Dynamic language models

Damir Ćavar

Agenda

Agenda

- Language models

Agenda

- Language models
- Dynamic models

Agenda

- Language models
- Dynamic models
- Induction of language regularities

Agenda

- Language models
- Dynamic models
- Induction of language regularities
- Research directions

Modeling Language

Modeling Language

- Symbolic rule-based approaches

Modeling Language

- Symbolic rule-based approaches
  - Direct linguistic foundation

Modeling Language

- Symbolic rule-based approaches
  - Direct linguistic foundation
    - Grammars, rules and linguistic heuristics

Modeling Language

- Symbolic rule-based approaches
  - Direct linguistic foundation
    - Grammars, rules and linguistic heuristics
  - Complexity on the level of grammar development

Modeling Language

- Symbolic rule-based approaches
  - Direct linguistic foundation
    - Grammars, rules and linguistic heuristics
  - Complexity on the level of grammar development
  - Simple implementations with potentially complex computations

Modeling Language

- Problems:
  - Coverage and robustness
    - Complexity of grammars and deviation of real language data
  - Bias and lack of flexibility/interoperability
    - Theory driven and with a specific formalism.

Modeling Language

- Statistical approaches
  - Indirect linguistic foundation or
    - Corpora (e.g. audio, text)
  - Direct linguistic foundation with quantification of grammars and rule-sets
  - Sparse data problem

Problems

- Static models: grammar-based and statistical (empiricist or connectionist)
- Dynamic language properties: changes
  - lexical (e.g. morphological, semantic)
  - grammar (e.g. likelihood of constructions, new constructions types)
  - domains (e.g. pragmatics)

Possible Solutions

- Dynamic models:
    - Adaptive
    - Deductive and inductive
    - Symbolic and/or statistical

Concepts

- Deduction
  - Logic and rule-based
  - Meta-knowledge driven
  - Core statistical model
- Induction
  - Empirically and data-oriented

Concepts

- Induction:
  - Identification of basic strategies with broad coverage
    - for language types
    - for linguistic levels
  - Intuition:
    - Language properties are full of regularities and patterns, at some levels these should be learnable

Research Questions

- Which language properties can be induced with what kind of strategies and effort?
  - Specification of strategies for learning of regularities at different linguistic levels (e.g. phonology, morphology, syntax, semantics).
  - Typologies of languages on a technical and formal learning strategies scale.

Bootstrapping Cues

- Various hypotheses about what kind of language properties serve as cues for (induction or deduction of) linguistic knowledge
  - Phonological bootstrapping
  - Role of lexical items (e.g. function words)
  - Semantic bootstrapping

Bootstrapping Cues

- Other possible cues:
  - Morphological regularities
  - Used successfully in language technology:
    - Samuelsson (1994), later in Brants' (2000) TnT

Acquisition of Morphology

- Acquisition of morphological regularities:
  - Incremental
  - Phases with deviations from target grammar
  - Persistence of learners: corrections ignored, mismatch between parsing/processing and production
  - Stable target grammar intuition

Theoretical Concepts

- Principles and Parameters Model
- Optimality Theory Approach
- Connectionist Models
- Here:
    - Purely empiricist approach

