Computational Linguistics: History & Comparison of Formal Grammars

Raffaella Bernardi

KRDB, Free University of Bozen-Bolzano

Via della Mostra, Room: 1.06, e-mail: bernardi@inf.unibz.it

Contents

1 Formal Grammars 3
2 History of Formal Grammars 4
2.1 Constituency-based vs. Dependency-based 5
2.2 Constituency vs. Dependencies 6
3 DG & CFPSG & CG 8
3.1 Combining Constituency and Dependencies 9
4 Generative Power and Complexity of FGs 10
4.1 DG, CG, CTL, CCG, and TAG 11
5 Meaning entered the scene 12
5.1 Different ongoing efforts 13
5.2 Montague and the development of formal semantics 14
6 Grammars meet Logic & 15
7 .. Computation 16
7.1 Unification 17
8 FG and applications 18
9 HCI via Natural Language 19
10 Natural Language Interfaces to Data Bases 20

10.1 Advantages & Disadvantages 21
10.2 Experiments 22
10.3 Linguistic problems 23
10.4 Sample Architecture 24
10.5 Which approach 25
10.6 Response generation 26
10.7 Restricted NL input 27
10.8 Online Demos 28
10.9 Attempto Controlled English (ACE) 29
10.10 Ambiguity 30
11 Complexity of NL fragments 31
11.1 "Which" from the ontology perspective 32
11.3 User: specification and queries 33
12 English lite 34
13 Questions 35
13.1 Strategies 36
14 Conclusion 37

1. Formal Grammars

- We have seen that Formal Grammars play a crucial role in the research on Computational Linguistics.
- We have looked at Context Free Grammars/Phrase Structure Grammars, Categorial Grammar and Categorial Type Logic

But through the years, computational linguists have developed other formal grammars too.

Today, we will look at the most renown ones, at their generative capacity and their complexity. Then we mention some applications.

2. History of Formal Grammars

Important steps in the historical developments of Formal grammar started in the 1950's and can be divided into five phases:

1. Formalization: Away from descriptive linguistics and behavioralism (performance vs. competence) [1950's 1960's]
2. Inclusion of meaning: Compositionality [1970's]
3. Problems with word order: Need of stronger formalisms [1970's 1980's]
4. Grammar meets logic & computation [1990's]
5. Grammar meets statistic [1990's 2000's]

In these phases, theoretical linguists addressed similar issues, but worked them out differently depending on the perspective they took:

- constituency-based or
- dependency-based.

2.1. Constituency-based vs. Dependency-based

Constituency (cf. structural linguists like Bloomfield, Harris, Wells) is a **horizontal** organization principle: it groups together constituents into phrases (larger structures), until the entire sentence is accounted for.

- Terminal and non-terminal (phrasal) nodes.
- Immediate constituency: constituents need to be adjacent (CFPSG).
- But we have seen that meaningful units may not be adjacent –Discontinuous constituency or long-distance dependencies.
- This problem has been tackled by allowing flexible constituency: "phrasal re-bracketing"

Dependency is an asymmetrical relation between a head and a dependent, i.e. a **vertical** organization principle.

2.2. Constituency vs. Dependencies

Dependency and constituency describe different dimensions.

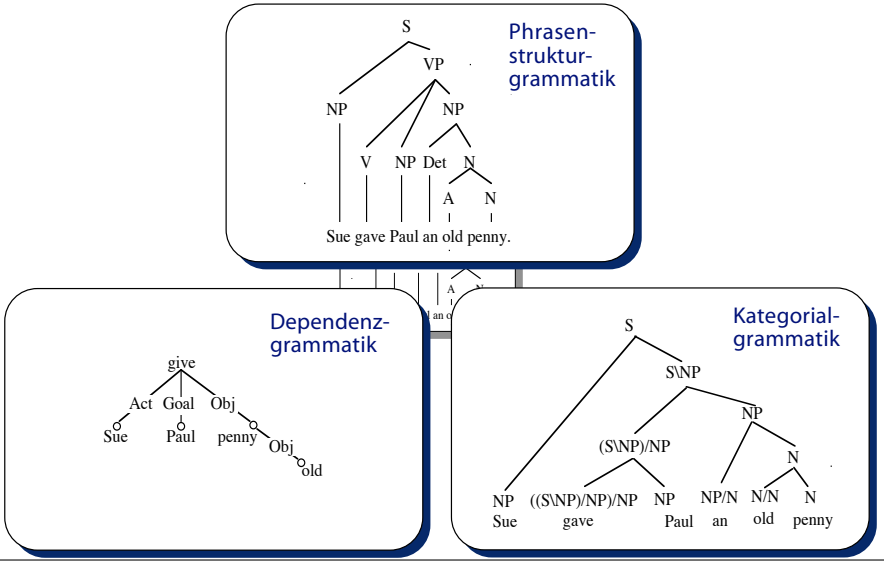
1. A phrase-structure tree is closely related to a derivation, whereas a dependency tree rather describes the product of a process of derivation.
2. Usually, given a phrase-structrue tree, we can get very close to a dependency tree by constructing the transitive collapse of headed structures over nonterminals.

