MACHINE TRANSLATION REVIEW

The Periodical
of the
Natural Language Translation Specialist Group
of the
British Computer Society
Issue No. 8
October 1998

The *Machine Translation Review* incorporates the Newsletter of the Natural Language Translation Specialist Group of the British Computer Society and appears twice yearly.

The Review welcomes contributions, articles, book reviews, advertisements, and all items of information relating to the processing and translation of natural language. Contributions and correspondence should be addressed to:

Derek Lewis
The Editor
Machine Translation Review
School of Modern Languages
Queen's Building
University of Exeter
Exeter
EX4 4QH
United Kingdom

Tel: +44 (0)1392 264296
Fax: +44 (0)1392 264377
E-mail: D.R.Lewis@exeter.ac.uk

ISSN 1358–8346

Contents

Group News and Information

Letter from the Chairman

72 Brattle Wood
Sevenoaks
Kent, TN13 1QU

Tel: 01732 455446
Office: 0171 815 7472
Fax: 0171 815 7550
E-mail: wiggjd@sbu.ac.uk

2 December 98

I'm glad to say that the Proceedings of our International Machine Translation Conference at Cranfield in 1994 have now been printed. They cost £28.00 plus postage and packing (UK £4.00). They should be ordered from Douglas Clarke or myself (see Committee list for contact numbers). A list of the contents of the Proceedings is available.

As a taster we include, by kind permission of our keynote speaker, Professor Yorick Wilks, a copy of his opening address at the Conference in this issue.

The other paper in this issue is reprinted by kind permission of the authors, Daniel Sleator and Davy Temperely. The paper was originally presented at the Third International Workshop on Parsing Technologies in August 1993. The LINK software is obtainable, free of charge, via *www.cs.cmu.edu/~sleator* or *www.link.cs.cmu.edu/link*. I believe this is an original approach to parsing language and should be of interest to academics and students alike. I would certainly welcome discussing the merits of the system with anyone interested in evaluating it.

Still welcome are more articles, papers and reports on the subject of machine translation and related subjects such as computer assisted language teaching, computer-based dictionaries and aspects of multilinguality in computing etc. We would welcome papers from academic staff and students in linguistics and related disciplines, and from translators and any other users of MT software.

Perhaps I could remind members that they do not need to live near London to assist the Committee. We do not have sufficient funds to pay travel expenses for all Committee members to attend meetings, but we still welcome Correspondent members who are otherwise treated as full members of the committee and kept advised of all Committee business. Anyone interested in helping should contact me or any other Committee member.

Finally, we are considering organising another Conference in the Autumn of 2000, probably at Exeter University. If you would like to take any part therein or if you have any comment to make about it, please contact Derek Lewis, me or any other Committee member.

All opinions expressed in this *Review* are those of the respective writers and are not necessarily shared by the BCS or the Group.

J.D.Wigg

The Committee

The telephone numbers and e-mail addresses of the Committee are as follows:

David Wigg (Chair)	Tel.: +44 (0)1732 455446 (H)
	Tel.: +44 (0)171 815 7472 (W)
	E-mail: wiggjd@sbu.ac.uk
Monique L'Huillier (Secretary)	Tel.: +44 (0)1276 20488 (H)
	Tel.: +44 (0)1784 443243 (W)
	E-mail: m.lhuillier@vms.rhbnc.ac.uk
Derek Lewis (Editor)	Tel.: +44 (0)1404 814186 (H)
	Tel.: +44 (0)1392 264296 (W)
	E-mail: d.r.lewis@exeter.ac.uk
Douglas Clarke	Tel.: +44(0)1908 373141
Ian Kelly	Tel.: +44(0)1276 857599
	E-mail: 100350.3046@compuserve.com
Veronica Lawson	Tel.: +44(0)171 7359060
	E-mail: 100733.504@compuserve.com
Roger Harris (Rapporteur)	Tel.: +44 (0)181 800 2903 (H)
	E-mail: rwsh@dircon.co.uk
Correspondent Members:	
Gareth Evans (Minority Languages)	Tel.: +44 (0)1792 481144
	E-mail: g.evans@sihe.ac.uk
Ruslan Mitkov	Tel: +44 (0)1902 322471 (W)
	E-mail: R.Mitkov@wlv.ac.uk

BCS Library

Books kindly donated by members are passed to the BCS library at the IEE, Savoy Place, London, WC2R 0BL, UK (tel: +44 (0)171 240 1871; fax: +44 (0)171 497 3557). Members of the BCS may borrow books from this library either in person or by post. All they have to provide is their membership number. The library is open Monday to Friday, 9.00 am - 5.00 pm.

Website

The website address of the BCS-NLTSG is: http://www.bcs.org.uk/siggroup/sg37.htm

Some Notes on the State of the Art: Where are we now in MT, what works and what doesn't? And the role of MT as an international collaborative activity

by

Yorick Wilks
Computer Science Department,
University of Sheffield, UK

Abstract

The paper examines briefly the impact of the 'statistical turn' in machine translation (MT) R&D in the last decade, and particularly the way in which it has made large scale language resources (lexicons, text corpora etc.) more important than ever before and reinforced the role of evaluation in the development of the field. But resources mean, almost by definition, co-operation between groups and, in the case of MT, specifically co-operation between language groups and states.

The paper then considers what alternatives there are now for MT R&D. One is to continue with interlingual methods of translation, even though those are not normally thought of as close to statistical methods. The reason is that statistical methods, taken alone, have almost certainly reached a ceiling in terms of the proportion of sentences and linguistic phenomena they can translate successfully. Interlingual methods remain popular within large electronics companies in Japan, and in a large US Government funded project (PANGLOSS).

The question then discussed is what role there can be for interlinguas and interlingual methods in co-operation in MT across linguistic and national boundaries. The paper then turns to evaluation and asks whether, across national and continental boundaries, it can become a co-operative or a 'hegemonic' enterprise. Finally the paper turns to resources themselves and asks why co-operation on resources is proving so hard, even though there are bright spots of real co-operation.

Introduction: The Debate over the 'Statistical MT' Hypothesis

In the last ten years, empiricism has struck computational linguistics in general and MT in particular, where by empiricism I mean a move to methods based on large scale language data, usually corpora of texts, sometimes including dictionary texts, available on computers, rather than on a priori linguistic theories and rules. One of the most striking examples was the purely statistical approach to machine translation at the IBM Watson Research Laboratories which made use of the very large Canadian English/French parliamentary corpus (Brown et al., 1988). The results were striking: with virtually none of the conventional sources of linguistic knowledge (lexicons, syntax, semantics, etc.), the system produced figures of between 50 and 65% of sentences correctly translated, depending on the relationship of the training to the experimental corpus. Although the result was astonishing to many, more detailed critiques (e.g. Wilks, 1994) have pointed out that the figure has remained static if only pure statistical methods are used, that some linguistic phenomena are seemingly resistant to this approach, that the system, CANDIDE, has never actually beaten SYSTRAN in a direct

competition of unseen texts from areas different from the training corpus, and that the economics of corpus availability and production are probably against any commercial and general development of CANDIDE for new languages.

All that is now in the past, and we can ask what the effect of the IBM work has been on MT and computational linguistics in general. One could say the alternatives are the following:

* Going on with theoretical linguistic development, which one could deem 'linguistics as chemistry', in search of the correct and devastating formula.

* Machine-aided translation, which supplements computational lacunae by having a human in the translation loop, and has been much used in commercial systems.

* Keep on hacking in the hope that, like SYSTRAN, a system can grow to an acceptable level of performance, perhaps by blending the best of statistical and symbolic components.

There are systems, still under development, in both commercial environments and research laboratories that have adopted all these latter day strategies, sometimes more than one at once. One could also argue that all those strategies agree on most of the following morals that can and have been drawn from where we are now for future MT systems.

Future MT Systems

* Unaided statistical methods will probably not be enough for any viable system, commercial or otherwise, since they do not lead to a system that beats SYSTRAN, which is available for a large range of languages.

* One should be more sceptical than ever about a system that works on some data, because all MT systems work to some degree, whatever their assumptions: word-for-word MT as much as pure statistical MT. Coverage is as much a criterion as quality of translation.

* There are proven bags of tricks in MT, as Bar Hillel always argued (1960) and no amount of theoretical research is going to diminish their importance.

* Symbolic and statistical methods can be combined, and that seems to be where most MT research is at the moment.

* Interlingual methods remain popular, in spite of the above, at least in Japan and the US.

* Evaluation continues to drive MT, and helps keep old systems alive. The last ARPA evaluations showed SYSTRAN still very much in the game, but with small commercial upstarts beating the research systems, and much closer to the old, established, and more expensive ones than the latter find comfortable.

* Thanks to IBM, resource driven systems are here to stay, at least for the foreseeable future and Big-Data-Small-Program may still be a good ideal, from SYSTRAN to IBM. Here one can take for contrast theoretically motivated systems like EUROTRA (Johnson et al. 1985).

Let us now turn to some issues at the junction of resources, evaluation and interlinguas.

Modalities of International Cross-Language Co-Operation

Co-operation is now crucial to MT because resource creation demands it, and resources are now considered crucial to MT by all except those still firmly committed to formal linguistic methods, and who have therefore effectively withdrawn from empirical and evaluation-driven MT.

Obvious types of co-operation are:

* between monolingual groups within states (usually monolingual)

* between monolingual groups within the (multilingual) EU

* between groups or state organisations within blocs (US, EU, Japan), where one of those blocs is monolingual, one multilingual, and one (The US) with aspects of both.

The next question is: what should be the basis of that co-operation if it is across languages and cultures (e.g. in writing the analysis, generation and transfer modules of a conventionally structured MT system)?

Should it be on the basis of:

* each partner doing what they do best (as opposed to everyone doing and redoing everything)?

* each partner doing their own language (as opposed to 'I'll help you with yours')?

* each partner doing their own interlinguas (as opposed to 'I'll believe more in mine if you can use it too')?

* each partner doing their own evaluation of their own modules (as opposed to 'I'll evaluate yours and you mine')?

But, historically not all insight is from inside a language: one has only to think of the early keyboards for Chinese, which came from the West, and the fact that Jespersen, a Dane, produced the first full descriptive linguistic grammar for English. The recent morpholympics competition was, I think, won by a Finnish analyser of German which beat all the groups from Germany.

Genuine co-operation, on the other hand, can include offers such as the free availability of JUMAN, the Japanese segmenter from Kyoto University, which is of the 'I'll help you do my language' type, and which is quite different from 'I'll do mine and you do yours', an attitude which drastically limits possible forms of co-operation. On the other hand, the new Finnish constraint parser for English (Karlsson, 1990) is 'I'll help you do yours'. If one doubts the need for this kind of thing, I can cite from personal experience the project at CRL-NMSU which built a Spanish lexicon from an English one largely because we could not find a Spanish machine-readable lexicon at all.

Consider, as part of this issue, the problem of the mutual perceptions of Japanese and English speakers: each group sees their own language as mysterious and hard to specify by rules. The proof of this, for English speakers, is that vast numbers of foreigners speak English but find it so hard to get the language right, as opposed to communicate adequately with it. Yet, and as a way of reaching the same conclusion from the opposite evidence, the Japanese sometimes infer, from the fact that so few foreigners speak Japanese at all, let alone perfectly, that they cannot. One imagines that this attitude will soon change, as foreigners speaking Japanese, at least adequately, become commonplace. This situation creates a paradox for speakers of English because it is so widely used; with the result that native speakers often implicitly divide the language into two forms: where one is the 'International English' which they understand but cannot speak.

A side-effect of the IBM statistical methods for MT was that they showed the surprising degree to which you do not have to understand ANYTHING of the language you are processing. Most workers in the language industries find this conclusion intuitively unacceptable, even if they do not subscribe to what one might call the 'meaning and

knowledge' analysis still popular within many Japanese systems, as it used to be for English during the 'artificial intelligence' period in the 1970's. Its basis in both languages was what is usually called paucity of structural information, or some such phrase, which opposes the two languages to, say, Spanish or German, whose speakers tend to believe their language rule governed. Most commentators on recent MT developments contrast as radically opposed the IBM statistical methods to those earlier AI methods explored in the US. But that contrast can disguise the closeness of Meaning-Knowledge systems to statistical systems: both rest on quantifiable notions of information or knowledge. AI systems for MT like 'preference semantics' (Wilks 1977) can be seen as quantitative systems that, at the time, lacked the empirical data, since provided by more recent approaches like (Grishman and Sterling 1989).

