

# Алгоритмы автоматического словоизменения.

Алексей Сорокин<sup>1,2</sup>

<sup>1</sup>МГУ им. М. В. Ломоносова, механико-математический факультет

<sup>2</sup>МФТИ, факультет инноваций и высоких технологий

Яндекс, 30 августа 2017г.

# Введение

- Основные задачи автоматического словоизменения:
  - Автоматическое словопорождение (синтез).
  - Автоматическая лемматизация (анализ).
  - Автоматическое пополнение парадигм.

# Введение

- Основные задачи автоматического словоизменения:
  - Автоматическое словопорождение (синтез).
  - Автоматическая лемматизация (анализ).
  - Автоматическое пополнение парадигм.
- Применения:
  - Автоматическая обработка текста: лемматизация, морфологический анализ.
  - Машинный перевод: анализ и синтез.
  - Пополнение лексических ресурсов, моделирование обучения языку.

## Соревнования по автоматическому словоизменению

SIGMORPHON-2016:

- Порождение (inflection):

*корова*, pos=N, case=INS, num=SG  $\mapsto$  ?

2) и 3) отражают реальную языковую практику (известна одна словоформа слова, нужно породить остальные).

# Соревнования по автоматическому словоизменению

SIGMORPHON-2016:

- Порождение (inflection):

*корова*, pos=N, case=INS, num=SG  $\mapsto$  ?

- Репорождение (reinflection):

pos=N, case=ESS, num=PL, *коровах*  $\mapsto$   
?, pos=N, case=INS, num=SG

2) и 3) отражают реальную языковую практику (известна одна словоформа слова, нужно породить остальные).

## Соревнования по автоматическому словоизменению

## SIGMORPHON-2016:

- Порождение (inflection):

*корова*, pos=N, case=INS, num=SG  $\mapsto$  ?

- Репорождение (reinflection):

pos=N, case=ESS, num=PL, *коровах*  $\mapsto$   
 ?, pos=N, case=INS, num=SG

- Обобщённое репорождение (unlabeled reinflection):

*коровах*  $\mapsto$  ?, pos=N, case=INS, num=SG

2) и 3) отражают реальную языковую практику (известна одна словоформа слова, нужно породить остальные).

## Соревнования по автоматическому словоизменению

SIGMORPHON-2017:

- Порождение

*корова*, pos=N, case=INS, num=SG  $\mapsto$  ?

## Соревнования по автоматическому словоизменению

## SIGMORPHON-2017:

- Порождение

*корова*, pos=N, case=INS, num=SG  $\mapsto$  ?

- Пополнение парадигм.

корова	?
?	?
?	коровах
корову	?
?	?
корове	?

?	пишу	?	?
пиши	?	?	писала
?	пишет	?	?
			?



## SIGMORPHON-2016: методы

- Автоматическое словопорождение — типичная задача преобразования последовательностей (sequence-to-sequence learning).
- Можно решать отдельно для каждого набора грамем.

# SIGMORPHON-2016: методы

- Автоматическое словопорождение — типичная задача преобразования последовательностей (sequence-to-sequence learning).
- Можно решать отдельно для каждого набора грамем.
- Репорождение сводится к двум задачам порождения: обратной (лемматизация) и прямой.

# SIGMORPHON-2016: методы

- Автоматическое словопорождение — типичная задача преобразования последовательностей (sequence-to-sequence learning).
- Можно решать отдельно для каждого набора грамем.
- Репорождение сводится к двум задачам порождения: обратной (лемматизация) и прямой.
- Возможные алгоритмы решения:
  - Свести задачу порождения к задаче классификации (закодировать словоизменяемые шаблоны).

## SIGMORPHON-2016: методы

- Автоматическое словопорождение — типичная задача преобразования последовательностей (sequence-to-sequence learning).
- Можно решать отдельно для каждого набора грамем.
- Репорождение сводится к двум задачам порождения: обратной (лемматизация) и прямой.
- Возможные алгоритмы решения:
  - Свести задачу порождения к задаче классификации (закодировать словоизменяемые шаблоны).
  - Преобразовывать последовательности, угадывая выравнивание (условные случайные поля).

