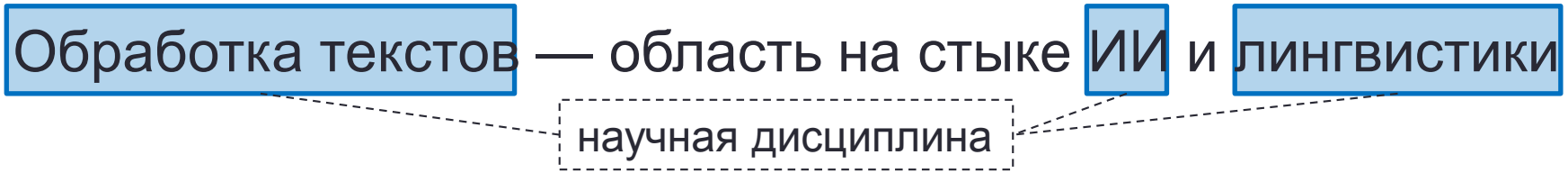
ОСНОВЫ ОБРАБОТКИ ТЕКСТОВ

Лекция #6:
Базовые задачи обработки текстов

Лектор: м.н.с. ИСП РАН Майоров Владимир Дмитриевич

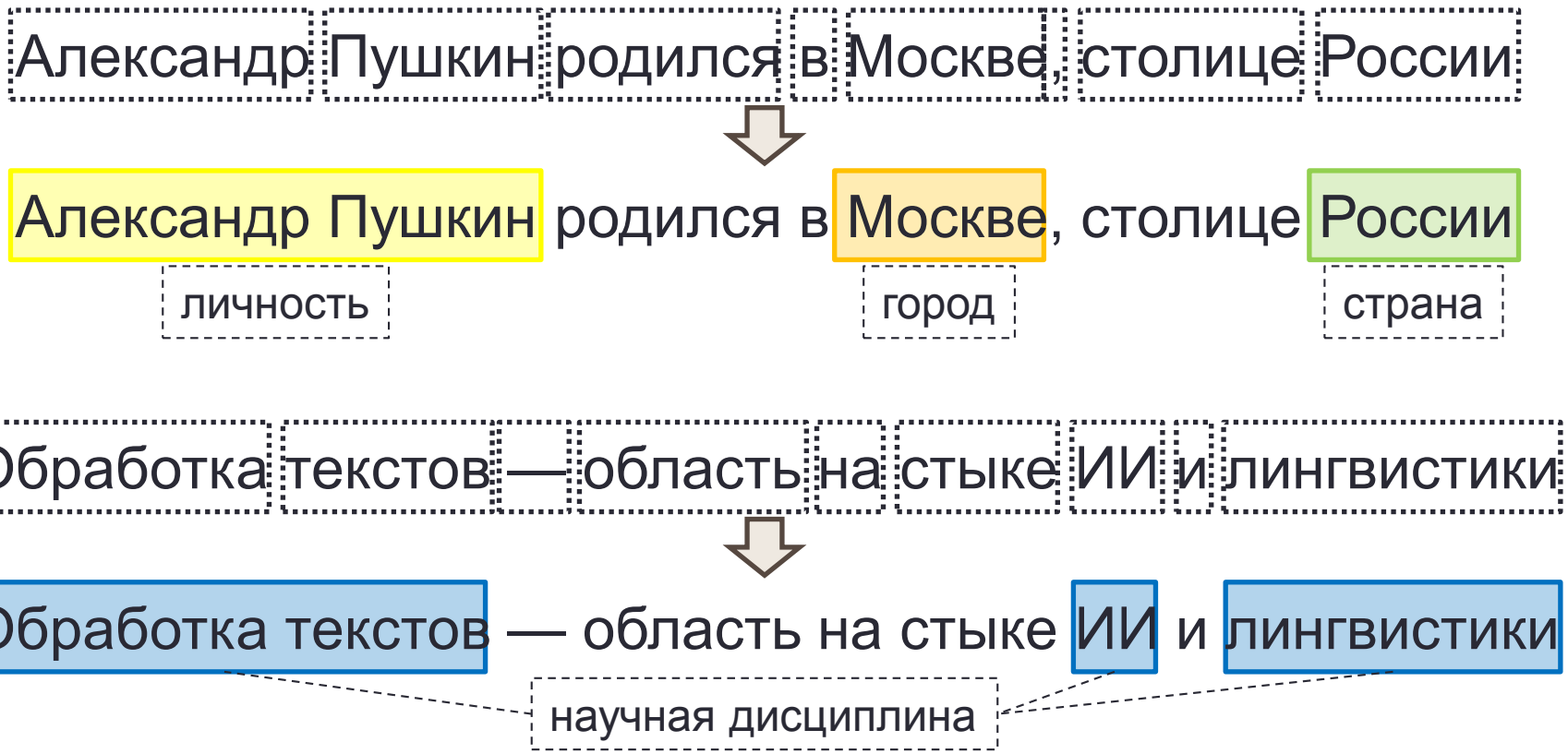
Задача NERC

- На входе: текст, разбитый на предложения и токены
- На выходе: множество сущностей (начало, конец, тип)



Задача NERC

- На входе: текст, разбитый на предложения и токены
- На выходе: множество сущностей (начало, конец, тип)



Сегментация текста

- На входе: текст (последовательность символов)
- На выходе: текст, разбитый на предложения и токены

Александр Пушкин родился в Москве, столице России

Александр | Пушкин | родился | в | Москве | , | столице | России

Обработка текстов — область на стыке ИИ и лингвистики

Обработка | текстов | — | область | на | стыке | ИИ | и | лингвистики

Сегментация текста

- На входе: текст (последовательность символов)
- На выходе: текст, разбитый на предложения и токены

- Предложение – это единица языка, которая представляет собой грамматически организованное соединение слов, обладающее смысловой законченностью.

- Токен – минимальная лингвистическая единица текста (речи), обладающая смыслом.

  {слово, пунктуация, число}

Токенизация

- На входе: текст (последовательность символов)
- На выходе: текст, разбитый на токены

Александр Пушкин родился в Москве, столице России.

- Решение (RegExp, регулярные выражения)

Токенизация

- На входе: текст (последовательность символов)
- На выходе: текст, разбитый на токены

Александр Пушкин родился в Москве, столице России.

- Решение (RegExp, регулярные выражения):
  - Разделить входную строку по пробельным символам

Александр | Пушкин | родился | в | Москве, | столице | России.

Токенизация

- На входе: текст (последовательность символов)
- На выходе: текст, разбитый на токены

Александр Пушкин родился в Москве, столице России.

- Решение (RegExp, регулярные выражения):
  - Разделить входную строку по пробельным символам

Александр | Пушкин | родился | в | Москве, | столице | России.

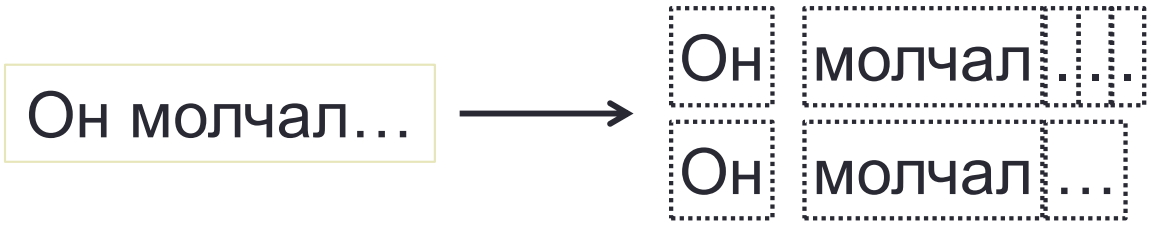
  - Отделить префиксные и постфиксные знаки пунктуации

Александр | Пушкин | родился | в | Москве | , | столице | России | .

Токенизация

- Пунктуация бывает внутри слова

- Может быть несколько символов пунктуации подряд



Токенизация

- Будем считать токенами подстроки, содержащие только буквы и цифры (alphanumeric)

- Остальные подстроки попробуем разделить по не alphanumeric символам, используя классификатор

Признаки:
- Длины токенов-кандидатов
- Символы (n-gram)
- Словарь
- …

Классификаторы:
- SVM
- Лог. регрессия
- Нейронные сети
- …

Токенизация

- Некоторые токены должны состоять из нескольких слов
  - Русский

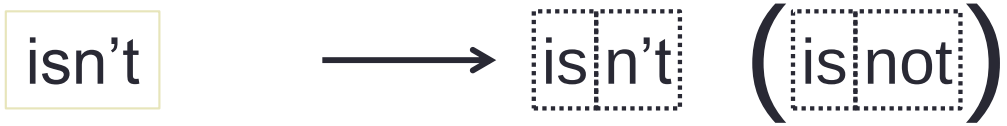
  - Испанский

Токенизация

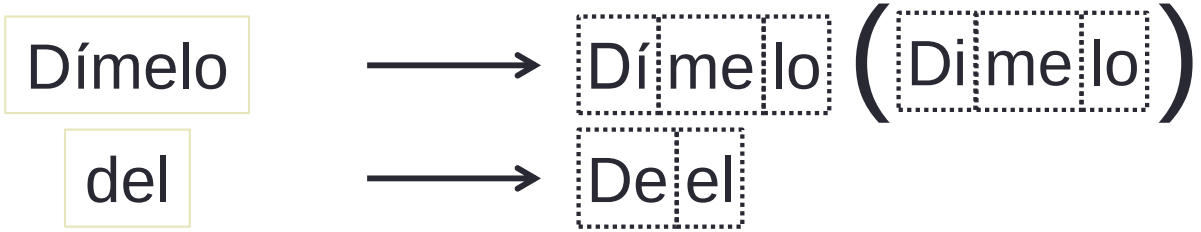
- Некоторые слова склеиваются
  - Английский isn’t → is n’t (is not)
  - Испанский Dímelo → Dí me lo (Di me lo)
    del → De el

