

Link Grammar

Subjects: Computer Science, Information Systems

Contributor: HandWiki Zheng

Link grammar (LG) is a theory of syntax by Davy Temperley and Daniel Sleator which builds relations between pairs of words, rather than constructing constituents in a phrase structure hierarchy. Link grammar is similar to dependency grammar, but dependency grammar includes a head-dependent relationship, whereas Link Grammar makes the head-dependent relationship optional (links need not indicate direction). Colored Multiplanar Link Grammar (CMLG) is an extension of LG allowing crossing relations between pairs of words. The relationship between words is indicated with link types, thus making the Link grammar closely related to certain categorial grammars. For example, in a subject–verb–object language like English, the verb would look left to form a subject link, and right to form an object link. Nouns would look right to complete the subject link, or left to complete the object link. In a subject–object–verb language like Persian, the verb would look left to form an object link, and a more distant left to form a subject link. Nouns would look to the right for both subject and object links.

Keywords: dependency grammar ; multiplanar ; subject–object–verb

1. Overview

Link grammar connects the words in a sentence with links, similar in form to a catena. Unlike the catena or a traditional dependency grammar, the marking of the head-dependent relationship is optional for most languages, becoming mandatory only in free-word-order languages (such as Turkish,^[1] Finnish, Hungarian, Lithuanian^[2]). That is, in English, the subject-verb relationship is "obvious", in that the subject is almost always to the left of the verb, and thus no specific indication of dependency needs to be made. In the case of subject-verb inversion, a distinct link type is employed. For free word-order languages, this can no longer hold, and a link between the subject and verb must contain an explicit directional arrow to indicate which of the two words is which.

Link grammar also differs from traditional dependency grammars by allowing cyclic relations between words. Thus, for example, there can be links indicating both the head verb of a sentence, the head subject of the sentence, as well as a link between the subject and the verb. These three links thus form a cycle (a triangle, in this case). Cycles are useful in constraining what might otherwise be ambiguous parses; cycles help "tighten up" the set of allowable parses of a sentence.

For example, in the parse

```
+---->WV--->+ +-Wd---+Ss---Pa---+ | | | | LEFT-WALL he runs fast
```

the LEFT-WALL indicates the start of the sentence, or the root node. The directional **WV** link (with arrows) points at the head verb of the sentence; it is the Wall-Verb link.^[3] The **Wd** link (drawn here without arrows) indicates the head noun (the subject) of the sentence. The link type **Wd** indicates both that it connects to the wall (W) and that the sentence is a declarative sentence (the lower-case "d" subtype).^[4] The **Ss** link indicates the subject-verb relationship; the lower-case "s" indicating that the subject is singular.^[5] Note that the WV, Wd and Ss links form a cycle. The Pa link connects the verb to a complement; the lower-case "a" indicating that it is a predicative adjective in this case.^[6]

2. Parsing Algorithm

Parsing is performed in analogy to assembling a jigsaw puzzle (representing the parsed sentence) from puzzle pieces (representing individual words).^{[7][8]} A language is represented by means of a dictionary or lexis, which consists of words and the set of allowed "jigsaw puzzle shapes" that each word can have. The shape is indicated by a "connector", which is a link-type, and a direction indicator + or - indicating right or left. Thus for example, a transitive verb may have the connectors **S-** & **O+** indicating that the verb may form a Subject ("**S**") connection to its left ("**-**") and an object connection ("**O**") to its right ("**+**"). Similarly, a common noun may have the connectors **D-** & **S+** indicating that it may connect to a

determiner on the left ("D-") and act as a subject, when connecting to a verb on the right ("S+"). The act of parsing is then to identify that the **S+** connector can attach to the **S-** connector, forming an "S" link between the two words. Parsing completes when all connectors have been connected.

