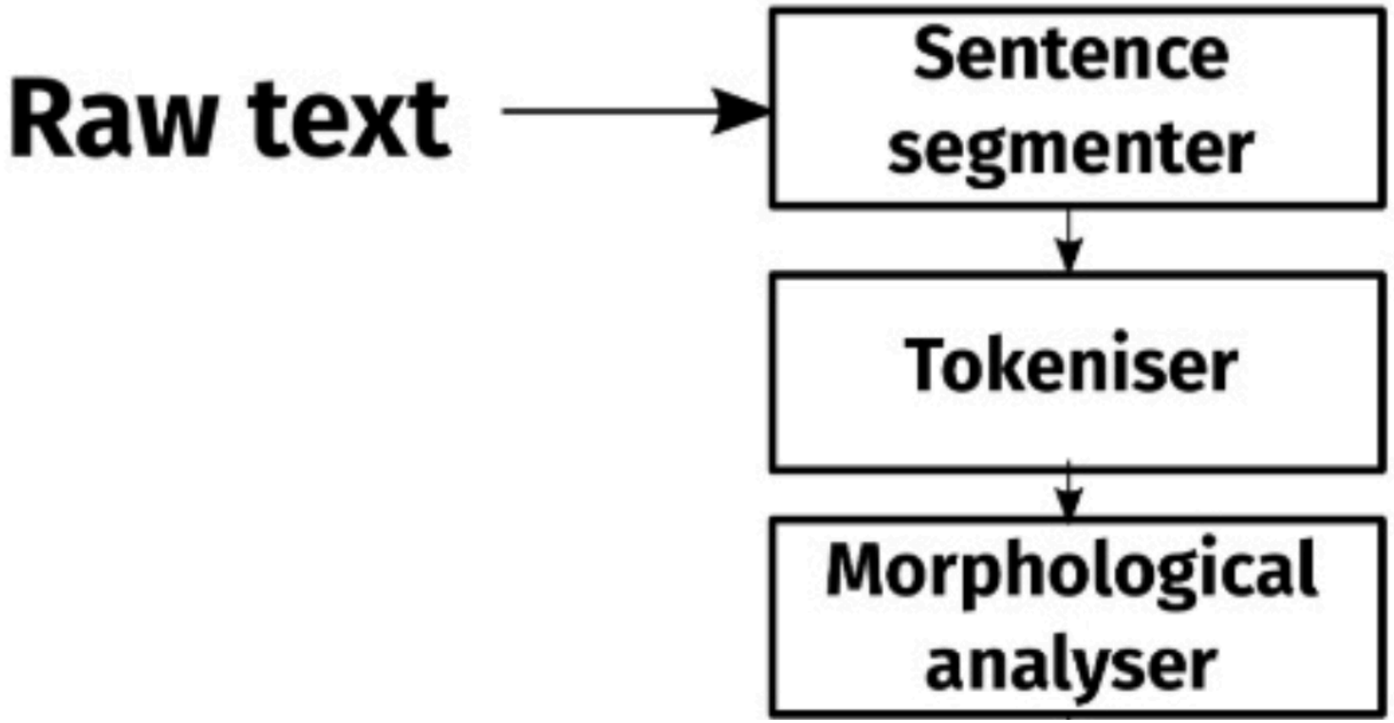
Морфологический парсинг

Занятие 5

Морфологический анализ



Морфологический анализ

Дано: текст, поделённый на токены.

Цель:

- Лемматизировать
- Разметить части речи (PoS-tagging)
- Определить грамматические характеристики токенов (features)

В чем сложность?

Омонимия, зависимость от контекста, OOV и слова, которых не было в обучающих данных, если у нас парсер с обучением и вообще как это реализовать?

Применение:

- лемматизация — практически везде
- морфо-теггинг — задачи, где морфо-признаки значимы

Словари парадигм

- Грамматический словарь Зализняка
- Леммы, полная парадигма, часть речи и грамматические характеристики.

Как быть с омонимией лемм/ словоформ?

```
53225:душ
душ NOUN Animacy=Inan|Case=Nom|Gender=Masc|Number=Sing
душами  NOUN Animacy=Inan|Case=Ins|Gender=Masc|Number=Plur
душам NOUN Animacy=Inan|Case=Dat|Gender=Masc|Number=Plur
душем NOUN Animacy=Inan|Case=Ins|Gender=Masc|Number=Sing
душу  NOUN Animacy=Inan|Case=Dat|Gender=Masc|Number=Sing
душа  NOUN Animacy=Inan|Case=Gen|Gender=Masc|Number=Sing
душ NOUN Animacy=Inan|Case=Acc|Gender=Masc|Number=Sing
душе  NOUN Animacy=Inan|Case=Loc|Gender=Masc|Number=Sing
душей NOUN Animacy=Inan|Case=Gen|Gender=Masc|Number=Plur
души  NOUN Animacy=Inan|Case=Nom|Gender=Masc|Number=Plur
души  NOUN Animacy=Inan|Case=Acc|Gender=Masc|Number=Plur
душах NOUN Animacy=Inan|Case=Loc|Gender=Masc|Number=Plur

53226:душа
душа  NOUN Animacy=Inan|Case=Nom|Gender=Fem|Number=Sing
душами  NOUN Animacy=Inan|Case=Ins|Gender=Fem|Number=Plur
душам NOUN Animacy=Inan|Case=Dat|Gender=Fem|Number=Plur
```

Разрешение омонимии

Дизамбигуация

<normal_form=мыть; word=мыла; pos=VERB; tag=Gender=Fem|Mood=Ind|Number=Sing|
<normal_form=мыло; word=мыла; pos=NOUN; tag=Case=Gen|Gender=Neut|Number=Sing;

Разрешение омонимии

Дизамбигуация

У нас есть токены с вариантами анализа (по словарю парадигм)

Как выбрать верный вариант?

