Parsing gaps with rules and parsers.

Alexey Sorokin[1,2]

[2]Moscow Institute of Science and Technology, [1]Moscow State University

"Dialogue", International Conference
on Computational Linguistics,
Moscow, May, 30th, 2019

Gapping: problem setting

- Gapping refers to restoration of “subsumed” verbs and their arguments:

  Бакр стал президентом Ирака, а Саддам Хуссейн – вице-президентом.

  Bacr became the president of Iraque and Saddam Hussein – the vice-president.

- Gapping includes:
  - Binary gap detection and location:

Gapping: problem setting

- Gapping refers to restoration of “subsumed” verbs and their arguments:

  Бакр стал президентом Ирака, а Саддам Хуссейн – вице-президентом.

  Bacr became the president of Iraque and Saddam Hussein – the vice-president.

- Gapping includes:
  - Binary gap detection and location:

    Бакр стал президентом Ирака, а Саддам Хуссейн – вице-президентом.

    Bacr became the president of Iraque and Saddam Hussein – the vice-president.

Gapping: problem setting

- Gapping refers to restoration of “subsumed” verbs and their arguments:

  Бакр стал президентом Ирака, а Саддам Хуссейн – вице-президентом.

  Bacr became the president of Iraque and Saddam Hussein – the vice-president.

- Gapping includes:
  - Binary gap detection and location:
  - Gap argument location:

  Бакр стал президентом Ирака, а Саддам Хуссейн – вице-президентом.

  Bacr became the president of Iraque and Saddam Hussein – the vice-president.

Gapping: problem setting

- Gapping refers to restoration of “subsumed” verbs and their arguments:

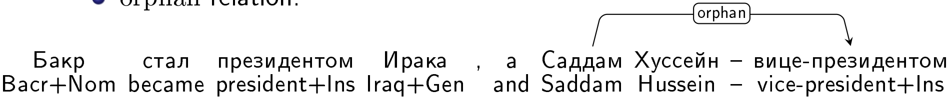
Бакр стал президентом Ирака, а Саддам Хуссейн – вице-президентом.
Bacr became the president of Iraque and Saddam Hussein – the vice-president.

- Gapping includes:
  - Binary gap detection and location:
  - Gap argument location:
  - Remnants matching:

Бакр стал президентом Ирака, а Саддам Хуссейн – вице-президентом.
Bacr became the president of Iraque and Saddam Hussein – the vice-president.

Rule-based approach

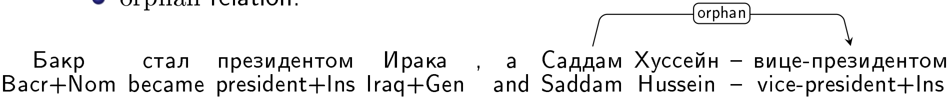
- orphan relation:



Бакр	стал	президентом	Ирака	,	а	Саддам	Хуссейн	–	вице-президентом
Bacr+Nom	became	president+Ins	Iraq+Gen		and	Saddam	Hussein	–	vice-president+Ins

Rule-based approach

- orphan relation:



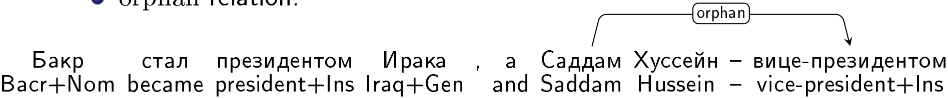
Бакр	стал	президентом	Ирака	,	а	Саддам	Хуссейн	–	вице-президентом
Bacr+Nom	became	president+Ins	Iraq+Gen		and	Saddam	Hussein	–	vice-president+Ins

- , *a* ... – construction:

(, *а Саддам Хуссейн – вице-президентом*)

Rule-based approach

- orphan relation:

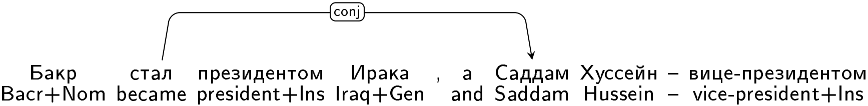


Бакр	стал	президентом	Ирака	,	а	Саддам	Хуссейн	–	вице-президентом
Bacr+Nom	became	president+Ins	Iraq+Gen		and	Saddam	Hussein	–	vice-president+Ins