Токенизация

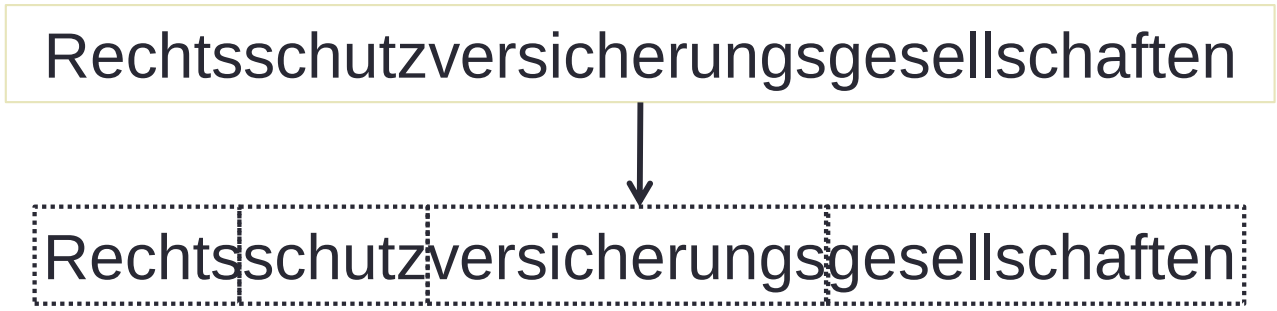
- Некоторые слова склеиваются
  - Английский



  - Испанский

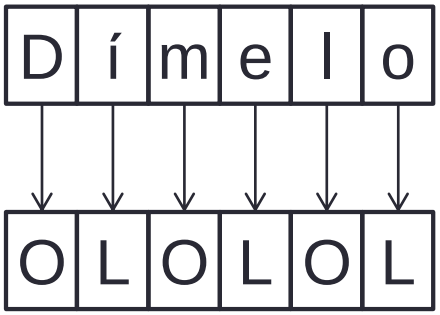


  - Немецкий



Токенизация

- Задача разметки последовательности. Каждый символ необходимо отнести к одному из двух классов:
  - Последний символ в токене (L)
  - Не последний символ в токене (O)
- Методы:
  - RNN (Рекурентные нейронные сети)
  - BiRNN
  - …



- После разделения на символы замена токенов по словарю

Определение границ предложений

- На входе:
  - текст (последовательность токенов)
  - текст (последовательность символов)
- На выходе: текст, разбитый на предложения

- Идея:
  - В естественных языках предложения отделяются друг от друга знаками препинания (**. ! ?**)

- Решение:
  - Классифицировать каждый токен (**. ! ?**) является ли он концом предложения

Сегментация текста

- На входе: текст (последовательность символов)
- На выходе: текст, разбитый на предложения и токены

- Идея
  - Одновременно находить границы токенов и предложений

- Решение
  - Разметка последовательности. Каждый символ классифицируется на один из **трех** классов:
    - Последний символ предложения
    - Последний символ токена
    - Обычный символ

Оценка качества

- Gold

Александр	Пушкин	родился	в	Москве,	столице	России.

- Predicted

Александр	Пушкин	родился	в	Москве,	столице	России.

Оценка качества

- Gold

Александр Пушкин родился в Москве, столице России.

- Predicted

Александр Пушкин родился в Москве, столице России.

- Метрики
  - $Precision = \frac{|correct|}{|predicted|};$
  - $Recall = \frac{|correct|}{|gold|};$
  - $F_1 = \frac{2*Precision*Recall}{Precision+Recall};$

$Precision = \frac{5}{7};$

$Recall = \frac{5}{9};$

$F_1 = \frac{10}{16};$

Определение языка

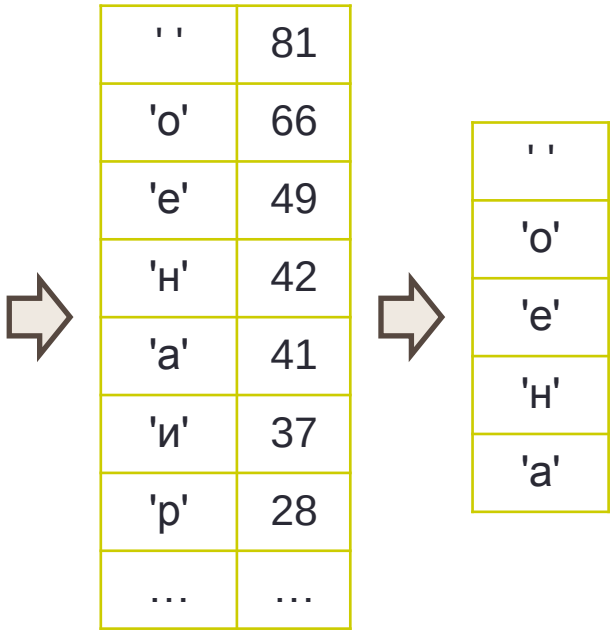
- На входе: текст (последовательность символов)
- На выходе: метка языка

Александр Сергеевич Пушкин родился в Москве 26 мая 1799 года.	русский
Александр Сергей улы Пушкин 1799 елның 26 маенда Мәскәү шәһәрендә туа.	татарский
Alexander Pushkin was born in Moscow on may 26, 1799.	английский
Alexander Pushkin nació en Moscú el 26 de mayo de 1799.	испанский
Alexandre Pouchkine est né à Moscou le 26 mai 1799.	французский

Профили языка*

- Для каждого языка извлекается его профиль
  - Профиль – список (top-K) символьных n-gram отсортированный по невозрастанию частоты

Происхождение Александра Сергеевича Пушкина идёт от разветвлённого нетитулованного дворянского рода Пушкиных, восходившего по генеалогической легенде к «мужу честну» Ратше. Пушкин неоднократно писал о своей родословной в стихах и прозе; он видел в своих предках образец истинной «аристократии», древнего рода, честно служившего отечеству, но не снискавшего благосклонности правителей и «гонимого». Не раз он обращался (в том числе в художественной форме) и к образу своего прадеда по матери — африканца Абрама Петровича Ганнибала, ставшего слугой и воспитанником Петра I, а потом военным инженером и генералом.

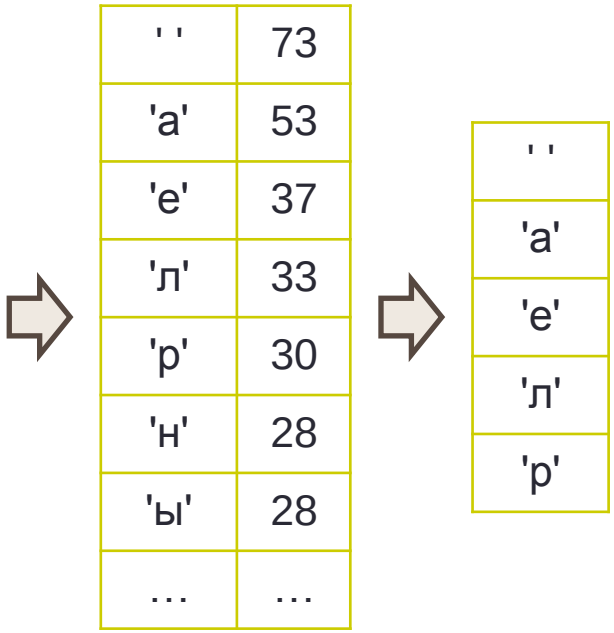


* W. B. Cavnar, J. M. Trenkle. "N-gram-based text categorization." (1994)

Профили языка

- Для каждого языка извлекается его профиль
  - Профиль – список (top-K) символьных n-gram отсортированный по невозрастанию частоты

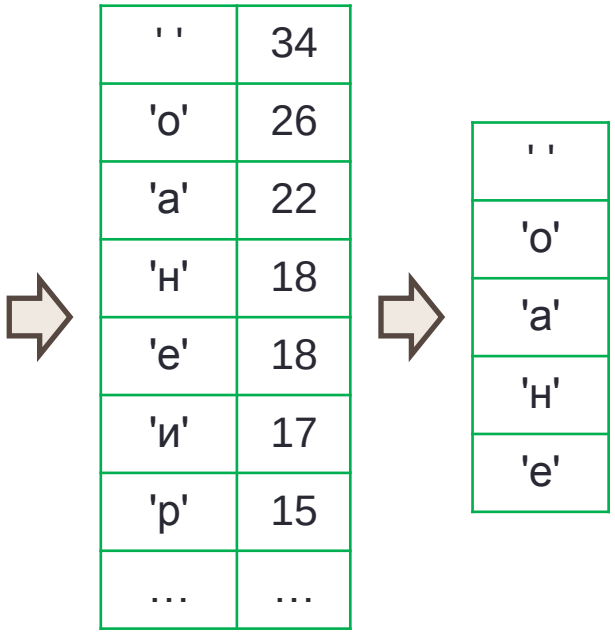
Үсмер чагының 6 елын Пушкин 1811 елның 19 октябрендә дәрәҗәле кешеләр өчен ачылган лицейда үткәрә. Лицейда укыгында усмернең таланты ачыла һәм башкалар тарафыннан югары бәяләнә. Лицейда үткәрелгән еллар, дуслары шагыйрь күңелендә мәңгегә саклана һәм иҗатында чагылыш таба. Александр Сергеевичның яшьлеге шушында үтә, биредә шагыйрь Пушкин туа: 130лап шигырь иҗат ителә, «Руслан һәм Людмила» поэмасы языла башлый, танылган журналларда беренче язмалары басыла. Лицей хәзерге урта яки югары уку йортлары дәрәҗәсендә белем бирергә тиеш була.