Theoretical Concepts

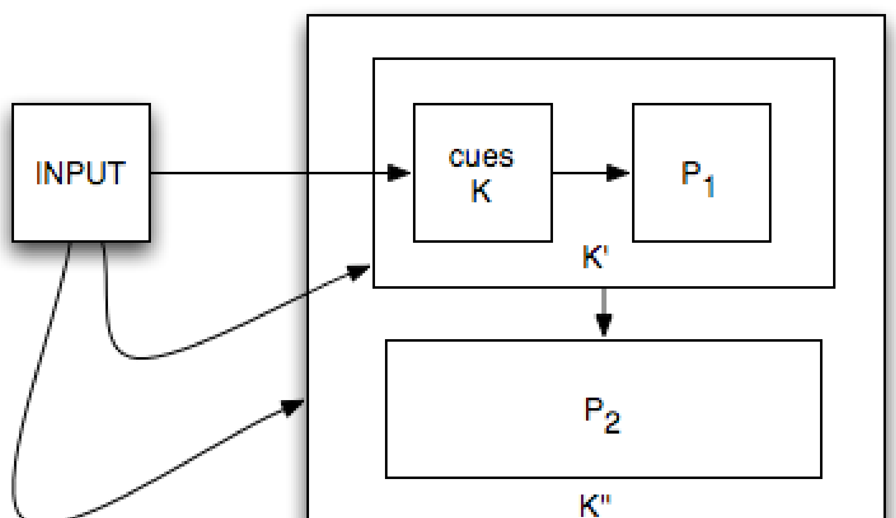
- What kind of language properties can be identified from just language data?
- How can these properties be used to learn/ induce higher level linguistic knowledge?
  - What kind of linguistic knowledge is needed to achieve this?
  - What are crucial differences between languages?

Applied Context

- Use for language technologies:
  - From approx. 5500 languages, only 1% is adequately described and has more or less adequate technological resources.
  - Universal (dynamic, adaptive, extensible) solutions (minimally language specific) can increase the development speed of NLP tools.

Cue-based Learning

- Incremental Cue-based Learning
  - Initial Bootstrapping Phase: An initial set of cues K identifies specific constraints and their ranking P_1 given some input.
  - Subsequent Bootstrapping Phases: Together with the set of cues K and the induced knowledge P_1 a new set of cues K' is derived, and so on.



Cue-based Learning

- Elementary Cues
  - e.g. phones, morphemes, phrases and their statistical, distributional, and information theoretic properties
- Secondary Cues
  - e.g. phonemes, categories (types) and their statistical, distributional, and information theoretic properties

Cue Identification

- Secondary level cue-identification:
  - Sparse data problem on the token level.
    - Solution:
      - Typing: identifying properties of elementary units (e.g. morphemes) on the basis of:
        - morphological properties
        - syntactic properties

Alternative

- Basic constraints are fundamental and not "symptom" related.
  - Information Theory (e.g. Entropy)
  - Statistical (e.g. Frequency)
  - Distributional (e.g. absolute or relative position and relation to others)
- Language specific constraints can be induced.

Architecture

- General principles:
  - Incremental input with incremental grammar induction and optimization
  - Minimum revisions via restricted memory (short term memory)
  - Learning only from previous experience

Fundamental Constraints

- Language properties: equilibrium between
  - size of grammar
  - usability

$$\text{size} \longleftarrow \text{grammar} \longrightarrow \text{usability}$$

Description Length

Description Length

sleeps
sleep
dream
dreams

Data

Description Length

sleeps
sleep
dream
dreams

Data

sleeps $P(sleeps)$
sleep $P(sleep)$
dream $P(dream)$
dreams $P(dreams)$

Hypothesis 1
size: 38 bytes

Description Length

sleeps
sleep
dream
dreams

Data

sleeps *P*(*sleeps*)
sleep *P*(*sleep*)
dream *P*(*dream*)
dreams *P*(*dreams*)

Hypothesis 1
size: 38 bytes

sleep *P*(*sleeps*) Ptr(-s)
dream *P*(*dream*) Ptr(-s)
-s *P*(*-s*)

Hypothesis 2
size: 33 bytes

Minimum Description Length

- Evaluation in a constraint satisfaction system:
  - Minimum Description Length Principle: Minimize the description length of the language model, including the size of the described data. (Occam's razor) (Gruenwald et al. 2005)
  - Trade off goodness-of-fit on the observed data with the *complexity* or *richness* of the data.

