Referring expression generation for question answering and graph visualization

Ivan Rygaev, Dialogue 2018
Laboratory of Computational Linguistics
Institute for Information Transmission Problems RAS, Moscow, Russia

Referring expressions

- A pronoun or a definite noun phrase which the speaker uses to refer to an individual
  - he
  - Ivan
  - the speaker
  - the tall man in a white t-shirt
  - the one who is standing over there
- Should allow the hearer to uniquely identify the referent (in the context)

Referring expression generation

- A subtask of the natural language generation
  - A scene (context) is given in some formal structure
  - Contains many objects with their properties and relations
  - The task is to produce description of the given target object in the scene context
- Requirements
  - Uniqueness (unambiguous target identification)
  - Naturalness (human-likeness)
- Two stages
  - Description content selection
  - Surface linguistic realization

Question-answering system

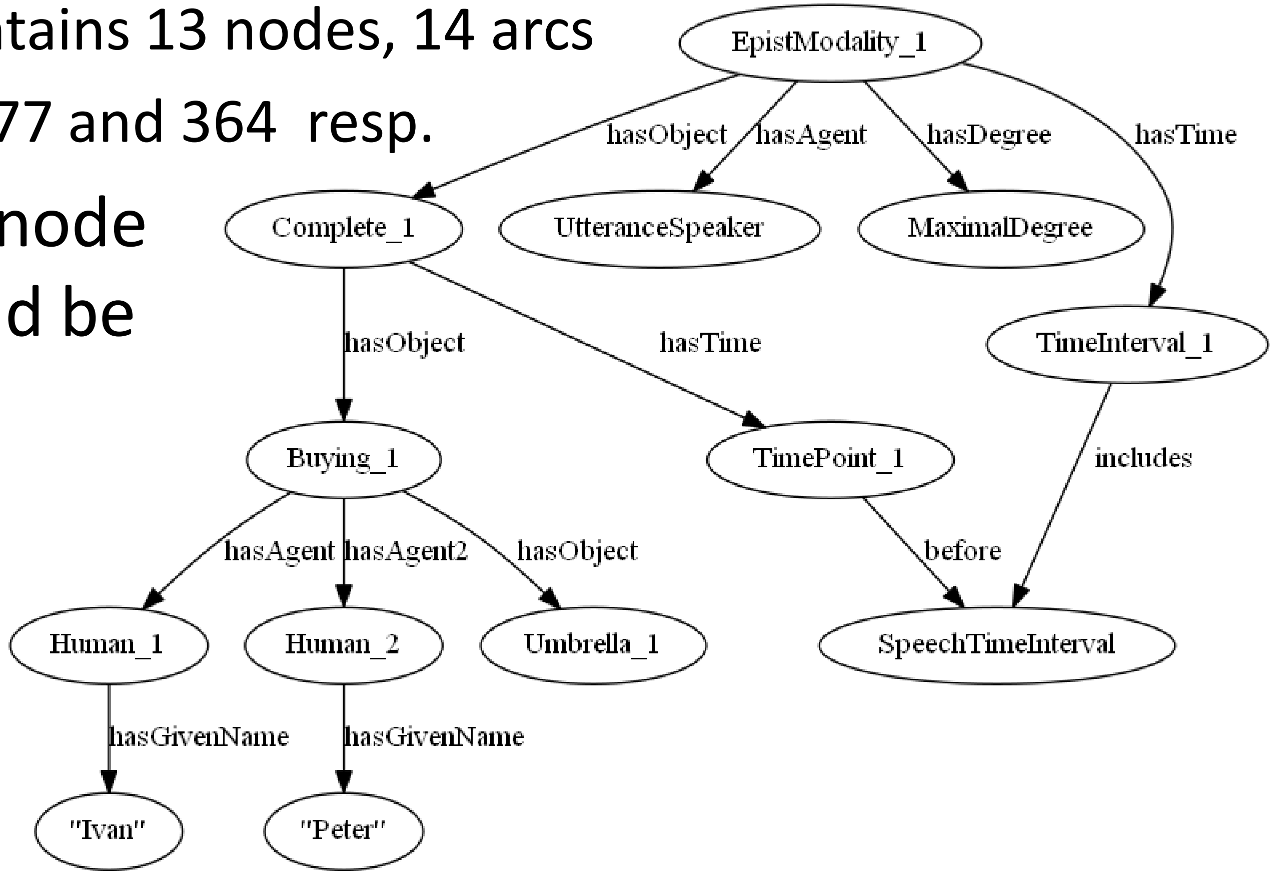
- SemETAP semantic analyzer is a part of ETAP-3
  - Translates an original sentence to a language-independent semantic representation (a semantic graph).
  - Applies logical rules to infer new knowledge.
  - Answers questions by converting them into a semantic graph and using it as a pattern for search.
  - Returns an internal name of the object and some limited set of properties – object type and name (if exists).
    Agent_2_1 = Human_1_1
    Agent_2_1_FamilyName = “Иванов”
  - Easily convertible to a noun phrase but it is not enough