Systems that emphasise the core role of verb meaning (all those going back to Fillmore and case in AI and computational linguistics generally, and beyond him to the verb centred tradition of classical logic) have to deal, in the end, with the vacuity of much verb meaning ('Kakeru' in Japanese or 'Make' in English are classic examples) and the reliance for understanding their use on the types of things you can do with, say, keys and locks, or scrolls and branches (in the case of Kakeru). Similar situations for English arise when only the object (bed, versus book, versus point etc.) of the verb give any content at all to the meaning of 'make' when used with them.

Perhaps, as with DO, BE, HAVE, in English, those verbs are almost entirely redundant and the verb name is no more than a pointer to constrain abnormal uses: you could delete such verbs from a text and still guess rightly what was going on; or at least you could with Kakeru if you could distinguish open and close (a lock with a key) from the wider context available. One could put this in symbolic terms as 'nouns prefer verbs as well as vice versa', but that is no more, in the case of the vacuous verbs above, than recapitulating the basics of information theory, in that these verbs carry little or no information. Text statistics, of the IBM type, reflect this and so should our analysis.

My point here is that, with these phenomena, symbolic and statistical analyses are saying the same thing in different ways, though the symbolic tradition inherits various prejudices (like the structural primacy of verbs in English), where statistical methods are simply unprejudiced.

The Relationship to MT Evaluation

Certain issues to do with MT evaluation follow from the discussion of the last section, particularly in connection with international co-operation in MT, particularly projects that require modules of a single system to be built in different countries, as is standard in EU R&D. Let us consider module interfaces (which may or may not be considered as interlinguas, which raise other, special, issues). One of these is 'how can you evaluate an international/intermodule project properly?'.

The EU MT project EUROTRA (Johnson et al. 1985) was designed on the assumption that national/language groups built modules for their own language(s) and the system was held together by a strong structure of software design and, above all, agreed interfaces. But how could one assign blame for error (if any) inside an overall project designed like this after a bad evaluation of overall performance. In fact no serious evaluations of that project based on quantitative assessment of output were ever done, but that is beside the point for this abstract discussion.

EUROTRA was not, in its final form an interlingual system, but imagine a two module interlingual system. Some have certainly written about the possibility of evaluating the modules:

Source Language--> INTERLINGUA and INTERLINGUA ----> Target Language

separately. But could this method for assignment of error be of more than internal team interest if this were an international co-operative project? Or, more precisely, for a given bad translation, how could one know for certain which of those modules was at fault, if each chose, chauvinistically, to blame the other? Clearly, that would only be possible if they had a clear way of deciding for a given sentence what was its correct interlingual representation. If he could do that it would be clear whether or not the first module produced that representation: if it did, the error must be in module two, and if not it would be in module one.

Although not interlingua based, the EUROTRA groups had to agree on module interfaces that are, in effect, interlinguas in the sense of this discussion; it was just there was more than one of them, because there were more than two modules required for a translation. In any case the groups there shared similar language-family assumptions so the interface was not too hard to define. But could Japanese and English speakers agree on a joint interlingua without an indefinite number of arbitrary decisions, such as what are the base meanings of kakeru?

One possible way out of the problem of agreeing on an interlingua between two very different languages, and assuming one did not take the 'third way' out of selecting another existing language as an interlingua: might it be possible to define two interlinguas (one J-orientated; one E-orientated) and use both, perhaps comparing translations achieved by the two routes from source to target? That would at least have the virtue of having to have an interlingua based only on one of the two languages and which might therefore not be comprehensible to the other team.

But we will always have the residual problem, rarely mentioned, that one cannot program the module Source-->INTERLINGUA unless one is a 'native speaker' of that interlingua (i.e. a native speaker of the language on which it is based), but then the other team will not be able to program the module INTERLINGUA-->Target. A moment's reflection should show that the 'two directions' solution is not a solution at all, because both teams can only program one module for each route, so there is no translation produced. In practice, this would just become a blame shifting mechanism: 'Our part was fine, so the problem must be in your generation!'.

Suppose we retain the earlier assumption that everyone does analysis and generation of their own native language, and see what the possible models would be if we did have both a J-based interlingua (JINT) and an E-based one (EINT):

i. J source---> (J group)---> JINT-------> (E group)---> E target
ii. R R EINT R R
iii. E source---> (E group)---->EINT------>(J group)---> J target
iv. R R JINT R R

The question we raised above was whether, say, an English-speaking group could do task (iv). It is crucial to recall at this point that some Japanese-speaking groups do perform tasks like (ii): the NEC MT group has used an English-like interlingua, and the EDR lexical group in Tokyo has certainly produced large numbers of codings in an E-based interlingua for Japanese word sense, which is effectively task (ii) without any generation to follow.

The solution may then be that we should learn enough of each other's languages to use each other's interlinguas, and then compare the effectiveness of the routes above. And we would probably want to add a safety clause that the evaluation of any module into or out of an interlingua based on language X should be done by the speakers of language Y.

If there are also to be rules going between the interlinguas we shall have what some Japanese groups are calling semantic transfer. Whatever that is, it is quite distinct from syntactic transfer, which is right or wrong and capable of extraction from data, as in the work of Matsumoto and colleagues (e.g. Utsuro et al. 1994). This relativist notion of an interlingua, explicitly dependent on actual natural languages, is one quite separate from the classical notion, of the sort once advocated by Schank (1973) where there could not be more than one interlingua, almost by definition. The tradition being explored in this paper (cf. Wilks et al. 1995) is that if interlinguas in fact have characteristics of natural languages, then the relativist tradition may be the only one with a future.

Relativism and Interlinguas in MT

I would suggest that one can no longer continue to say, as many still do with straight faces, that items in an interlingua look like words but are in fact 'just labels'. This ignores the degree to which they are used as a language along with assumptions brought in from languages. They always look like languages, like particular languages, as we saw above, so maybe they are languages.

Remember Ogden's Basic English (Ogden 1942): a reduced primitive language of some thousand words, about the size of the inventory of head notions in a thesaurus like Roget, and about half the size of the LDOCE defining vocabulary (Procter 1978). The words of basic English were also highly ambiguous because of the small size of the set, as is the LDOCE defining vocabulary, a task Guo set out to rectify by a handtagging of the LDOCE defining vocabulary, to produce what he called Mini-LDOCE (Guo 1992). Interlingual items are ambiguous in exactly the same way, though this fact is rarely discussed or tackled. It did surface briefly during discussion at a Pennsylvania seminar on the EDR dictionary, when EDR colleagues explained how hard they sometimes found it to understand the EINT structures they had created in the conceptual part of EDR, and this was in part because the EINT words have senses they did not know. This may be a paradoxical advantage, as I shall discuss in a moment.

If this point of view has merit, then many empirical possibilities arise immediately: one would be to adapt to this task some of the systems for producing and checking controlled languages (e.g. Carnegie Group's CLE). These could be adapted to check not only the well-formedness of formulae in an interlingua but the distribution and usage of the primitive terms. Again, a range of techniques have been developed at research centres to sense-tag texts against some given division of the lexical senses of words; so that each word in a text is tagged with one and only one sense tag that resolves its lexical ambiguity (e.g. Bruce et al. 1993). This technique could probably be extended to interlinguas, if their formulae were viewed as texts, so as to control the non-ambiguity of the interlingual forms. As we noted above, Guo has already performed this task for the prose definitions of LDOCE, and that task is not different in principle from what we are discussing here.

The motivation for all this, remember, is so that interlingual expressions can be controlled so that they are understood by native speakers of the language from which the interlingual

was drawn and by others, where the latter group are far more important for accessibility of interlingual MT as a technique.

None of this is an argument against interlinguas, but a suggestion for treating them seriously, making them more tractable, in the way MRD-based research has made lexicons more serious and consistent than the old, purely a priori, ones.

Another possible way of dealing with the difficulty we diagnosed is Hovy and Nirenburg's (1992) argument that an interlingua could be extended by the union of primitives from the classifying ontologies for the relevant languages under definition. This would abolish at a stroke the difficulty of an interlingua as a whole being based upon a single natural language, but would not help any users understand the parts not in their language. The gain would be in equity: all users would now be in the same position of not believing they understood all the symbols in the interlingua, but the basic problem would not be resolved.

It is vital to remember here that none of the above makes any sense if you are able to cling firmly to the belief that interlinguas are not using natural language symbols at all, but only manipulating words as 'labels for concepts'. If you believe that, then all the above is, for you, unnecessary and irrelevant, and some of my close colleagues are in that position. I appeal to them, however, to look again and see that the position is sheer self deception: and we have no access at all to concepts other than through their language names which are, irreducibly, in some language. Because of the convenience that computers, say, are objects to which we can all point, we may persuade ourselves that we all have the concept of computer and the name doesn't matter. This, consolation, however does not last once one notices that of the words used to define other words (e.g. the 2000 words of the LDOCE defining vocabulary - the very words that appear in interlinguas, of course) virtually none are the least like 'computer': state, person, type, argument, form are not open to simple ostensive definition and their translations are matters of much dispute and complexity. I rest my case.

Evaluation as Hegemony

I want now to move from one undiscussible subject to another, but at shorter length. We neglect at our peril the international aspects of evaluation systems and the way in which they become, or are perceived to be 'hegemonic': in the sense of attempts to assert control over the R&D of another culture. There is strong resistance in the European Commission to any general regime for the evaluation of MT based on open competitions between entrants of the kind that has developed research so rapidly, at least in its initial stages, in the ARPA community in the US. There is a belief in the Commission that such competitions are wasteful and divisive, and that belief has clearly helped to keep some substandard research in Europe alive and well for many years.

Protracted negotiations on sharing linguistic resources (lexicons and corpora) between the US and the EU have not progressed well largely because of this issue of evaluation, largely because the US side wanted to tie exchange of resources to the idea of common evaluation. The US side stressed the value of competitive evaluations between groups that accepted the same regime (usually imposed by the funding agency).

The EU side stressed co-operative R&D and downplayed evaluation, pointing out the incestuous effects of groups that compete and co-operate too intensely . Evidence of the latter are the unexpected successes of EU groups that entered ARPA MUC and Speech competitions (Sussex, Siemens, Philips, LIMSI): one could say they opened up a gene pool that had become too incestuous.

The Commission side saw the US position as hegemonic in the sense defined here: the US saw the European position as wanting to be shielded from open competition and ungrateful in that it expected to get US resources (chiefly speech data) for no return. I retail this history not to show a right and wrong side--it is not so simple--but to note that international co-operation is a complex cultural matter, in MT as anywhere else, and we should be aware of the complex links between evaluation and resources as well as the more technical issues to do with the representations and interfaces we noted above.

Resource Sharing in the Future

Nonetheless, resources will be essential to the future of MT and resources for MT, almost by definition, come from diverse languages and so states and cultures. Ways round these difficulties must be found, and in a range of areas:

Resources:	corpora, lexicons, dictionaries
Standards:	(mark-up (e.g. SGML)), tag sets, for lexicon interchange
Software modules:	alignment, taggers etc.

In all of these areas there is progress: the EU has actively encouraged the spread of the first type, and the inhibitions tend to come far more from the commercial concerns of publishers than from governments. Resource and software distribution centres have sprung up (e.g. CLR and LDC in the US, Saarbrücken in the EU). Software modules like taggers from the US and segmenters like Kyoto university's JUMAN have become widely available through individual acts of corporate and individual good citizenship. The EDR in Japan and Cambridge University Press (with its new lexicon) in the EU have announced plans to make lexical data far more available than was normally the case.

The EU has a crucial role to play in future resource provision for MT, not only because, with its twelve major languages, its need for MT is so great but because it has funded such substantial resource projects (and tool projects to use resource) already: NERC, ELRA, MULTEXT, GENELEX, AQUILEX, PAROLE, EAGLES, the names are legion.

