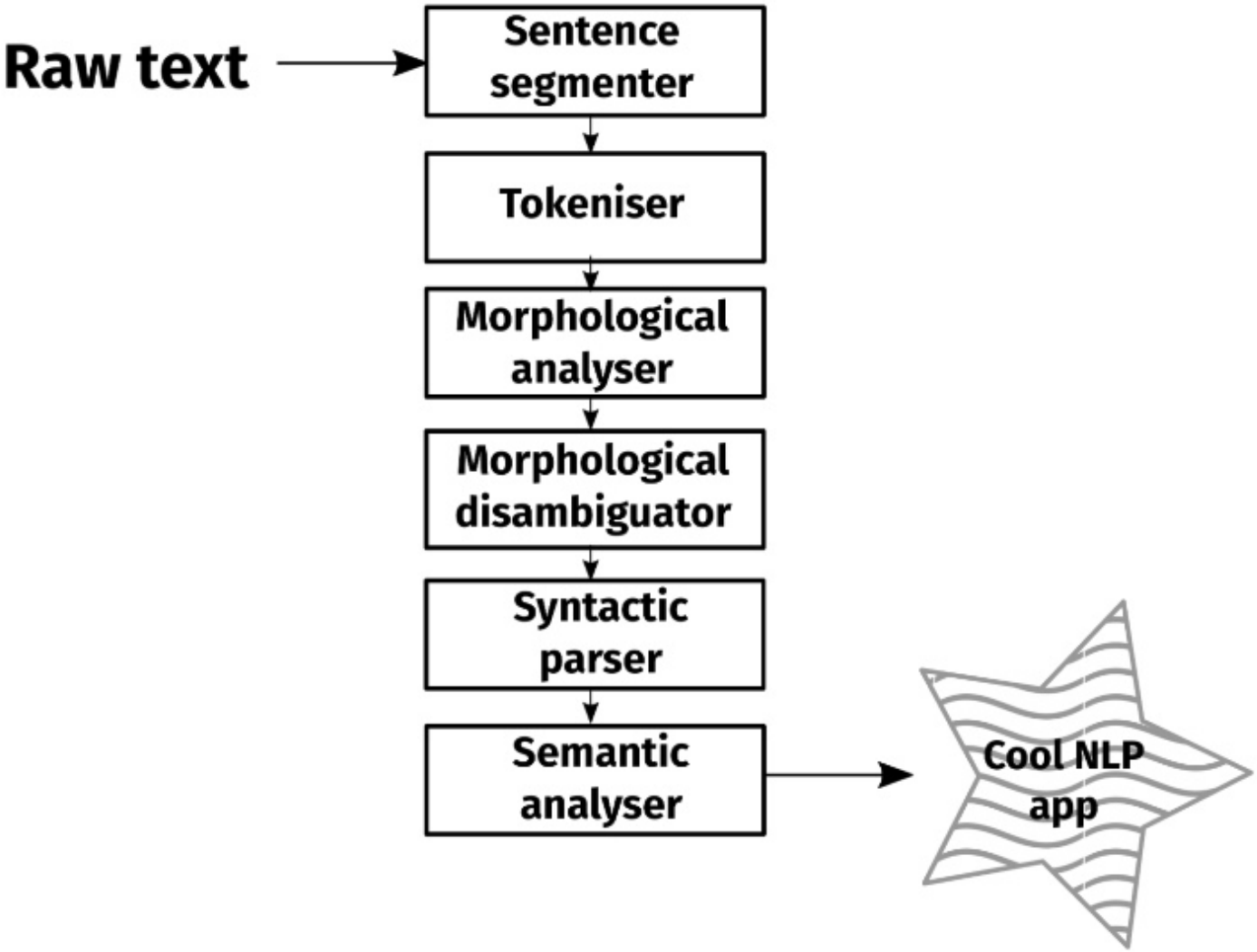
МОРФОЛОГИЧЕСКИЙ АНАЛИЗ

Маша Шеянова, masha.shejanova@gmail.com

November 20, 2018

НИУ ВШЭ

ШАГИ ПАЙПЛАЙНА



АНАЛИЗ

INTRO. ПОВТОРЕНИЕ.

