not only against the eight elements of the collocate list associated with the 'perform' reading of *go through*, but with all the collocate lists that belong to all the readings of *go through* that are posited by the dictionary we use (the Oxford Dictionary of Current Idiomatic English, with additional collocates taken over from the Oxford/Hachette and Robert/Collins bilingual dictionaries - see Appendix D.4). The list numbers 44 items, given below in alphabetical order:
[apprenticeship, argument, beer, ceremony, clothes, cupboard, document, drink, edition,

2 In the case of our *go through* entries, such identity between textual filler of the argument and corresponding collocate is most likely in the very restricted reading of *go through* as **be published,** where the collocates *printing* and *edition* will often be found as textual exponents of the argument of the preposition.

experience, experiment, fact, file, fire, food, formality, fortune, initiation, lesson, list, luggage, mail, marriage, money, operation, ordeal, pain , paper, phase, pocket, printing, procedure, process, programme, room, scene, stage, stock, store, subject, suitcase, text, trunk, wardrobe]
The task of running through all these collocates, matching them all against the textual candidate, is likely to be heavy on computer resources, as indeed it is[3]. This is even more the case if LEXDIS is embedded in a parser such as VERBA, whose design features are dictated entirely by perspicuity and ease of use by a linguist and/or lexicographer. VERBA is indeed purely incremental, building structure on top of structures established in a previous pass, and making use of feature unification all through.
However, we have got accustomed to seeing the availability of computer resources increase dramatically over time, so that an emphasis on clear design principles rather than efficiency is a reasonable choice to make, in a world where you can't have your cake and eat it, to end this paper on a multi-word unit (... avoiding a near miss).

References

Lexicographical resources

CIDE = *Cambridge International Dictionary of English*. Cambridge University Press, Cambridge, England, 1995
COBUILD = *The Collins COBUILD English Language Dictionary,* edited by J. Sinclair *et al.,* HarperCollins, London and Glasgow, 1987
LDOCE = *Longman Dictionary of Contemporary English* edited by Paul Procter, Longman, Harlow, 1978
ODCIE = *Oxford Dictionary of Current Idiomatic English* (Vol.1: Verbs With Prepositions and Particles), edited by A.P. Cowie and R. Mackin, Oxford University Press, London, 1975
OH = Oxford/Hachette English/French pair (*The Oxford-Hachette French Dictionary French-English English-French* edited by Marie-Hélène Corréard and Valerie Grundy, Oxford University Press, Oxford, Hachette, Paris, 1994)
RC = Le Robert and Collins English/French pair (*Collins Robert French/English, English/French Dictionary*, Unabridged, Third Edition, edited by Beryl T. Atkins, Alain Duval and Rosemary C. Milne, Harper-Collins Publishers, 1993 (First ed. 1978)
WordNet = WordNet 3.0 Prolog files (see Miller 1990)

Other references

Church, K., and Hanks, P. (1989). 'Word Association Norms, Mutual Information and Lexicography,' *Association for Computational Linguistics,* Vancouver, Canada
Ide, N. and Véronis, J. (1998). 'Word Sense Disambiguation: The State of the Art', *Computational Linguistics*, 1998, 24(1)
Kilgarriff, A. (1999), ' “I don't believe in word senses” ', ITRI-97-12, *Information Technology Research Institute Technical Report Series*, ITRI, University of Brighton, Brighton, March, 1999
Lesk, M. (1986). 'Automated Sense Disambiguation Using Machine-readable Dictionaries: How to Tell a Pine Cone from anIce Cream Cone.' *Proceedings of the 1986 SIGDOC Conference,* Toronto, Canada, June 1986, 24-26.
Michiels, A. (1999). 'An Experiment in Translation Selection and Word Sense Discrimination**'**, in

3 LEXDIS takes only about 6 seconds and a half to answer all the 13 queries of the **quicktest** file given in Appendix A, but the full parsing of *I know the rituals inspectors are expected to go through* (Appendix D), producing five parses corresponding to five 'readings' of *go through,* takes about 20 seconds cputime (cputime : 21.4033). The parsing of the parallel sentence *I know the books inspectors are expected to write,* yielding a single parse and making no call on LEXDIS, takes 2.5 seconds cputime.