These are still early days, even though so much has been spent, in that it is still hard to actually get hold of genuinely reusable resources and tools: interface and format problems still bedevil real reuse. the EU is also haunted by the spectre of English: it is more than one of the twelve languages: it is the superlanguage, that provokes both utilisation and fear of take-over, and all tied in with the mixed attitudes to US culture that we noticed in connection with evaluation. This complex attitude has worked against the EU funding of specifically English resources, on the grounds that they are available from the US and that the UK has already put such great efforts into its learner's dictionaries (LDOCE, OALD, COBUILD, the new Cambridge Dictionary etc.) and its national corpora (The Bank of English, the British National Corpus etc.). Were it not for these last, English could easily be in the extraordinary position of being the only EU language, all of whose resources were from or controlled by sources outside the EU.

All this effort and activity has tended to downplay the ultimate need to build resources in major languages (e.g. Russian, Chinese, Arabic) that are neither ones own nor, at the moment, seem inclined to build their own electronic resources. Russia has such resources but they seem to have deteriorated in the short term with the economy itself. The issue of who builds such resources is also relevant, of course, and in the real world, tied up with perceived threats, commercial and military.

In spite of all this, we can be sure the resource issue will not now go away from MT, and that commercial and government interests will ensure that greater resources are built and maintained. What we, as researchers, need to work for is maximum availability and the way that such resources can serve international communication, politically, of course, but, crucially, within interlingual aspects of the R&D process itself.

References

Bar-Hillel, Y. (1960) 'The Present Status of Automatic Translation of Languages', in Alt (ed.) *Advances in Computers,* Vol. 1, Academic Press: New York

Brown, P., Cocke, J., Della Pietra. S. et al. (1988), 'A Statistical Approach to Language Translation' in *Proceedings of the 12th International Conference on Computational Linguistics*, Budapest

Bruce, B., Guthrie L. and Wilks, Y. A. (1993) 'Automatic Lexical Extraction: Theories and Applications', in Beckmann, F. and Heyer, G., *Theorie und Praxis des Lexikons: Festschrift fürHelmut Schnelle,* De Gruyter: Berlin

Guo, C. M. (ed.) (1992) *Machine Tractable Dictionaries: Design and Construction*, Ablex: Norwood, NJ

Grishman, R. and Sterling, J. (1989) 'Preference Semantics for Message Understanding' in *Proceedings DARPA Speech and Natural Language Workshop*, New York University

Hovy, E. and Nirenburg, S. (1992) 'Approximating an Interlingua in a Principled Way', in *Proceedings DARPA Speech and Language Workshop*, Harriman: New York

Johnson, R., King, M. and des Tombes, L. (1985) 'EUROTRA: a Multi-Lingual System under Development', in *Computational Linguistics*, Vol. 11

Karlsson, F. (1990) 'Constraint Grammar as a Framework for Parsing Running Text', in *Proceedings COLING90*, Helsinki

Ogden, C. K. (1942) *The General Basic English Dictionary*, W. W. Norton: New York.

Procter, P. (1978) *Longman Dictionary of Contemporary English (LDOCE)*, Longman: Harlow, Essex, UK

Schank, R. C. (1973) 'Identification of Conceptualisations Underlying Natural Language', in Schank, R. C. and Colby, K. M. (eds), *Computer Models of Thought and Language*, W. H. Freeman: San Francisco

Utsuro, T., Ikeda, H., Yamanae, M. et al. (1994) 'Bilingual Text Matching Using Bilingual Dictionary and Statistics', in *Proceedings COLING94*, Kyoto

Wilks, Y. A. (1978) 'Making Preferences more Active', in *Artificial Intelligence*, Vol. 11

Wilks, Y. A. (1994) 'Developments in Machine Translation Research in the US', in *The ASLIB Proceedings*, Vol. 46, The Association for Information Management

Wilks, Y. A., Slator, B. and Guthrie, L. (1995) *Electronic Words: Dictionaries, Computers and Meanings*, MIT Press: Cambridge, MA

Parsing English with a Link Grammar

by

Daniel D. Sleator and Davy Temperley

School of Computer Science, Carnegie Mellon University, Pittsburg
Music Department, Columbia University, New York

1 Introduction

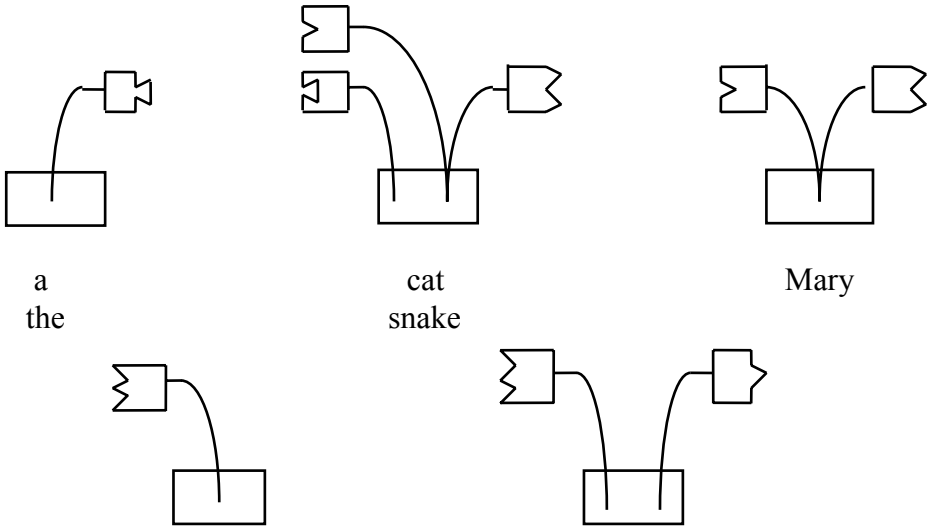
1.1 What is a link grammar?

Most sentences of most natural languages have the property that if arcs are drawn connecting each pair of words that relate to each other, then the arcs will not cross (Melcuk 1988:36). This well-known phenomenon, which we call *planarity*, is the basis of *link grammars*, our new formal language system.

A link grammar consists of a set of *words* (the terminal symbols of the grammar), each of which has a *linking requirement*. A sequence of words is a *sentence* of the language defined by the grammar if there exists a way to draw *links* among the words so as to satisfy the following conditions:

Planarity:	The links do not cross (when drawn above the words).
Connectivity:	The links suffice to connect all the words of the sequence together.
Satisfaction:	The links satisfy the linking requirements of each word in the sequence.

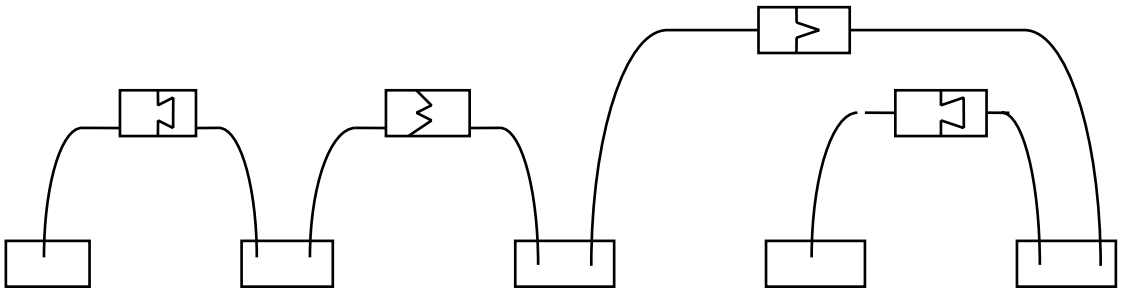
The linking requirements of each word are contained in a *dictionary*. To illustrate the linking requirements, the following diagram shows a simple dictionary for the words *a, the, cat, snake, Mary, ran,* and *chased.* The linking requirement of each word is represented by the diagram above the word.



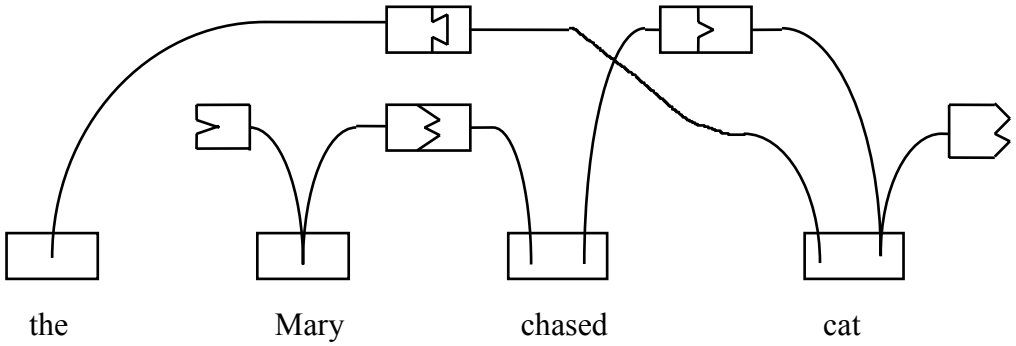
ran chased

Each of the intricately shaped labeled boxes is a *connector*. A connector is satisfied by matching it to a compatible connector (one with the appropriate shape, facing in the opposite direction). Exactly one of the connectors attached to a given black dot must be satisfied (the others, if any, must not be used). Thus *cat* requires a **D** connector to its left and either an **O** connector to its left or a **S** connector to its right. Plugging a pair of connectors together corresponds to drawing a link between that pair of words.

The following diagram shows how the linking requirements are satisfied in the sentence, *The cat chased a snake* (the unused connectors have been suppressed here).

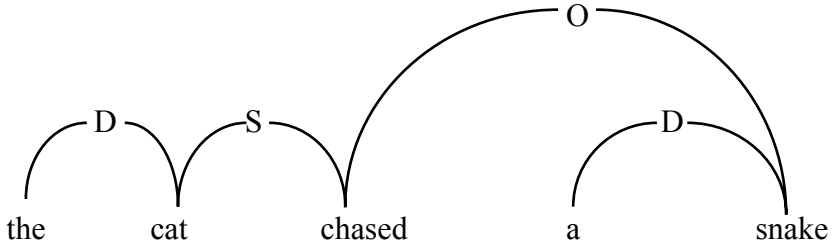


It is easy to see that *Mary chased the cat,* and *the cat ran* are also sentences of this grammar. The sequence of words, *the Mary chased cat*, is not in this language. Any attempt to satisfy the linking requirements leads to a violation of one of the three rules. Here is one attempt.



Similarly *ran Mary* and *cat ran chased* are not part of this language.

A set of links that proves that a sequence of words is in the language of a link grammar is called a *linkage*. From now on we shall use simpler diagrams to illustrate linkages. Here is the simplified form of the diagram showing that *the cat chased a snake* is part of this language.



We have a succinct, computer-readable notation for expressing the dictionary of linking requirements. The following dictionary encodes the linking requirements of the previous example.

words	formula
a the	**D+**
snake cat	**D– & (O– or S+)**
Mary	**O– or S+**
ran	**S–**
chased	**S– & O+**

The linking requirement for each word is expressed as a formula involving the operators **&**, and **or**, parentheses, and connector names. The + or – suffix on a connector name indicates the direction (relative to the word being defined) in which the matching connector (if any) must lie. The **&** of two formulae is satisfied by satisfying both the formulae. The **or** of two formulae requires that exactly one of its formulae be satisfied. The order of the arguments of an **&** operator is significant. The farther left a connector is in the expression, the nearer the word to which it connects must be. Thus, when using *cat* as an object, its determiner (to which it is connected with its **D–** connector) must be closer than the verb (to which it is connected with its **O–** connector).

We can roughly divide our work on link grammars into three parts: the link grammar formalism and its properties, the construction of a wide-coverage link grammar for English, and efficient algorithms and techniques for parsing link grammars. We now touch briefly on all three of these aspects.

Link grammars are a new and elegant context-free grammatical formalism and have a unique combination of useful properties, which are listed below. They also resemble dependency grammars and categorial grammars, although there are also many significant differences; some light is shed on the relationship in section 6. The proof of the context-freeness of link grammars is not included in this paper, but appears in our technical report (Sleator and Temperley 1991). Note that context-free systems can differ in many ways, including the ease with which the same grammar can be expressed, the efficiency with which the same grammar can be parsed, and the usefulness of the output of the parser for futher processing.

1. In a link grammar each word of the lexicon is given a definition describing how it can be used in a sentence. The grammar is distributed among the words. Such a system is said to be *lexical*. This has several important advantages. It makes it easier to construct a large grammar, because a change in the definition of a word only affects the grammaticality of sentences involving that word. The grammar can easily be constructed incrementally. Furthermore, expressing the grammar of the irregular verbs of English is easy — there is a separate definition for each word.