У нас есть: текст, поделённый на токены.
Наша цель:

- каждый токен привести к начальной форме (ветров -> ветра); это называется **лемматизация**
- выяснить, к какой части речи он относится (существительное, глагол); это называется **POS-tagging**
- определить его грамматические характеристики (падеж, число, время)

Проблемы: тысячи их. Это вообще очень нетривиальная задача.
Применение:

- лемматизация — практически везде
- анализ — задачи, где грамматичиские признаки значимы

ЕЩЁ НЕМНОГО ТЕРМИНОЛОГИИ

- **стемминг** — нахождение основы слова; основа слова не обязательно совпадает с морфологическим корнем слова
- агглютинативность: каждое морфологическое значение обозначается одной морфемой — прийти-1sg-pst-fem (татарский)
- флективность: все морфологические смыслы стремятся уместиться в одну морфему — прийти-1sg.pst.fem (русский)
- синтетические языки: стараются выразить как можно больше значений в морфологии (адыгейский)
- аналитические языки: стараются выразить как можно больше значений словах и не трогать морфологию (английский, китайский)

Когда хорош стемминг, а не лемматизация?

Стемминг. Некоторая терминология.

Когда хорош стемминг, а не лемматизация?

- в аналитических языках типа английского — морфологии всё равно почти нет, некоторые слова появляются сразу в виде основ
- в языках, где морфология выражается в основном в суффиксах, нет нетривиальных явлений в середине слова
- где нет омонимии основ (агглютинативные склонны к такому куда меньше, чем флективные)

СТЕММИНГ VS ЛЕММАТИЗАЦИЯ

Некоторая терминология:

- **стемминг** — нахождение основы слова; основа слова не обязательно совпадает с морфологическим корнем слова
- агглютинативность: каждое морфологическое значение обозначается одной морфемой (прийти-1sg-pst-fem)
- флективность: все морфологические смыслы стремятся уместиться в одну морфему (прийти-1sg.pst.fem)

ПОДХОДЫ: СЛОВАРЬ

В анализаторе хранится словарь с парадигмами.

падеж	ед. ч.	мн. ч.
Им.	парадйгма	парадйгмы
Р.	парадйгмы	парадйгм
Д.	парадйгме	парадйгмам
В.	парадйгму	парадйгмы
Тв.	парадйгмой парадйгмою	парадйгмами

ПОДХОДЫ: МОРФОЛОГИЧЕСКИЙ ТРАНСДЬЮСЕР

Но что, если морфология **слишком** богатая?

В агглютинативных языках у одного слова может быть около сотни форм — словарь будет занимать слишком много места, большая часть — дублирование.

Будем использовать более умный способ хранения данных — **морфологический трансдъюсер**.

КОНЕЧНЫЙ АВТОМАТ: ФОРМАЛЬНО

Конечный автомат — математическая модель, у которой есть таблица переходов, текущее состояние автомата, стартовое состояние и заключительное состояние.