A given word may have dozens or even hundreds of allowed puzzle-shapes (termed "disjuncts"): for example, many verbs may be optionally transitive, thus making the **O+** connector optional; such verbs might also take adverbial modifiers (**E** connectors) which are inherently optional. More complex verbs may have additional connectors for indirect objects, or for particles or prepositions. Thus, a part of parsing also involves picking one single unique disjunct for a word; the final parse must satisfy (connect) *all* connectors for that disjunct.^[9]

2.1. Dependency

Connectors may also include head-dependent indicators **h** and **d**. In this case, a connector containing a head indicator is only allowed to connect to a connector containing the dependent indicator (or to a connector without any h-d indicators on it). When these indicators are used, the link is decorated with arrows to indicate the link direction.^[8]

A recent extension simplifies the specification of connectors for languages that have little or no restrictions on word-order, such as Lithuanian. There are also extensions to make it easier to support languages with concatenative morphologies.

2.2. Planarity

The parsing algorithm also requires that the final graph is a planar graph, i.e. that no links cross.^[9] This constraint is based on empirical psycho-linguistic evidence that, indeed, for most languages, in nearly all situations, dependency links really do not cross.^{[10][11]} There are rare exceptions, e.g. in Finnish, and even in English; they can be parsed by link-grammar only by introducing more complex and selective connector types to capture these situations.

2.3. Costs and Selection

Connectors can have an optional floating-point cost markup, so that some are "cheaper" to use than others, thus giving preference to certain parses over others.^[8] That is, the total cost of parse is the sum of the individual costs of the connectors that were used; the cheapest parse indicates the most likely parse. This is used for parse-ranking multiple ambiguous parses. The fact that the costs are local to the connectors, and are not a global property of the algorithm makes them essentially Markovian in nature.^{[12][13][14][15][16][17]}

The assignment of a log-likelihood to linkages allows link grammar to implement the semantic selection of predicate-argument relationships. That is, certain constructions, although syntactically valid, are extremely unlikely. In this way, link grammar embodies some of the ideas present in operator grammar.

Because the costs are additive, they behave like the logarithm of the probability (since log-likelihoods are additive), or equivalently, somewhat like the entropy (since entropies are additive). This makes Link Grammar compatible with machine learning techniques such as hidden Markov models and the Viterbi algorithm, because the link costs correspond to the link weights in Markov networks or Bayesian networks.

2.4. Type Theory

The Link Grammar link types can be understood to be the types in the sense of type theory.^{[9][18]} In effect, the Link Grammar can be used to model the internal language of certain (non-symmetric) compact closed categories, such as pregroup grammars. In this sense, Link Grammar appears to be isomorphic or homomorphic to some categorial grammars. Thus, for example, in a categorial grammar the noun phrase "*the bad boy*" may be written as

the bad boy
NP/N, N/N, N

whereas the corresponding disjuncts in Link Grammar would be

the: D+; bad: A+; boy: D- & A-;

The contraction rules (inference rules) of the Lambek calculus can be mapped to the connecting of connectors in Link Grammar. The **+** and **-** directional indicators correspond the forward and backward-slashes of the categorial grammar. Finally, the single-letter names **A** and **D** can be understood as labels or "easy-to-read" mnemonic names for the rather more verbose types *NP/N*, etc.

The primary distinction here is then that the categorical grammars have two type constructors, the forward and backward slashes, that can be used to create new types (such as *NP/N*) from base types (such as *NP* and *N*). Link-grammar omits the use of type constructors, opting instead to define a much larger set of base types having compact, easy-to-remember mnemonics.

3. Examples

3.1. Example 1

A basic rule file for an SVO language might look like:

```
<determiner> D+; <noun-subject> {D-} & S+; <noun-object> {D-} & O-; <verb> S- & {O+};
```

Thus the English sentence, "The boy painted a picture" would appear as:

```
+-----O-----+ +-D---S--+ +-D--+ | | | | The boy painted a picture
```

Similar parses apply for Chinese.^[19]

3.2. Example 2

Conversely, a rule file for a null subject SOV language might consist of the following links:

```
<noun-subject> S+; <noun-object> O+; <verb> {O-} & {S-};
```

And a simple Persian sentence, *man nAn xordam* (من نان خوردم) 'I ate bread' would look like:^{[20][21][22]}

```
+-----S-----+ | +-O--+ | | | man nAn xordam
```