   Another nice feature of a lexical system is that it allows the construction of useful probabilistic language models. This has led researchers to construct lexical versions of other grammatical systems, such as tree-adjoining grammars (Schabes 1992). Lafferty and the present authors have also constructed such a probabilistic model for link grammars (Oehrle et al.1988).

2. Unlike a phrase structure grammar, after parsing a sentence with a link grammar, words that are associated semantically and syntactically are directly linked. This makes it easy to enforce agreement, and to gather statistical information about the relationships between words.

3. In English, whether or not a noun needs a determiner is independent of whether it is used as a subject, an object, or even if it is part of a prepositional phrase. The algebraic notation we developed for expressing a link grammar takes advantage of this orthogonality. Any lexical grammatical system, if it is to be used by a human being, must have such a capability. In our current on-line dictionary the word *cat* can be used in 369 different ways, and for *time* this number is 1689. A compact link grammar formula captures this large number of possibilities and can easily be written and comprehended by a human being.
4. Another interesting property of link grammars is that they have no explicit notion of constituents or categories. In most sentences parsed with our dictionaries, constituents can be seen to emerge as contiguous connected collections of words attached to the rest of the sentence by a particular type of link. For example, in the dictionary above, **S** links always attach a noun phrase (the connected collection of words at the left end of the link) to a verb (on the right end of the link). **O** links work in a similar fashion. In these cases the links of a sentence can be viewed as an alternative way of specifying the constituent structure of the sentence. On the other hand this is not the way we think about link grammars, and we see no advantage in taking that perspective.

Our second result is the construction of a link grammar dictionary for English. The goal we set for ourselves was to make a link grammar that can distinguish, as accurately as possible, syntactically correct English sentences from incorrect ones. We chose a formal or newspaper-style English as our model. The result is a link grammar of roughly 800 definitions (formulae) and 25,000 words that capture many phenomena of English grammar. It handles: noun-verb agreement, questions, imperatives, complex and irregular verbs, many types of nouns, past and present participles in noun phrases, commas, a variety of adjective types, prepositions, adverbs, relative clauses, possessives, coordinating conjunctions, unbounded dependencies, and many other things.

The third result described in this paper is a program for parsing with link grammars. The program does an exhaustive search — it finds every way of parsing the given sequence with the given link grammar. It is based on our own $O(n^3)$ algorithm (n is the number of words in the sentence to be parsed). The program also makes use of several very effective data structures and heuristics to speed up parsing. The program is comfortably fast, parsing typical newspaper sentences in a few seconds on a modern workstation.

Both our program (written in ANSI-C) and our dictionary are available via anonymous ftp through the internet. The directory is /usr/sleator/public on the host spade.pc.cs.cmu.edu (138.2.209.226). Our technical reports (Lafferty et al.1992, Sleator and Termperly 1991) are also available at this address. Having the program available for experimentation may make it easier to understand this paper.

1.2 The organization of this paper

In section 2 we define link grammars more formally and explain the notation and terminology used throughout the rest of the paper. In section 3 we describe the workings of a small link grammar for English. Our $O(n^3)$ algorithm is described in section 4, and the data structures and heuristics that make it run fast are described in section 5. In section 6 we explain the relationship between link grammars, dependency syntax, and categorial grammars. We show how to automatically construct a link grammar for a given categorial grammar. This construction allows our efficient parsing algorithms and heuristics to be applied to categorial

grammars. Section 7 mentions several other research projects that are based on link grammars.

Space limitations prevent us from presenting details of a number of other aspects of our work. The following paragraphs mention a few of these. More details on all of these matters are contained in our technical report (Sleator and Temperley 1991).

There are a number of common English phenomena that are not handled by our current system. Our technical report contains a list of these, along with the reason for this state of affairs. The reasons range from the fact that ours is a preliminary system to the fact that some phenomena simply do not fit well into the link grammar framework.

Coordinating conjunctions such as *and* pose a problem for link grammars. This is because in a sentence like *The dog chased and bit Mary* there should logically be links between both *dog* and *bit* and *chased* and *Mary.* Such links would cross. We have devised a scheme that handles the vast majority of uses of such conjunctions and incorporated it into our program. The existence of such a conjunction in a sentence modifies the grammar of the words in it. The same parsing algorithm is then used on the resulting modified grammar.

Certain other constructs are difficult to handle only using the basic link grammar framework. One example is the non-referential use of *it*: *It is likely that John will go* is correct, but *The cat is likely that John will go* is wrong. It is possible — but awkward — to distinguish between these with a link grammar. To deal with this (and a number of other phenomena) we extended the basic link grammar formalism with a *post-processor* that begins with a linkage, analyzes its structure, and determines if certain conditions are satisfied. This allows the system to correctly judge a number of subtle distinctions (including that mentioned here).

2 Notation and Terminology

2.1 Meta-rules

The link grammar dictionary consists of a collection of entries, each of which defines the linking requirements of one or more words. These requirements are specified by means of a *formula* of *connectors* combined by the binary associative operators **&** and **or**. Precedence is specified by means of parentheses. Without loss of generality we may assume that a connector is simply a character string ending in + or -.

When a link connects to a word, it is associated with one of the connectors of the formula of that word, and it is said to *satisfy* that connector. No two links may satisfy the same connector. The connectors at opposite ends of a link must have names that *match*, and the one on the left must end in + and the one on the right must end in -. In basic link grammars, two connectors match if and only if their strings are the same (up to but not including the final + or -). A more general form of matching will be introduced later.

The connectors satisfied by the links must serve to satisfy the whole formula. We define the notion of satisfying a formula recursively. To satisfy the **&** of two formulae, both formulae must be satisfied. To satisfy the **or** of two formulae, one of the formulae must be satisfied, and *no* connectors of the other formula may be satisfied. It is sometimes convenient to use the empty formula (‘( )’), which is satisfied by being connected to no links.

A sequence of words is a *sentence* of the language defined by the grammar if there exists a way to draw links among the words so as to satisfy each word’s formula and the following *meta-rules*:

Planarity: The links are drawn above the sentence and do not cross.

Connectivity: The links suffice to connect all the words of the sequence together.

Ordering: When the connectors of a formula are traversed from left to right, the words to which they connect proceed from near to far. In other words, consider a word and consider two links connecting that word to words to its left. The link connecting the nearer word (the shorter link) must satisfy a connector appearing to the left (in the formula) of that of the other word. Similarly, a link to the right must satisfy a connector to the left (in the formula) of a longer link to the right.

Exclusion: No two links may connect the same pair of words.

2.2 Disjunctive Form

The use of formulae to specify a link grammar dictionary is convenient for creating natural language grammars but it is cumbersome for mathematical analysis of link grammars and in describing algorithms for parsing link grammars. We therefore introduce a different way of expressing a link grammar called *disjunctive form*.

In disjunctive form each word of the grammar has a set of *disjuncts* associated with it. Each disjunct corresponds to one particular way of satisfying the requirements of a word. A disjunct consists of two ordered lists of connector names: the *left list* and the *right list.* The left list contains connectors that connect to the left of the current word (those connectors end in **-**) and the right list contains connectors that connect to the right of the current word. A disjunct will be denoted:

$$((L_1, L_2,..., L_m)\ (R_n, R_{n-1}, ..., R_1))$$

Where $L_1, L_2,..., L_m$ are the connectors that must connect to the left, and $R_1, R_2, ..., R_n$ are connectors that must connect to the right. The number of connectors in either list may be zero. The trailing + or **-** may be omitted from the connector names when using disjunctive form since the direction is implicit in the form of the disjunct.

To satisfy the linking requirements of a word, one of its disjuncts must be satisfied (and no links may attach to any other disjunct). To satisfy a disjunct all of its connectors must be satisfied by appropriate links. The words to which L_1, L_2 ... are linked are to the left of the current word and are monotonically increasing in distance from the current word. The words to which R_1, R_2 ... are linked are to the right of the current word and are monotonically increasing in distance from the current word.

It is easy to see how to translate a link grammar in disjunctive form to one in standard form. This can be done simply by rewriting each disjunct as

$$(L_1 \ \& \ L_2 \ \& \ ... \ \& \ L_m \ \& \ R_1 \ \& \ R_2 \ \& \ ... \ \& \ R_n)$$

and combining all the disjuncts together with the **or** operator to make an appropriate formula.

It is also easy to translate a formula into a set of disjuncts. This is done by enumerating all ways that the formula can be satisfied. For example, the formula

(**A-** or ( )) **& D- & (B**+ or ( )) **&** (**O-** or **S**+)

corresponds to the following eight disjuncts:

((A,D) (S,B))

((A,D,O) (B))
((A,D) (S))
((A,D,O) ( ))
((D) (S,B))
((D,O) (B))
((D) (S))
((D,O) ( ))

2.3 Our Dictionary Language

To streamline the difficult process of writing the dictionary we have incorporated several other features to the dictionary language. Examples of all of these features can be found in section 3.

It is useful to consider connector matching rules that are more powerful than simply requiring the strings of the connectors to be identical. The most general matching rule is simply a table — part of the link grammar — that specifies all pairs of connectors that match. The resulting link grammar is still context-free.

In the dictionary presented later in this paper and in our larger on-line dictionary, we use a matching rule that is slightly more sophisticated than simple string matching. We shall now describe this rule.

A connector name begins with one or more upper case letters followed by a sequence of lower case letters or *s. Each lower case letter (or *) is a *subscript*. To determine if two connectors match, delete the trailing + or - and append an infinite sequence of *s to both connectors. The connectors match if and only if these two strings match under the proviso that * matches a lower case letter (or *).

For example, **S** matches both **Sp** and **Ss**, but **Sp** does not match **Ss**. Similarly, **D*u** matches **Dmu** and **Dm**, but not **Dmc**. All four of these connectors match **Dm**.

The formula (**A+ & B+**) is optional. Since this occurs frequently we denote it with curly braces, as follows: {**A+ & B+**}.

It is useful to allow certain connectors to be able to connect to one or more links. This makes it easy, for example, to allow any number of adjectives to attach to a noun. We denote this by putting an '?' before the connector name, and call the result a *multi-connector*.

Our dictionaries consist of a sequence of *entries*, each of which is a list of words separated by spaces, followed by a colon, followed by the formula defining the words, followed by a semi-colon.

3 An Example

Perhaps the best way to understand how to write a link grammar for English is to study an example. The following dictionary does not cover the complete grammar of the words it contains, but it does handle a number of phenomena: verb-noun agreement, adjectives, questions, infinitives, prepositional phrases, and relative clauses.

```
the:            D+;
a:              Ds+;
John Mary:
    J- or O- or (({C- or CL-} & S+) or SI-);
```