Профили языка

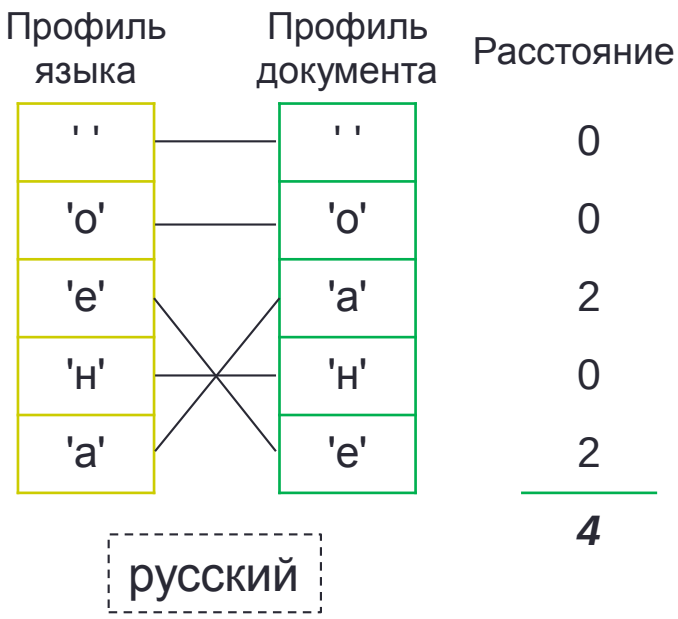
- Для классифицируемого текста извлекается профиль

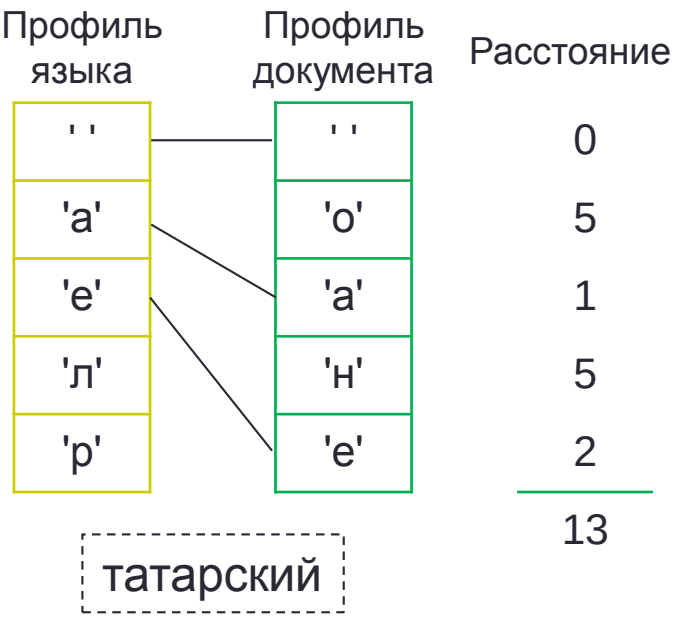
Весной 1820 года Пушкина вызвали к военному генерал-губернатору Петербурга графу М. А. Милорадовичу для объяснения по поводу содержания его стихотворений (в том числе эпиграмм на Аракчеева, архимандрита Фотия и самого Александра I), несовместимых со статусом государственного чиновника.



Профили языка

- Профиль классифицируемого текста сравнивается с профилями языков (out-of-place), выбирается ближайший язык





Naïve Bayes Classifier. Теория

- Теорема Байеса

$$p(c|d) = \frac{p(d|c) * p(c)}{p(d)};$$

  - $p(c|d)$ – вероятность $d \in c$
  - $p(d|c)$ – вероятность встретить d в c
  - $p(c)$ – вероятность встретить документ класса c
  - $p(d)$ – вероятность встретить документ d

- Классификатор

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}}\, p(c|d) = \underset{c \in C}{\operatorname{argmax}} \frac{p(d|c) * p(c)}{p(d)} = \underset{c \in C}{\operatorname{argmax}}\, p(d|c) * p(c)$$

Naïve Bayes Classifier. Теория

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}}\, p(d|c) * p(c)$$

- Представим d как $[f_1, f_2, \dots, f_n]$
  (набор признаков)

$$p(d|c) = p\left(f_1, f_2, \dots, f_{d_n} \middle| c\right) = p(f_1|c)p(f_2, \dots, f_n|c, f_1)$$
$$= p(f_1|c)p(f_2|c, f_1)p(f_3, \dots, f_n|c, f_1, f_2)$$
$$= p(f_1|c)p(f_2|c, f_1) \dots p(f_n|c, f_1, f_2, \dots, f_{n-1})$$

Naïve Bayes Classifier. Теория

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}}\, p(d|c) * p(c)$$

- Представим d как $[f_1, f_2, \dots, f_n]$
  (набор признаков)

$$p(d|c) = p(f_1, f_2, \dots, f_{d_n}|c) = p(f_1|c)p(f_2, \dots, f_n|c, f_1)$$
$$= p(f_1|c)p(f_2|c, f_1)p(f_3, \dots, f_n|c, f_1, f_2)$$
$$= p(f_1|c)p(f_2|c, f_1) \dots p(f_n|c, f_1, f_2, \dots, f_{n-1})$$

- Предположим, что f_i взаимно независимы при условии c
  $$p(f_i|c, f_j) = p(f_i|c), i \neq j$$

Naïve Bayes Classifier. Теория

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}}\, p(d|c) * p(c)$$

- Представим d как $[f_1, f_2, \dots, f_n]$
  (набор признаков)

$$p(d|c) = p(f_1|c)p(f_2|c\,) \dots p(f_n|c) = \prod_{i=1}^{n} p(f_i|c)$$

- Предположим, что f_i взаимно независимы при условии c

$$p(f_i|c, f_j) = p(f_i|c), i \neq j$$

Naïve Bayes Classifier. Теория

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \prod_{i=1}^{n} p(f_i|c) * p(c);$$

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \left[\sum_{i=1}^{n} \log p(f_i|c) + \log p(c) \right];$$

Определение языка

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \left[\sum_{i=1}^{n} \log p(f_i|c) + \log p(c) \right];$$

- Признаки – буквенные n-gram'ы
- Оценка параметров:
  - $p(c) = \frac{|D_c|}{|D|}$;
  - $p(f_i|c) = \frac{|\{f_i \in D_c\}|}{\sum_j |\{f_j \in D_c\}|}$;

Naïve Bayes Classifier. Сглаживание

- Проблема: некоторые n-gram'ы могут не встретиться в обучающем корпусе
  - $p(f_i|c) = \frac{|\{f_i \in D_c\}|}{\sum_j |\{f_j \in D_c\}|} = \frac{0}{\sum_j |\{f_j \in D_c\}|} = 0;$
- Решение: аддитивное сглаживание
  - $p(f_i|c) = \frac{|\{f_i \in D_c\}| + \varepsilon}{\sum_j (|\{f_j \in D_c\}| + \varepsilon)} \neq 0;$

Naïve Bayes Classifier. Обучение

Происхождение Александра Сергеевича Пушкина идёт от разветвлённого нетитулованного дворянского рода Пушкиных, восходившего по генеалогической легенде к «м
св
пр
ро
бл
об
об
Пе
Пе

русский

Үсмер чагының 6 елын Пушкин 1811 елның 19 октябрендә дәрәҗәле кешеләр өчен ачылган лицейда үткәрә. Лицейда укыгында усмернең таланты ачыла һәм башкалар тарафыннан югары бәяләнә. Лицейда үткәрелгән еллар, дуслары шагыйрь күңелендә мәңгегә саклана һәм иҗатында чагылыш таба. Александр Сергеевичның яшьлеге шушында үтә, биредә шагыйрь Пушкин туа: 130лап шигырь иҗат ителә, «Руслан һәм Людмила» поэмасы языла башлый, танылган журналларда беренче язмалары басыла. Лицей хәзерге урта яки югары уку йортлары дәрәҗәсендә белем бирергә тиеш була.