Tops, Guy, Devriendt, Betty and Geukens, Steven (eds), *Thinking English Grammar,* Orbis/Supplementa, Tome 12, Peeters, Leuven-Paris, 1999, pp. 383-407
Michiels, A. (2000). 'New Developments in the DEFI Matcher', *International Journal of Lexicography,* Vol. 13, No 3, 2000, pp.151-167
Michiels, A. (2001).' DEFI, un outil d'aide à la compréhension', in *Actes du Congrès TALN 2001*, Tours, 2001, pp. 283-293
Michiels, A. (2002). 'Le traitement de la phraséologie dans DEFI', Van Vaerenbergh, Leona (ed), *Linguistics and Translation Studies. Translation Studies and Linguistics*. Linguistica Antverpiensia. New Series 1/2002, 2002, pp. 349-364
Michiels, A. (2006). 'Les lexies en TAL', in Bracops,M., Dalcq, A.-E., Goffin, I., Jabé, A., Louis, V. and Van Campenhoudt, M., eds., 2006 : *Des arbres et des mots. Hommage à Daniel Blampain*, Bruxelles, Éditions du Hazard, ISBN : 2-930154-14-4. (http://hdl.handle.net/2268/1884)
Michiels, A (2009). ', a Multi-word-unit-oriented Feature-unification-based Parser', Unpublished paper, University of Liège, 2009 (available as

http://promethee.philo.ulg.ac.be/engdep1/download/prolog/lexdis/verba.pdf)
Miller, G. (1990). 'Wordnet: An on-line lexical database.' *International Journal of Lexicography (special issue)*, 3(4):235–312.
Montemagni, S., Federici, S. and Pirrelli, V. (1996). 'Example-based Word Sense Disambiguation: a Paradigm-driven Approach', in *Euralex'96 Proceedings*, Göteborg University, 151-160.
Pustejovsky, J. (1995). *The Generative Lexicon* The MIT Press 1995.
Wielemaker, J. (2003). 'An overview of the SWI-Prolog Programming Environment', in Fred Mesnard, F. and Serebenik, A., eds, *Proceedings of the 13th International Workshop on Logic Programming Environments,* Katholieke Universiteit Leuven, Heverlee, Belgium, 2003

APPENDICES

A. *Sample LEXDIS Session*

Computing lexical distance...
A. Michiels, University of Liège

Input file? [stdin. or filename.] --> quicktest.
Results file? [filename.] --> quicktest.

[gun, n, hunter]
gun with POS=n is related to hunter with POS=n with weight=19 as follows:
Shared Labels: [hfzh] -> weight: 4
Shared words in definition: [protect, metal, sport, hunting] -> weight: 12
Shared words in examples: [big] -> weight: 1
Sharing WordNet path -> weight: 2

[gun, n, hunger]
gun with POS=n is related to hunger with POS=n with weight=1 as follows:
Shared words in examples: [started] -> weight: 1

[bread, n, hunger]
bread with POS=n is related to hunger with POS=n with weight=4 as follows:
Shared words in definition: [food, formal] -> weight: 4

[bread, n, gun]

bread with POS=n is ***not*** related to gun with POS=n (Adjusted Weight = 0).

[cat, n, dog]
cat with POS=n is related to dog with POS=n with weight=72 as follows:
Shared Labels: [am] -> weight: 4
Shared words in definition: [animal, man, term, metal, moving, woman, catch, four-legged, pet, member, animals, life] -> weight: 36
Shared words in examples: [dogs, pet] -> weight: 2
Cooccurrence in collocate lists -> weight: 10
Cooccurrence in Roget's thesaurus -> weight: 3
Cooccurrence in R/C-Oxf/Hach indic db -> weight: 4
Cooccurrence in R/C-Oxf/Hach extended lemma db -> weight: 8
Sharing WordNet path -> weight: 5

[cat, n, fox]
cat with POS=n is related to fox with POS=n with weight=47 as follows:
Shared Labels: [am] -> weight: 4
Shared words in definition: [animal, belonging, family, wild, woman, mammal, thick, fur, small, furry, tail] -> weight: 33
Cooccurrence in Roget's thesaurus -> weight: 1
Cooccurrence in R/C-Oxf/Hach indic db -> weight: 4
Sharing WordNet path -> weight: 5

[serpent, n, snake]
serpent with POS=n is related to snake with POS=n with weight=114 as follows:
Shared Labels: [am] -> weight: 4
Shared words in definition: [resembles, snake, serpent, ophidian, limbless, scaly, elongate, reptile, venomous, large] -> weight: 70
Cooccurrence in Roget's thesaurus -> weight: 8
Sharing WordNet path -> weight: 32

[serpent, n, bird]
serpent with POS=n is related to bird with POS=n with weight=4 as follows:
Shared words in definition: [creature] -> weight: 2
Cooccurrence in Roget's thesaurus -> weight: 2