Constituency and dependency are not adversaries, they are complementary notions.

Using them together we can overcome the problems that each notion has individually.

3. DG & CFPSG & CG

THREE TRADITIONS



3.1. Combining Constituency and Dependencies

In 1975, Joshi et ali introduced a grammatical formalism called Tree-Adjoining Grammars (TAGs), which are tree-generating systems. The application of TAGs to natural language is known as LTAGs.

- New way of thinking of domain of dependencies
- Localization of dependencies : elementary structures of a formalisms over which dependencies such as agreement, subcategorization and filler-gap relation can be specified.

Septina is going to review LTAG.

4. Generative Power and Complexity of FGs

Recall, every (formal) grammar generates a unique language. However, one language can be generated by several different (formal) grammars.

Formal grammars differ with respect to their **generative power**:

One grammar is of a greater generative power than another if it can recognize a language that the other cannot recognize.

Two grammars are said to be

- **weakly** equivalent if they generate the same string language.
- **strongly** equivalent if they generate both the same string language and the same tree language.

4.1. DG, CG, CTL, CCG, and TAG

- DG: Gross (1964)(p.49) claimed that the dependency languages are **exactly** the context-free languages. This claim turned out to be a mistake, and now there is new interested in DG. (Used in QA)
- CG: Chomsky (1963) conjectured that **Lambek calculi** were also **context-free**. This conjectured was proved by Pentus and Buszkowski in 1997.
- TAG and CCG: have been proved to be Mildly Context Free.
- CTL has been proved to be Mildly Sensitive (Moot), or Context Sensitive (Moot) or Turing Complete (Carpenter), accordingly to the structural rules allowed.
- LG has been proved to be Mildly Context Free. (Moot 2008)

5. Meaning entered the scene

Chomsky was, in general, **sceptical of efforts to formalize semantics**. Interpretative semantics or the autonomy of syntax: Syntax can be studied without reference to semantics (cf. also Jackendoff).

Criticism on both transformational and non-transformational approaches:

- Transformations do not correspond to syntactic relations, relying too much on linear order.
- Similarly, Curry (1961; 1963) criticized Lambek for the focus on order (directionality).

5.1. Different ongoing efforts

- Developing a notion of (meaningful) logical form, to which a syntactic structure could be mapped using transformations. Efforts either stayed close to a constituency-based notion of structure, like in generative semantics (Fodor, Katz), or were dependency-based (Sgall et al, particularly Panevová (1974; 1975); Fillmore (1968)). Cf. also work by Starosta, Bach, Karttunen.
- Montague's formalization of semantics – though Montague and the semanticists in linguistics were unaware of one another, cf. (Partee, 1997)

5.2. Montague and the development of formal semantics

The foundational work by Frege, Carnap, and Tarski had led to a rise in work on modal logic, tense logic, and the analysis of **philosophically interesting issues in natural language**. Philosophers like Kripke and Hintikka added model theory.

These developments went hand-in-hand with the **logical syntax** tradition (Peirce, Morris, Carnap), distinguishing syntax (well-formedness), from semantics (interpretation), and pragmatics (use).

Though the division was inspired by language, **few linguists attempted to apply the logician's tools in linguistics as such**.

This changed with **Montague**.

> "I reject the contention that an important theoretical difference exists between formal and natural languages." (Montague, 1974)(p.188)

A compositional approach, using a "rule-by-rule" translation (Bach) of a syntactic structure into a first-order, intensional logic. This differed substantially from transformational approaches (generative or interpretative semantics).