VSO order can be likewise accommodated, such as for Arabic.^[23]

3.3. Example 3 (Morphology)

In many languages with a concatenative morphology, the stem plays no grammatical role; the grammar is determined by the suffixes. Thus, in Russian, the sentence 'вверхуплыли редкие облачка' might have the parse:^{[24][25]}

```
+-----Wd-----+-----Sip-----+ | +-----EI-----+ +---  
-----Api-----+ | | +-LLCZD-+ +-LLAQZ+ +-LLCAO-+ | | | | | | | LEFT-WALL вверху.e  
плы.= =ли.vnndpp ре.= =дкие.api облачка.= =a.ndnpi
```

The subscripts, such as 'vnndpp', are used to indicate the grammatical category. The primary links: Wd, EI, Sip and Api connect together the suffixes, as, in principle, other stems could appear here, without altering the structure of the sentence. The Api link indicates the adjective; Sip denotes subject-verb inversion; EI is a modifier. The Wd link is used to indicate the head noun; the head verb is not indicated in this sentence. The LLXXX links serve only to attach stems to suffixes.

3.4. Example 4 (Phonology)

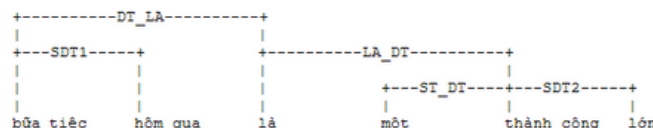
The link-grammar can also indicate phonological agreement between neighboring words. For example:

```
+-----Ost-----+ +----->WV----->+ +-----Ds**x-----+ +-----Wd-----+Ss*b-+ +---  
PHv-+-----A-----+ | | | | | | LEFT-WALL that.j-p is.v an abstract.a concept.n
```

Here, the connector 'PH' is used to constrain the determiners that can appear before the word 'abstract'. It effectively blocks (makes it costly) to use the determiner 'a' in this sentence, while the link to 'an' becomes cheap. The other links are roughly as in previous examples: S denoting subject, O denoting object, D denoting determiner. The 'WV' link indicates the head verb, and the 'W' link indicates the head noun. The lower-case letters following the upper-case link types serve to refine the type; so for example, Ds can only connect to a singular noun; Ss only to a singular subject, Os to a singular object. The lower-case v in PHv denotes 'vowel'; the lower-case d in Wd denotes a declarative sentence.

3.5. Example 5 - Vietnamese

The Vietnamese language sentence "Bữa tiệc hôm qua là một thành công lớn" - "The party yesterday was a great success" may be parsed as follows:^[26]



Vietnames link grammar example. <https://handwiki.org/wiki/index.php?curid=1288136>