- , *а* … – construction:

(, *а Саддам Хуссейн – вице-президентом*)

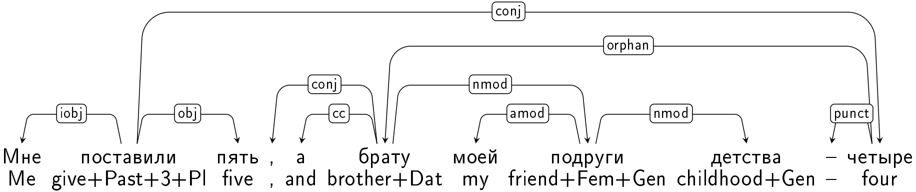
- conj relation between clauses:



Бакр	стал	президентом	Ирака	,	а	Саддам	Хуссейн	–	вице-президентом
Bacr+Nom	became	president+Ins	Iraq+Gen		and	Saddam	Hussein	–	vice-president+Ins

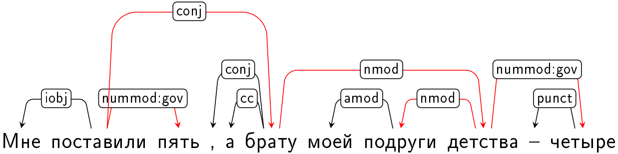
Rule-based approach: failure

- expected parse:



Мне	поставили	пять	,	а	брату	моей	подруги	детства	–	четыре
Me	give+Past+3+Pl	five	,	and	brother+Dat	my	friend+Fem+Gen	childhood+Gen	–	four

- automatically obtained parse (UDPipe 2.3):



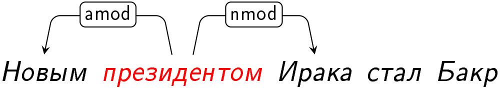
Мне поставили пять , а брату моей подруги детства – четыре

From spans to heads

- We decide to train a neural model on annotated gap sentences.
- But arguments and remnants are consituents, not single words.
- We predict span heads instead of spans.

From spans to heads

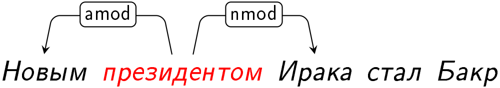
- We decide to train a neural model on annotated gap sentences.
- But arguments and remnants are consituents, not single words.
- We predict span heads instead of spans.
- During training span head is calculated using syntactic trees:



Новым президентом Ирака стал Бакр

From spans to heads

- We decide to train a neural model on annotated gap sentences.
- But arguments and remnants are consituents, not single words.
- We predict span heads instead of spans.
- During training span head is calculated using syntactic trees:



Новым президентом Ирака стал Бакр

- The spans are restored as descendants of detected heads.
- The quality of last two stages depends on the external parser.

Neural model: notation

- Mostly we work with tuples

$$(L, V, R; l, g, r)$$

L V R l g r

Бакр стал президентом , а Саддам Хуссейн – вице-президентом.

Neural model: notation

- Mostly we work with tuples

$$(L, V, R; l, g, r)$$

L	V	R			l	g	r
Бакр	стал	президентом	,	а	Саддам Хуссейн	–	вице-президентом.

- There can be multiple gap groups:

L	V	R	l_1	g_1, r_1		l_2	g_2	r_2
Я	пью	чай,	Петя	кофе,	а	Маша	–	молоко

Neural model: pipeline

- For each verb v find the positions of its gapped occurrences g_i.
- If no gaps were found, return that sentence does not contain gaps.

Neural model: pipeline

- For each verb v find the positions of its gapped occurrences g_i.
- If no gaps were found, return that sentence does not contain gaps.
- For each found gap g and its main verb v find its left (l) and right (r) arguments.