6. Grammars meet Logic & ...

Logics to specify a grammar framework as a mathematical system:

- Feature logics: HPSG, cf. (King, 1989; Pollard and Sag, 1993; Richter et al., 1999)
- Categorial Type Logics (Kurtonina, 1995; Moortgat, 1997)

We will hear more about Feature Logics by Valia Kordoni (28th of May).

Dimitriy is going to review CLT.

Logics to interpret linguistically realized meaning:

- Montague semantics: used in early LFG, GPSG, Montague Grammar, Categorial Type Logic, TAG (Synchronous LTAG)
- Modal logic: used in dependency grammar frameworks, e.g. (Broeker, 1997; Kruijff , 2001).
- Linear logic: used in contemporary LFG, (Crouch and van Genabith, 1998).

7. .. Computation

Computation of linguistic structures

- Unification (constraint-based reasoning): LFG, HPSG, categorial grammar (UCG), dependency grammar (UDG, DUG, TDG)
- "Parsing as deduction": CTL
- Optimality theory: robust constraint-solving, e.g. LFG

7.1. Unification

The development of Unification Grammars has strongly been influenced by the:

- use of tools developed in Logics and in AI;
- the progress made in the area of Natural Language Processing;
- Development of Logic Programming: Prolog.

    1. Declarative character: grammar is not a set of rules, but a set of constraints that a sequence needs to satisfy in order for it to be a grammatical phrase.
    2. Constraints do not need to be ordered.

Transformational grammars are inadequate if faced with implementation problems. Derivations proceed from deep structures while automatic sentence analysis requires the inverse process.

Unification grammars or constraint based grammars represent the new syntactic models of the 80's.

We will hear more on this on the 28th from Valia Kordoni.

8. FG and applications

Wide coverage: Syntax-Semantics interface... with all the "compromise" needed to go wide. Statistically based parser.

- Steedman (and Szabolcsi): theory of CCG.
- Julia Hockenmaier: CCG Bank
- Curran, Clark, Bos: softwares http://svn.ask.it.usyd.edu.au/trac/candc/wiki

E.g. Used in QA and Textual Entailment. Could be useful for many applications!

We will hear more about FG and statistics on the 21st from Yannick Versley.

9. HCI via Natural Language

In the '50 Machine Translation work pointed out serious problems in trying to deal with unrestricted, extended text in open domains. This led researchers in the '60 and early '70 to focus on question-answering dialogues in restricted domain.

Attention shifted from developing NL systems to solving individual language-related problems, e.g., to develop faster, and more efficient parsers.

Now, researchers are back to deal with unrestricted extended text and dialogues.

1. NLDB
2. Dialogue Systems,
3. QA
4. IQA

All of them aim at assisting users to access data from some source. Today we speak of NLDB, next time of Dialogue Systems and IQA.

10. Natural Language Interfaces to Data Bases

NLDB refers to systems that allow the user to access information stored in a database by typing requests in some natural language. Its history (see Androutsopoulos for more details):

'60/'70 they were built having a particular DB in mind. No interest in portability issues. E.g., LUNAR

late '70 Dialogues; large DB; semantic grammars (domain dependent - no portable). E.g. LADDER

early '80 From English into Prolog evaluated against Prolog DB. Eg., CHAT-80

mid '80 popular research area. Research focused on portability issues. E.g. TEAM

'90 NLIDBs did not gain the expected commercial acceptance. Alternative solutions were successful (graphical or form-based interface). Decrease in the nubmer of papers on the topic.

10.1. Advantages & Disadvantages

- Advantages:
  - NLDB should be easier to use. But: currently only limited subsets of NL. Hence, training is needed.
  - It supports anaphoric and elliptic expressions.
- Disadvantages:
  - The NL coverage is not clear to the user. False positive expectation and False negative expectation
  - It is not clear to the user whether the rejected question is outside the system's linguistic coverage or the system's conceptual coverage. Need of diagnostic messages.
  - User assume intelligence of the NLIDBs.
  - NL is verbose and ambiguous.
  - Tedious configuration.

10.2. Experiments

- Training of the interfaces (graphical, SQL, NL). Then ask queries most of which are similar to the ones used in the training period.

  Results: NLIDBs seem to be better in queries where data from many tables have to be combined and in queries that were not similar to the ones the users had encountered during the training period.