# SIGMORPHON-2016: методы

- Автоматическое словопорождение — типичная задача преобразования последовательностей (sequence-to-sequence learning).
- Можно решать отдельно для каждого набора грамем.
- Репорождение сводится к двум задачам порождения: обратной (лемматизация) и прямой.
- Возможные алгоритмы решения:
  - Свести задачу порождения к задаче классификации (закодировать словоизменяемые шаблоны).
  - Преобразовывать последовательности, угадывая выравнивание (условные случайные поля).
  - Преобразовывать последовательности (нейронные сети, encoder-decoder).

## SIGMORPHON-2016: результаты

## Задание 1.

Язык	Kahn, Schütze (enc-dec)	RNN	CRF	Sorokin (класс.)	Aharoni (2017)
Арабский	95.47	93.34	72.42	83.52	97.0
Финский	96.80	95.14	88.65	87.86	97.0
Грузинский	98.50	97.55	94.9	94.8	98.9
Немецкий	95.8	95.11	92.64	94.22	96.6
Венгерский	99.3	98.38	91.05	89.82	99.3
Навахо	91.48	88.43	56.33	57.42	95.4
Русский	91.46	90.62	89.13	89.67	92.2
Испанский	98.84	98.41	98.35	98.76	98.9
Турецкий	98.93	97.93	90.84	92.03	98.0

## SIGMORPHON-2016: результаты

## Задание 3.

Язык	Kahn, Schütze (enc-dec)	другие RNN	CRF	Sorokin (класс.)
Арабский	96.52	70.43	59.4	80.57
Финский	96.56	87.55	85.77	84.34
Грузинский	98.87	92.08	94.13	88.05
Немецкий	95.6	89.58	91.15	91.99
Венгерский	99.5	96.46	89.12	90.41
Навахо	96.2	83.21	84.38	52.12
Русский	89.91	84.59	85.48	86.59
Испанский	97.96	94.85	96.27	97.30
Турецкий	99.31	91.25	84.38	92.44

## Классификационный подход (Sorokin, 2017)

- Для каждой проблемы классификационная задача решается отдельно.



## Классификационный подход (Sorokin, 2017)

- Для каждой проблемы классификационная задача решается отдельно.
- Из каждой пары  $\langle \text{лемма} \rangle$ - $\langle \text{словоформа} \rangle$  извлекается шаблон (абстрактная парадигма):

$$\text{volver-vuelvo} \mapsto 1+o+2\#1+ue+2+o,$$

$$1=v, 2=lv$$

## Классификационный подход (Sorokin, 2017)

- Для каждой проблемы классификационная задача решается отдельно.
- Из каждой пары ⟨лемма⟩-⟨словоформа⟩ извлекается шаблон (абстрактная парадигма):

$$\text{volver-vuelvo} \mapsto 1+o+2\#1+ue+2+o, \\ 1=v, 2=lv$$

- Для извлечения шаблона используется наибольшая общая подпоследовательность, элементы которой объявляются переменными.

## Метод абстрактных парадигм

- Для извлечения шаблона используется наибольшая общая подпоследовательность, элементы которой объявляются переменными.
- В наибольшей общей подпоследовательности ограничивается максимальная длина начального разрыва и разрыва в середине в зависимости от языка.

## Метод абстрактных парадигм

- Для извлечения шаблона используется наибольшая общая подпоследовательность, элементы которой объявляются переменными.
- В наибольшей общей подпоследовательности ограничивается максимальная длина начального разрыва и разрыва в середине в зависимости от языка.
- Параметры разрыва:

Язык	Разрыв в начале	Разрыв в середине
Испанский	2	2
Турецкий	0	1
Арабский	5	1

## Метод абстрактных парадигм

- Для извлечения шаблона используется наибольшая общая подпоследовательность, элементы которой объявляются переменными.
- В наибольшей общей подпоследовательности ограничивается максимальная длина начального разрыва и разрыва в середине в зависимости от языка.
- Параметры разрыва:

Язык	Разрыв в начале	Разрыв в середине
Испанский	2	2
Турецкий	0	1
Арабский	5	1

- Задача предсказания словоформы сводится к “угадыванию” метки шаблона.
- После этого ещё надо “расставить” элементы шаблона внутри слова, но чаще всего это задача однозначная.

## Порождение словоформ с помощью абстрактных парадигм

- Признаки: суффиксы слова (максимальная длина 5-6), префиксы слова (максимальная длина зависит от языка).
- Отделяются только 10% наиболее полезных признаков.