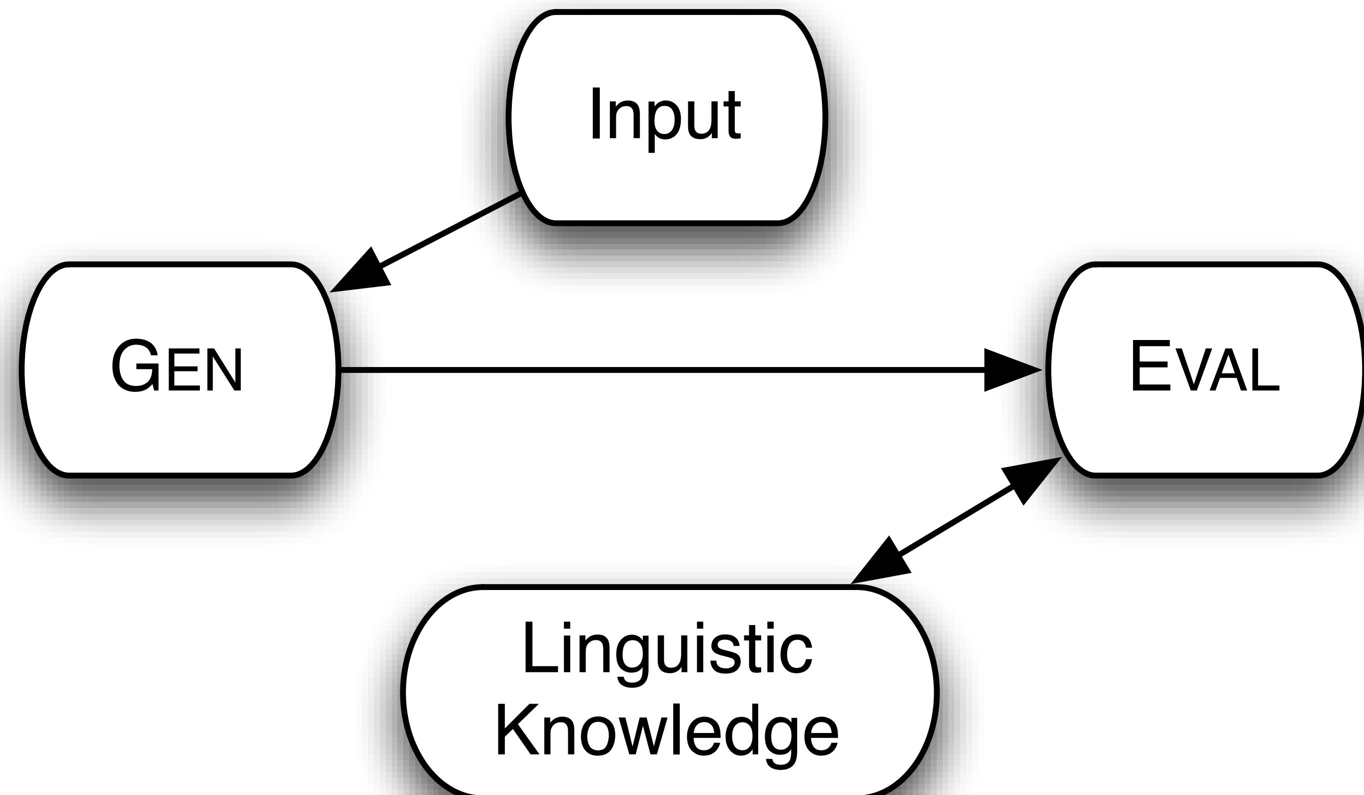
Minimum Description Length

- Let H_1, H_2 ... H_n be a list of candidate models. The best hypothesis $H \in H_1 \cup H_2 \cup \ldots H_n$ to explain the data D is the one which minimizes the sum $L(H)+L(D|H)$.
  - $L(H)$ is the length, in bits, of the description of the hypothesis; and
  - $L(D|H)$ is the length, in bits, of the description of the data when encoded with the help of the hypothesis.

Architecture

- General processes:
  - Generation of hypotheses for a given input
  - Selection of appropriate hypotheses
  - Induction of grammar rules/constraints and their ranking

General Induction Architecture



Architecture

- Hypothesis generation:
  - Random or complete
  - Statistical:
    - Transitional probabilities (Harris, 1955)
    - EM-based (Brent, et al.)
  - Alignment based (ABL) (van Zaanen, 2001)

Architecture

- Hypothesis generation: ABL
  - Substitutability and Complementarity
    - Given two words (one known word and one unknown input word), the edges of matching substrings mark morphological boundaries.
  - Advantage:
    - Learning from previous knowledge.