[pool, n, water]
pool with POS=n is related to water with POS=n with weight=127 as follows:
Shared words in definition: [water, large, ground, swimming, plants, liquid, amount, surface, pool, lake, body, area, supply, goods, thin, covered] -> weight: 88
Shared words in examples: [pool, swimming, blood, water, body, pounds, men] -> weight: 17
Cooccurrence in R/C-Oxf/Hach indic db -> weight: 5
Cooccurrence in R/C-Oxf/Hach collocates db -> weight: 5
Sharing WordNet path -> weight: 12

[pool, n, house]
pool with POS=n is related to house with POS=n with weight=41 as follows:
Shared words in definition: [large, small, money, organization, purpose, body, goods, take, gambling, business] -> weight: 30
Shared words in examples: [football, won, small, stood, cheap] -> weight: 5
Sharing WordNet path -> weight: 6

[pool, n, football]
pool with POS=n is related to football with POS=n with weight=74 as follows:

Shared Guide Words: [game] -> weight: 12
Shared words in definition: [filled, large, game, games, played, try, win, football, players] -> weight: 47
Shared words in examples: [garden, local, want, play] -> weight: 9
Cooccurrence in Roget's thesaurus -> weight: 1
Cooccurrence in R/C-Oxf/Hach extended lemma db -> weight: 5

[pool, n, tennis]
pool with POS=n is related to tennis with POS=n with weight=27 as follows:
Shared words in definition: [small, game, played, area, players, hit] -> weight: 18
Shared words in examples: [play] -> weight: 6
Cooccurrence in Roget's thesaurus -> weight: 1
Sharing WordNet path -> weight: 2

[pool, n, football, n, w:t, m:l]
pool with POS=n and Idnum=co44102 is related to football with POS=n and
Idnum=ci27328>ci69956*co22637#co22638/lg29686&football%1:04:00:: with weight=24 as follows:
Shared words in definition: [try, win] -> weight: 24

[pool, n, water, n, w:t, m:l]
pool with POS=n and Idnum=lg43110 is related to water with POS=n and Idnum=ci83118 with weight=52 as follows:
Shared words in definition: [area] -> weight: 42
Shared words in examples: [] -> weight: 10

nadamas
cputime : 6.42724

End of input... Always glad to be able to help... Bye!

As an afterthought: LEXDIS on the link between 'picture' and 'photograph'

'Neither of these preprocesses, though, can help highlight the "natural" similarity between nouns such as "picture" and "photograph". Although one might imagine a preprocess that would help in this particular case, there will probably always be a class of generalizations that are obvious to an intelligent lexicographer, but lie hopelessly beyond the objectivity of a computer.'
Church and Hanks (1989:82).

QUERY: [picture, n, photograph, n, m:g]

picture with POS=n is related to photograph with POS=n with weight=195 as follows:
Shared Labels: [pg] -> weight: 4
Shared Guide Words: [image, photo] -> weight: 24
Shared words in definition: [film, representation, picture, pic, form, image, object, scene, produced, view, camera, printed, photo, exposure, print, transparent, slide, recorded, light-sensitive, material] -> weight: 100
Shared words in examples: [good, colour, photographs, black-and-white, aerial, nude, signed, elvis, presley, autographed, parents, took, small, worlds, french, inventor, j, n, ni,pce] -> weight: 24
Cooccurrence in collocate lists -> weight: 3
Cooccurrence in Roget's thesaurus -> weight: 1
Cooccurrence in R/C-Oxf/Hach indic db -> weight: 5
Cooccurrence in R/C-Oxf/Hach extended lemma db -> weight: 2
Sharing WordNet path -> weight: 32

B. Friends of the doctor...

List derived from the results of the LEXDIS query : `friends(doctor,n,w:24).`

i.e. return all the pairs Word, Pos for which the match with the noun 'doctor' reaches the threshold of 24 in global mode