## Порождение словоформ с помощью абстрактных парадигм

- Признаки: суффиксы слова (максимальная длина 5-6), префиксы слова (максимальная длина зависит от языка).
- Отделяются только 10% наиболее полезных признаков.
- Классификатор: линейная регрессия.
- Обучение: извлечь из пар лемма-словоформа признаки леммы и шаблон словоизменения, обучить классификатор на этих данных.

## Порождение словоформ с помощью абстрактных парадигм

- Признаки: суффиксы слова (максимальная длина 5-6), префиксы слова (максимальная длина зависит от языка).
- Отделяются только 10% наиболее полезных признаков.
- Классификатор: линейная регрессия.
- Обучение: извлечь из пар лемма-словоформа признаки леммы и шаблон словоизменения, обучить классификатор на этих данных.
- Предсказание (даны только леммы):
  - Извлечь из лемм признаки, предсказать по ним метки шаблонов.



## Порождение словоформ с помощью абстрактных парадигм

- Признаки: суффиксы слова (максимальная длина 5-6), префиксы слова (максимальная длина зависит от языка).
- Отделяются только 10% наиболее полезных признаков.
- Классификатор: линейная регрессия.
- Обучение: извлечь из пар лемма-словоформа признаки леммы и шаблон словоизменения, обучить классификатор на этих данных.
- Предсказание (даны только леммы):
  - Извлечь из лемм признаки, предсказать по ним метки шаблонов.
  - В порядке убывания вероятности шаблонов проверить, применим ли он к лемме, если да, извлечь переменные.
  - Если несколько вариантов извлечения, выбрать наиболее вероятный по локальным контекстам.

# Порождение словоформ с помощью абстрактных парадигм

- Признаки: суффиксы слова (максимальная длина 5-6), префиксы слова (максимальная длина зависит от языка).
- Отделяются только 10% наиболее полезных признаков.
- Классификатор: линейная регрессия.
- Обучение: извлечь из пар лемма-словоформа признаки леммы и шаблон словоизменения, обучить классификатор на этих данных.
- Предсказание (даны только леммы):
  - Извлечь из лемм признаки, предсказать по ним метки шаблонов.
  - В порядке убывания вероятности шаблонов проверить, применим ли он к лемме, если да, извлечь переменные.
  - Если несколько вариантов извлечения, выбрать наиболее вероятный по локальным контекстам.
  - Подставить переменные в шаблон, вернуть значение словоформы.

## Порождение словоформ с помощью абстрактных парадигм: энграммные вероятности

- При классификации учитываются только символы на концах слов.
- Результат применения шаблона может нарушать фонологические законы (гармония гласных и т. д.).

## Порождение словоформ с помощью абстрактных парадигм: энграммные вероятности

- При классификации учитываются только символы на концах слов.
- Результат применения шаблона может нарушать фонологические законы (гармония гласных и т. д.).
- Если нужная информация содержится в корне слова, алгоритм не может её извлечь.

## Порождение словоформ с помощью абстрактных парадигм: энграммные вероятности

- При классификации учитываются только символы на концах слов.
- Результат применения шаблона может нарушать фонологические законы (гармония гласных и т. д.).
- Если нужная информация содержится в корне слова, алгоритм не может её извлечь.
- Решение: дополнительно применить энграммную модель.

## Порождение словоформ с помощью абстрактных парадигм: энграммные вероятности

- При классификации учитываются только символы на концах слов.
- Результат применения шаблона может нарушать фонологические законы (гармония гласных и т. д.).
- Если нужная информация содержится в корне слова, алгоритм не может её извлечь.
- Решение: дополнительно применить энграммную модель.
- “Поверх” базовой модели дополнительный классификатор с двумя признаками:
  - Логарифм вероятности базового классификатора.
  - Логарифм вероятности словоформы по энграммной модели.

## Порождение словоформ с помощью абстрактных парадигм: энграммные вероятности

- При классификации учитываются только символы на концах слов.
- Результат применения шаблона может нарушать фонологические законы (гармония гласных и т. д.).
- Если нужная информация содержится в корне слова, алгоритм не может её извлечь.
- Решение: дополнительно применить энграммную модель.
- “Поверх” базовой модели дополнительный классификатор с двумя признаками:
  - Логарифм вероятности базового классификатора.
  - Логарифм вероятности словоформы по энграммной модели.
- Этот классификатор переранжирует гипотезы.