Neural model: pipeline

- For each verb v find the positions of its gapped occurrences g_i.
- If no gaps were found, return that sentence does not contain gaps.
- For each found gap g and its main verb v find its left (l) and right (r) arguments.
- For each gap argument (l, r) detect the corresponding remnant (L, R) of the main verb.

Neural model: pipeline

- For each verb v find the positions of its gapped occurrences g_i.
- If no gaps were found, return that sentence does not contain gaps.
- For each found gap g and its main verb v find its left (l) and right (r) arguments.
- For each gap argument (l, r) detect the corresponding remnant (L, R) of the main verb.
- Restore the spans for detected constituent heads.

Neural model: ideas

- Each stage of the model recovers the unknown position of word that better matches the known word:
    - The gap position m matches the main verb v.

Neural model: ideas

- Each stage of the model recovers the unknown position of word that better matches the known word:
    - The gap position m matches the main verb v.
    - The left and right gap arguments l, r match the main gap g.

Neural model: ideas

- Each stage of the model recovers the unknown position of word that better matches the known word:
  - The gap position m matches the main verb v.
  - The left and right gap arguments l, r match the main gap g.
  - The left and right verb arguments L, R match the verb arguments l, r and the verb v itself.

Neural model: ideas

- Each stage of the model recovers the unknown position of word that better matches the known word:
    - The gap position m matches the main verb v.
    - The left and right gap arguments l, r match the main gap g.
    - The left and right verb arguments L, R match the verb arguments l, r and the verb v itself.
- For each stage we prepare embeddings of two types $g_1, \ldots, g_n$ and $h_1, \ldots, h_n$ and train target h-embeddings to be similar to source g-embeddings.
- Matching can be detected via attention (dot-product) or binary classification.

Neural model: common

- There are three separate models (gap detection, argument location, remnant matching) of similar structure but with different weights. The heads of the models also differ.
- All models take as source ELMo embeddings $e_1, \dots, e_n$.
- Embeddings are processed via two BiLSTMs to obtain two sequences of context embeddings $g_1, \dots, g_n,\ h_1, \dots, h_n$.

Neural model: common

- There are three separate models (gap detection, argument location, remnant matching) of similar structure but with different weights. The heads of the models also differ.
- All models take as source ELMo embeddings $e_1, \ldots, e_n$.
- Embeddings are processed via two BiLSTMs to obtain two sequences of context embeddings $g_1, \ldots, g_n,\ h_1, \ldots, h_n$.
- Gap detection head:

$$\begin{array}{rcl} S_{ij} & = & W_g[g_i, h_j, g_i - h_j, g_i \odot h_j], \\ p_{ij} & = & \sigma(S_{ij}), \\ p_{ij} \geqslant \frac{1}{2} & - & w_j \text{ is the gap for main verb } w_i. \end{array}$$

- Sigmoid is used to detect multiple gaps.

Neural model: argument location

- Argument location:

$$\begin{aligned} S_{ijk} &= W_{arg}[g_i, g_j, h_k], \\ p_{ijk} &= \mathrm{softmax}_k S_{ijk}. \end{aligned}$$

- $p_{ijk} > \frac{1}{2}$ – w_k is an argument for gap w_j and its main verb w_i.

Neural model: argument location

- Argument location:

$$\begin{aligned} S_{ijk} &= W_{arg}[g_i, g_j, h_k], \\ p_{ijk} &= \mathrm{softmax}_k S_{ijk}. \end{aligned}$$

- $p_{ijk} > \frac{1}{2}$ – w_k is an argument for gap w_j and its main verb w_i.
- Models for left and right argument are separate.
- We also check $i < k < j$ for left argument and $j \geqslant k$ for right.

Neural model: argument location

- Argument location:

$$\begin{aligned} S_{ijk} &= W_{arg}[g_i, g_j, h_k], \\ p_{ijk} &= \mathrm{softmax}_k S_{ijk}. \end{aligned}$$

- $p_{ijk} > \frac{1}{2}$ – w_k is an argument for gap w_j and its main verb w_i.
- Models for left and right argument are separate.
- We also check $i < k < j$ for left argument and $j \geqslant k$ for right.
- We use softmax since there should be only one matching arguments.
- If there are no such p_{ijk}, the model decides that gap was erroneous.