['Church Father', n, 26,'Dr.', n, 105,'Father', n, 26,'Father of the Church', n, 26,'MD', n, 91,abortionist, n, 34,academic, adj, 30,academic, n, 50,academician, n, 29,alum, n, 24,alumnus, n, 24,anaesthetist, n, 40,anesthesiologist, n, 30,anesthetist, n, 30,anteroom, n, 24,application, n, 25,attend, v, 53,autopsy, n, 24,be, v, 51,'bedside manner', n, 38,better, adj, 26,bleep, v, 28,bonesetter, n, 36,call, n, 28,call, v, 29,'call in', v, 24,case, n, 48,'check-up', n, 31,chemist, n, 37,clinic, n, 30,clinician, n, 31,college, n, 68,conjurer, n, 25,consultant, n, 34,consulting, adj, 25,couch, n, 34,'country doctor', n, 29,cut, v, 37,degree, n, 45,dentist, n, 42,dermatologist, n, 32,diagnostician, n, 30,diet, n, 30,do, v, 57,doc, n, 121,doctor, n, 280,doctoral, adj, 28,doctorate, n, 42,dope, n, 26,dr, n, 94,'ethical drug', n, 28,examine, v, 30,extern, n, 32,'eye doctor', n, 30,'family doctor', n, 38,fee, n, 28,fellow, n, 25,fireman, n, 27,fix, v, 32,'flying doctor', n, 26,forceps, n, 28,full, adj, 25,game, n, 32,general, adj, 40,general, n, 33,general, v, 25,'general practice', n, 25,'general practitioner', n, 49,geriatrician, n, 34,get, v, 40,go, v, 43,good, adj, 25,gp, n, 37,graduate, n, 25,gynaecologist, n, 28,gynecologist, n, 28,haematologist, n, 28,have, v, 41,hematologist, n, 28,high, adj, 30,historian, n, 24,hold, v, 31,homoeopath, n, 31,honorary, adj, 32,house, n, 34,'house physician', n, 38,houseman, n, 36,hurt, v, 24,ill, adj, 37,ill, adv, 28,ill, n, 28,injection, n, 31,institution, n, 38,intern, n, 43,job, n, 38,job, v, 26,leech, n, 37,licentiate, n, 24,locum, n, 47,lpn, n, 29,make, v, 26,malpractice, n, 36,man, n, 33,man, v, 27,master, n, 59,matron, n, 31,md, n, 33,medic, n, 45,medical, adj, 41,medical, n, 50,'medical extern', n, 32,medicine, n, 44,medicine, v, 29,medico, n, 105,neurologist, n, 32,neurosurgeon, n, 28,nurse, n, 93,nurse, v, 50,nursing, n, 27,obstetrician, n, 40,oculist, n, 35,office, n, 59,operation, n, 28,ophthalmologist, n, 35,order, v, 30,paediatrician, n, 36,paramedic, n, 24,pass, v, 26,pathologist, n, 30,patient, n, 69,pediatrician, n, 36,pharmacy, n, 26,phd, n, 53,physician, n, 133,placebo, n, 35,play, v, 36,practice, n, 57,practise, v, 52,practitioner, n, 54,prescribe, v, 42,prescription, n, 42,'prescription drug', n, 28,'prescription medicine', n, 28,pretend, adj, 27,pretend, n, 25,pretend, v, 29,proctologist, n, 28,profession, n, 56,prognosis, n, 30,psychiatrist, n, 33,pulse, n, 29,put, v, 31,quack, n, 61,qualified, adj, 53,radiologist, n, 30,receptionist, n, 28,record, n, 26,refer, v, 33,registrar, n, 37,regular, adj, 24,repair, n, 24,repair, v, 45,resident, n, 34,restore, v, 28,run, v, 24,rx, n, 24,sawbones, n, 42,scholar, n, 26,school, n, 40,scrub, v, 27,section, n, 25,see, v, 55,set, v, 33,shot, n, 27,sick, adj, 34,sick, n, 25,skill, n, 24,'skin doctor', n, 28,spatula, n, 26,specialist, n, 51,stethoscope, n, 28,stitch, n, 25,stitch, v, 26,'strike off', v, 27,student, n, 32,surgeon, n, 72,surgery, n, 67,swab, n, 44,syringe, v, 28,take, adv, 26,take, n, 28,take, v, 77,theologian, n, 28,title, n, 35,train, v, 32,treat, n, 26,treat, v, 38,university, n, 44,use, v, 24,vet, n, 44,vet, v, 35,veterinarian, n, 40,veterinary, n, 32,'veterinary surgeon', n, 43,visit, n, 29,visit, v, 29,'waiting room', n, 26,'witch doctor', n, 24,'x-ray', n, 33]