# Порождение словоформ с помощью абстрактных парадигм: энграммные вероятности

- Алгоритм обучения классификатора:
  - Для всех пар  $(l, w)$  в обучающей выборке извлечь пару  $p_{base}(l, w), p_{ngram}(w)$ .
  - Аналогично поступить для других возможных ответов  $w_1, \dots, w_n$  базового классификатора.



# Порождение словоформ с помощью абстрактных парадигм: энграммные вероятности

- Алгоритм обучения классификатора:
  - Для всех пар  $(l, w)$  в обучающей выборке извлечь пару  $p_{base}(l, w), p_{ngram}(w)$ .
  - Аналогично поступить для других возможных ответов  $w_1, \dots, w_n$  базового классификатора.
  - Поместить  $[p_{base}(l, w) - p_{base}(l, w_i), p_{ngram}(l, w) - p_{ngram}(l, w_i)]$  в положительный класс, противоположные вектора в отрицательный.

## Порождение словоформ с помощью абстрактных парадигм: энграммные вероятности

- Алгоритм обучения классификатора:
  - Для всех пар  $(l, w)$  в обучающей выборке извлечь пару  $p_{base}(l, w), p_{ngram}(w)$ .
  - Аналогично поступить для других возможных ответов  $w_1, \dots, w_n$  базового классификатора.
  - Поместить  $[p_{base}(l, w) - p_{base}(l, w_i), p_{ngram}(l, w) - p_{ngram}(l, w_i)]$  в положительный класс, противоположные вектора в отрицательный.
- При предсказании выбирать гипотезу с наивысшей стоимостью (score) в соответствии с моделью переранжирования.

# Переранжирование: эффективность

Эффективность переранжирования на development set.

Language	Verbs			Adjectives			Nouns		
	Recall	Accur.	Prec.	Recall	Accur.	Prec.	Recall	Accur.	Prec.
Arabic	88.2	80.4	91.16%	95.9	93.6	97.60%	83.9	76.2	90.82%
	85.5	76.1	89.01%	96.1	94.5	98.34%	84.4	83.1	98.46%
Finnish	95.1	94.1	98.95%	63.9	62.9	98.44%	96.0	87.8	91.46%
	96.1	90.6	94.28%	65.7	62.9	95.74%	98.0	92.7	94.59%
Georgian	59.8	42.3	70.74%	100.0	99.2	99.2%	98.5	97.6	98.09%
	62.9	51.3	81.56%	100.0	100.0	100.0%	99.8	99.4	99.60%
German	93.3	90.0	96.46%	98.1	97.2	99.08%	94.8	91.2	96.20%
	94.3	94.3	100.0%	98.4	98.1	99.70%	96.6	94.1	97.41%
Hungarian	97.5	92.5	94.87%				82.1	75.9	92.45%
	99.1	98.7	99.60%				99.1	99.1	100.0%
Navajo	61.8	56.4	91.26%				97.8	94.5	96.63%
	67.9	64.8	95.43%				95.6	63.3	66.21%
Russian	91.4	83.2	91.03%	98.5	95.8	97.26%	98.0	91.9	93.78%
	88.2	86.6	98.19%	97.7	95.2	97.44%	96.8	94.1	97.21%
Spanish	98.8	98.6	99.80%	100.0	100.0	100%	100.0	100.0	100%
	98.6	98.5	99.90%	100.0	100.0	100%	99.5	97.2	97.69%
Turkish	87.7	83.5	95.21%				89.5	87.3	97.54%
	93.8	93.8	100.0%				96.8	96.4	99.59%

## Attention-RNN подход

- Наилучшие результаты в SIGMORPHON-2016 показал Attention-RNN-подход (Bahdanau et al., 2015).

## Attention-RNN подход

- Наилучшие результаты в SIGMORPHON-2016 показал Attention-RNN-подход (Bahdanau et al., 2015).
- Сеть состоит из нескольких слоёв:
  - $[x_1, \dots, x_T]$  — входное представление.