The task

- Cover REG for answers in general (when name is not available and type is not enough):
  - *Arshavin could not save the match. Who lost the match?*
  - *Arshavin's team*
- Scope of this paper:
  - Only specific singular reference
  - Only content selection task (no surface realization yet)
- Graph-based approach:
  - The scene is a semantic graph
  - A description subgraph should be selected

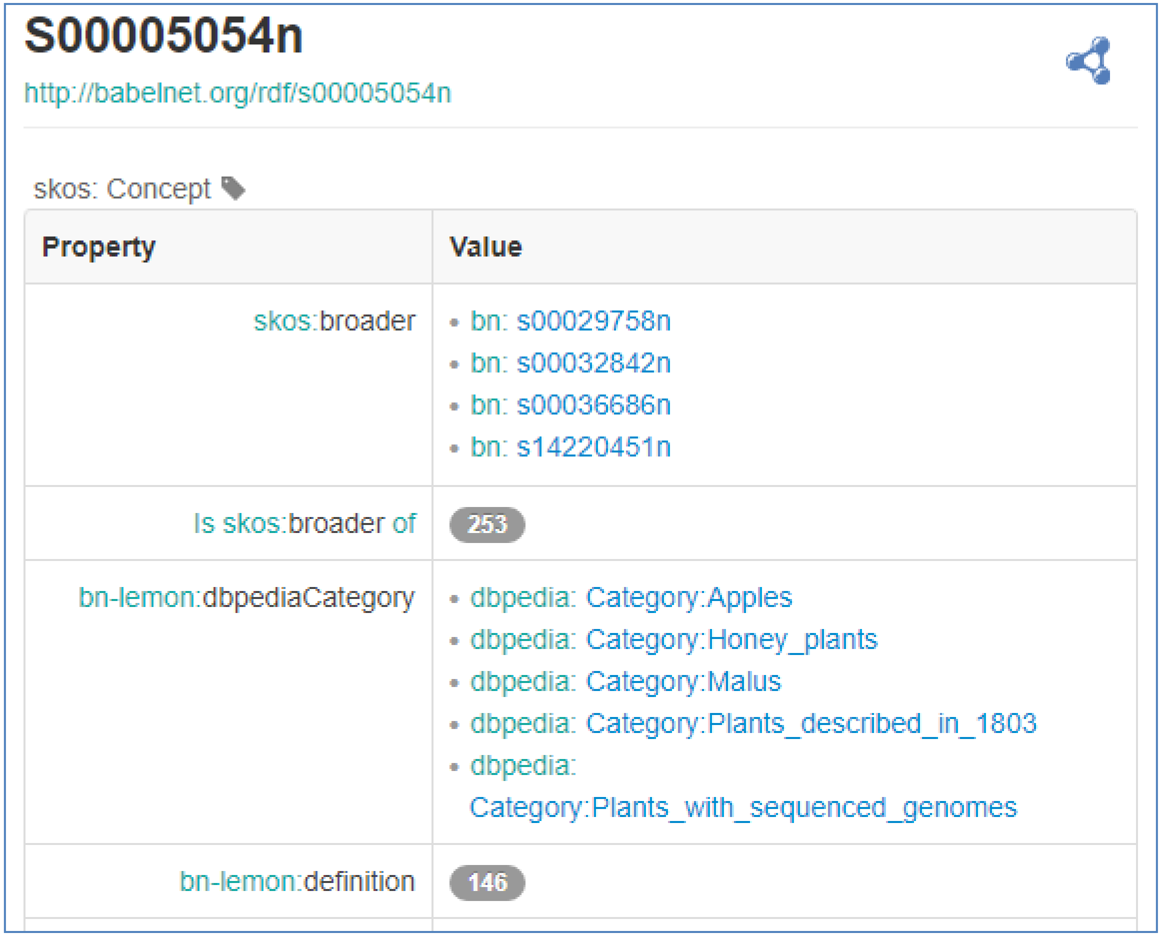
Graph visualization

- *Ivan bought an umbrella from Peter*
  - BSemS contains 13 nodes, 14 arcs
  - EnSemS – 77 and 364 resp.
- Descriptive node names would be very useful



Semantic Web

- BabelNet linked data interface:
  - Semantic concept of Apple (fruit):
- Descriptive node names would be very useful



S00005054n

http://babelnet.org/rdf/s00005054n

skos: Concept

Property	Value
skos:broader	• bn: s00029758n • bn: s00032842n • bn: s00036686n • bn: s14220451n
Is skos:broader of	253
bn-lemon:dbpediaCategory	• dbpedia: Category:Apples • dbpedia: Category:Honey_plants • dbpedia: Category:Malus • dbpedia: Category:Plants_described_in_1803 • dbpedia: Category:Plants_with_sequenced_genomes
bn-lemon:definition	146

Existing algorithms

- Full Brevity (Dale 1989)
  - Generates shortest possible description
  - Computationally expensive and of low human-likeness
  - People often produce non-minimal descriptions
- Greedy Heuristics (Dale 1989, 1992)
  - Incrementally adds the most discriminating property
- The Incremental Algorithm (Reiter and Dale 1992)
  - Incrementally adds a property based on preference order
  - People prefer certain properties over others
  - Polynomial complexity and best human-likeness

Existing algorithms

- Relational data
  - The Incremental Algorithm works well for simple properties but not for relational ones
  - *the dog next to the tree in front of the garage*
    the dog in front of the garage
- Graph-based algorithm (Krahmer et al 2003)
  - Branch and bound algorithm
  - Uses cost function to guide the search
  - Can mimic all the three algorithms with some changes
  - Fits well into our graph-based semantic representations