Neural model: remnants matching

- Remnants matching:

$$\begin{aligned} S_{ijk} &= W_{arg}[g_i, g_j, h_k, g_j - h_k, g_j \odot h_k], \\ p_{ijk} &= \mathrm{softmax}_k S_{ijk}. \end{aligned}$$

- $p_{ijk} > \frac{1}{2}$ – w_k is the argument of main verb w_i, that matches the gap argument w_j.
- We also check $k < i$ for left argument and $i < k < j$ for the right.

Neural model: training

- Models are trained separately and work as pipeline.
- Therefore recall is more important than precision.

Neural model: training

- Models are trained separately and work as pipeline.
- Therefore recall is more important than precision.
- Training objective: cross-entropy (standard).
- Additional modifications:
    - Penalize false negatives more severely in crossentropy.
    - Set selection threshold θ to 0.25 instead of 0.5.
    - Penalize the arguments that do now show the highest probability.

$$L_{aux} = -\alpha(\log s_{ij_g} - \log \max_j s_{ij})$$

Results

Model	Binary			Full F1
	P	R	F1	
Rule-based (gaps only)	80.6	82.5	81.5	-
Rule-based (full)	93.4	64.6	76.3	60.2
Neural (single, gaps)	96.2	92.5	94.3	-
Neural (ensemble, gaps)	97.0	92.9	95.0	-
Neural (single, full)	97.3	89.1	93.0	85.3
Neural (ensemble, full)	97.9	90.1	93.9	87.1
Neural (small, gaps)	98.2	81.2	88.9	-
Neural (small, full)	98.6	71.5	82.9	69.4
Best			95.9	89.2

Results: individual stages

Stage	Rule			Neural		
	P	R	F1	P	R	F1
Gap detection	71.7	75.5	73.6	94.2	93.1	93.7
Argument location	94.8	88.1	91.3	96.2	95.6	95.9
Remnant matching	95.0	68.1	79.4	95.2	81.9	88.0
Span prediction	91.9	77.5	84.1	91.9	77.5	84.1

Instance-wise scores for individual pipeline stages.

Основные ошибки

- For full resolution: parsing errors.
- Difficult to detect:
    - Single remnant gapping.

Основные ошибки

- For full resolution: parsing errors.
- Difficult to detect:
  - Single remnant gapping.
  - Gapping in single clause of a complex sentence (especially in subordinate):

    Я смотрел на него, он на меня, и мы …
  - Main verb in main clause, gap in subordinate clause:

    Я буду любить кого-нибудь так, как она меня
  - Homonymy:

    Они называли тебя другом, а её подругой.

Future work and conclusions

- Main conclusion: neural models beat rule-based.

Future work and conclusions

- Main conclusion: neural models beat rule-based.
- Future directions:
    - Other types of ellipsis:

Я люблю чай и Вася тоже.
Я не люблю кофе, а Маша обожает.

Future work and conclusions

- Main conclusion: neural models beat rule-based.
- Future directions:
    - Other types of ellipsis:

      Я люблю чай и Вася тоже.
      Я не люблю кофе, а Маша обожает.
    - Other embeddings (BERT, ...),
    - Morphological and syntactic features, require better parsers.

Future work and conclusions

- Main conclusion: neural models beat rule-based.
- Future directions:
  - Other types of ellipsis:

Я люблю чай и Вася тоже.

Я не люблю кофе, а Маша обожает.

  - Other embeddings (BERT, ...),
  - Morphological and syntactic features, require better parsers.
  - Test for other languages and smaller datasets.

Future work and conclusions

- Main conclusion: neural models beat rule-based.
- Future directions:
    - Other types of ellipsis:

      Я люблю чай и Вася тоже.

      Я не люблю кофе, а Маша обожает.
    - Other embeddings (BERT, ...),
    - Morphological and syntactic features, require better parsers.
    - Test for other languages and smaller datasets.
- Downstream application: semantic parsing, information extraction.

Спасибо за внимание!

Thank you for your attention!