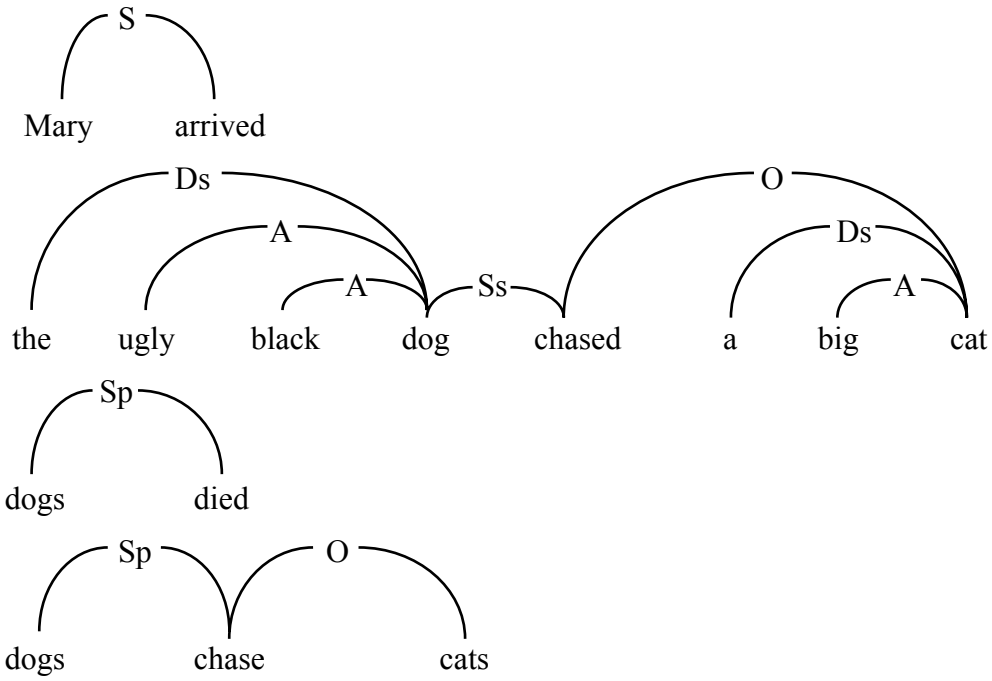
```
dog cat park bone stick:
    {?A-} & Ds-
    & {?M+ or (C+ & Bs+)}
    & (J- or  O- or ({C- or CL-} & Ss+) or SIs-);
dogs cats parks bones sticks:
    {?A-} & Dm-
    & {?M+ or (C+ & Bp+)}
    & (J- or O- or ({C- or CL-} & Sp+) or SIp-);
has:
    (SIs+ or Ss- or (Z- & B-))
    & (((B- or O+) & {?EV+}) or T+);
did:
    (SIs+ & I+)
  or   ((S- or (Z- & B-))
       & (((B- or O+) & {?EV+}) or I+));
can may will must:
    (SI+ or S- or (Z- & B-)) & I+;
is was:
    (Ss- or (Z- & Bs-) or SIs+)
    & (AI+ or O+ or B- or V+ or MP+);
touch chase meet:
    (Sp- or (Z- & Bp-) or I-)
    & (O+ or B-) & {?EV+};
touches chases meets:
    (Ss- or (Z- & Bs-)) & (O+ or B-) & {?EV+};
touched chased met:
    (V- or M-
    or ((S- or (Z- & B-) or T-) & (O+ or B-)))
    & {?EV+};
touching chasing meeting:
    (GI- or M-) & (O+ or B-) & {?EV+};
die arrive:
    (SP- or (Z- & Bp-) or I-) & {?EV+};
dies arrives:
    (Ss- or (Z- & Bs-)) & {?EV+};
died arrived:
    (S- or (Z- & Bs-) or T-) & {?EV+};
dying arriving:
    (GI- or M-) & {?EV+};
with in by:
    J+ & (Mp- or EV-);
big black ugly:
    A+ or (AI- & {?EV+});
who:
    (C- & {Z+ or CL+}) or B+ or Ss+;
```

3.1 Some Simple Connectors

We develop an explanation of how this works in stages. Let us first restrict our attention to the following connectors: **S**, **O**, **A**, **D**. (Imagine the dictionary with all of the other connectors removed.) The **S** is used to connect a noun to its verb. The **O** connector is used to connect a verb to its object. The **A** connector is used to connect an adjective to its noun. The **D** is for connecting a determiner to its noun. Notice that this connector is omitted from proper nouns, is optional on plural nouns, and is mandatory on singular nouns. Similarly, the **S** connector is subscripted to ensure verb-noun agreement.

The ordering of the terms in these expressions is often important. For example, the fact that on nouns, the **A-** occurs to the left of the **D-** means that the adjective must be closer to the noun than the determiner.

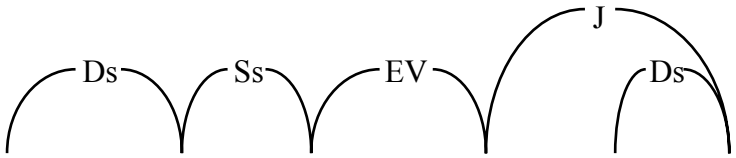
Here are some judgements that can be rendered by what we have described so far:

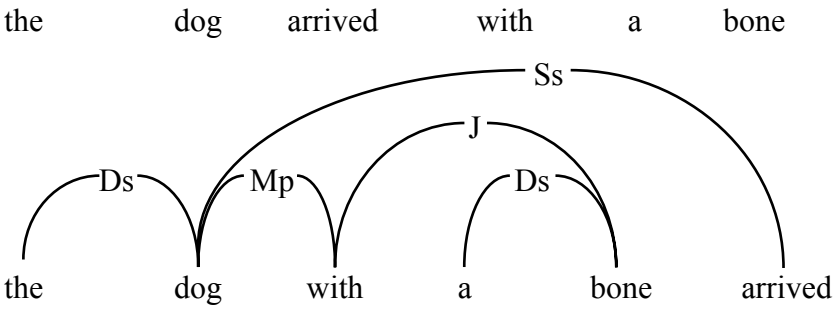


*a dog chase a cat
*black the dog died
*a/*the Mary chased the cat
*a dogs died
*dog died

3.2 Prepositions

The **J**, **M** and **EV** connectors allow prepositional phrases. The **J** connector connects a preposition to its object. Notice that in nouns, the **J-** is an alternative to the **O-**. This means that a noun cannot be an object of a verb and of a preposition. The **M** connector is used when a prepositional phrase modifies a noun and the **EV** connector is used when a prepositional phrase modifies a verb. The following two examples illustrate this:

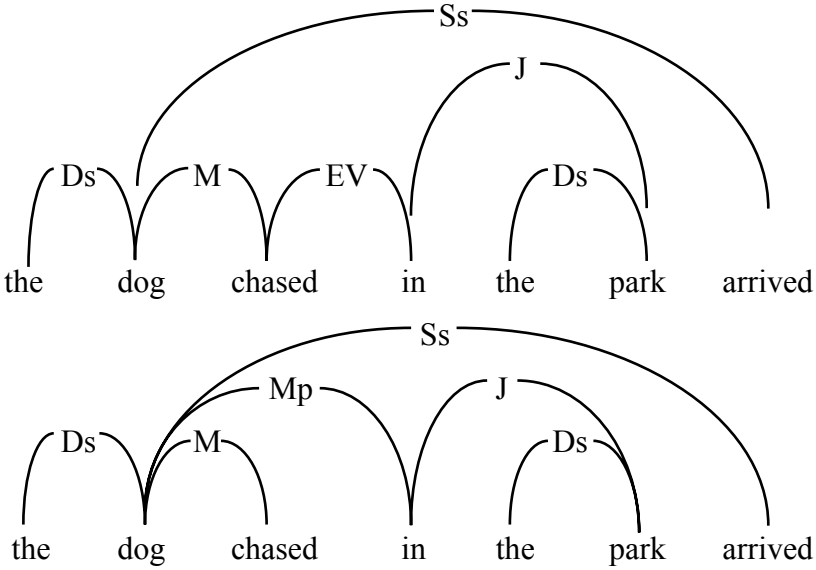




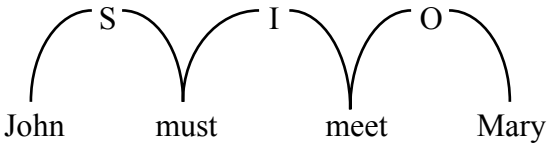
Notice that, as with **A-** connectors on nouns, a **?** is used for **M-** connectors on nouns and **EV-** connectors on verbs, allowing multiple prepositional phrases, such as *John chased a dog in the park with a stick*.

3.3 Participles

The **M-** connector on *chased* allows it to act as a participle phrase modifying a noun, as shown in these examples:



The **I** connector is used for infinitives, as in:

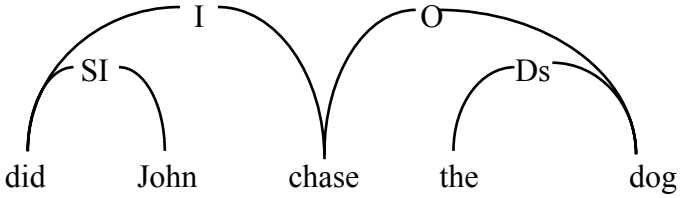


Notice that the **I** connector is an alternative to the **S** connector on plural verb forms. Thus we take advantage of the fact that plural verb forms are usually the same as the infinitive forms and include them both in a single dictionary entry.

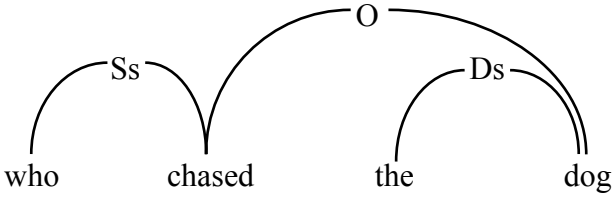
In a similar way, the **T** connector is used for past participles. Past participles have a **T-**; forms of the verb *have* have a **T+**. The **GI** connector is used for present participles. Present participles have a **GI-** connector; forms of the verb *be* have a **GI+**. The **AI** connector is used for predicative adjectives. Adjectives have a **AI-** connector; forms of *be* have a **AI+** connector.

3.4 Questions

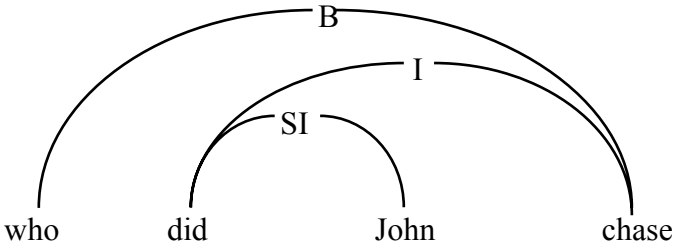
The **SI** connector is used for questions where there is subject-verb inversion. On nouns **SI-** is an alternative to **S+**, and on invertible verbs (is, has, did, must, etc.) **SI+** is an alternative to **S-** . This allows



Wh- questions work in various different ways; only questions involving *who* will be discussed here. For subject-type questions, where *who* is substituting for the subject, *who* simply has an **S+** connector. This allows



For object-type questions, where *who* is substituting for the object, the **B** connector is used. Transitive verbs have **B-** connectors as an alternative to their **O+** connectors. *Who* has a **B+** connector. This allows



The following incorrect sentences are rejected:

*Did John chase
*Who did John chase Mary
*John did Mary chase
*Chased John Mary

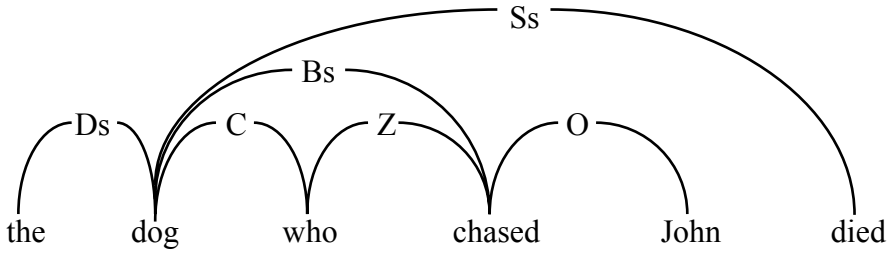
The following incorrect construction is accepted. In our on-line system, post-processing is used to eliminate this.

*Who John chased

3.5 Relative Clauses

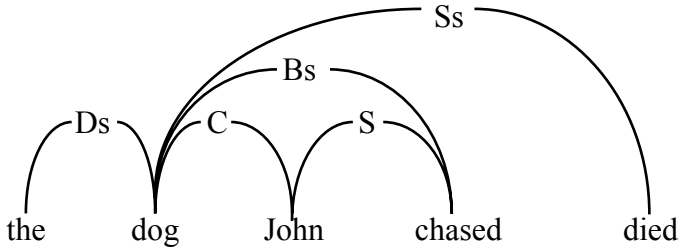
For subject-type relative clauses, where the antecedent is acting as the subject of the clause, a **B** connector serves to connect the noun to the verb of the relative clause. Nouns have a **B+** connector. Notice that this is optional; it is also **&-**ed with the **S+**, **SI-**, **O+**, and **J+** connectors, meaning that one of these connectors must be used whether or not the noun takes a relative clause. Verbs have a **B-** connector which is **or**-ed with their **S-** connectors, if a verb is in a subject-type relative clause, it may not make an **S** connection as well.