C. *Working on a triplet in local mode*

A 'triplet' query in LEXDIS can take either of the following two formats:
t(ArgBearer, PosArgBearer, Arg1, PosArg, Arg2, PosArg), e.g. *t(wear, v, tie, n, watch, n)*

or
t(ArgBearer, Arg1, Arg2) e.g. *t(wear, tie, watch)*
where v is filled in by the system as default pos for the ArgBearer and n for the Args.
LEXDIS computes the triplets of best matches for the following relations:
ArgBearer – FirstArg
ArgBearer – Second Arg
FirstArg – SecondArg
and selects the word senses (if any) that occur in more than one relation.
Example : QUERY: t(wear, tie, watch)
Here are the three triplets (the numerical value is the weight of the match between the two word senses labelled by the Identification Numbers in the *p* structures):
I . WEAR – TIE
[3-p(wear, ci83351, tie, ci77397/lg53208#co60228&tie%1:06:01::),
2-p(wear, ci83339, tie, co60229),
2-p(wear, co64915, tie, co60233)]
II. WEAR – WATCH
[22-p(wear, ci83339, watch, ci83082#co64601),

22-p(wear, wear%2:29:01::, watch, ci83082#co64601),
22-p(wear, wear%2:35:00::, watch, ci83082#co64601)]

III. TIE-WATCH
[2-p(tie, ci77397/lg53208#co60228&tie%1:06:01::, watch, ci83082#co64601),
2-p(tie, ci77397/lg53208#co60228&tie%1:06:01::, watch, lg55831),
2-p(tie, ci77397/lg53208#co60228&tie%1:06:01::, watch, lg55833)]

The simple algorithm outlined above leads to the selection of the correct readings:

Wear ci83339 is selected because it appears in both relations I and II:
wear v [body] [] ci83339 [tracey, wearing, simple, black, dress, cotton, lycra, mix, carolines, wedding, complaining, musicians, rings, theyre, playing, wears, glasses, reading]
Def: to have clothing or jewellery on your body

Tie ci77397/lg53208#co60228&tie%1:06:01:: is selected because it appears in both relations I and III:
tie n [necktie] [cl] ci77397/lg53208#co60228&tie%1:06:01:: [stood, front, mirror, tightening, necktie, wore, vest, took, jacket, loosened, wearing, jumper, sports, suit, silk, gift]
Def: [neckwear consisting of a long narrow piece of material worn (mostly by men) under a collar and tied in knot at the front; "he stood in front of the mirror tightening his necktie"; "he wore a vest and tie", a tie is a long narrow piece of cloth that is worn round the neck under a shirt collar and tied in a knot at the front ties are worn mainly by men see also bow tie old school tie, also esp ame necktie a band of cloth worn round the neck usu inside a shirt collar and tied in a knot at the front, a tie also esp am necktie is a long thin piece of material that is worn under a shirt collar esp by men and tied in a knot at the front]

Watch ci83082#co64601 is selected because it appears in both relations II and III:
watch n [clock] [] ci83082#co64601 [digital, analogue, ian, bought, elaine, gold, christmas, seems, says, sure, later, glanced, nervously, looked, stopped]
Def: [a small clock which is worn on a strap around the wrist or sometimes connected to a piece of clothing by a chain, a watch is a small clock which you wear on a strap on your wrist or on a chain]

It would thus seem that – in this particular case at least - the simple algorithm outlined above is powerful enough to enable LEXDIS to cash in on the word sense connectivity assumed by the connector linking the two arguments to the argument bearer. But we do not claim that LEXDIS is appropriate as a word sense discriminator on its own – it should be used in conjunction with other tools (such as in the first place a parser) and is first and foremost meant to enhance the usefulness of lexical information provided by the very sources from which LEXDIS is built, as we have tried to show with indicators and collocate lists.

D. *A sample parse with word-sense identification.*

1. String :

I know the rituals inspectors are expected to go through.

2. WordList:

[0/i, 1/know, 2/the, 3/rituals, 4/inspectors, 5/are, 6/expected, 7/to, 8/go, 9/through, endpos(10)]

3. Pretty-printed parse

cat:pred
  voice:active
  weight_coll:0
  c_str
    head
      cat:vg
      pos:v
      lex:know
      tense:present
      voice:active
    subject
      cat:np
      sem:[hum]
      lex:i
      index:i(0, 1)
      c_str
        head
          lex:i
          sem:[hum]
    object
      cat:np
      weight_coll:**25-ceremony-10**
      index:i(2, 10)
      sem:[abstract]
      lex:ritual
      c_str
        head
          cat:np
          sem:[abstract]
          lex:ritual
          index:i(2, 4)
          c_str
            det
              pos:det
              lex:the
            head
              pos:n
              lex:ritual
              sem:[abstract]

```
rel_clause
        index:i(2, 4)
        sem:[abstract]
        weight_coll:0
        c_str
            head
                auxgroup:[tense:present]
                prop:[voice:passive]
                pos:v
                lex:expect
                tense:untensed
                voice:passive
            subject
                cat:np
                sem:[hum]
                lex:inspector
                index:i(4, 5)
                c_str
                    det
                        det
                        zero
                    head
                        pos:n
                        lex:inspector
                        sem:[hum]
            object
                cat:pred
                voice:active
                weight_coll:0
                c_str
                    head
                        auxgroup:[tense:untensed]
                        pos:v
                        lex:go_through_3_perform_rehearse
                        tense:untensed
                        voice:active
                    subject
                        e:i(4, 5)
                    arg_prep
                        e:i(2, 4)
```