Our modifications

- Breadth-first search instead of depth-first search
  - Much faster for finding short descriptions in a big graph
- Cost function based on relation preference order
  - hasName, hasFamilyName, hasObject, hasAgent, etc.
- Node types are automatically added for free
  - *'Something on something'* vs *'The cup on the table'*
  - Not only more natural but faster
- Hard length limit
  - Not to hang when there is no unique description at all
  - Long descriptions are not very useful anyway

Graph visualization example

- *Ivan bought an umbrella from Peter* (EnSemS)

- (EpistModality hasObject (Complete hasObject (Buying)))
  - hasAgent UtteranceSpeaker
  - hasTime (TimeInterval isTimeOf (EpistModality))
  - hasObject
    - (Complete hasObject (Buying))
      - hasTime (TimePoint isTimeOf (Complete hasObject (Buying)))
      - hasObject
        - (Buying)
          - hasAgent (Human hasGivenName Ivan)
          - hasTime (TimeInterval isTimeOf (Buying))
          - hasObject (Umbrella)
          - hasAgent2
            - (Human hasGivenName Peter)
              - hasGivenName Peter
              - isAgentOf (Selling)
              - isAgentOf (Exchange hasObject (Umbrella))
              - isAgentOf (Giving hasObject (Umbrella))
              - isAgentOf (Own hasObject (CurrencyMeasure) isResultOf (Giving))
              - isAgentOf (Own hasObject (Umbrella) isPreconditionOf (Giving))
              - isAgent2Of (Buying)
              - isAgent2Of (Exchange hasObject (CurrencyMeasure))

Evaluation

- Performance evaluation
- Human evaluation
  - Informativeness (target object is identified unambiguously)
  - Naturalness (people can choose similar expression)
- No linguistic realization yet
  - Only preliminary human evaluation possible
  - Expressions are presented in Etalog formal language
  - Evaluated by linguists who are familiar with Etalog

Performance evaluation

Description length	Average generation time, ms	Average number of iterations per target	Average time per iteration, ms
1	9.80	1.00	9.80
3	37.40	14.93	2.50
5	369.35	113.96	3.24
7	2565.00	772.85	3.32