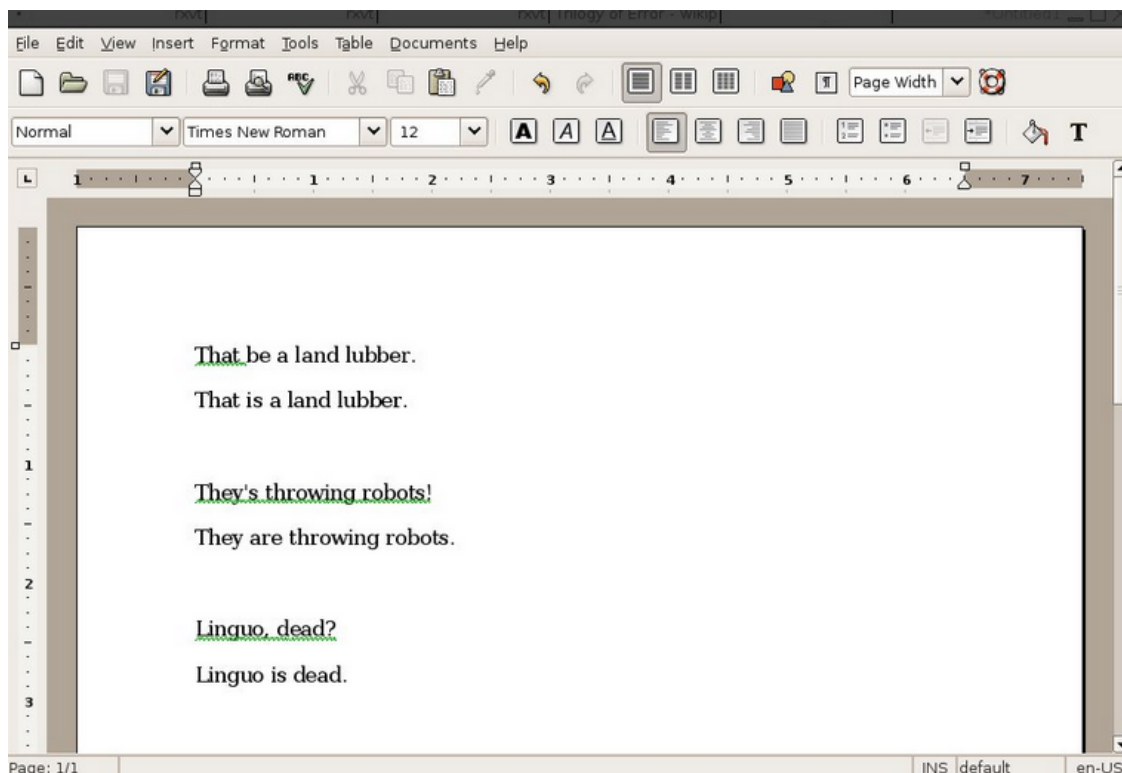
4. Implementations

The link grammar syntax parser is a library for natural language processing written in C. It is available under the LGPL license. The parser^[27] is an ongoing project. Recent versions include improved sentence coverage, Russian, Persian and Arabic language support, prototypes for German, Hebrew, Lithuanian, Vietnamese and Turkish, and programming API's for Python, Java, Common LISP, Autolt and OCaml, with 3rd-party bindings for Perl,^[28] Ruby^[29] and JavaScript node.js.

A current major undertaking is a project to learn the grammar and morphology of new languages, using unsupervised learning algorithms.^{[31][32]}

The *link-parser* program along with rules and word lists for English may be found in standard Linux distributions, e.g., as a Debian package, although many of these are years out of date.^[33]

5. Applications



AbiWord checks grammar using Link Grammar. <https://handwiki.org/wiki/index.php?curid=1732832>

AbiWord,^[27] a free word processor, uses Link Grammar for on-the-fly grammar checking. Words that cannot be linked anywhere are underlined in green.

The semantic relationship extractor ReLex,^[34] layered on top of the Link Grammar library, generates a dependency grammar output by making explicit the semantic relationships between words in a sentence. Its output can be classified as being at a level between that of SSyntR and DSyntR of Meaning-Text Theory. It also provides framing/grounding,

anaphora resolution, head-word identification, lexical chunking, part-of-speech identification, and tagging, including entity, date, money, gender, etc. tagging. It includes a compatibility mode to generate dependency output compatible with the Stanford parser,^[35] and Penn Treebank^[36]-compatible POS tagging.

Link Grammar has also been employed for information extraction of biomedical texts^{[37][38]} and events described in news articles,^[39] as well as experimental machine translation systems from English to German, Turkish, Indonesian,^[40] and Farsi.^{[41][42]}

The Link Grammar link dictionary is used to generate and verify the syntactic correctness of three different natural language generation systems: NLGen,^[43] NLGen2^[44] and microplanner/surreal.^[45] It is also used as a part of the NLP pipeline in the OpenCog AI project.