4. VERBA lexical entries for GO THROUGH:

adapted from ***Oxford Dictionary of Current Idiomatic English***
Vol 1 : Verbs with Prepositions and Particles
(with additional collocates from RC/OH E->F dictionaries)

```
% it didn't take Albert very long to go through his inheritance
verb([v(goes,go,went,gone,going,go_through_1_consume)],v_mwu_prep,
arglist:[subject:[type:np, canon:0,gappable:yes, oblig:yes,
constraints:[sem:[hum]]],
     athematic:[type:prep,canon:2,gappable:no,
                  oblig:yes,constraints:[lex:through]],
          arg_prep:[type:np,canon:3,gappable:yes,oblig:yes,
constraints:[c_str:[head:[lex:Lex]]]]],
     ft:[pc:[coll(arg_prep,Lex,[money, food, drink])]]).

% it was obvious that the room had been gone through by an intruder
verb([v(goes,go,went,gone,going,go_through_2_search)],v_mwu_prep,
arglist:[subject:[type:np, canon:0,gappable:yes, oblig:yes,
constraints:[sem:[hum]]],
     athematic:[type:prep,canon:2,gappable:no,
                  oblig:yes,constraints:[lex:through]],
          arg_prep:[type:np,canon:3,gappable:yes,oblig:yes,
constraints:[c_str:[head:[lex:Lex]]]]],
     ft:[pc:[coll(arg_prep,Lex,[room, pocket, clothes, cupboard, wardrobe,
                   luggage, suitcase, trunk])]]).

% they finally went through the marriage ceremony for the sake of their children
% they went through the scene over and over again
verb([v(goes,go,went,gone,going,go_through_3_perform_rehearse)],v_mwu_prep,
arglist:[subject:[type:np, canon:0,gappable:yes, oblig:yes,
constraints:[sem:[hum]]],
     athematic:[type:prep,canon:2,gappable:no,
                  oblig:yes,constraints:[lex:through]],
     arg_prep:[type:np,canon:3,gappable:yes,oblig:yes,
                    constraints:[c_str:[head:[lex:Lex]]]]],
ft:[pc:[coll(arg_prep,Lex,[marriage,initiation,scene, lesson,programme,
               ceremony, formality, procedure])]]).

% their proposal should be gone through with the utmost care
verb([v(goes,go,went,gone,going,go_through_4_examine)],v_mwu_prep,
arglist:[subject:[type:np, canon:0,gappable:yes, oblig:yes,
constraints:[sem:[hum]]],
     athematic:[type:prep,canon:2,gappable:no,
                  oblig:yes,constraints:[lex:through]],
          arg_prep:[type:np,canon:3,gappable:yes,oblig:yes,
constraints:[c_str:[head:[lex:Lex]]]]],
     ft:[pc:[coll(arg_prep,Lex,[fact, argument, subject, file,
                   mail, text, list, document])]]).

% his book went through ten editions in a year
verb([v(goes,go,went,gone,going,go_through_5_be_published)],v_mwu_prep,
arglist:[subject:[type:np, canon:0,gappable:yes, oblig:yes,
constraints:[sem:[document],c_str:[head:[lex:LexSubj]]]],
     athematic:[type:prep,canon:2,gappable:no,
                  oblig:yes,constraints:[lex:through]],
          arg_prep:[type:np,canon:3,gappable:yes,oblig:yes,
constraints:[c_str:[head:[lex:LexPrepArg]]]]],
     ft:[pc:[coll(subject,LexSubj,[book, title, article]),
```

```
       coll(arg_prep,LexPrepArg,[printing, edition])]]).

% he would have gone through fire for the girl he loved
verb([v(goes,go,went,gone,going,go_through_6_endure_experience)],v_mwu_prep,
arglist:[subject:[type:np, canon:0,gappable:yes, oblig:yes,
constraints:[sem:[hum]]],
     athematic:[type:prep,canon:2,gappable:no,
                oblig:yes,constraints:[lex:through]],
          arg_prep:[type:np,canon:3,gappable:yes,oblig:yes,
constraints:[c_str:[head:[lex:Lex]]]]],
     ft:[pc:[coll(arg_prep,Lex,[operation, pain, ordeal, apprenticeship,
                 fire, phase, stage, process,
                 experience, experiment])]]).

% go through somebody's hands
verb([v(goes,go,went,gone,going,go_through_somebodys_hands)],v_mwu_prep,
arglist:[subject:[type:np, canon:0,gappable:yes, oblig:yes,
constraints:[c_str:[head:[lex:LexSubj]]]],
     athematic:[type:prep,canon:2,gappable:no,
                oblig:yes,constraints:[lex:through]],
          arg_prep:[type:np,canon:3,gappable:yes,oblig:yes,
constraints:[c_str:[det:[type:or([poss_adj,genitive])]],
        c_str:[head:[txt:hands]]]]],
     ft:[pc:[coll(subject,LexSubj,
     [pound, jewellery, diamond, paper, document, patient, case])]]).

% go through the mill
verb([v(goes,go,went,gone,going,go_through_the_mill)],v_mwu_prep,
arglist:[subject:[type:np, canon:0,gappable:yes, oblig:yes,
constraints:[sem:[hum]]],
     athematic:[type:prep,canon:2,gappable:no,
                oblig:yes,constraints:[lex:through]],
          arg_prep:[type:np,canon:3,gappable:no,oblig:yes,
constraints:[ c_str:[det:[lex:the]],
         c_str:[head:[txt:mill]]]]],
     ft:[]).

% go through the motions
verb([v(goes,go,went,gone,going,go_through_the_motions)],v_mwu_prep,
arglist:[subject:[type:np, canon:0,gappable:yes, oblig:yes,
constraints:[sem:[hum]]],
     athematic:[type:prep,canon:2,gappable:no,
                oblig:yes,constraints:[lex:through]],
          arg_prep:[type:np,canon:3,gappable:no,oblig:yes,
constraints:[c_str:[det:[lex:the]],
        c_str:[head:[txt:motions]]]]],
     ft:[]).

% go through the proper channels
verb([v(goes,go,went,gone,going,go_through_the_proper_channels)],v_mwu_prep,
arglist:[subject:[type:np, canon:0,gappable:yes, oblig:yes,
constraints:[c_str:[head:[lex:LexSubj]]]],
     athematic:[type:prep,canon:2,gappable:no,
                oblig:yes,constraints:[lex:through]],
          arg_prep:[type:np,canon:3,gappable:no,oblig:yes,
          constraints:[c_str:[adjp:[c_str:[head:[lex:proper]]]],
                c_str:[head:[txt:channels]]]]],
     ft:[pc:[coll(subject,LexSubj,[application, complaint, letter])]]).
```