## Attention-RNN подход

- Наилучшие результаты в SIGMORPHON-2016 показал Attention-RNN-подход (Bahdanau et al., 2015).
- Сеть состоит из нескольких слоёв:
  - $[x_1, \dots, x_T]$  — входное представление.
  - $[h_1, \dots, h_t]$  — состояния LSTM-энкодера,  
 $h_i = [\overleftarrow{h}_i, \overrightarrow{h}_i]$  — двусторонняя сеть.

## Attention-RNN подход

- Наилучшие результаты в SIGMORPHON-2016 показал Attention-RNN-подход (Bahdanau et al., 2015).
- Сеть состоит из нескольких слоёв:
  - $[x_1, \dots, x_T]$  — входное представление.
  - $[h_1, \dots, h_t]$  — состояния LSTM-энкодера,  
 $h_i = [\overleftarrow{h}_i, \overrightarrow{h}_i]$  — двусторонняя сеть.
  - Декодированный Attention-слой:

$$\begin{aligned}
 y_t &= g(y_{t-1}, s_t, c_t), \\
 c_t &= \sum_j a_{tj} h_t, \\
 a_{tj} &= \frac{e_{tj}}{\sum_k e_{tk}}, \\
 e_{tj} &= a(s_{t-1}, h_t), \\
 a &— \text{feedforward-сеть,}
 \end{aligned}$$

## Attention-RNN подход

- Наилучшие результаты в SIGMORPHON-2016 показал Attention-RNN-подход (Bahdanau et al., 2015).
- Сеть состоит из нескольких слоёв:
  - $[x_1, \dots, x_T]$  — входное представление.
  - $[h_1, \dots, h_t]$  — состояния LSTM-энкодера,  
 $h_i = [\overleftarrow{h}_i, \overrightarrow{h}_i]$  — двусторонняя сеть.
  - Декодированный Attention-слой:

$$\begin{aligned}
 y_t &= g(y_{t-1}, s_t, c_t), \\
 c_t &= \sum_j a_{tj} h_t, \\
 a_{tj} &= \frac{e_{tj}}{\sum_k e_{tk}}, \\
 e_{tj} &= a(s_{t-1}, h_t), \\
 a &— \text{feedforward-сеть,}
 \end{aligned}$$

- Дополнительный выходной maxout-слой.



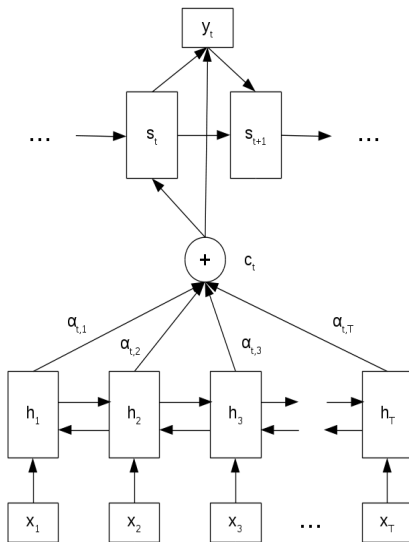
## Attention-RNN подход

- Наилучшие результаты в SIGMORPHON-2016 показал Attention-RNN-подход (Bahdanau et al., 2015).
- Сеть состоит из нескольких слоёв:
  - $[x_1, \dots, x_T]$  — входное представление.
  - $[h_1, \dots, h_t]$  — состояния LSTM-энкодера,  
 $h_i = [\overleftarrow{h}_i, \overrightarrow{h}_i]$  — двусторонняя сеть.
  - Декодированный Attention-слой:

$$\begin{aligned}
 y_t &= g(y_{t-1}, s_t, c_t), \\
 c_t &= \sum_j a_{tj} h_t, \\
 a_{tj} &= \frac{e_{tj}}{\sum_k e_{tk}}, \\
 e_{tj} &= a(s_{t-1}, h_t), \\
 a &= \text{feedforward-сеть},
 \end{aligned}$$

- Дополнительный выходной maxout-слой.
- Входные и выходные граммы хранятся в стартовом элементе последовательности  $x_1$ .