татарский

$$p(\text{русский}) = \frac{1}{2}; \quad p(\text{татарский}) = \frac{1}{2};$$

Naïve Bayes Classifier. Обучение

русский

Происхождение Александра Сергеевича Пушкина идёт от разветвлённого нетитулованного дворянского рода Пушкиных, восходившего по генеалогической легенде к «мужу честну» Ратше. Пушкин неоднократно писал о своей родословной в стихах и прозе; он видел в своих предках образец истинной «аристократии», древнего рода, честно служившего отечеству, но не снискавшего благосклонности правителей и «гонимого». Не раз он обращался (в том числе в художественной форме) и к образу своего прадеда по матери — африканца Абрама Петровича Ганнибала, ставшего слугой и воспитанником Петра I, а потом военным инженером и генералом.

$$p(\text{'П'}|\text{русский}) = \frac{6+1}{610+46} = 0.01067; \quad p(\text{'и'}|\text{русский}) = \frac{37+1}{610+46} = 0.05793;$$

$$p(\text{'р'}|\text{русский}) = \frac{28+1}{610+46} = 0.04421; \quad p(\text{'с'}|\text{русский}) = \frac{27+1}{610+46} = 0.04268;$$

$$p(\text{'о'}|\text{русский}) = \frac{66+1}{610+46} = 0.10213; \quad p(\text{'ж'}|\text{русский}) = \frac{5+1}{610+46} = 0.00914;$$

…

Naïve Bayes Classifier. Обучение

татарский

Үсмер чагының 6 елын Пушкин 1811 елның 19 октябрендә дәрәҗәле кешеләр өчен ачылган лицейда үткәрә. Лицейда укыгында усмернең таланты ачыла һәм башкалар тарафыннан югары бәяләнә. Лицейда үткәрелгән еллар, дуслары шагыйрь күңелендә мәңгегә саклана һәм иҗатында чагылыш таба. Александр Сергеевичның яшьлеге шушында үтә, биредә шагыйрь Пушкин туа: 130лап шигырь иҗат ителә, «Руслан һәм Людмила» поэмасы языла башлый, танылган журналларда беренче язмалары басыла. Лицей хәзерге урта яки югары уку йортлары дәрәҗәсендә белем бирергә тиеш була.

$$p(\text{'П'}|\text{татарский}) = \frac{2+1}{538+54} = 0.00507; \quad p(\text{'и'}|\text{татарский}) = \frac{16+1}{538+54} = 0.02872;$$

$$p(\text{'р'}|\text{татарский}) = \frac{30+1}{538+54} = 0.05236; \quad p(\text{'с'}|\text{татарский}) = \frac{9+1}{538+54} = 0.01689;$$

$$p(\text{'о'}|\text{татарский}) = \frac{3+1}{538+54} = 0.00675; \quad p(\text{'ж'}|\text{татарский}) = \frac{1+1}{538+54} = 0.00337;$$

…

Naïve Bayes Classifier. Применение

Весной 1820 года Пушкина вызвали к военному генерал-губернатору Петербурга графу М. А. Милорадовичу для объяснения по поводу содержания его стихотворений (в том числе эпиграмм на Аракчеева, архимандрита Фотия и самого Александра I), несовместимых со статусом государственного чиновника.

$$\hat{c}_j = \log p(c_j) + \sum_{i=1}^{n} \log p(f_i|c_j);$$

$$\hat{c}_{\text{русский}} = \underset{\log p('\text{B}'|c)}{-0.69315} + \underset{\log p('\text{e}'|c)}{(-6.48768)} + \underset{\log p('\text{c}'|c)}{(-2.57566)} + \underset{\log p('\text{.}'|c)}{(-3.15547)} + \cdots + (-5.10139)$$

$$\hat{c}_{\text{русский}} = \mathbf{-953.05578}$$

$$\hat{c}_{\text{татарский}} = \underset{\log p('\text{B}'|c)}{-0.69315} + \underset{\log p('\text{e}'|c)}{(-6.40026)} + \underset{\log p('\text{c}'|c)}{(-2.76267)} + \underset{\log p('\text{.}'|c)}{(-4.09767)} + \cdots + (-4.60850)$$

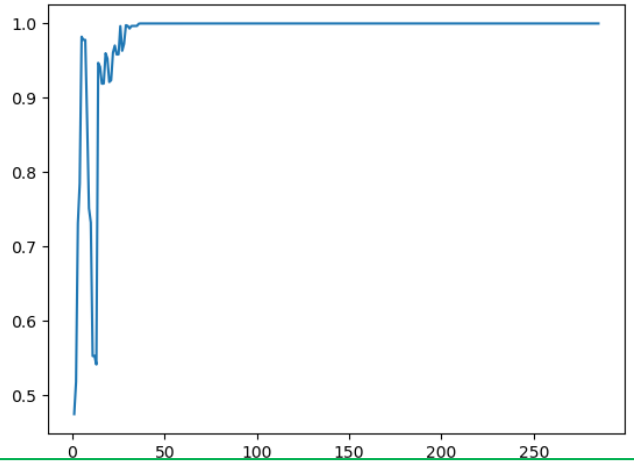
$$\hat{c}_{\text{татарский}} = -1066.18426$$

Naïve Bayes Classifier. Применение

- Назад к вероятностям:

$$p(c_i|d) = \frac{e^{\hat{c}_i}}{\sum_j e^{\hat{c}_j}} = \frac{1}{1 + \sum_{j \neq i} e^{\hat{c}_j - \hat{c}_i}};$$

- Наш пример:
  - $\hat{c}_{\text{русский}} = \mathbf{-953.05578}$
  - $\hat{c}_{\text{татарский}} = -1066.18426$
  - $p(\text{русский}|d) = \frac{1}{1+e^{-113.12848}} \approx 1$
- Для первых 30 символов:
  - $\hat{c}_{\text{русский}} = \mathbf{-113.25854}$
  - $\hat{c}_{\text{татарский}} = -118.98256$
  - $p(\text{русский}|d) = \frac{1}{1+e^{-5.72402}} \approx 0.99674$



Весной 1820 года Пушкина вызва

Определение языка

- Задача многоклассовой классификации (multiclass classification)
  - Объекты: тексты
  - Классы: метки языков
- Признаки классификации
  - N-gram по символам
  - Слова (из словаря)
- Классификаторы
  - Multiclass SVM
  - Naïve Bayes
  - Нейронные сети
    - Perceptron
    - Convolutional NN
  - …

Оценка качества (binary classification)

- $Accuracy = \frac{tp+tn}{tp+tn+fp+fn};$
- $Precision = \frac{tp}{tp+fp};$
- $Recall = \frac{tp}{tp+fn};$
- $F_1 = \frac{2*Precision*Recall}{Precision+Recall};$

Confusion table

		Ожидаемые классы	
		спам	не спам
Предсказанные классы	спам	**tp**	fp
	не спам	fn	**tn**

Оценка качества (multiclass classification)

Confusion matrix

		Ожидаемые классы		
		русский	татарский	английский
Предсказанные классы	русский	**5**	2	0
	татарский	3	**3**	2
	английский	0	1	**10**

Оценка качества (multiclass classification)

Confusion matrix

Предсказанные классы	Ожидаемые классы: русский	татарский	английский
русский	**5**	2	0
татарский	3	**3**	2
английский	0	1	**10**

Confusion table
(класс «русский»)

Предсказанные классы	Ожидаемые классы: русский	не русский
русский	**5**	2
не русский	3	**17**

$$Precision = \frac{5}{7};$$

$$Recall = \frac{5}{8};$$

$$F_1 = \frac{2}{3};$$

Оценка качества (multiclass classification)

Confusion matrix

Предсказанные классы	Ожидаемые классы: русский	татарский	английский
русский	**5**	2	0
татарский	3	**3**	2
английский	0	1	**10**

Confusion table
(класс «татарский»)

Предсказанные классы	Ожидаемые классы: татарский	не татарский
татарский	**3**	5
не татарский	3	**15**

$$Precision = \frac{3}{8};$$

$$Recall = \frac{1}{2};$$

$$F_1 = \frac{3}{7};$$

Оценка качества (multiclass classification)

Confusion matrix

Предсказанные классы	Ожидаемые классы: русский	татарский	английский
русский	**5**	2	0
татарский	3	**3**	2
английский	0	1	**10**

Confusion table (класс «английский»)

Предсказанные классы	Ожидаемые классы: английский	не английский
английский	**10**	1
не английский	2	**13**

$$Precision = \frac{10}{11};$$

$$Recall = \frac{5}{6};$$

$$F_1 = \frac{20}{23};$$

Оценка качества (multiclass classification)

- Макро усреднение
  - $Precision = \frac{1}{|C|}\sum Precision(c)$;
  - $Recall = \frac{1}{|C|}\sum Recall(c)$;
  - $F_1 = \frac{1}{|C|}\sum F_1(c)$;

$$Precision = \frac{1231}{1848} \approx 0.6661;$$
$$Recall = \frac{47}{72} \approx 0.6528;$$
$$F_1 = \frac{949}{1449} \approx 0.6549;$$

		Ожидаемые классы	
		русский	не русский
Предсказанные классы	русский	**5**	2
	не русский	3	**17**

		Ожидаемые классы	
		татарский	не татарский
Предсказанные классы	татарский	**3**	5
	не татарский	3	**15**

		Ожидаемые классы	
		английский	не английский
Предсказанные классы	английский	**10**	1
	не английский	2	**13**

Оценка качества (multiclass classification)

- Взвешенное макро усреднение
  - $Precision = \frac{\sum(tp_c+fn_c)Precision(c)}{\sum(tp_c+fn_c)};$
  - $Recall = \frac{\sum(tp_c+fn_c)Recall(c)}{\sum(tp_c+fn_c)};$
  - $F_1 = \frac{\sum(tp_c+fn_c)F_1(c)}{\sum(tp_c+fn_c)};$

$$Precision = \frac{5813}{8008} \approx 0.7259;$$
$$Recall = \frac{9}{13} \approx 0.6923;$$
$$F_1 = \frac{4429}{6279} \approx 0.7054;$$

8

Предсказанные классы	Ожидаемые классы: русский	Ожидаемые классы: не русский
русский	**5**	2
не русский	3	**17**

6

Предсказанные классы	Ожидаемые классы: татарский	Ожидаемые классы: не татарский
татарский	**3**	5
не татарский	3	**15**

12

Предсказанные классы	Ожидаемые классы: английский	Ожидаемые классы: не английский
английский	**10**	1
не английский	2	**13**

Оценка качества (multiclass classification)

- Микро усреднение
  - $Precision = \frac{\sum_c tp_c}{\sum_c tp_c + fp_c};$
  - $Recall = \frac{\sum_c tp_c}{\sum_c tp_c + fn_c};$
  - $F_1 = \frac{2 * Precision * Recall}{Precision + Recall};$

$$Precision = \frac{9}{13} \approx 0.6923;$$
$$Recall = \frac{9}{13} \approx 0.6923;$$
$$F_1 = \frac{9}{13} \approx 0.6923;$$

		Ожидаемые классы		
		русский	татарский	английский
Предсказанные классы	русский	**5**	2	0
	татарский	3	**3**	2
	английский	0	1	**10**

Определение языка: сложности

- Очень короткий текст

Один два три четыре пять	русский
Адзін два тры чатыры пяць	белорусский
Один два три чотири п’ять	украинский
Един два три четири пет	болгарский
Еден два три четири пет	македонский
Један два три четири пет	сербский

Определение языка: сложности

- Текст сразу на нескольких языках

I heard “El que quiera entender que entienda” by Mägo de Oz?