5. Test results

In this test we use a WordNet-free LEXDIS. The point is to enable comparison with results obtained using a WordNet baseline.

Our test sentences are derived from genuine citations, but which have had to be drastically simplified for our experimental parser to be able to deal with them. In each case the lexeme of the head of the relevant argument in the user's text (the argument for which the argument-bearing predicate provides a collocate list) is matched against every member of the collocate list. The table records the best match as well as the average lexical distance between the textual item and the whole of the collocate list. For instance, in

*He thought that she had gone through the **bins**.*

bin will be matched against the various collocate lists for the prepositional object. Let's concentrate on the list providing the match that ought to come top of the class, namely the one associated with the word sense individuated by the near-synonym 'search'. The list is the following:

[room, pocket, clothes, cupboard, wardrobe, luggage, suitcase, trunk]

LEXDIS therefore runs through the following matches :

bin/room, bin/pocket, bin/clothes, bin/cupboard, bin/wardrobe, bin/luggage, bin/suitcase and bin/trunk

the best match among the eight being *bin/pocket*, with a lexical proximity factor of 19. The average value of the eight matches is 8.

A rough-and-ready measure of the quality of the match is provided by adding up the two values, the best and the average. In the table, the winner according to LEXDIS is in *italics*, whereas the 'right' (i.e. human-selected) triple is printed in **bold** (the winner is consequently in ***bold italics*** when LEXDIS makes the right selection). We can see that LEXDIS is also useful in indicating how 'far' from the 'right' reading a 'wrong' reading is: compare the results for the discarded interpretations of *go through/training* and *go through/ritual*.