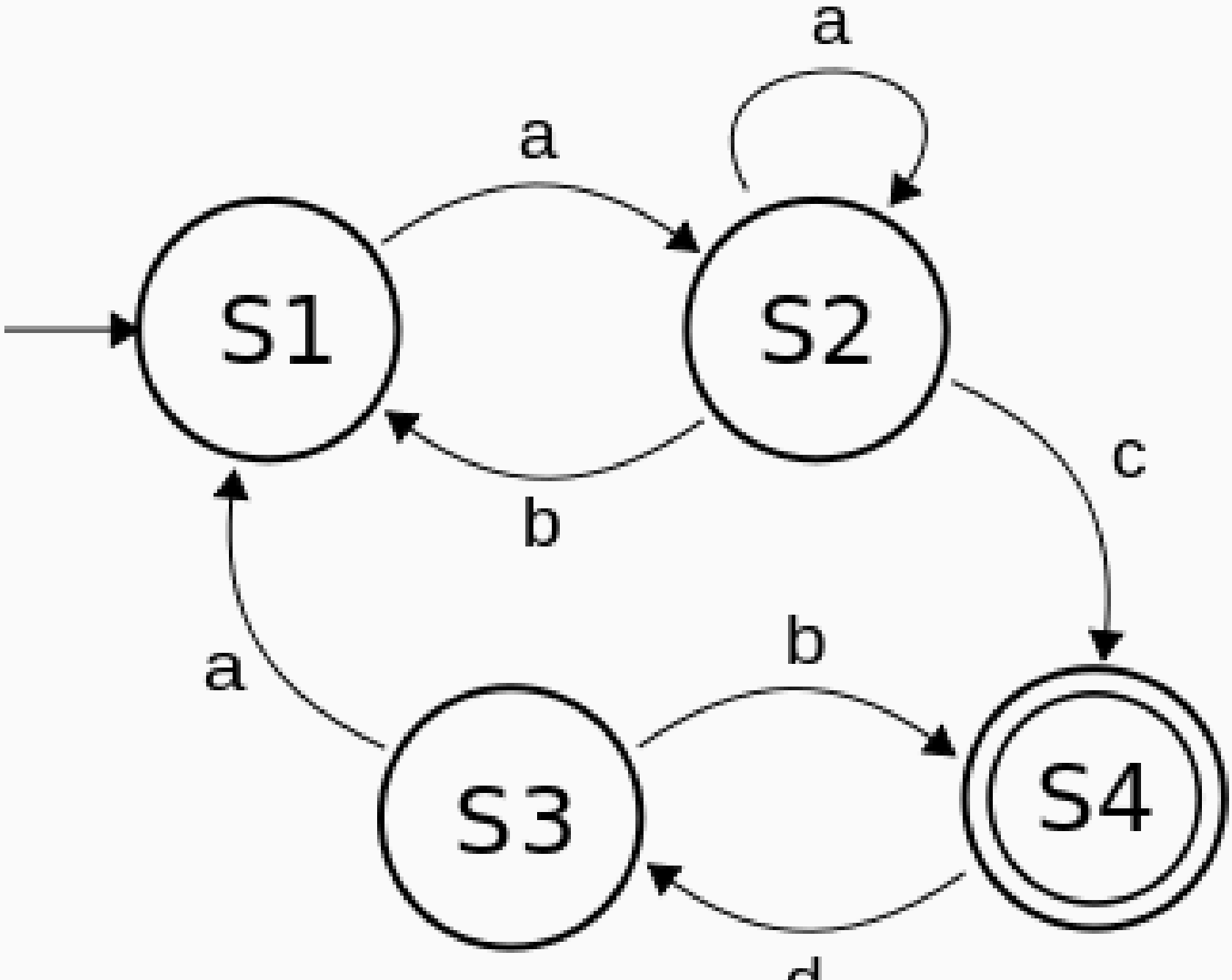
Таблица переходов — в ней хранятся переходы для текущего состояния и входного символа.

Текущее состояние — множество состояний в котором автомат может находиться в данный момент времени.

Стартовое состояние — состояние откуда КА начинает свою работу.

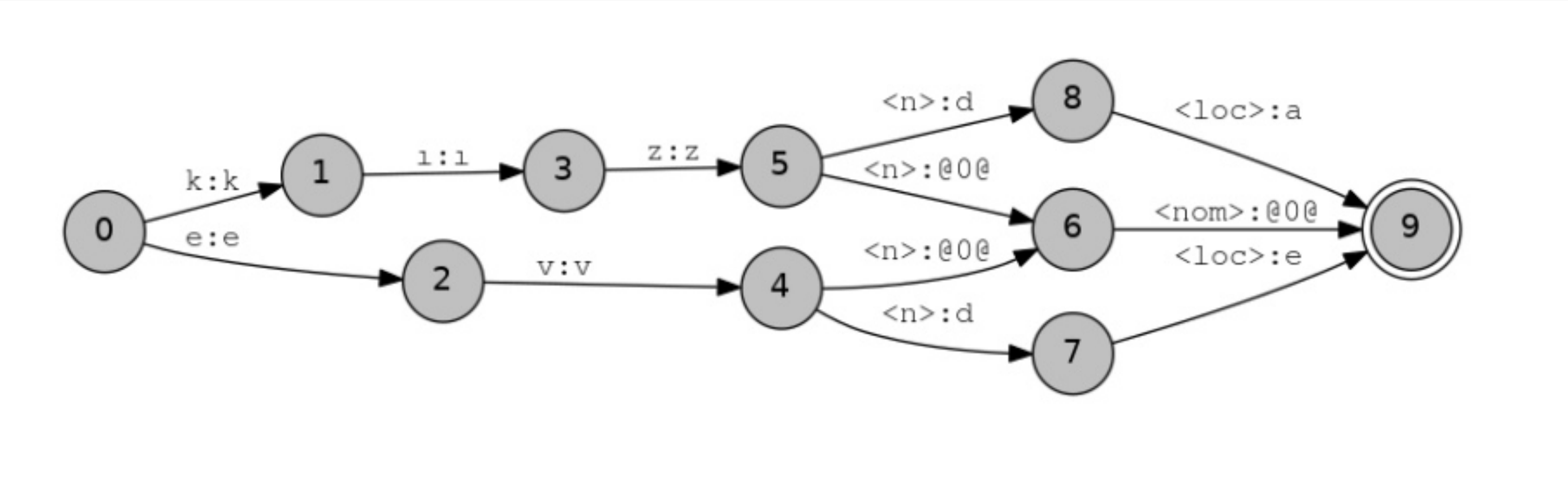
Заключительное состояние — множество состояний в которых автомат принимает определенную цепочку символов, в ином случае отвергает.