испанский

английский

¿Conoces “Born to Touch Your Feelings” de Scorpions?

английский

испанский

I am at Международный аэропорт Пулково in Санкт-Петербург

английский

русский

Определение языка: сложности

- Текст сразу на нескольких языках

I heard “El que quiera entender que entienda” by Mägo de Oz? — английский

¿Conoces “Born to Touch Your Feelings” de Scorpions? — испанский

I am at Международный аэропорт Пулково in Санкт-Петербург — английский

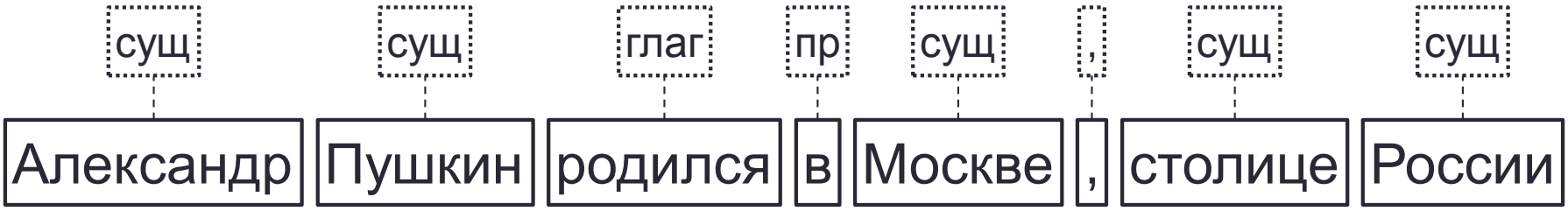
Определение языка: сложности

- Текст сразу на нескольких языках

Морфологический анализ

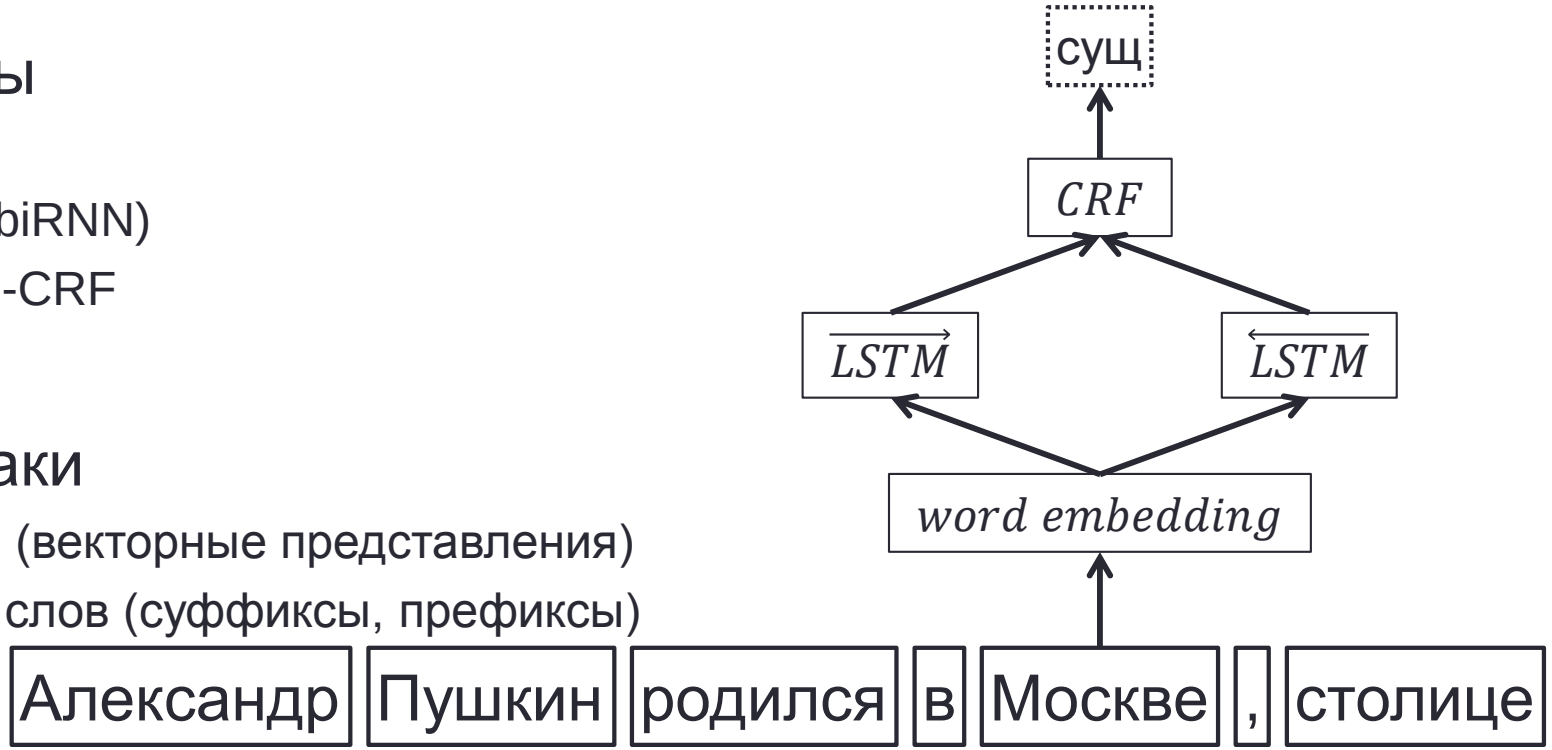
- Задача
  - На входе: предложение (последовательность токенов)
  - На выходе: морфологическая метка для каждого слова

- Морфологическая метка – часть речи
  - Существительное
  - Прилагательное
  - Глагол
  - …



Морфологический анализ

- Задача разметки последовательности
  - На входе: последовательность токенов
  - На выходе: морфологическая метка для каждого слова
- Методы
  - CRF
  - RNN (biRNN)
  - biRNN-CRF
- Признаки
  - Слова (векторные представления)
  - Части слов (суффиксы, префиксы)



Методы оценки качества

- Метрики многоклассовой классификации

- Sentence Accuracy
  - Количество правильно разобранных предложений среди всех предложений

Морфологический анализ

- Задача
  - На входе: предложение (последовательность токенов)
  - На выходе: морфологическая метка для каждого слова

- Морфологическая метка – часть речи + граммемы

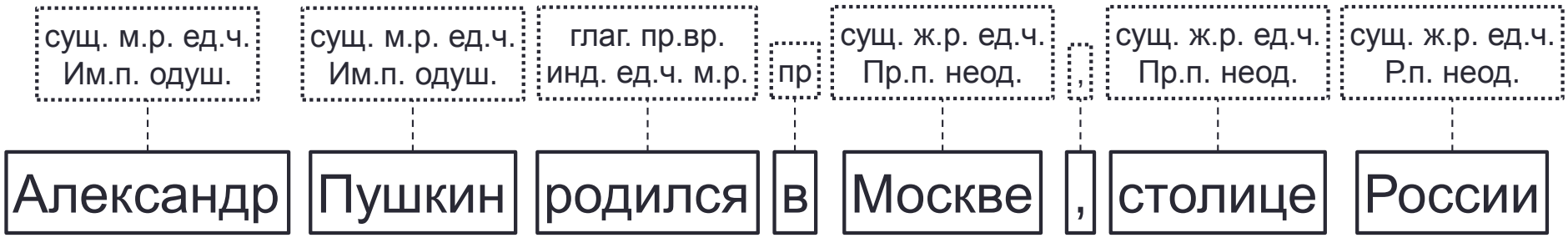
Части речи
- Существительное
- Прилагательное
- Глагол
- …