References

1. Özlem İstek, "A Link Grammar for Turkish", Thesis, Bilkent University, Ankara, Turkey (2006)
http://www.cs.bilkent.edu.tr/~ilyas/PDF/THESES/ozlem_istek_thesis.pdf
2. Lietuvių Kalbos Gramatika <http://www.abisource.com/projects/link-grammar/lithuanian/>
3. WV Link type <http://www.abisource.com/projects/link-grammar/dict/section-WV.html>
4. W link type <http://www.abisource.com/projects/link-grammar/dict/section-W.html>
5. S link type <http://www.abisource.com/projects/link-grammar/dict/section-S.html>
6. P link type <http://www.abisource.com/projects/link-grammar/dict/section-P.html>
7. Daniel D. K. Sleator, Davy Temperley, "Parsing English with a Link Grammar" Carnegie Mellon University Computer Science technical report CMU-CS-91-196 (1991) [1] (ArXiv) <https://arxiv.org/abs/cmp-lg/9508004>
8. An Introduction to the Link Grammar Parser <http://www.abisource.com/projects/link-grammar/dict/introduction.html>
9. Dennis Grinberg, John Lafferty, Daniel Sleator, "A Robust Parsing Algorithm for Link Grammar", Carnegie Mellon University Computer Science technical report CMU-CS-95-125, and Proceedings of the Fourth International Workshop on Parsing Technologies, Prague (1995) [2]
10. Havelka, J. (2007). "Beyond projectivity: multilingual evaluation of constraints and measures on non-projective structures". In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL-07): 608-615. Prague, Czech Republic: Association for Computational Linguistics.
11. R. Ferrer i Cancho "Why do syntactic links not cross?" EPL 76, 6 (2006), pp. 1228-1234.
12. John Lafferty, Daniel Sleator, Davey Temperley, "Grammatical Trigrams: a Probabilistic Model of Link Grammar" Proceedings of the AAAI Conference on Probabilistic Approaches to Natural Language (1992)[3]
13. Ramon Ferrer-i-Cancho (2013) "Hubiness, length, crossings and their relationships in dependency trees", ArXiv 1304.4086
14. D. Temperley, (2008). "Dependency length minimization in natural and artificial languages". Journal of Quantitative Linguistics, 15(3):256-282.
15. E. Gibson, (2000). "The dependency locality theory: A distance-based theory of linguistic complexity." In Marantz, A., Miyashita, Y., and O'Neil, W., editors, Image, Language, Brain. Papers from the first Mind Articulation Project Symposium. MIT Press, Cambridge, MA.
16. Haitao Liu "Dependency distance as a metric of language comprehension difficulty", 2008, Journal of Cognitive Science, v9.2 pp 159-191. <http://www.lingviko.net/JCS.pdf>
17. Richard Futrell, Kyle Mahowald, and Edward Gibson, "Large-scale evidence of dependency length minimization in 37 languages" (2015), doi:10.1073/pnas.1502134112 <https://doi.org/10.1073/pnas.1502134112>
18. Daniel Sleator, Davey Temperley, "Parsing English with a Link Grammar" Third International Workshop on Parsing Technologies (1993) [4] (See section 6 on categorial grammar.)
19. Carol Liu. "Towards A Link Grammar for Chinese." Computer Processing of Chinese and Oriental Languages - the Journal of the Chinese Language Computer Society. (2001) <https://www.link.cs.cmu.edu/link/papers/index.html>
20. John Dehdari, Deryle Lonsdale, "A Link Grammar for Persian", (2005)[5]
https://web.archive.org/web/20081203084810/http://www.ling.ohio-state.edu/~jonsafari/papers/dehdari_lonsdale_2005.pdf