For subject-type relative clauses, the relative pronoun *who* is mandatory. For this purpose, verbs have a **Z-** connector **and-**ed with their **B-** connector. *Who* has a **Z+** connector; therefore it can fulfill this need. However, it also has a **C-** connector anded with its **Z+** connector; this must connect back to the **C+** connector on nouns. This allows the following:

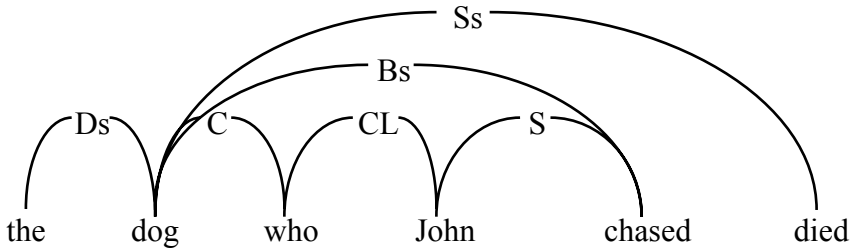


For object-type relative clauses, the same **B+** connector on nouns is used. However, this time it connects to the *other* **B-** connector on verbs, the one which is **or-**ed with the **O+** connector and which is also used for object-type *wh-* questions.

In this case, the relative pronoun *who* is optional. Notice that nouns have optional **C+** and **CL-** connectors which are **and-**ed with their **S+** connectors. These are used when the noun is the subject of an object-type relative clause. When *who* is not present, the **C+** connector on the antecedent noun connects directly to the **C-** on the subject of the relative clause:



When *who* is present the **C+** on the antecedent connects to the **C-** on *who*; this forces the **CL+** to connect to the **CL-** on the subject of the clause:



This system successfully rejects the following incorrect sentences:

*The dog chased cats died
*The dog who chase cats died
*The dog who John chased cats died
*The dog John chased cats died

*The dog who chased died

The following incorrect constructions are accepted but can be weeded out in post-processing:

*The dog did John chase died
*The dog who John died Mary chased died

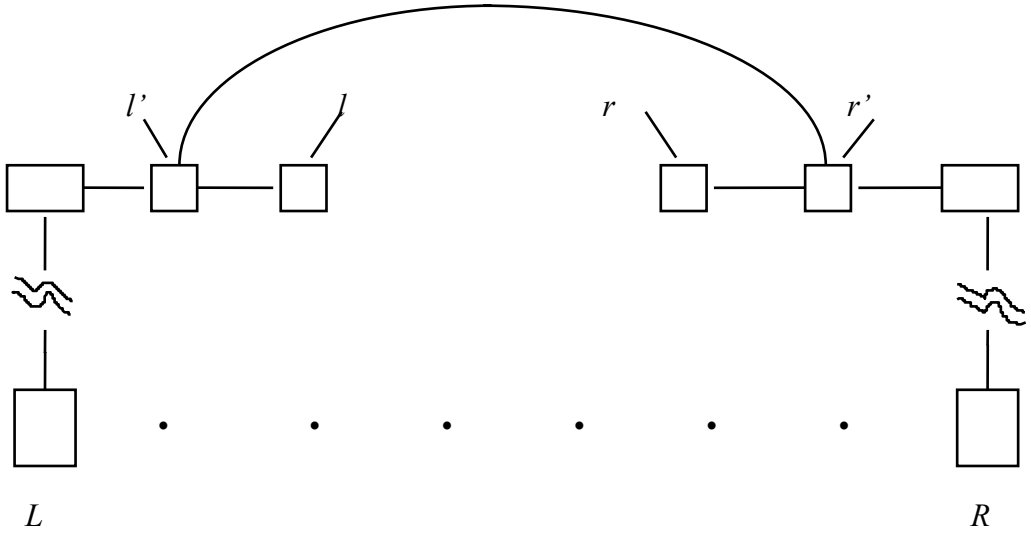
4 The Algorithm

Our algorithm for parsing link grammars is based on dynamic programming. Perhaps its closest relative in the standard literature is the dynamic programming algorithm for finding an optimal triangulation of a convex polygon (Cormen 1990: 320). It tries to build up a linkage (which we call a *solution* in this section) in a top-down fashion: It will never add a link (to a partial solution) that is above a link already in the partial solution.

The algorithm is most easily explained by specifying a data structure for representing disjuncts. A disjunct *d* has pointers to two linked lists of connectors. These pointers are denoted *left[d]* and *right[d]*. If *c* is a connector then *next[c]* will denote the next connector after *c* in its list. The next field of the last pointer of a list has the value NIL.

For example, suppose the disjunct *d* =((**D,O**) ( )) (using the notation of section 2). Then *left[d]* would point to the connector **O**, and *next*[left[d]] would point to the connector **D**, and *next*[next[*left*[*d*]]] would be NIL. Similarly, *right*[*d*] = NIL.

To give some intuition of how the algorithm works, consider the situation after a link has been proposed between a connector l^l on word *L* and a connector r^l on word *R*. (The words of the sequence to be parsed are numbered from O to N - 1.) For convenience we define *l* and *r* to be *next*[l^l] and *next*[r^l] respectively. The situation is shown in the following diagram:



Here the square boxes above the words *L* and *R* represent a data structure node corresponding to the word. The rectangular box above each of these represents one of the (possibly many) disjuncts for the word. The small squares pointed to by the disjuncts represent connectors.

How do we go about extending the partial solution into the region strictly between *L* and *R*? (This region will be denoted (*L,...,R*).) First of all, if there are no words in this region (*i.e. L* =

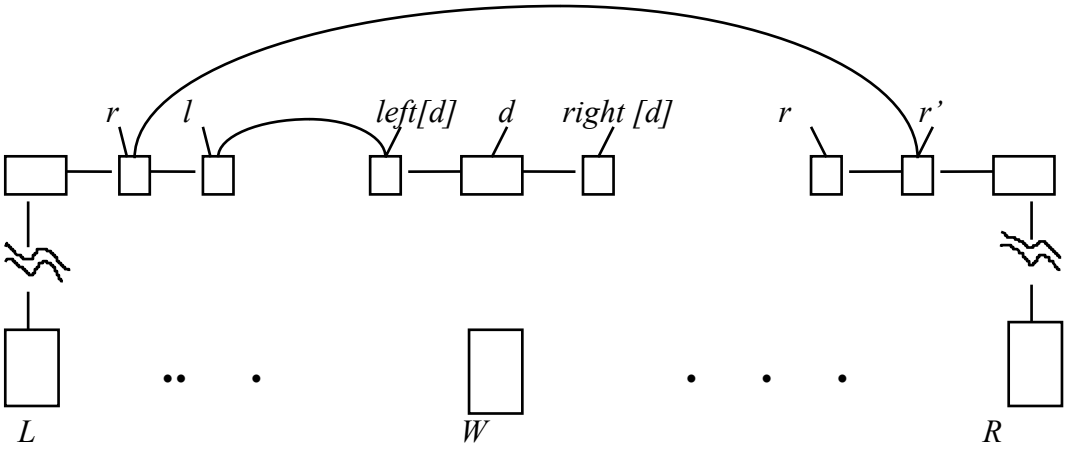
$R + 1$) then the partial solution we have built is certainly invalid if either $l \neq$ NIL or $r \neq$ NIL. If $l = r =$ NIL then this region is ok, and we may proceed to construct the rest of the solution.

Now suppose that the region between L and R contains at least one word. In order to attach the words of this region to the rest of the sentence there must be at least one link either from L to some word in this region, or from R to some word in this region (since no word in this region can link to a word outside of the [$L,...,R$] range, and something must connect these words to the rest of the sentence).

Since the connector l^1 has already been used in the solution being constructed, this solution must use the rest of the connectors of the disjunct in which l^1 resides. The same holds for r^1. The only connectors of these disjuncts that can be involved in the ($L,...,R$) region are those in the lists beginning with l and r. (The use of any other connector on these disjuncts in this region would violate the ordering requirement.) In fact all of the connectors of these lists must be used in this region in order to have a satisfactory solution.

Suppose for the moment that l is not NIL. We know that this connector must link to some disjunct on some word in the region ($L,...,R$). (It cannot link to R because of the exclusion rule.) The algorithm tries all possible such words and disjuncts. Suppose it finds a word W and a disjunct d on W such that the connector l matches *left[d]*. We can now add this link to our partial solution.

The situation is shown in the following diagram.



How do we determine if this partial solution can be extended to a full solution? We do this by solving two problems similar to the problem we started with. In particular we ask if the solution can be extended to the word range ($L,...,W$) using the connector lists beginning with *next[l]* and *next*[left[d]]. We also ask if the solution can be extended to the word range ($W,...,R$) using the connector lists beginning with *right[d]* and r. Notice that in the latter case the problem we are solving seems superficially different: the boundary words have not already been connected together by a link. This difference is actually of no consequence because the pair of links (L to R and L to W) play the role that a direct link from W to R would play: (1) they separate the region ($W,...,R$) from all the other words and (2) they serve to connect the words W and R together.

We need to consider one other possibility. That is that there might be a solution with a link between words L and W and a link between words W and R. (This results in a solution where the word/link graph is cyclic.) The algorithm handles this possibility by also attempting to form a link between *right [d]* and r. If these two match it does a third recursive call, solving a

third problem analogous to our original problem. In this problem the word range is (*W,...,r*) and the connector lists to be satisfied begin with *next*[right[d]] and *left[r]*. A very similar analysis suffices to handle the case when *l* is NIL.

The algorithm described has an exponential worst-case running time as a function of *N*, the number of words in the sequence to be parsed. This can easily be transformed into an efficient dynamic programming algorithm by using *memoization* (Cormen 1990: 312).

The running time is now bounded by the number of different possible recursive calls multiplied by the time used by each call. A recursive call is completely determined by specifying the pointers *l* and *r*. (These uniquely determine *L* and *R*.) The cost of a given call is bounded by the total number of disjuncts in the sequence of words.

If we let *d* be the number of disjuncts and *c* be the number of connectors, then the running time is $O(c^2d)$. For a fixed link grammar, $d = O(N)$ and $c = O(N)$, so the running time is $O(N^3)$.

Our technical reports describe this algorithm in more detail. They contain pseudo-code for the algorithm, an argument for its correctness (Sleator and Temperely 1991) and an elegant recurrence for the number of linkages of a sentence (Lafferty et al.1992).

After the algorithm above was implemented, we were interested in seeing how well it would work on sentences taken from newspapers and other natural sources. It quickly became clear that something else was needed to make the algorithm run faster on long sentences.

5 Speeding Things Up

As pointed out in the introduction, in a link grammar dictionary with significant coverage of English grammar the number of disjuncts on many words gets rather large. Thus the constant *d* in the analysis at the end of the last section is quite large. We devised and implemented several time-saving schemes that run in conjunction with the algorithm of the previous section.

5.1 Pruning

Our first approach is based on the following observation: in any particular sequence of words to be parsed most of the disjuncts are irrelevant for the simple reason that they contain a connector that does not match any other connector on a word in the sequence. To be more precise, suppose that a word *W* has a disjunct *d* with a connector **C** in its right list. If no word to the right of *W* has a connector (pointing to the left) that matches **C**, then the disjunct *d* cannot be in any linkage. This disjunct can therefore be deleted without changing the set of linkages. Deleting such a disjunct is called a *pruning step*. *Pruning* consists of repeating the pruning step until it can no longer be applied.

The set of disjuncts left (after pruning is complete) is independent of the order in which the steps are applied. (The pruning operation has the Church-Rosser property.) We therefore choose an ordering that can be efficiently implemented. It would be ideal if we could achieve a running time for pruning that is linear in the number of connectors. The scheme we propose satisfies no useful a-priori bound on its running time, but in practice it appears to run in linear time.

A series of sequential passes through the words is make, alternately left-to-right and right-to-left. The two types of passes are analogous, so it suffices to describe the left-to-right pass.

The pass processes the words sequentially, starting with word 1. Consider the situation after words i,...,*W* - 1 have been processed. A set of *S* connectors has been computed. This is the set of connectors that exists on the right lists of the disjuncts of words 1,...*W* -1 that have not been deleted. To process word *W*, we consider each disjunct *d* of *W* in turn. For each connector *c* on the left list of *d*, we search the set *S* to see if it contains a connector that matches *c*. If one of the connectors of *d* matches nothing in *S*, then we apply the pruning step to *d* (we remove *d*). Each right connector of each remaining disjunct of *W* is now incorporated into the set *S*. This completes the processing of word *W*.

The function computed by this left-to-right pass in indempotent, which is another way of saying that doing the operation twice in a row will be the same as doing it once. Therefore if (as we alternate left-to-right and right-to-left passes) a pass (after the first one) does nothing, then all further passes will do nothing. This is how the algorithm decides when to stop.