Род
- Мужской
- Женский
- Средний

Число
- Единственное
- Множественное

Падеж
- Именительный
- Родительный
- Дательный
- …

…



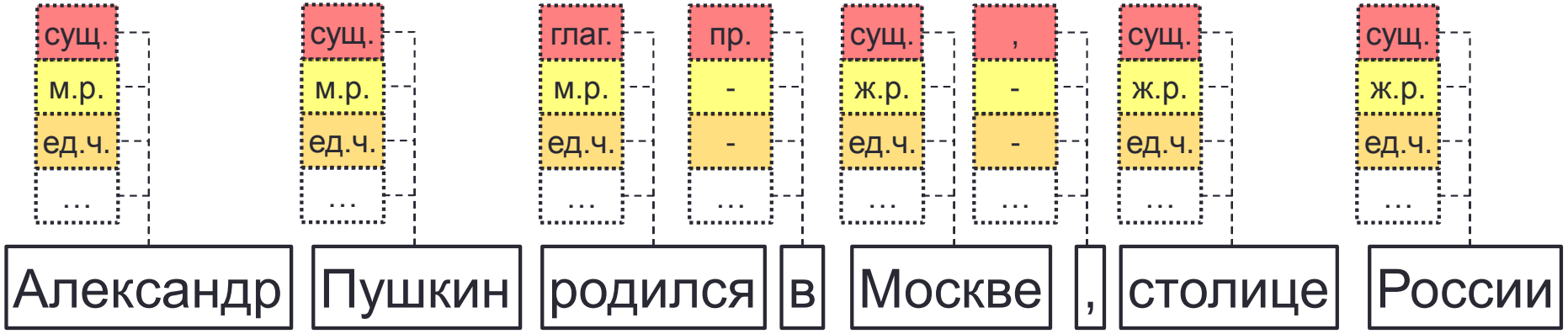
Морфологический анализ

- Проблемы
  - Много классов (UD SynTagRus: 733)
  - Некоторые классы очень редкие ⇒ не получится обучиться

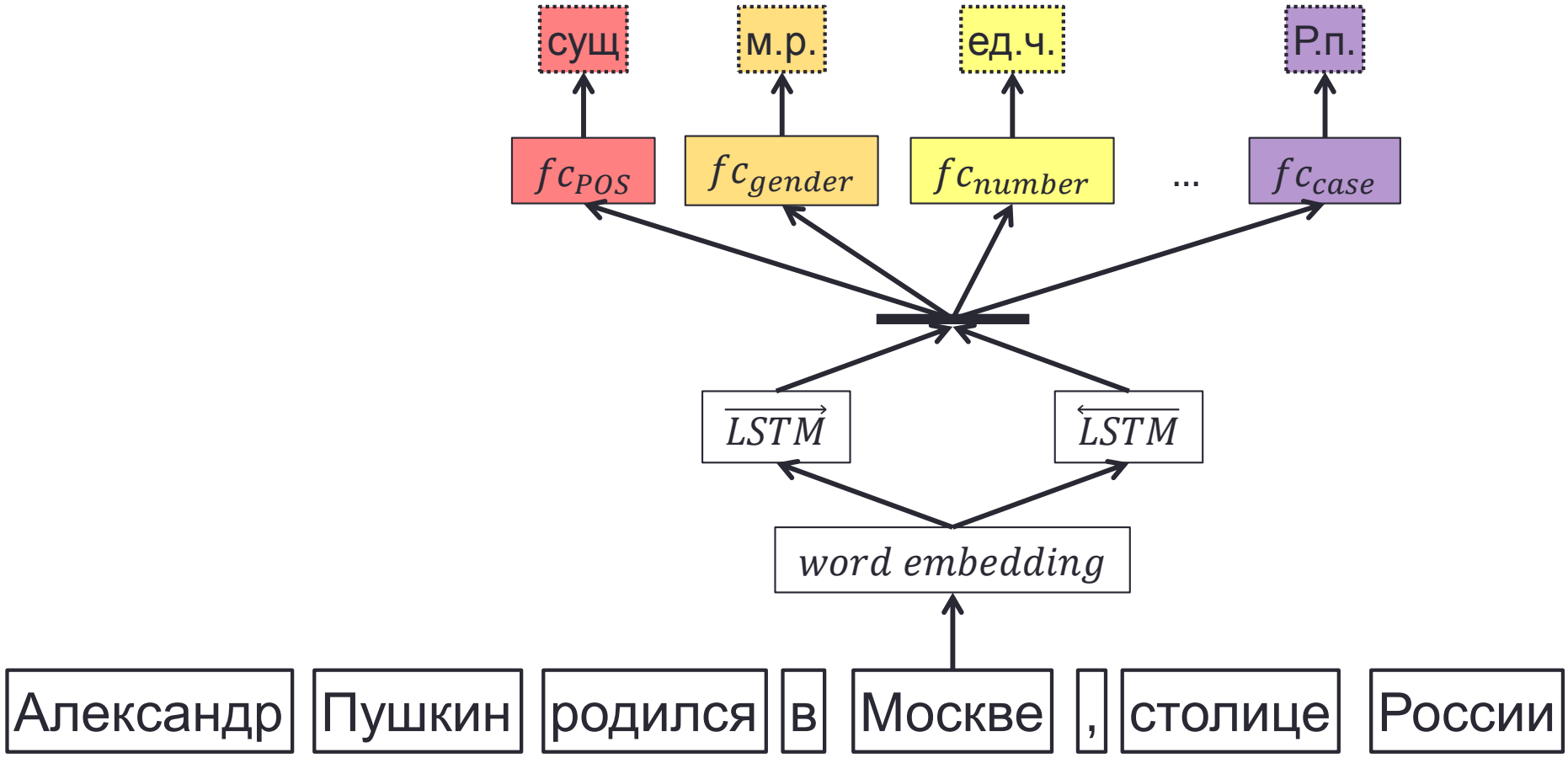
Морфологический анализ

- Проблемы
  - Много классов (UD SynTagRus: 733)
  - Некоторые классы очень редкие ⇒ не получится обучиться

- Решение
  - Разделить метки по грамматическим категориям
  - Назначить каждому токену по одной метке из каждой категории
  - Multilabel classification



Морфологический анализ



Морфологический анализ

- При таком подходе может получиться неправильная метка:
  - Род, число падеж у предлогов/союзов
  - Род у глаголов настоящего времени
  - Время у существительных

- Решение:
  - Собрать множество возможных меток (из корпуса/руками)
  - Определять наиболее вероятную метку из возможных

$$p(\text{сущ. м.р. ед.ч.}|w) = p(\text{сущ}|w)p(\text{м. р. }|w)p(\text{ед. ч}|w)$$

Нормализация (лемматизация) слов

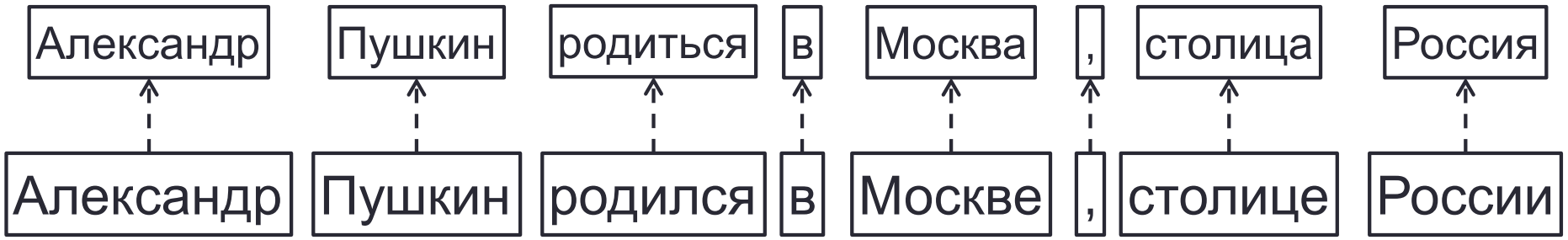
- Задача
  - На входе: предложение/последовательность слов (токенов)
  - На выходе: последовательность лемм – слов (токенов) в нормальной форме

- Нормальная форма слова – каноническая форма слова
  - Для существительных:
    - единственное число
    - именительный падеж
  - Для глаголов:
    - инфинитив
  - …

Нормализация (лемматизация) слов

- Задача
  - На входе: предложение/последовательность слов (токенов)
  - На выходе: последовательность лемм – слов (токенов) в нормальной форме

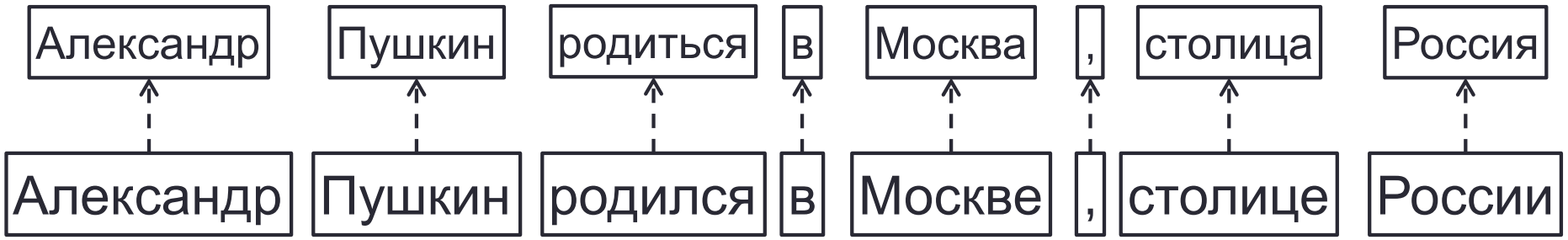
- Задача похожа на разметку последовательности



Нормализация (лемматизация) слов

- Задача
  - На входе: предложение/последовательность слов (токенов)
  - На выходе: последовательность лемм – слов (токенов) в нормальной форме