Go through the proper channels and *go through the motions* present no problems at all; LEXDIS results are not to be taken into account when a multi-word unit reading is captured thanks to the presence in the user's text of lexical elements permitting the identification of the multi-word-unit (*proper channels* and *the motions*).

Perhaps a remark is in order regarding the relationship between the highest value for proximity and the average one. If the two values diverge widely (let's say something like a 10:1 ratio), we might enquire whether we do not have a problem, either with the collocate list or with LEXDIS itself. The various elements of a collocate list should describe a lexical universe displaying a certain degree of coherence; this hypothesis is at the very basis of using co-presence in a collocate list in order to relate two lexical items (cf. Montemagni et al. 1996), a property that LEXDIS itself uses (by exploiting the MT database - cf. above). In fact, the relationship between the top value and the

average, once the coherence of LEXDIS itself is better established, could be used as a marker for a need to revise the collocate lists, and perhaps the division of the lexical space covered by the item the collocate lists are attached to.

Results using LEXDIS

GO THROUGH	consume	search	perform_ rehearse	examine	endure _experience	be_ published
He thought that she had gone through the **bins**.	3 drink 1	***19*** ***pocket*** ***8***	3 ceremony 1	17 file 5	5 fire 1	_
The inspector wanted the teachers to go through his **report**.	12 money 6	6 pocket 1	21 programme 8	***34*** ***document*** ***15***	14 process 6	_
She was expected to go through a daily **ritual**.	2 food 1	5 room 1	***25*** ***ceremony*** ***10***	5 text 1	12 operation 4	_
We know the **suffering** you went through.	2 food 1	2 room 0	3 scene 1	5 subject 1	***31*** ***pain*** ***6***	_
It is the **training** the students will be expected to go through.	15 money 9	12 pocket 4	17 scene 7	7 subject 1	***21*** ***experience*** ***8***	_
You don't know the **torture** I have gone through.	4 food 2	6 room 1	6 scene 2	7 subject 3	***32*** ***pain*** ***6***	_
I think that they went through five **bottles**.	***53*** ***drink*** ***20***	27 trunk 9	4 scene 1	19 file 4	8 pain 2	1 printing 0
The report went through five **editions**.	_	_	_		_	subj***:33*** ***article*** ***22*** *pobj****:1000*** ***edition*** ***508***
Your report should go through **the proper channels.**	go	through	***the***	***proper***	***channels***	
She liked his refusal to go through **the motions**.	go	through	***the***	***motions***		

'Baseline' results using WordNet relations[4]

Note that in 'I think that they went through five **bottles**', the wrong reading is selected. But it is only when we attempt to measure the semantic proximity between items with different parts of speech (as in the 'triplet' experiment) that LEXDIS really comes into its own: WordNet is POS-bound, like Roget's, whereas LEXDIS has means of crossing the POS boundaries.

GO THROUGH	consume	search	perform_ rehearse	examine	endure _experience	be_ published
He thought that she had gone through the **bins**.	2 drink 0	***6*** ***luggage*** ***1***	0 nil 0	5 mail 0	3 phase 0	_
The inspector wanted the teachers to go through his **report**.	0 nil 0	0 nil 0	21 programme 6	***24*** ***text*** ***10***	7 operation 1	_
She was expected to go through a daily **ritual**.	0 nil 0	0 nil 0	***60*** ***marriage*** ***10***	2 mail 0	31 operation 4	_
We know the **suffering** you went through.	0 nil 0	0 nil 0	4 marriage 1	0 nil 0	***30*** ***pain*** ***4***	_
It is the **training** the students will be expected to go through.	0 nil 0	0 nil 0	25 ceremony 6	5 mail 0	***43*** ***operation*** ***6***	_
You don't know the **torture** I have gone through.	0 nil 0	0 nil 0	2 procedure 0	0 nil 0	***26*** ***pain*** ***3***	_
I think that they went through five **bottles**.	**2** **drink** **0**	*10* *luggage* *3*	0 nil 0	4 mail 0	3 phase 0	0 nil 0
The report went through five **editions**.	_	_	_		_	*subj*:***32*** ***book*** ***25*** *pobj*:***1000*** ***edition*** ***505***

4 As stated in the section on resources, the WordNet relations used are *cs*, *ent*, *ins*, *hyp*, *mm*, *mp* and *ms*. They are recursively followed down to depth 3; the weights are assigned intuitively, as in LEXDIS, and would certainly have to be revised in a proper implementation.