The data structure used for the set of *S* is simply a hash table, where the hash function only uses the initial upper-case letters of the connector name. This ensures that if two connectors get hashed to different locations, then they definitely do not match.

Although we know of no non-trivial bound on the number of passes, we have never seen a case requiring more than five.

5.2 The Fast-Match Data Structure

The inner loop in the algorithm described in section 4 searches for a word *W* and a disjunct *d* of this word whose first left connector matches *l* or whose first right connector matches *r*. If there were a fast way to find all such disjuncts significant savings might be achieved. The fast-match data structure, which is based on hashing, does precisely this. The speed-up afforded by this technique is roughly the number of different connector types, which is roughly 30 in our current dictionary.

5.3 Power Pruning

Power pruning is a refinement of pruning that takes advantage of the ordering requirement of the connectors of a disjunct, the exclusion rule, and other properties of any valid linkage. It also interacts with the fast-match data structure in a beautiful way. Unfortunately these details are beyond the scope of the paper (although they appear in our technical report: see Sleator and Temperley 1991).

Each of the refinements described in this section significantly reduced the time required to do search for a linkage. The operations of pruning, power pruning, and searching for a linkage all take roughly the same amount of time.

6 Dependency and Categorial Grammars

6.1 Dependency Formalisms

There is a large body of work based on the idea that linguistic analysis can be done by drawing links between words. These are variously called *dependency systems* (Gaifman 1965), *dependency syntax* (Melcuk 1988), *dependency grammar* (Fraser 1989, 1990) or *word grammar* (Hudson 1984, 1989).

In dependency grammar a grammatical sentence is endowed with a *dependency structure*, which is very similar to a linkage. This structure, as defined by Melcuk (1988), consists of a set of planar directed arcs among the words that form a tree. Each word (except the *root word*) has an arc out to exactly one other word, and no arc may pass over the root word. In a linkage (as opposed to a dependency structure) the links are labeled, undirected, and may form cycles; there is no notion of a root word.

Gaifman (1965) was the first to actually give a formal method of expressing a dependency grammar. He shows that his model is context-free. Melcuk's definition of a dependency structure and Gaifman's proof that dependency grammar is context free imply that there is a very close relationship between these systems and link grammars. This is the case.

It is easy to take a dependency grammar in Gaifman's notation and generate a link grammar that accepts the same language. In this correspondence the linkage that results from parsing a sentence is the same as the corresponding dependency structure. This means that our algorithm for link parsing can easily be applied to dependency grammars. The number of disjuncts in the resulting link grammar is at most quadratic in the number of rules in the dependency grammar. None of the algorithms that have been described for dependency parsing (Fraser 1989, van Zuijlen 1989, Hudson 1989) seem to bear any resemblance to ours. It is therefore plausible to conjecture that our algorithms and techniques could be very useful for directly parsing dependency grammars.

Gaifman's result shows that it is possible to represent a link grammar as a dependency grammar (they are both context-free). But this correspondence is of little use if the parsed structures that result are totally different.

One problem with constructing a dependency grammar that is in direct correspondence with a given link grammar is that a linkage in a link grammar may have cycles, whereas cycles are not allowed in dependency grammar. If we restrict ourselves to acyclic linkages we run into another problem. This is that there is an exponential blow-up in the number of rules required to express the same grammar. This is because each disjunct of each word in the link grammar requires a separate rule in the dependency grammar.

Gaifman's model is not lexical. The method classifies the words into categories. One word can belong to many categories. Roughly speaking, for each disjunct that occurs in the dictionary there is a category of all words that have that disjunct. The notation is therefore in a sense orthogonal to the link grammar notation.

We are not aware of any notation for dependency systems that is lexical, or that is as terse and well suited for a natural language grammar as link grammars. There has been work on creating dependency grammars for English (Hudson 1989, Fraser 1989), but we are not aware of an implementation of a dependency grammar for any natural language that is nearly as sophisticated as ours.

6.2 Categorial Grammars

Another grammatical system, known as a *categorial grammar* (Bar-Hillel 1964) bears some resemblance to link grammars. Below we show how to express any categorial grammar concisely as a link grammar. It appears to be more difficult to express a link grammar as a categorial grammar.

Just as in a link grammar, each word of a categorial grammar is associated with one or more symbolic expressions. An expression is either an atomic symbol or a pair of expressions

combined with one of two types of binary operators: / and \.. A sentence is in the language defined by the categorial grammar if, after choosing one expression associated with each word, there is a *derivation* which transforms the chosen sequence of expressions into **S**, a single expression consisting of a special atomic symbol. The derivation proceeds by combining two neighbouring expressions into one using one of the following rules:

$$\frac{e \quad e\backslash f}{f} \qquad\qquad \frac{f/e \quad e}{f}$$

Here *e* and *f* are arbitrary expressions, and *f**e* and *f*/e* are other expressions built using *e* and *f*. In both cases the two expressions being combined (the ones shown above the line) must be adjacent in the current sequence of expressions. Each combinational operation produces one expression (the one below the line), and reduces the number of expressions by one. After *n* - 1 operations have been applied, a sentence of length *n* has been reduced to one expression.

For example, consider the following categorial grammar [9]:

Harry:	**NP, S/(S\NP)**
likes:	**(S\NP)/NP**
peanuts:	**NP**
passionately:	**(S\NP)\(S\NP)**

Here is the derivation of *Harry likes peanuts passionately*.

```
Harry        likes        peanuts     passionately
S/(S\NP)  (S\NP)/NP        NP       (S\NP)\(S\NP)
          -------------------
                S/NP
                ------------------------------------
                                    S\NP
          ----------------------------------------
                S
```

The set of languages that can be represented by categorial grammars (as they are described here) is the set of context-free languages (Bar-Hillel 1964; there are other variants of categorial grammars which are mildly context-sensitive (Joshi 1991), but, of course, the construction presented here does not work for those languages). This fact alone sheds no light on the way in which the formalism represents a language. To get a better understanding of the connection between categorial grammars and link grammars, the following paragraphs explain a way to construct a link grammar for a given categorial grammar. The reverse (constructing a categorial grammar from a given link grammar) seems to be more difficult, and we do not know of an elegant way to do this.

To simplify the construction we use a modified definition of a link grammar called a *special link grammar*. This differs from an ordinary link grammar in two ways: the links are not allowed to form cycles; and there is a special word at the beginning of each sentence called *the wall*. The wall will not be viewed as being part of any sentence.

Let *d* be a categorial grammar expression. We will show how to build an equivalent link grammar expression *E(d)*. If a word *w* has the set $\{d_1,d_2,...,d_k\}$ of categorial expressions, then we give that word the following link grammar expression:

$E(d_1)$**or**$E(d_2)$**or**...**or**$E(d_k)$

The function $E(\cdot)$ is defined recursively as follows:

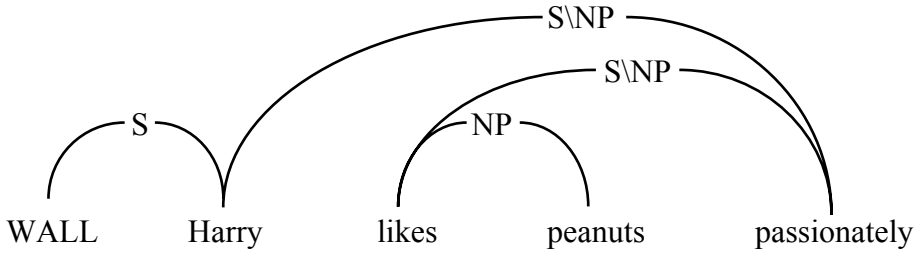
E)f/e) = *f/e-* **or** *f/e+* **or** *(e+ & E(f))*
E(e\f) = *e\f-* **or** *e\f+* **or** *(e- & E(f))*
E(A) = *A-* **or** *A+*

Here *A* stands for any atomic symbol from the categorial grammar; *A, f/e* and *e\f* are connector names in the link grammar formulae. The wall has the formula **S+**. Here is the link grammar corresponding to the categorial grammar above:

WALL: **S+;**
peanuts: **NP+** or **NP-;**
Harry:
(NP- or **NP+)**
or **(S/<S\NP>-**
or **(S<S\NP>+**
or **(S<S\NP+ & (S+** or **S-)));**
likes:
<S\NP>- or **<S\NP>+**
or **(NP+ & (S\NP-** or **S\NP+**
or **(S- & (NP-** or **NP+))))**;
passionately:
<S\NP>/<S\NP>- or **<S\NP>/<S\NP>+**
or **(S\NP- & (S\NP-** or **S\NP+**
or **(S- & (NP-** or **NP+))))**;

(Here we have replaced parentheses in the categorial grammar expressions with brackets when using them inside of a link grammar expression.)

This link grammar gives the following analysis of the sentence shown above:



Notice that in this construction both the size of the link grammar formula and the number of disjuncts it represents are linear in the size of the original categorial grammar expressions. This suggests that a very efficient way to parse a categorial grammar would be to transform it to a link grammar, then apply the algorithms and heuristics described in this paper.

7 Remarks

Link grammars have become the basis for several other research projects. John Lafferty (1992) proposes to build and automatically tune a probabilistic language model based on link grammars. The proposed model gracefully encompasses trigrams and grammatical

constraints in one framework. Andrew Hunt (personal communicaton) has developed a new model of the relationship of prosody and syntax based on link grammars. He has implemented the model, and in preliminary tests the results are much better than with other models. Tom Brehony (personal communication) has modified our parser to detect the kinds of errors that Francophones make when they write in English.

Acknowledgement

This paper was originally presented at the Third International Workshop on Parsing Technologies in August 1993 and is reprinted here by permission of the authors.

References

Bar-Hillel, Y. (1964) *Language and Information. Selected Essays on their Theory and Application*, Addison-Wesley

Carston, R. (ed.) (1989) *UCL Working Papers in Linguistics 1*, University College London

Carston, R. (ed.) (1990) *UCL Working Papers in Linguistics 2*, University College London

Cormen, T.H., Leiserson, C.E. and Rivest, R.L. (1990) *Introduction to Algorithms*, MIT Press and McGraw-Hill

Fraser, N. (1989) 'Parsing and Dependency Grammar', in Carston, R. (1989): 296-319

Fraser, N. (1990) 'Prolegomena to a Formal Theory of Dependency Grammar', in Carston, R. (1990): 298-319

Gaifman, H. (1965) 'Dependency Systems and Phrase-Structure Systems', in *Information and Control* Vol. 8: 304-337

Hudson, R. (1984) *Word Grammar*, Basil Blackwell

Hudson, R. (1989) 'Towards a Computer Testable Word Grammar of English', in Carston, R. (1989): 321-339

Hudson, R. (1990) *English Word Grammar*, Basil Blackwell

Joshi, A.K. (1991) 'Natural Language Processing', in *Science*, Vol. 253, No. 5025, September 13: 1242-49

Lafferty, J., Sleator, D., D. and Temperley, D. (1992) 'Grammatical Trigrams: A Probabilistic Model of Link Grammar', in *Proceedings of the 1992 AAAI Fall Symposium on Probabilistic Approaches to Natural Language*; also *Technical Report CMU-CS-92-181*, School of Computer Science, Carnegie Mellon University, September 1992

Melcuk, I.A. (1988) *Dependency Syntax: Theory and Practice*, State University of New York Press

Oehrle, R.T., Bach, E. and Wheeler, D. (eds) (1988) *Categorial Grammars and Natural Language Structures*, D. Reidel Publishing Company

Schabes, Y. (1992) 'Stochastic Lexicalized Tree-Adjoining Grammars', in *Proceedings of COLING-91*, Nantes, France, July 1992

Sleator, D. D. and Temperley, D. (1991) 'Parsing English with a Link Grammar,' *Technical Report CMU-CS-91-196*, Carnegie Mellon University, School of Computer Science, October 1991

Zuijlen, J., M. van (1989) 'Probabilistic Methods in Dependency Grammar Parsing', in *P2oceedings of the International Workshop on Parsing Technologies*, Carnegie Mellon University: 142-250

Book Review

Colin Haynes (1998) *Breaking Down the Language Barriers*, London: ASLIB, the Association for Information Management. Paperback xxii + 173p. Price unknown. ISBN 0-85142-381-7.