Evaluator

- Weighted voting constraints:
  - Minimum Description Length
  - Mutual Information (point-wise, average, left- and right)
  - Relative Entropy
  - Surface constraints: morph. length, frequency, segment count, etc.

Architecture

- Grammar size
  - Minimum Description Length Principle (MDL)
    - From n grammars that describe the same data, chose the grammar with the smallest size (e.g. number of symbols, length of terminals)

Architecture

- Grammar size
  - Relative Entropy
    - From a set of hypotheses about the structure of an input i, add the hypothesis h to the set of grammar rules/hypotheses that results in lowest divergence from the original grammar.

Architecture

- Grammar size
  - Relative Entropy
    - We calculate RE as a variant of the Kullback-Leibler Divergence
    - Given grammars G1 and G2, choose the grammar that has the smallest divergence from the initial grammar G0.

Architecture

- Grammar size - Relative Entropy
  - Kullback-Leibler Divergence

$$\sum_{x \in X} P(x) lg \frac{P(x)}{Q(x)}$$

$$\sum_{x \in X} P(x) lg \frac{1}{P(x)}$$

Architecture

- Hypothesis evaluation: Mutual Information

$$\sum_{y \in \{<xY>\}} p(< xy > |x) lg \frac{p(< xy >)}{p(x)p(y)}$$

- Pairwise summation of left MI of x and right MI of y.
- Accepting morpheme boundaries at local MI-maxima.

Architecture

- Mutual Information
  - symmetric: MI(<xy>) = MI(<yx>)
  - frequency sensitive
- Relative Entropy
  - asymmetric: given <xy>, RE(y) ≠ RE(x)

Architecture

- Usability related criteria:
  - Frequency of Morpheme Boundaries
  - Number of Morpheme Boundaries
  - Length of Morphemes

Architecture

- Restricted grammar optimization:
  - Small short-term memory window (e.g. 100 utterances).
  - Optimization of the sub-grammar within the window.
    - Significance of the generated rules: elimination of rules with low significance scores.

Architecture

- Voting-based architecture:
    - Every component votes for a hypothesis (= grammar)
    - The hypotheses with the highest votes win.