## Attention-RNN подход

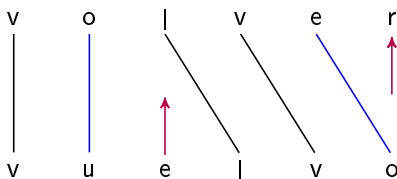


## Monotonic-attention-RNN подход

- Attention-веса  $a_{tj}$  кодируют выравнивание (вероятность получения перевода  $\mathbf{v}_t$  из исходного слова  $\mathbf{u}_j$ ).

# Monotonic-attention-RNN подход

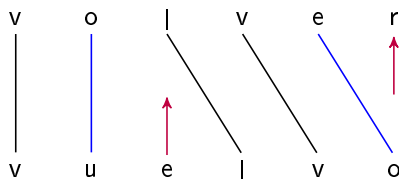
- Attention-веса  $a_{tj}$  кодируют выравнивание (вероятность получения перевода  $\mathbf{v}_t$  из исходного слова  $\mathbf{u}_j$ ).
- Естественное выравнивание есть и при словоизменении:



- Но оно обязательно монотонно и содержит только  $1-0$ ,  $0-1$ ,  $1-1$  фрагменты.

## Monotonic-attention-RNN подход

- Attention-веса  $a_{tj}$  кодируют выравнивание (вероятность получения перевода  $\mathbf{v}_t$  из исходного слова  $\mathbf{u}_j$ ).
- Естественное выравнивание есть и при словоизменении:



- Но оно обязательно монотонно и содержит только  $1-0$ ,  $0-1$ ,  $1-1$  фрагменты.
- Нельзя ли ограничиться только такими выравниваниями (Aharoni, Goldberg, 2017)?

## Monotonic-attention-RNN подход

- Также используется двухслойная нейронная сеть.

## Monotonic-attention-RNN подход

- Также используется двухслойная нейронная сеть.
- Выходной алфавит — элементарные операции выравнивания (`writeS` для выписывания символа  $S$  и `step` для сдвига).
- Выравнивания строятся для всех пар ⟨лемма⟩-⟨словоформа⟩.

## Monotonic-attention-RNN подход

- Также используется двухслойная нейронная сеть.
- Выходной алфавит — элементарные операции выравнивания ( $\text{write}_S$  для выписывания символа  $S$  и  $\text{step}$  для сдвига).
- Выравнивания строятся для всех пар  $\langle \text{лемма} \rangle$ - $\langle \text{словоформа} \rangle$ .
- Алгоритм предсказывает не выходное слово, а выравнивание.



## Monotonic-attention-RNN подход

- Также используется двухслойная нейронная сеть.
- Выходной алфавит — элементарные операции выравнивания ( $\text{write}_S$  для выписывания символа  $S$  и  $\text{step}$  для сдвига).
- Выравнивания строятся для всех пар  $\langle \text{лемма} \rangle$ - $\langle \text{словоформа} \rangle$ .
- Алгоритм предсказывает не выходное слово, а выравнивание.
- Энкодер — двусторонняя LSTM (как в Kahn, Schütze).

## Monotonic-attention-RNN подход

- Также используется двухслойная нейронная сеть.
- Выходной алфавит — элементарные операции выравнивания ( $\text{write}_S$  для выписывания символа  $S$  и  $\text{step}$  для сдвига).
- Выравнивания строятся для всех пар ⟨лемма⟩-⟨словоформа⟩.
- Алгоритм предсказывает не выходное слово, а выравнивание.
- Энкодер — двусторонняя LSTM (как в Kahn, Schütze).
- Декодер — рекуррентная сеть:

$$y_i = \text{softmax}(W \cdot \text{LSTM}(z_1, \dots, x_i) + b),$$

$$z_i = [h_j, \mathbf{f}, s_{i-1}]$$

$\mathbf{f}$  — представление грамем,

$h_j$  — энкодинг текущего символа,

$s_{i-1}$  — представление  $y_{i-1}$ .