21. Armin Sajadi, Abdollahzadeh, A., "Farsi Syntactic Analysis using Link Grammar " (In Farsi), Letter of Research Center of Intelligent Signal Processing, Vol 1(9), 25-37 (In Farsi), 2006. <http://web.cs.dal.ca/~sajadi/wp-content/uploads/2012/10/sajadi2006.pdf>
22. Sajadi, A., Homayounpour, M. "Representation of Farsi Morphological Knowledge using Link Grammar"(In Farsi), Letter of Research Center of Intelligent Signal Processing, Vol 1(9), 41-55, 2006.
23. Warren Casbeer, Jon Dehdari, and Deryle Lonsdale, " A Link Grammar parser for Arabic" in Perspectives on Arabic Linguistics: Papers from the annual symposium on Arabic linguistics. Volume XX: Kalamazoo, Michigan, March 2006, Ed. Mustafa A. Mughazy (2006) https://web.archive.org/web/20140512101153/http://www.ling.ohio-state.edu/~jonsafari/papers/casbeer-etal2007_preprint.pdf
24. Документация по связям и по классам слов доступна. <http://www.abisource.com/projects/link-grammar/russian/doc/>
25. Грамматика связей (Link Grammar) <http://slashzone.ru/parser/>
26. Nguyễn Thị Thu Hương, Nguyễn Thúc Hải, Nguyễn Thanh Thủy "Parsing complex - compound sentences with an extension of Vietnamese link parser combined with discourse segmenter" Journal of Computer Science and Cybernetics, Vol 28, No 4 (2012) <http://vjs.ac.vn/index.php/jcc/article/view/1451>
27. AbiWord — Link Grammar Parser <http://www.abisource.com/projects/link-grammar/>
28. Lingua-LinkParser (Perl interfaces) <http://search.cpan.org/~dbrian/Lingua-LinkParser/>
29. "Ruby Link Parser interfaces". <http://www.deveiate.org/projects/Ruby-LinkParser>.
30. JavaScript node.js library <https://github.com/dijs/link-grammar>
31. OpenCog Language Learning http://wiki.opencog.org/w/Language_learning
32. Learning Language from a Large (Unannotated) Corpus <https://arxiv.org/abs/1401.3372>
33. Debian - Package Search Results - link-grammar <http://packages.debian.org/link-grammar>
34. "RelEx Dependency Relationship Extractor". http://opencog.org/wiki/RelEx_Dependency_Relationship_Extractor.
35. The Stanford Parser: A statistical parser <http://nlp.stanford.edu/software/lex-parser.shtml>
36. The Penn Treebank Project <http://www.cis.upenn.edu/~treebank/>
37. Jing Ding; Daniel Berleant; Jun Xu; Andy W. Fulmer (November 2003). "Extracting biochemical interactions from MEDLINE using a link grammar parser". pp. 467–471. ISBN 0-7695-2038-3. <https://ualr.edu/jdberleant/papers/LGPmanuscript8-8-03a.pdf>. Retrieved 2009-09-26.
38. Sampo Pyysalo, Tapio Salakoski, Sophie Aubin and Adeline Nazarenko, "Lexical Adaptation of Link Grammar to the Biomedical Sublanguage: a Comparative Evaluation of Three Approaches", BMC Bioinformatics 7(Suppl 3):S2 (2006). <http://www.biomedcentral.com/1471-2105/7/S3/S2>
39. Harsha V. Madhyastha; N. Balakrishnan; K. R. Ramakrishnan (2003). "Event Information Extraction Using Link Grammar". pp. 16. doi:10.1109/RIDE.2003.1249841. <https://dx.doi.org/10.1109%2FRIDE.2003.1249841>
40. Teguh Bharata Adji; Baharum Baharudin; Norshuhani Zamin (2008). "Applying Link Grammar Formalism in the Development of English-Indonesian Machine Translation System". pp. 17–23. doi:10.1007/978-3-540-85110-3_3. https://dx.doi.org/10.1007%2F978-3-540-85110-3_3
41. A.Sajadi and M.R Borujerdi, "Machine Translation Using Link Grammar", Submitted to the Journal of Computational Linguistics, MIT Press (Feb 2009)
42. Sajadi, A., Borujerdi, M. "Machine Translation Based on Unification Link Grammar" Journal of Artificial Intelligence Review. DOI=10.1007/s10462-011-9261-7, Pages 109-132, 2013.
43. Ruiting Lian, et al, "Sentence generation for artificial brains: a glocal similarity matching approach", Neurocomputing (Elsevier) (2009, submitted for publication).
44. Blake Lemoine, NLGen2: A Linguistically Plausible, General Purpose Natural Language Generation System (2009) <http://www.louisiana.edu/~bal2277/NLGen2.doc>
45. Microplanner and Surface Realization (SuReal) <http://wiki.opencog.org/w/Microplanner>