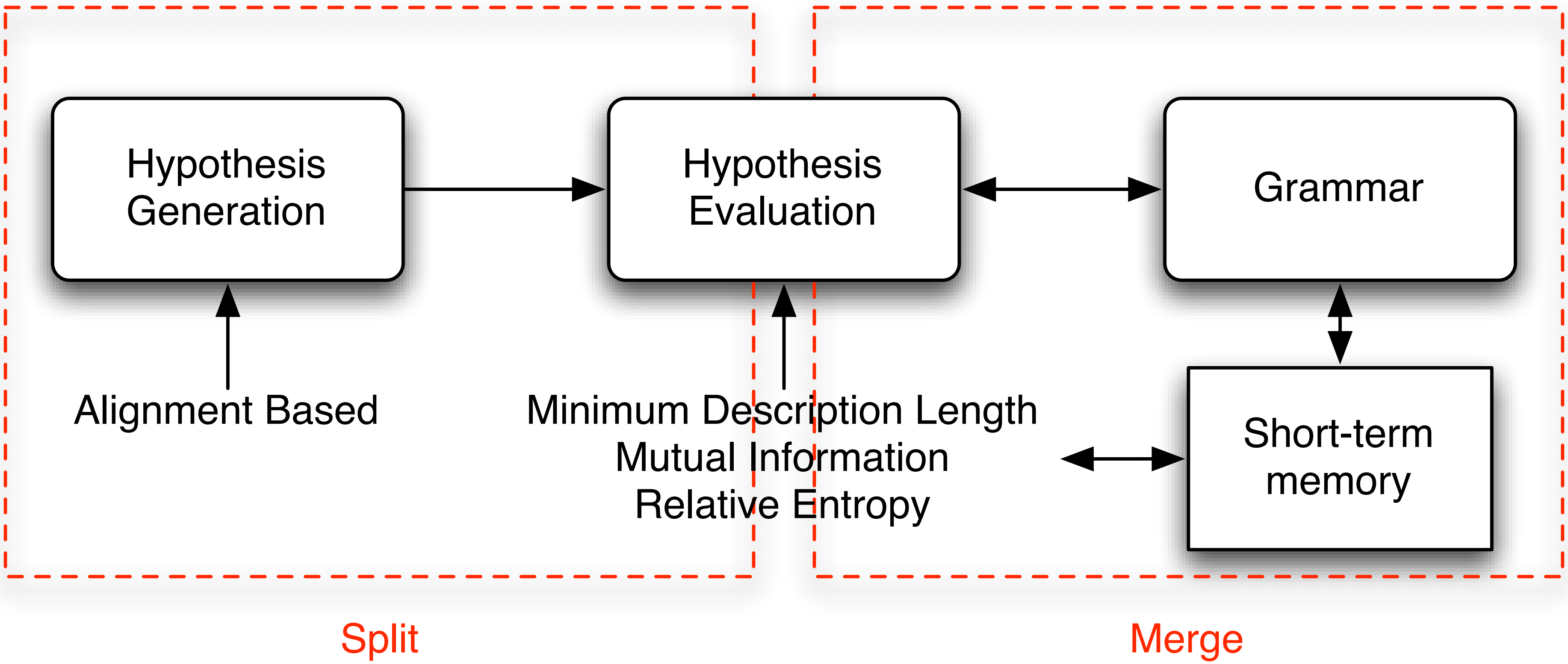
Architecture

- Weighting of constraints:
  - Every voter is weighted (0-1)
    - Compatible to constraint ranking

Architecture

- Weighting of constraints:
  - Means of self-supervision:
    - Online adjustment of the weights of the constraints that produce hypotheses that do not enter grammar.
    - Partially equivalent to Error-driven Constraint Demotion

ABUGI



Hypothesis Generation
Hypothesis Evaluation
Grammar
Alignment Based
Minimum Description Length
Mutual Information
Relative Entropy
Short-term memory
Split
Merge

Architecture

- Input: Utterances with word boundaries
  - *The cars are ugly.*
- Output:
  - Signature for every morpheme merged with previously generated signatures:
    - *#car*$ = [NONE, *s*$]
    - *s*$ = [*#car*$, …]

Morphology Induction

- Evaluation Gold-standard:
  - manual segmentation of:
    - CHILDES Peter corpus
    - 10% Brown corpus
  - CELEX

Morphology Induction

- Evaluation:
  - Online incremental self-evaluation
    - Parallel input: raw & bracketed words
    - Reason:
      - Evaluation of grammar development
      - Visualization of saturation curve