## Monotonic-attention-RNN подход

- Также используется двухслойная нейронная сеть.
- Выходной алфавит — элементарные операции выравнивания ( $\text{write}_S$  для выписывания символа  $S$  и  $\text{step}$  для сдвига).
- Выравнивания строятся для всех пар ⟨лемма⟩-⟨словоформа⟩.
- Алгоритм предсказывает не выходное слово, а выравнивание.
- Энкодер — двусторонняя LSTM (как в Kahn, Schütze).
- Декодер — рекуррентная сеть:

$$y_i = \text{softmax}(W \cdot \text{LSTM}(z_1, \dots, x_i) + b),$$

$$z_i = [h_j, \mathbf{f}, s_{i-1}]$$

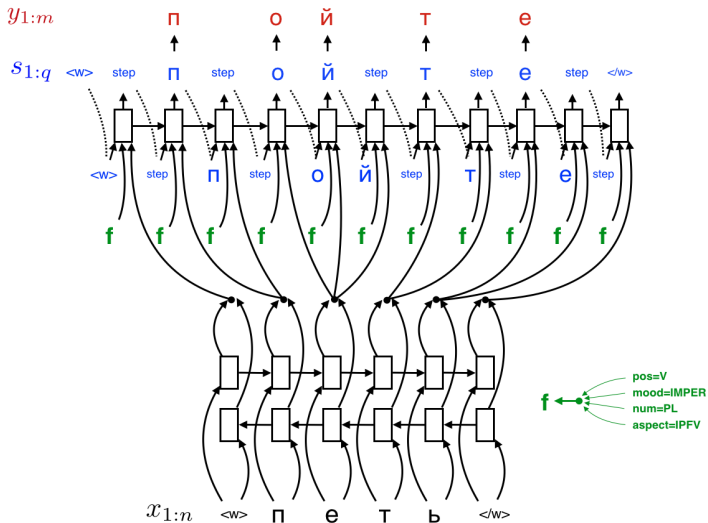
$\mathbf{f}$  — представление грамем,

$h_j$  — энкодинг текущего символа,

$s_{i-1}$  — представление  $y_{i-1}$ .

- Когда модель выдаёт  $\text{step}$ , выполняется  $j = j + 1$  и пересчитывается  $h_j$ .

# Monotonic-attention-RNN подход



# SIGMORPHON-2017

- Задание 1: автоматическое словоизменение.
- Разный размер обучающей выборки: 100/1000/10000 пар.

# SIGMORPHON-2017

- Задание 1: автоматическое словоизменение.
- Разный размер обучающей выборки: 100/1000/10000 пар.
- 56 языков разной степени сложности (разный размер парадигм).

# SIGMORPHON-2017

- Задание 1: автоматическое словоизменение.
- Разный размер обучающей выборки: 100/1000/10000 пар.
- 56 языков разной степени сложности (разный размер парадигм).
- Задание 2: автоматическое пополнение парадигм.

## SIGMORPHON-2017

- Задание 1: автоматическое словоизменение.
- Разный размер обучающей выборки: 100/1000/10000 пар.
- 56 языков разной степени сложности (разный размер парадигм).
- Задание 2: автоматическое пополнение парадигм.
- Результаты задания 1:

Язык	10000	1000	100
Арабский	94.5	79.7	37.00
Фарерский	87.8	68.1	42.4
Латинский	81.3	51.8	19.3
Литовский	95.6	62.8	23.3
Ньюнорск	92.8	65.6	54.6
Среднее	95.3	82.8	50.6



## SIGMORPHON-2017

- Задание 1: автоматическое словоизменение.
- Разный размер обучающей выборки: 100/1000/10000 пар.
- 56 языков разной степени сложности (разный размер парадигм).
- Задание 2: автоматическое пополнение парадигм.
- Результаты задания 1:

Язык	10000	1000	100
Арабский	94.5	79.7	37.00
Фарерский	87.8	68.1	42.4
Латинский	81.3	51.8	19.3
Литовский	95.6	62.8	23.3
Ньюнорск	92.8	65.6	54.6
Среднее	95.3	82.8	50.6

## SIGMORPHON-2017

- Задание 1: автоматическое словоизменение.
- Разный размер обучающей выборки: 100/1000/10000 пар.
- 56 языков разной степени сложности (разный размер парадигм).
- Задание 2: автоматическое пополнение парадигм.
- Результаты задания 1:

Язык	10000	1000	100
Арабский	94.5	79.7	37.00
Фарерский	87.8	68.1	42.4
Латинский	81.3	51.8	19.3
Литовский	95.6	62.8	23.3
Ньюнорск	92.8	65.6	54.6
Среднее	95.3	82.8	50.6

- Победитель на выборке 10000 — Kahn et al. (расширение модели 2016).
- Победитель на малых выборках — Makarov et al. (Zurich, расширение модели Aharoni-Goldberg).