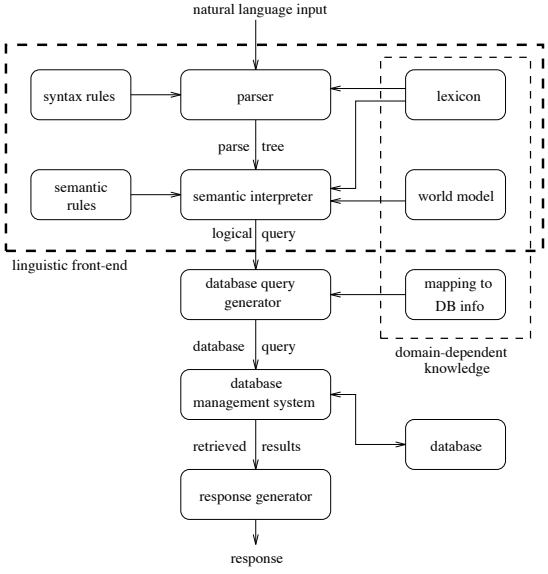
- NL is an effective method of interaction for casual users with a good knowledge of the DB, who perform question-answering tasks in a restricted domain.

- Another approach: Wizard of Oz experiment.

10.3. Linguistic problems

- Quantifier scoping. Ambiguous, ad hoc solutions (e.g., choose only one reading as possible, give different weights to QPs.)
- Conjunction and Disjunction: Sometime conjunctions in NL are actually interpreted as disjunction. E.g., List all applicants who live in California and Arizona. There are also cases of ambiguous use of "and", e.g., Which minority and female applicants know Fortran?"
- Nominal compound problem: E.g., "research department" vs. "research system". In the first case, the department carries out the research, in the second the system is used in research.
- Anaphora: Use of pronouns and possessive determiners or noun phrases to denote entities mentioned in the discourse. Solution: keep list of all entities, use the most recent one as link to the anaphora. Use of world knowledge.
- Elliptical sentences. E.g., U1: "Who is the manager of the largest department?" U2: "The smallest department?" Need of discourse model.

10.4. Sample Architecture



10.5. Which approach

Advantages of the last approach: modularity of the architecture

- the linguistic front-end is independent of the underlying DBMS
- domain knowledge is separated from the rest of the front end
- reasoning modules can be added between the semantic interpreter and the DB query generator.

10.6. Response generation

Failure Explain cause of failure to retrieve answer.

False Presupposition The system should report the false presupposition about the DB world.

Literal answers some time a literal answer would be "yes/no" but it won't be an acceptable answer. Cooperative answers can help. Sometime important to reason about the user's goal.

Misunderstandings translate the SQL query back to NL, (paraphrase modules)

10.7. Restricted NL input

Currently systems use limited subsets of NL.

Limitation user doesn't know which is this subset. Has to rephrase the question, does not know which questions could be handled.

Long term aim to broad the linguistic coverage.

Alternative approach deliberately and explicitly restrict the set of NL the user is allowed to input (controlled natural language.)

syntactic pattern

menu-based

ontology-driven See Enrico Franconi and Paolo Dongilli's work.

complexity of NL fragments See Ian Pratt and Camilo Thorne works

10.8. Online Demos

Examples of today NLDBs:

- ACE: http://attempto.ifi.unizh.ch/site/tools/
- Geo http://www.cs.utexas.edu/users/ml/geo-demo.html
- PENG: http://www.ics.mq.edu.au/~peng/PengEditor.html
- PRECISE

Ronell could give an overview of existing CNL systems. (See CNL 09 Proceedings).

10.9. Attempto Controlled English (ACE)

- Lexicon: limited set of type of words: e.g.

  "ACE verbs are in 3rd person singular or plural, in indicative mood, and in simple present tense. Both indicative and passive verbs can be used but passive constructions must include a prepositional phrase, e.g. '. . . by . . . '."

- Grammar: limited set of constructions.

  "Sentence are a concatentation of a NP with a VP. It is possible to create well-formed-sentence with a single NP prefixed by the fixed phrase "there is/are". Composite declarative sentences are recursively built from simpler sentence using the predefined constructors: coordination, negation, global quantification, if-then subordination."

10.10. Ambiguity

The sentences of ACE are handled by the parsers and receive always only one MR, even in case they could be ambiguous.

E.g., relative clauses always attach to the most recent noun.