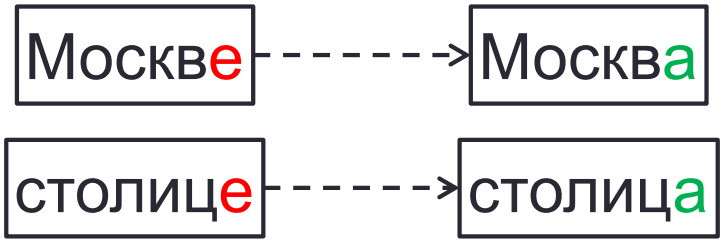
- Задача похожа на разметку последовательности, но
  - На выходе слова, а не классы
  - Слова на выходе не связаны между собой



Нормализация (лемматизация) слов

- Задача
  - На входе: слово (токен)
  - На выходе: лемма – слово (токен) в нормальной форме

- Решение
  - Заменять суффикс словоформы на суффикс канонической формы



Ripple Down Rules

- Способ инкрементального описания знаний

IF птица THEN летает EXCEPT
IF птенец THEN
не летает
ELSE IF пингвин THEN не летает EXCEPT
IF пингвин в самолете THEN летает
ELSE IF самолет THEN летает
…

LemmaGen*

- Основан на Ripple Down Rules

IF suffix THEN transformation EXCEPT exceptions

IF “” THEN “”→”” EXCEPT
IF “d” THEN “d”→””
ELSE IF “ote” THEN “ote”→”ite”
ELSE IF “ing” THEN “ing”→”e” EXCEPT
IF “ting” THEN “”→””
ELSE IF “ten” THEN “ten”→”e”
ELSE IF “s” THEN “s”→””
…

* Juršic M. et al. "Lemmagen: Multilingual lemmatisation with induced ripple-down rules." (2010)

LemmaGen. Обучение

- Собрать отсортированный словарь всех словоформ с леммами
- Отсортировать его по инвертированным словоформам
- Рекурсивно построить RDR

1. Взять весь список
2. Для текущего списка:
   a) Выделить самый длинный общий суффикс
   b) Выделить самое частое правило
   c) Добавить правило для общего суффикса
   d) Разделить список по суффиксу на 1 больше
   e) Для каждого списка повторить 2.

слова	слово
сова	сова
мама	мама
словом	слово
слово	слово
сову	сова
маму	мама
семья	семья

LemmaGen. Обучение

1. Взять весь список
2. Для текущего списка:
   a) Выделить самый длинный общий суффикс
   b) Выделить самое частое правило
   c) Добавить правило для общего суффикса
   d) Разделить список по суффиксу на 1 больше
   e) Для каждого списка повторить 2.

IF “” THEN “”→””

слова	слово
сова	сова
мама	мама
словом	слово
слово	слово
сову	сова
маму	мама
семья	семья

LemmaGen. Обучение

1. Взять весь список
2. Для текущего списка:
   a) Выделить самый длинный общий суффикс
   b) Выделить самое частое правило
   c) Добавить правило для общего суффикса
   d) Разделить список по суффиксу на 1 больше
   e) Для каждого списка повторить 2.

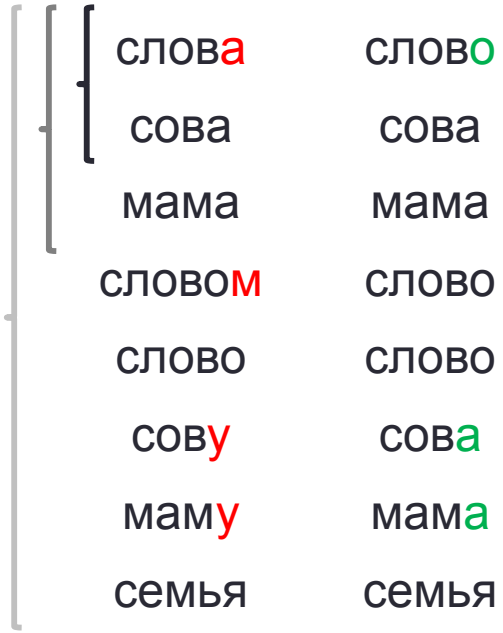
IF “” THEN “”→”” EXCEPT
IF “а” THEN “”→””

слова	слово
сова	сова
мама	мама
словом	слово
слово	слово
сову	сова
маму	мама
семья	семья

LemmaGen. Обучение

1. Взять весь список
2. Для текущего списка:
   a) Выделить самый длинный общий суффикс
   b) Выделить самое частое правило
   c) Добавить правило для общего суффикса
   d) Разделить список по суффиксу на 1 больше
   e) Для каждого списка повторить 2.

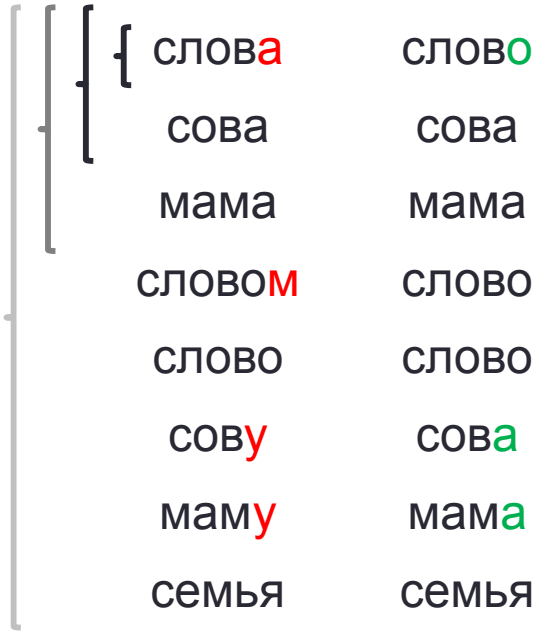
IF “” THEN “”→”” EXCEPT
IF “а” THEN “”→”” EXCEPT
IF “ова” THEN “а”→”о”



LemmaGen. Обучение

1. Взять весь список
2. Для текущего списка:
   a) Выделить самый длинный общий суффикс
   b) Выделить самое частое правило
   c) Добавить правило для общего суффикса
   d) Разделить список по суффиксу на 1 больше
   e) Для каждого списка повторить 2.

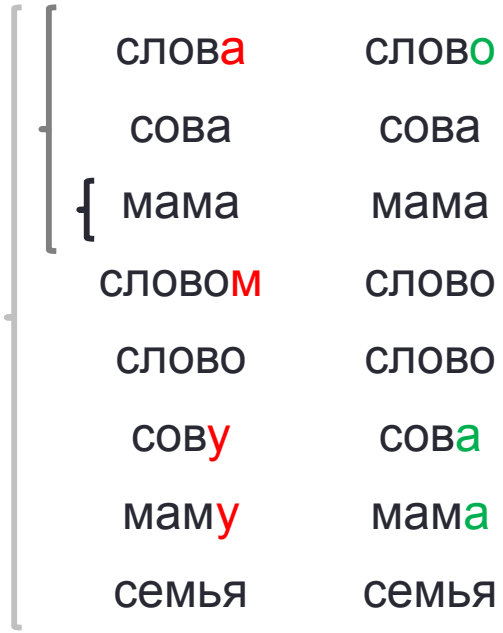
IF “” THEN “”→”” EXCEPT
IF “а” THEN “”→”” EXCEPT
IF “ова” THEN “а”→”о”



LemmaGen. Обучение

1. Взять весь список
2. Для текущего списка:
   a) Выделить самый длинный общий суффикс
   b) Выделить самое частое правило
   c) Добавить правило для общего суффикса
   d) Разделить список по суффиксу на 1 больше
   e) Для каждого списка повторить 2.

IF “” THEN “”→”” EXCEPT
IF “а” THEN “”→”” EXCEPT
IF “ова” THEN “а”→”о”



LemmaGen. Обучение

1. Взять весь список
2. Для текущего списка:
   a) Выделить самый длинный общий суффикс
   b) Выделить самое частое правило
   c) Добавить правило для общего суффикса
   d) Разделить список по суффиксу на 1 больше
   e) Для каждого списка повторить 2.

IF “” THEN “”→”” EXCEPT
IF “а” THEN “”→”” EXCEPT
IF “ова” THEN “а”→”о”

слова	слово
сова	сова
мама	мама
словом	слово
слово	слово
сову	сова
маму	мама
семья	семья

LemmaGen. Обучение

1. Взять весь список
2. Для текущего списка:
   a) Выделить самый длинный общий суффикс
   b) Выделить самое частое правило
   c) Добавить правило для общего суффикса
   d) Разделить список по суффиксу на 1 больше
   e) Для каждого списка повторить 2.