- Time grows exponentially with the description length
  - This is the main reason for hard length limit
  - Growth mostly comes from the number of iterations
  - Iteration time growth is moderate

Human evaluation

- 51 sentences from the football high spots corpus
- They were not previously used for REG development
- Manually created questions to these sentences
- Etalog answers presented to 4 linguists to evaluate
  - Informative? (yes/no)
  - Natural? (yes/no)
- A total of 287 question-answer pairs were evaluated

Human evaluation

Referred individual type	# of answers	Informativeness	Naturalness
Person (by name)	81	100.00%	98.42%
Person (by other means)	28	36.04%	18.02%
Football team	75	28.16%	17.20%
Place (penalty area, etc.)	41	62.73%	41.61%
Event (pass or shot)	32	58.59%	47.66%
Time	22	19.32%	3.41%
Ball	7	100.00%	42.86%
Body part	1	100.00%	50.00%
Total	**287**	**59.27%**	**47.04%**

Examples

- Good:
  - *После навеса в штрафную в исполнении Кержакова Аршавин блестящим ударом в падении вколачивает мяч в сетку.*
    After a pass by Kerzhakov into the penalty area Arshavin with a brilliant shot in the fall hammers the ball into the net.
  - *За какую команду играет Аршавин?*
    Which team does Arshavin play for?
  - *(FootballTeam isObjectOf (PlaysFor hasAgent (Human hasName "Кержаков")))*
    The team which Kerzhakov plays for

Examples

- Unnatural:
  - *Валладарес переправил мяч в перекладину, от которой тот покинул пределы поля!*
    Valladares repelled the ball into the crossbar, from which it left the field!
  - *Что Валладарес переправил в перекладину?*
    What did Valladares repel into the crossbar?
  - *(Ball isAgentOf (Leaving))*
    The leaving ball

Examples

- Uninformative:
  - *Ревякин спасает свою команду (сначала) после удара Кержакова, вытащив мяч из-под перекладины, (а затем и Семака).*
    Revyakin saves his team (first) after Kerzhakov's shot, pulling the ball from under the crossbar, (and then after Semak's one).
  - *Откуда вытаскивают мяч?*
    Where the ball is pulled from?
  - *(Region isObjectOf (Below))*
    Below something

Lessons learned

- For REG algorithm itself:
  - Cost function should be configured more carefully. Some descriptions were formally distinguishing but useless. Probably more complex cost function is required
- For the scene graph (EnSemS) construction:
  - Duplicated individuals are not always combined together by coreference rules. This leads to longer unnatural descriptions
  - Concept decomposition sometimes lacks necessary info. *'The team with a coach'* would not be seen distinguishing if all teams had a coach by decomposition.

Linguistic realization

- Etalog expression can serve as a template:
  - It is already tree-like, a template for a syntactic tree
  - *(Own hasObject (CurrencyMeasure) isResultOf (Giving))*
    The ownership of money as a result of a transfer
  - Concepts -> words, semantic relations -> syntactic ones
- Converting a graph to a tree with a given head:
  - Use inverse relations (hasResult – isResultOf) to direct all arcs from the head to the leaves.
  - Eliminate loops by splitting a node and marking the split with explicit variable
  - *(Human ?x isAgentOf (Shaving hasObject ?x))*
    A person who shaved (himself)

Conclusions

- Presented a practical solution for the content selection task in REG for:
  - Question answering
  - Graph visualization
- Improvements to an existing graph-based algorithm suggested and implemented
- Preliminary evaluation provided
- A sketch on linguistic realization outlined

References

- Dale, R. (1989) Cooking up referring expressions. Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics (ACL), p. 68–75.
- Dale, R. (1992) Generating Referring Expressions: Constructing Descriptions in a Domain of Objects and Processes. The MIT Press, Cambridge, MA.
- Reiter, E., Dale R. (1992) A fast algorithm for the generation of referring expressions. Proceedings of the 14th International Conference on Computational Linguistics (COLING), p. 232–238, Nantes.
- Krahmer, E., van Erk, S., Verleg, A. (2003). Graph-Based Generation of Referring Expressions. Computational Linguistics, 29(1), p. 53–72.

Thank you!