Every man owns a dog_d that_d likes a cat_c that_c likes a mouse and that_c eats a bone.

They also deal with anaphora resolution –the use Discourse Representation Structures (DRS).

They generate paraphrases of the sentence to make sure the system and the user agree in the assigned interpretation.

Paraphrases is becoming a hot topic for HCI via NL.

11. Complexity of NL fragments

The FOL meaning representation of the entailment above is:

$$\{\forall x(man(x) \rightarrow mortal(x)), man(socrates)\} \models mortal(socrates)$$

Pratt has proved that COP is **PTIME**

Fragment	Decision class for satisfiability
COP+TV+DTV	**PTIME**
COP+REL	**NP-Complete**
COP+REL+TV	**EXPTIME-Complete**
COP+REL+TV+DTV	**NEXPTIME-Complete**
COP+REL+TV+RA	**NEXPTIME-Complete**
COP+REL+TV+GA	undecidable

REL relative pronoun.
RA restricted anaphora, pronouns take their closest allowed antecedents.
GA general anaphora.

11.1. "Which" from the ontology perspective

Which fragment? Our proposal is to merge Pratt's approach with the research mentioned above and use, as controlled language for accessing ontologies, those fragments with a **desirable computational complexity**.

- Description Logics (DLs) are the logics that provide the formal underpinning to ontologies and the Semantic Web.
- They are a decidable fragment of FOL, and experience has shown that they have the right expressivity required by the most commonly used formalisms for conceptual modeling, e.g. `UML` class diagrams and entity-relationship schemas.
- DL-lite is the maximal DL that has the ability to efficiently and effectively manage very large data repositories by relying on industrial-strength relational database management systems (`RDBMS`). Moreover, DL-lite can still capture the essential features of both `UML` class diagrams and `ER` schemas.
- Hence, we use DL-lite as the starting point to answer the **which** part of our question, viz. to pinpoint the most suitable fragment to add **specifications** in the ontology.

11.3. User: specification and queries

We consider the case where the ABox is actually stored in a database, and hence managed by a `DBMS`.

Given a DL-lite TBox $\mathcal{T}$ and a *DB* (storing the ABox), a user can be interested in:

1. adding new specifications to the TBox,
2. adding new facts to the DB, or
3. querying the DB.

12. English lite

The constraints expressed in the `TBox` are universals. They are of the form $Cl \sqsubseteq Cr$ that translates into FOL as $\forall x.Cl(x) \rightarrow Cr(x)$ and in natural language as

(a) [Every $\underbrace{\texttt{NOUN}}_{Cl}$] $\underbrace{\texttt{VERB\_PHRASE}}_{Cr}$

(b) [[Everyone $\underbrace{[\text{who } \texttt{VERB\_PHRASE}]}_{Cl}$] $\underbrace{\texttt{VERB\_PHRASE}}_{Cr}$]

Hence, the determiner "every" and the quantifier "everyone" play a crucial role in determining the linguistic structures that belong to the natural language fragment corresponding to a DL-lite TBox.

We have to zoom into the `NOUN` and `VERB_PHRASE` constituents.

In other words, we spell out how the Cl and Cr of DL-lite can be expressed in English.

13. Questions

- Can we be satisfied?
- Can we do more, and define a grammar that recognizes “all and only” linguistic structures whose meaning representation is in DL-lite?
- But how can we define the “all”?
- Would a user be happy in using a Controlled Natural Language?
- How far is this CNL from the sentences that a user would naturally use to access Information Systems?
- Would we ever be able to bridge this gap?

13.1. Strategies

- Analyze corpus of questions to DB

  We have looked at `Geo880` (set of 880 queries to a US geography).

  Most of the queries were conjunctive queries, but the one involving: (i) aggregation functions (highest, most, longest etc.), and (ii) counting (how many, higher, etc) but the latter could be handled in some restricted form.

- Built a grammar able to recognize only `CQs` while building their meaning representation.

- Try experiments to test user satisfiability to enter specifications in the ontology and query a DB.

- Study the literature on Text Simplification for e.g. people with aphasia. Aim: to re-write users' questions into simplified and suitable ones.

- "All" sentences in DL-lite ... still a mystery.

14. Conclusion

Next time we will be looking at QA and IQA.

Projects: If we have not agree on your topic yet. Let's do it now.

Project Presentation day: 3rd of June?