КОНЕЧНЫЙ АВТОМАТ: ПРИМЕР



КОНЕЧНЫЙ ТРАНСДЬЮСЕР: ЧТО ЭТО

Почти то же самое, но проме того, чтобы переходить по состояниям, он нечто ”порождает”.



МОРФОЛОГИЧЕСКИЙ ТРАНСДЬЮСЕР: СЛОЖНОСТИ

Трансдьюсеры хорошо справляются с агглютинативной морфологией. Но что если

- есть гармония гласных
- какое-то чередование на стыке морфем
- чередование внутри слова
- нетривиальная орфография?

Для всех этих случаев приходится придумывать обходные пути.

ДИЗАМБИГУАЦИЯ

ЗАЧЕМ?

Главная проблема морфологического анализа — омонимия:

```
lemma: сорок, analysis: NUMR loct, score: 0.285714
lemma: сорока, analysis: NOUN,inan,femn sing,nomn,
score: 0.142857
lemma: сорока, analysis: NOUN,anim,femn sing,nomn,
score: 0.142857
lemma: сорок, analysis: NUMR gent, score: 0.142857
lemma: сорок, analysis: NUMR datv, score: 0.142857
lemma: сорок, analysis: NUMR ablt, score: 0.142857
```

ЗАДАЧА

У нас есть: токены с кандидатами на анализ.
Наша цель: выбрать правильный анализ для каждого слова.

Морфологическая дизамбигуация — необходимый шаг после анализа, обычно морфологический анализ подразумевает и дизамбигуацию тоже.

подходы

- основанные только на частотности разбора (e.g. pymorphy)
- основанные на контексте (e.g. mystem)
  - правиловая — Constraint Grammar
  - скрытые марковские модели (HMM)
  - рекуррентные нейронные сети (RNN)

ИНСТРУМЕНТЫ

MYSTEM

Технология Яндекса, авторы Илья Сегалович и Виталий Титов.

Как устроен:

- **морфологический парсер** mystem работает на словаре Зализняка. В словаре приблизительно 100 тыс. слов русского языка с их полным морфологическим описанием (указаны морфологические парадигмы каждого слова)
- **неизвестные слова** анализируются по аналогии с наиболее похожими знакомыми словами
- выбор наиболее **вероятных разборов** с опорой на **контекст**

Подробнее про принцип работы – в статье.

Как пользоваться Для питона есть удобная обёртка: pymystem3.

MYSTEM

Особенности: у него есть своя токенизация.

Плюсы

- есть статистическая дизамбигуация по контексту
- умеет лемматизировать незнакомые слова
- в отличие от pymorphy, честно заявляет, что не знает этого слова (‘bastard‘)

Минусы

- дизамбигуация по контексту — только со своей токенизацией
- работает медленно, особенно под виндой, особенно если подавать уже токенизированный текст
- может неправильно лемматизировать незнакомые слова
- закрытый код
- тэги кириллицей :(
- зачем-то пробелы – тоже токены

PYMORPHY

Написан Михаилом Коробовым на питоне.

Как устроен:

- для парсинга использует словарь проекта OpenCorpora
- для анализа незнакомых слов – набор правил, работающих на суффиксах и окончаниях
- подбирает наиболее вероятный разбор по его частотности

Подробнее об устройстве – в статье.

PYMORPHY

Плюсы:

- работает быстро
- есть ранжирование разборов-кандидатов по вероятности
- открытый код

Минусы:

- нет дизамбигуации по контексту
- нет втстроенной токенизации, нужно подавать по слову

Спасибо за внимание!

Вопросы?