Colin Haynes has written this book with the avowed intention of changing the still largely negative attitudes of professional translators and their organisations towards MT/MAT. He pulls no punches in naming the organisations who, when approached on the subject of MT, had little positive to offer: these range from UNESCO (who provided 'a lot of expensively printed pontificating and spreading knowledge ... but nothing of practical value as far as MT was concerned') to the London-based Translation Association whose members, primarily literary translators, continue to eschew on-line dictionaries or thesauri for traditional, paper-based reference works. Haynes regrets such 'professional negativity' and notes how it is forcing MT suppliers to bypass translators and target business managers and end users directly.

To an extent this book, which despite its studiedly non-academic tone is peppered with cultural and sociolinguistic observations, is as much about language in general as MT in particular. Genuinely concerned about the effects of electronic media on language and the implications for literature, the author draws attention to the globalisation of English, the prospect of its fragmentation into sub-varieties, and the dangers of a dumbing down of standards of literacy and written communication. At the same time he is dismissive of those who may wish to freeze dry a language in a particular era and of the 'plummy, aristocratic tradition of ... standard BBC speak' that survives today in the mouths of establishment figures such as Lady Thatcher and Sir Edward Heath (who are mentioned by name). Rather we must recognise the dynamics of language change and not expect to control them. Inevitably, the new electronic media and language technologies will compel us to re-evaluate traditional linguistic standards. But as the author points out, MT exists to remove the drudgery from commercial and technical translation, not to inhibit the creativity of literary translators with whom it cannot compete. Indeed, in view of widespread illiteracy in multilingual nations, Haynes argues that computers which can recognise the spoken word and translate between natural languages have a vital role to play in overcoming linguistic barriers and in contributing to socio-economic progress. A good example of this in South Africa is the Translator family of MT software and the Lexica Project (University of Pretoria). The aim here has been to develop MT for translating between European languages, Afrikaans, and varieties of Bantu. Incidentally the project also exemplifies how an MT system conceived originally for military purposes (i.e. to gather intelligence about the enemies of white-dominated South Africa) can be successfully adapted for peaceful purposes and genuine social benefit in a developing multilingual society. The average MT user may not have realised it, but in Haynes' view the new language technologies may also contribute positively to linguistic change by making language more dynamic and egalitarian and less the instrument of authority (who take the form of 'academics, bureaucrats who try to regulate it, and the snobs who use it to judge social status').

Turning to more technical aspects, the book describes the status of the principal technologies of speech recognition, speech synthesis and machine translation, whose

integration gives us the prospect of 'machine interpreting'. Mention is made of projects such as C-STAR (the Consortium for Speech Translation Advance Research), Verbmobil in Germany, ATR (Advanced Telecommunications Research) in Japan, and British Telecom's SALT, the world's first telephone interpreter. Speech recognition packages described include the Abbott system (Universities of Cambridge and Sheffield), ALPSpeak, the IBM and Philips Dictation Systems, DragonDictate, and Stanford Research Institute's Nuance Recognizer and Decipher, which has been incorporated into the Spoken Language Translator Project, a system for translating spoken English queries into Swedish, French and Spanish within the domain of passenger air travel enquiries. A significant commercial development is the 'Alliance' between Globalink, Translex (Canada), Eurosources (France) and IRIS (Japan) to create a single, global translation agency: incorporating over 5,000 translators, the Alliance will have at its core advanced tools such as MT, translation memory and terminology database management systems.

Haynes lists twenty reasons for adopting MT, ranging from time savings to consistency of output and even protection against industrial espionage. He refers to the need for documentation by business users, especially in the field of IT, and provides useful contact addresses for such organisations as the World Translations Index ('despite the remarkable progress made with MT, the fastest and easiest way to generate a competent translation is still to get a copy from somebody who has already done all the hard work'). He includes practical tips on integrating MT into the workplace and getting end users to define the quality they want, as well as on licensing and on choosing between an expensive MT system requiring customisation and a cheaper, cost-efficient package such as the Power Translator. He even suggests 'adding value' to MT output by formatting text and inserting graphic symbols in order to improve appearance and comprehension.

In a section entitled 'How it works' the book discusses informally the basic components of MT systems such as morphological analysis, dictionaries, and parsing. However, the assertion that a PC with 8MB RAM has 'More effective short-term capacity for your grammar checker to parse and your MT software to translate than the typical human brain' is a curious one. One of the shortest chapters has the longest title: 'Fully logic, artificial intelligence, neural networks and parallel processing'. The chapter provides thumbnail sketches of developments in these areas and emphasises their potential application to MT.

Considerable space is devoted to describing the usefulness of the wordprocessor for the translator. Modern wordprocessors now support work with multilingual texts, enable texts to be pre-formatted before translation and incorporate spelling and grammar checkers alongside on-line thesauri. Translators will also have to master the skills of object linking and embedding if there are to move graphics as well as text across language barriers: 'visual vocabularies', the graphic equivalents of lexicons, are increasingly used as comprehension aids in training and other promotional materials.

Other chapters include a questionnaire to help the reader choose and evaluate the most suitable MT system and tips to help the translator work faster and more accurately (these range from exhortations to use simple style and avoid culturally specific expressions to persuading the organisation to standardise on SGML for all document handling). The author also mentions (some) universities that include translation technology in their curricula and issues a long overdue plea for suppliers to make their programs more freely available to the education community, if only out of entrepreneurial self-interest. In a tour of 'MT software around the world' the author discusses various systems and their operational use: these include METAL, LOGOS, STYLUS, IBM's TRANSLATION MANAGER, LINITEXT, and

GLOBALINK. Readers can also download from the Internet a copy of the Translator program referred to earlier (free to purchasers of the book and restricted to English/French).

In summary, this book provides an entertaining and stimulating overview of MT and related technologies in a style that is informal and easy to read. It is designed less for the academic reader than for the business user and open-minded translator. The author is clearly a believer in MT, although he is well aware of the practical issues and constraints surrounding its use.

Derek Lewis, October 1998

Conferences and Workshops

The following is a list of recent (i.e. since the last edition of the MTR) and forthcoming conferences and workshops. Telephone numbers and e-mail addresses are given where known (please check area telephone codes).

6-7 April 1998
Second Workshop on Lexical Semantics Systems
Scuola Normale Superiore, Pisa, Italy
http://celi.sns.it/~wlss98

18–21 August 1998
NLP+IA98: International Conference on Natural Language Processing and Industrial Applications
Moncton, New-Brunswick, Canada
Tel: +33 4 76 51 4369, fax: +33 4 76 51 4405, e-mail: NLP+IA-98@imag.fr

8 October 1998
Workshop on Embedded MT Systems, Design, Construction, and Evaluation of Systems with an MT Component
Langhorne, Pennsylvania, USA
Tel: 301 394-5615, fax: 301 394-3903, e-mail: voss@arl.mil
http://rpstl.arl.mil/isb-south/

8–31 October 1998
AMTA98: Machine Translation and The Information Soup
Langhorne, Pennsylvania, USA
Tel/fax: +1 703 716 0912, e-mail: amta@clark.net
http://www.isi.edu/natural-language/AMTA98.html

11–14 November, 1998
RIFRA98: International Workshop on Extraction, Filtering and Automatic Summaraisation
Sfax, Tunisia
Tel: 216 4 278 777, fax: 216 4 279 139, e-mail: abdelmajid.benhamadou@fsegs.rnu.tn

22–27 November, 1998
Workshop on Cross Language Issues in Artificial Intelligence
Singapore
e-mail: mkleong@krdl.org.sg
http://jsaic.krdl.org.sg/pricai98/home.html

30 November–4 December 1998
ICSLP98: 5th International Conference on Spoken Language Processing
Sydney Convention Centre, Sydney, Australia
E-mail: icslp98@tourhosts.com.au
http://cslab.anu.edu.au/icslp98

7–8 December 1998
TWLT14: 14th Twente Workshop On Language Technology : Language Technology In Multimedia Information Retrieval
Tel: +31 53 893680, fax: +31 53 315283, e-mail: fdejong@cs.utwente.nl
http://wwwseti.cs.utwente.nl/Parlevink/Conferences/twlt14.html

7–18 December 1998
Conférence virtuelle: les nouvelles technologies dans l'enseignement du français langue étrangère
Ecole Internationale de la Francophonie, Bordeaux, France
E-mail: martine.jaudeau@francophonie.org ou
http://ciffad.francophonie.org/

14–16 December 1998
International Symposium on Computer Learner Corpora, Second Language Acquisition and Foreign Language Teaching
The Chinese University of Hong Kong,
E-mail: josephhung@cuhk.edu.hk
http://juppiter.fltr.ucl.ac.be/FLTR/GERM/ETAN/CECL/menu.html

13–15 January 1999
IWCS3: 3rd International Workshop on Computational Semantics
Tilburg, Netherlands
Tel: +31 13 466 30 60, fax: +31 13 466 31 10
http://cwis.kub.nl/~fdl/research/ti/Docs/IWCS/iwcs.htm

5–6 February 1999
Contrastive Linguistics and Translation Studies. Empirical Approaches
Louvain-la-Neuve, Belgium
http://www.fltr.ucl.ac.be/FLTR/GERM/ContraEngl.htm

19–23 April 1999
PA Expo99
Commonwealth Institute, London, UK
Tel: +44 1253 358081, fax: +44 1253 353811, e-mail: info@pap.com
http://www.practical-applications.co.uk/TPAC

30 April–2 May 1999
9th InPLA Computational Processing of Portuguese
Sao Paulo (PUCSP), Brazil.
E-mail: inpla@exatas.pucsp.br
http://sites.uol.com.br/tony4/homepage.html

23–25 August 1999
TMI99: 8th International Conference on Theoretical and Methodological Issues in Machine Translation, Chester, UK
Tel: 0774 93 5313 (+81), fax: 0774 93 5345 (+81)
http://www.ccl.umist.ac.uk/events/tmi99/

August 1999
ESSLI99: 11th European Summer School in Logic, Language and Information
Utrecht, Netherlands
Tel: +49 341 9735773, fax: +49 341 9735798
http://www.coli.uni-sb.de/esslli/

13–17 September 1999
Machine Translation Summit VII
Singapore
Tel: 65 874 2003, fax: 65 776 8109
E-mail: MT-SUMMIT-99-Sessions@mlist.ccm.cl.nec.co.jp
http://www.krdl.org.sg

20–22 September 1999
VExTAL Venezia per il Trattamento Automatico delle Lingue
Venice, Italy
E-mail: vextal@byron.cgm.unive.it
http://byron.cgm.unive.it/eventi/VEXTAL

MEMBERSHIP: CHANGE OF ADDRESS

If you change your address, please advise us on this form, or a copy, and send it to the following (this form can also be used to join the Group):

Mr. J.D.Wigg
BCS-NLTSG
72 Brattle Wood
Sevenoaks, Kent TN13 1QU
U.K.

Date: ...../...../.....

Name: .....
Address: .....
.....
Postal Code: ..... Country: .....
E-mail: ..... Tel.No: .....
Fax.No: .....

Note for non-members of the BCS: your name and address will be recorded on the central computer records of the British Computer Society.

Questionnaire

We would like to know more about you and your interests and would be pleased if you would complete as much of the following questionnaire as you wish (please delete any unwanted words).

1. a. I am mainly interested in the computing/linguistic/user/all aspects of MT.
   b. What is/was your professional subject? .....
   c. What is your native language? .....
   d. What other languages are you interested in? .....
   e. Which computer languages (if any) have you used? .....

2. What information in this Review (No. 8, October '98) or any previous Review, have you found:
   a. interesting? Date .....
      .....
      .....
   b. useful (i.e. some action was taken on it)? Date .....
      .....
      .....

3. Is there anything else you would like to hear about or think we should publish in the *MT Review*?
   .....
   .....
   .....
   .....

4. Would you be interested in contributing to the Group by,

   a. Reviewing MT books and/or MT/multilingual software
   b. Researching/listing/reviewing public domain MT and MNLP software .....
   c. Designing/writing/reviewing MT/MNLP application software .....
   d. Designing/writing/reviewing general purpose (non-application specific) MNLP .....
      procedures/functions for use in MT and MNLP programming .....
   e. Any other suggestions? .....
      .....
      .....
      .....

Thank you for your time and assistance.