Morphology Induction

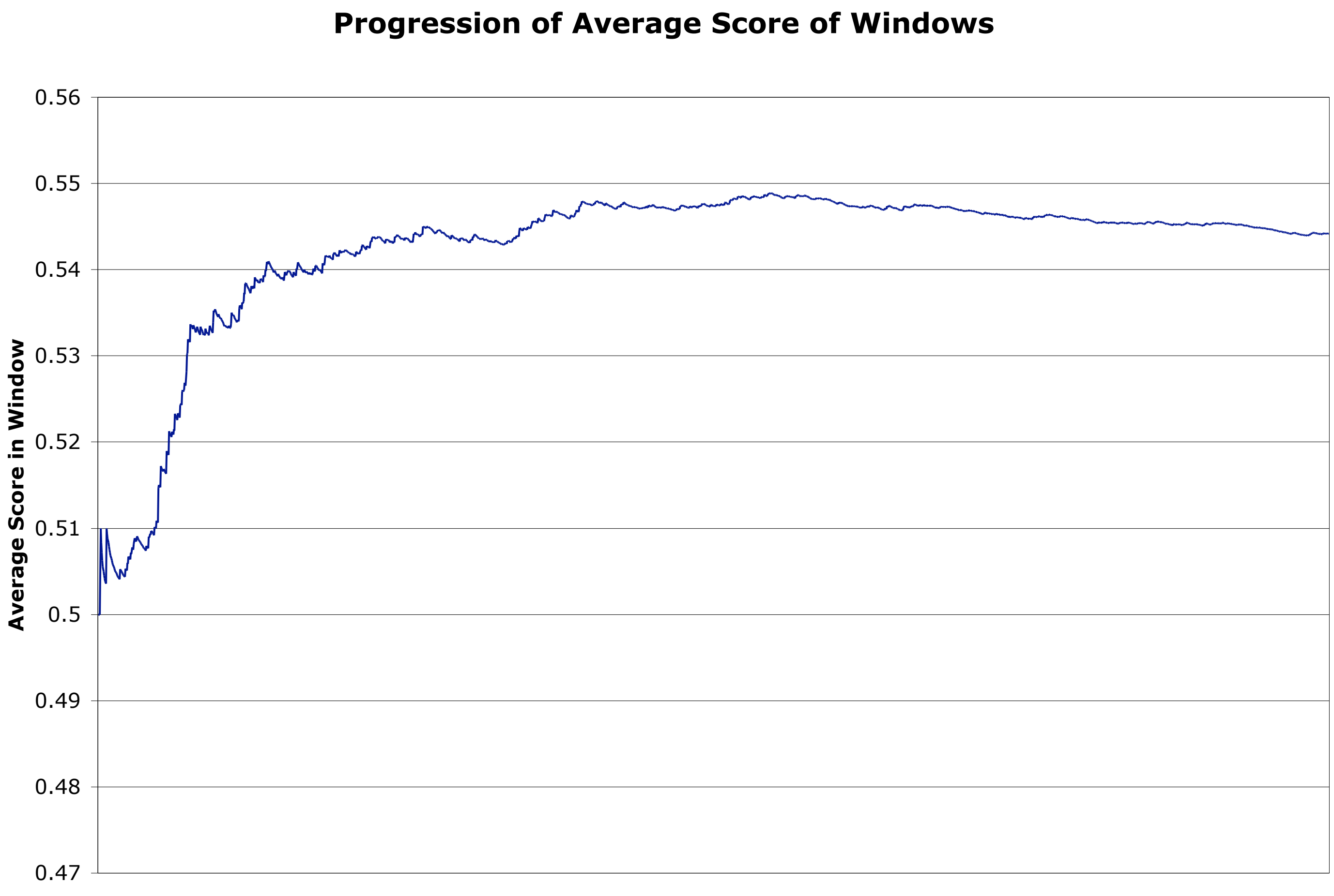
- Evaluation:
  - Offline incremental human evaluation
    - At every increment of grammar size s, dump the grammar.
    - Human annotation of paradigms and segmentation.

Morphology Induction

- Evaluation:
  - Corpora:
    - English: CHILDES, Brown corpus, Penn Treebank
    - Latin: Caesar “De Bello Gallico”
    - Japanese: “Genji Monogatari”

Morphology Induction

- Results: `F = (beta^2 + 1)*precsion*recall / ((beta^2*precision) + recall)`



Morphology Induction

- Brown & CHILDES Peter corpus (English):
  - Precision: 100%
  - Recall: ca. 80%
- Latin:
  - Precision: 99%
  - Recall: 35%

Morphology Induction

- No supervision wrt. notions of stem and affix:
  - Notions of stem or affix are derivable via clustering on the basis of the signatures.
    - s# = [$drink#, $sleep#, $dream#, ...]
    - $smoke# = [NONE, s#, ed#, ...]

Morphology Induction

- Acquisition Order (English):
  - Inflectional Morphology first
  - Derivational Morphology second
  - Prefixes and Infixes last
- Corresponds to observations from language acquisition
- Corresponds to the frequency distribution of these morpheme types