IF “” THEN “”→”” EXCEPT
IF “а” THEN “”→”” EXCEPT
IF “ова” THEN “а”→”о”
ELSE IF “у” THEN “у”→”а”

слова	слово
сова	сова
мама	мама
словом	слово
слово	слово
сову	сова
маму	мама
семья	семья

LemmaGen

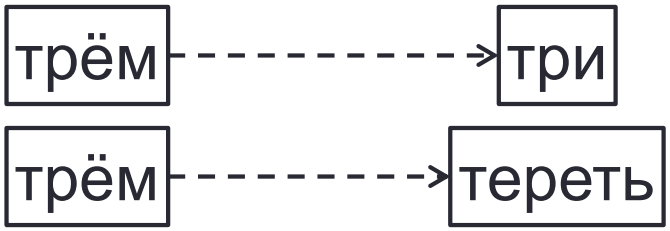
- Проблема: Слова исключения
  - Русский

  - Испанский

- Решение
  - Составить словарь исключений
  - Исключения можно записать в RDR

LemmaGen

- Проблема: Грамматическая омонимия
  - Русский



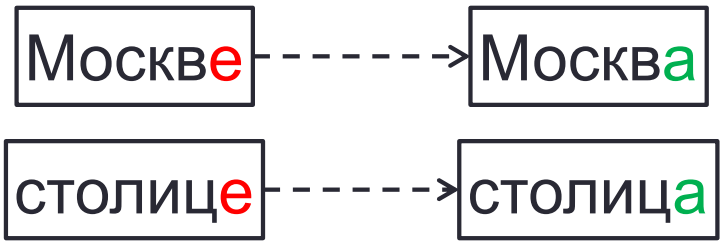
  - Английский

- Решение
  - Использовать информацию о морфологии слова
  - Построить для каждой части речи свой LemmaGen

Нормализация (лемматизация) слов

- Задача
  - На входе: слово (токен)
  - На выходе: лемма – слово (токен) в нормальной форме

- Процесс построения леммы
  - Удалить суффикс словоформы
  - Добавить суффикс леммы
  - Удалить префикс словоформы
  - Добавить префикс леммы



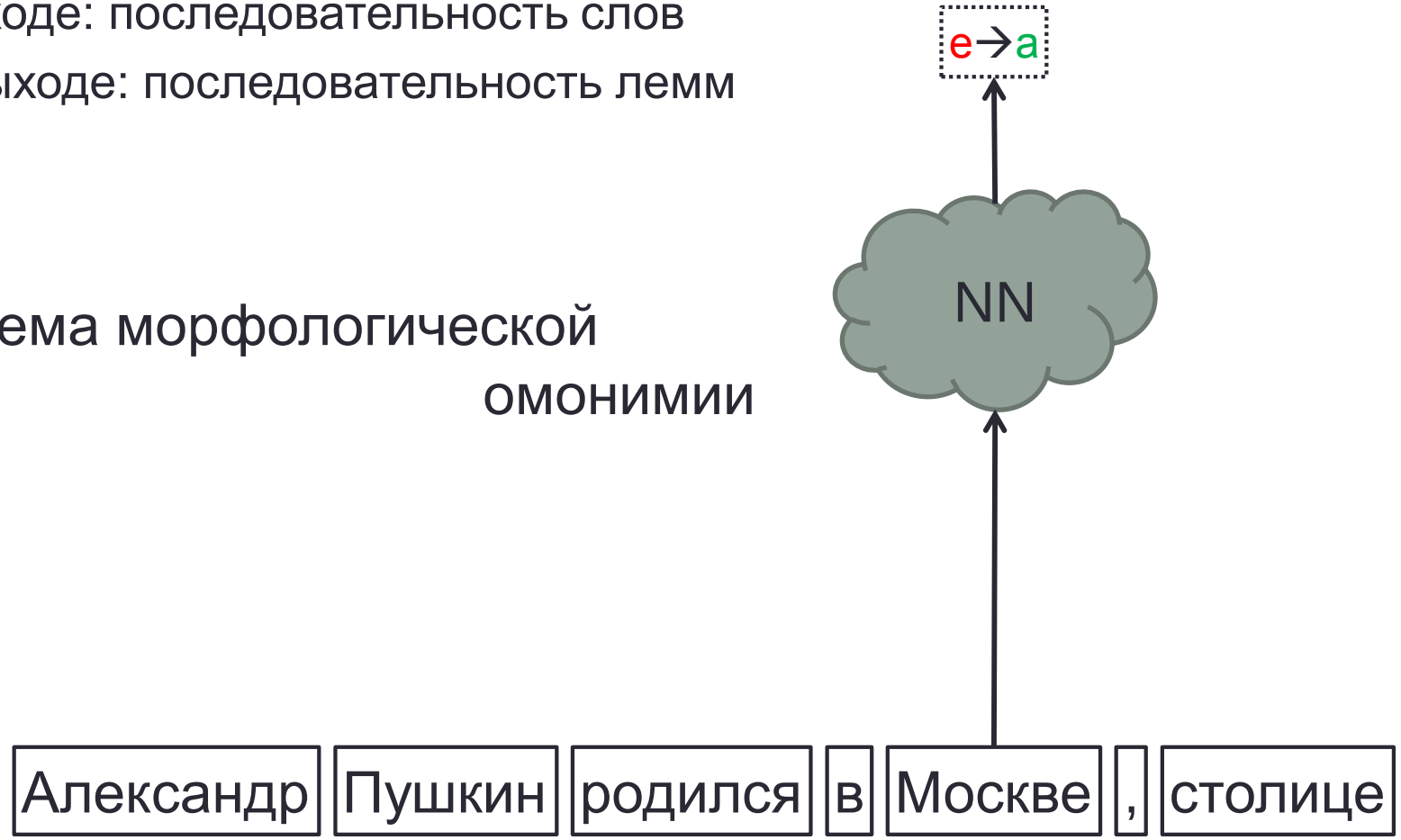
Нормализация (лемматизация) слов

Решаем задачу многоклассовой классификации

- Классы
  - суффиксы словоформ, суффиксы лемм, префиксы словоформ, префиксы лемм
  - Правила преобразования суффикса, правила преобразования префикса
- Признаки
  - Символы слова
  - Морфологические признаки
- Классификаторы
  - CNN
  - Perceptron
  - ...

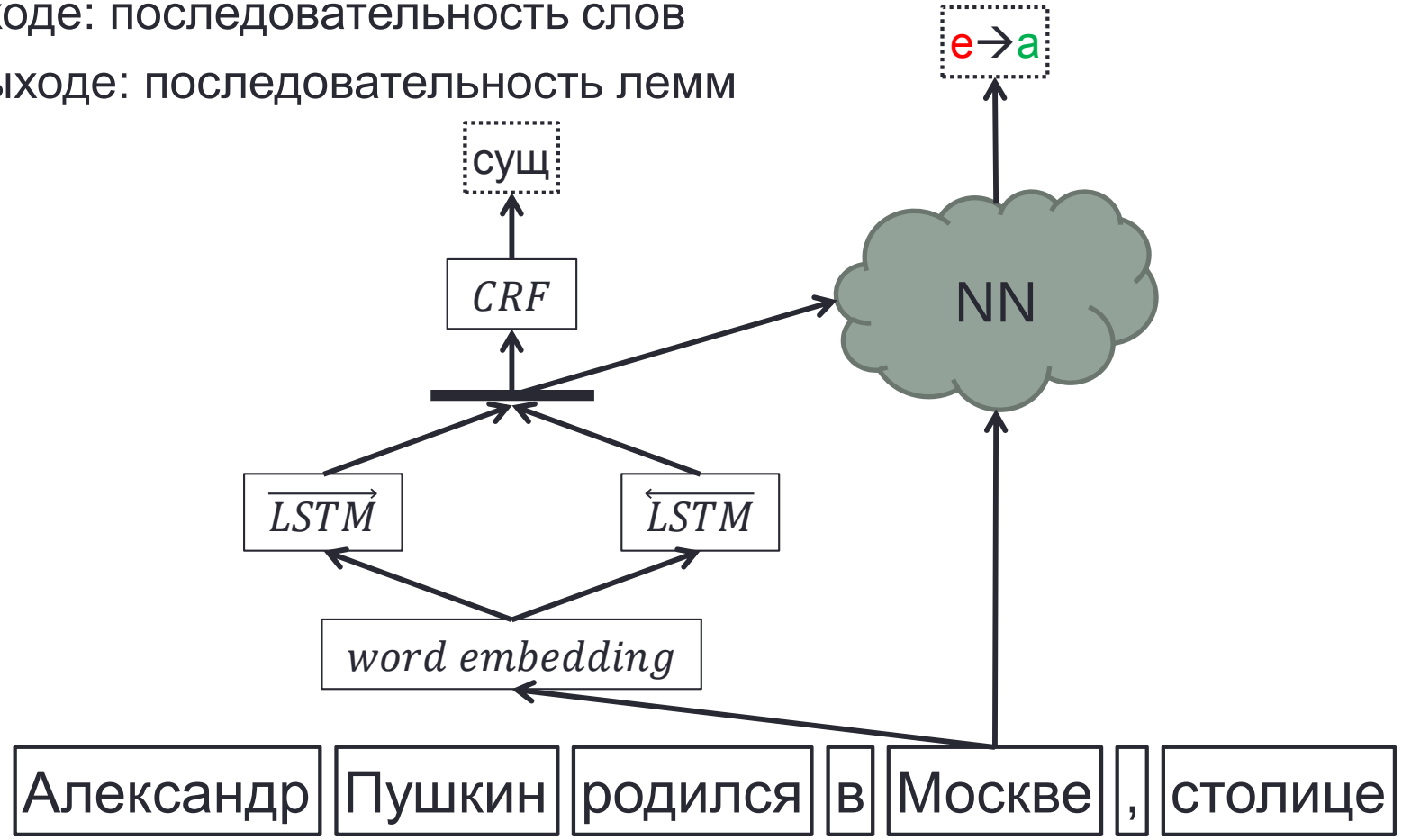
Нормализация (лемматизация) слов

- Задача
  - На входе: последовательность слов
  - На выходе: последовательность лемм
- Проблема морфологической омонимии



Нормализация (лемматизация) слов

- Задача
  - На входе: последовательность слов
  - На выходе: последовательность лемм



Оценка качества

- Классическая точность (Accuracy)

$$Accuracy = \frac{correct}{total};$$

- Точность среди неизвестных слов

$$Accuracy_{OOV} = \frac{correct_{OOV}}{total_{OOV}};$$

Следующая лекция

Синтаксический анализ