## Модель Makarov et al.

- При недостатке данных важно научить модель копировать.

## Модель Makarov et al.

- При недостатке данных важно научить модель копировать.
- Для этого в модели Aharoni, Goldberg вместо

$$y_i = \text{softmax}(W \cdot \text{LSTM}(z_1, \dots, z_i) + b),$$

- Используется:

$$\begin{aligned} y_i &= \sigma_i \text{softmax}(W \cdot \text{LSTM}(z_1, \dots, z_i) + b) + (1 - \sigma_i) \llbracket y_i = x_j \rrbracket, \\ s_i &= \text{sigmoid}(v \cdot [y_{i-1}, h_{i-1}, s_i, \mathbf{f}] + c) \end{aligned}$$

## Модель Makarov et al.

- При недостатке данных важно научить модель копировать.
- Для этого в модели Aharoni, Goldberg вместо

$$y_i = \text{softmax}(W \cdot \text{LSTM}(z_1, \dots, z_i) + b),$$

- Используется:

$$\begin{aligned} y_i &= \sigma_i \text{softmax}(W \cdot \text{LSTM}(z_1, \dots, z_i) + b) + (1 - \sigma_i) \llbracket y_i = x_j \rrbracket, \\ s_i &= \text{sigmoid}(v \cdot [y_{i-1}, h_{i-1}, s_i, \mathbf{f}] + c) \end{aligned}$$

- При недостатке данных важно предоставить их дополнительно.

## Модель Makarov et al.

- При недостатке данных важно научить модель копировать.
- Для этого в модели Aharoni, Goldberg вместо

$$y_i = \text{softmax}(W \cdot \text{LSTM}(z_1, \dots, z_i) + b),$$

- Используется:

$$\begin{aligned} y_i &= \sigma_i \text{softmax}(W \cdot \text{LSTM}(z_1, \dots, z_i) + b) + (1 - \sigma_i) \llbracket y_i = x_j \rrbracket, \\ s_i &= \text{sigmoid}(v \cdot [y_{i-1}, h_{i-1}, s_i, \mathbf{f}] + c) \end{aligned}$$

- При недостатке данных важно предоставить их дополнительно.
- Порождаются дополнительные пары лемма-словоформа:
  - Из текста извлекаются шаблоны словоизменения.

## Модель Makarov et al.

- При недостатке данных важно научить модель копировать.
- Для этого в модели Aharoni, Goldberg вместо

$$y_i = \text{softmax}(W \cdot \text{LSTM}(z_1, \dots, z_i) + b),$$

- Используется:

$$\begin{aligned} y_i &= \sigma_i \text{softmax}(W \cdot \text{LSTM}(z_1, \dots, z_i) + b) + (1 - \sigma_i) \llbracket y_i = x_j \rrbracket, \\ s_i &= \text{sigmoid}(v \cdot [y_{i-1}, h_{i-1}, s_i, \mathbf{f}] + c) \end{aligned}$$

- При недостатке данных важно предоставить их дополнительно.
- Порождаются дополнительные пары лемма-словоформа:
  - Из текста извлекаются шаблоны словоизменения.
  - Для слова (из корпуса или случайного) подбирается наиболее специальный шаблон, подходящий под данное слово.

## Модель Makarov et al.

- При недостатке данных важно научить модель копировать.
- Для этого в модели Aharoni, Goldberg вместо

$$y_i = \text{softmax}(W \cdot \text{LSTM}(z_1, \dots, z_i) + b),$$

- Используется:

$$\begin{aligned} y_i &= \sigma_i \text{softmax}(W \cdot \text{LSTM}(z_1, \dots, z_i) + b) + (1 - \sigma_i) \llbracket y_i = x_j \rrbracket, \\ s_i &= \text{sigmoid}(v \cdot [y_{i-1}, h_{i-1}, s_i, \mathbf{f}] + c) \end{aligned}$$

- При недостатке данных важно предоставить их дополнительно.
- Порождаются дополнительные пары лемма-словоформа:
  - Из текста извлекаются шаблоны словоизменения.
  - Для слова (из корпуса или случайного) подбирается наиболее специальный шаблон, подходящий под данное слово.
  - Этот шаблон применяется для порождения словоформы.