- Посадить разметчиков (долго и дорого, довольно точно)
- Смотреть по частотности корпуса (pymorphy)
- Опираться на контекст

Дизамбигуация по контексту

- скрытые Марковские цепи = HMM (TnT-Parser)
- Машинное обучение - ранжирование кандидатов (MatrixNet в Mystem)
- рекуррентные нейронные сети (rnnmorph с MorphoRuEval2017)
- трансформеры (парсер Анастасьева с GramEval2020)

Pymorphy 2

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

Подробнее в статье автора (Михаила Коробова) на Хабре:

https://habr.com/ru/post/176575/

Pymorphy 2

Плюсы:

- работает быстро
- есть ранжирование разборов-кандидатов по вероятности
- открытый код, можно покопаться

Минусы:

- нет разрешения омонимии по контексту
- нет встроенной токенизации, подаем токены (с другой стороны, можно кастомизировать токенизацию)

Mystem

Как устроен:

- Морфологический парсер mystem работает на словаре Зализняка в 200 лемм (дополнен НКРЯ?). С 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, исп. МО

Подробнее про принцип работы – в статье: https://ext-cachev2-m9mts04.cdn.yandex.net/download.yandex.ru/company/iseg-las-vegas.pdf?lid=1519

Для питона есть удобная обёртка: **pymystem3**.

Mystem

У mystem 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:

- работает медленно
- закрытый код
- есть претензии к качеству морфоразбора и лемматизации (но решение-то очень старое)

TnT-parser

Trigrams’n’Tags

NOUN / VERB?

Это была гравюра на стали

$$\frac{NOUN * PREP * NOUN}{NOUN * PREP} > \frac{NOUN * PREP * VERB}{NOUN * PREP}$$

Попробуем посчитать вероятность 2 вариантов PoS-тэггинга для этого предложения.

Тагсеты

Наборы тэгов

Пример - тагсет mystem

A	прилагательное	падеж, число, форма, степень сравнения, род	горячий, холодный
ADV	наречие		кисло, сладко
ADVPRO	местоименное наречие		почему, поэтому
ANUM	числительное-прилагательное	падеж, число, род	первый, третий
APRO	местоимение-прилагательное	падеж, число, род	мой, твой
COMP	часть композита		
CONJ	союз		и, но
INTJ	междометие		ах, ну
NUM	числительное	падеж	двадцать, пять
PART	частица		бы, же
PR	предлог		в, на
S	существительное	род, число, падеж, одушевленность	гусь, топор
SPRO	местоимение-существительное	лицо, число, падеж	ты, вы
V	глагод	лицо, число, время, вид, репрезентация, залог, пере-	идти, смотреть

Тагсеты

Раньше: почти у каждого решения - свой тагсет. Как их сравнивать?

Pymorphy

```
PARTS_OF_SPEECH = frozenset([
    'NOUN',  # имя существительное
    'ADJF',  # имя прилагательное (полное)
    'ADJS',  # имя прилагательное (краткое)
    'COMP',  # компаратив
    'VERB',  # глагол (личная форма)
    'INFN',  # глагол (инфинитив)
    'PRTF',  # причастие (полное)
    'PRTS',  # причастие (краткое)
    'GRND',  # деепричастие
    'NUMR',  # числительное
    'ADVB',  # наречие
    'NPRO',  # местоимение-существительное
    'PRED',  # предикатив
    'PREP',  # предлог
    'CONJ',  # союз
    'PRCL',  # частица
    'INTJ',  # междометие
])
```

НКРЯ

Части речи

S — существительное (*яблоня, лошадь, корпус, вечность*)
A — прилагательное (*коричневый, таинственный, морской*)
NUM — числительное (*четыре, десять, много*)
ANUM — числительное-прилагательное (*один, седьмой, восьмидесятый*)
V — глагол (*пользоваться, обрабатывать*)
ADV — наречие (*сгоряча, очень*)
PRAEDIC — предикатив (*жаль, хорошо, пора*)
PARENTH — вводное слово (*кстати, по-моему*)
SPRO — местоимение-существительное (*она, что*)
APRO — местоимение-прилагательное (*который, твой*)
ADVPRO — местоименное наречие (*где, вот*)
PRAEDICPRO — местоимение-предикатив (*некого, нечего*)
PR — предлог (*под, напротив*)
CONJ — союз (*и, чтобы*)
PART — частица (*бы, же, пусть*)
INTJ — междометие (*увы, батюшки*)

MSD

- **N** — Существительное (Noun)
- **A** — Прилагательное (Adjective)
- **V** — Глагол (Verb)
- **R** — Наречие (Adverb)
- **W** — Предикатив (Predicate)
- **P** — Местоимение (Pronoun)
- **M** — Числительное (Numeral)
- **S** — Предлог (Adposition)
- **C** — Союз (Conjunction)
- **H** — Вводная конструкция (Parenthesis)
- **I** — Междометие (Interjection)
- **Q** — Частица (Particle)
- **X** — Остальное (Residual)

Universal Dependencies

- UD — не самое оптимальное решение для отдельных языков, но облегченный вариант разметки.
- UD не является идеальным формальным представлением для парсинга, но полезна для сравнительного сопоставления результатов парсинга в разных языках
- UD — эсперанто в мире разметки
- > 100 языков и > 200 корпусов с разметкой.

Корпуса UD для русского

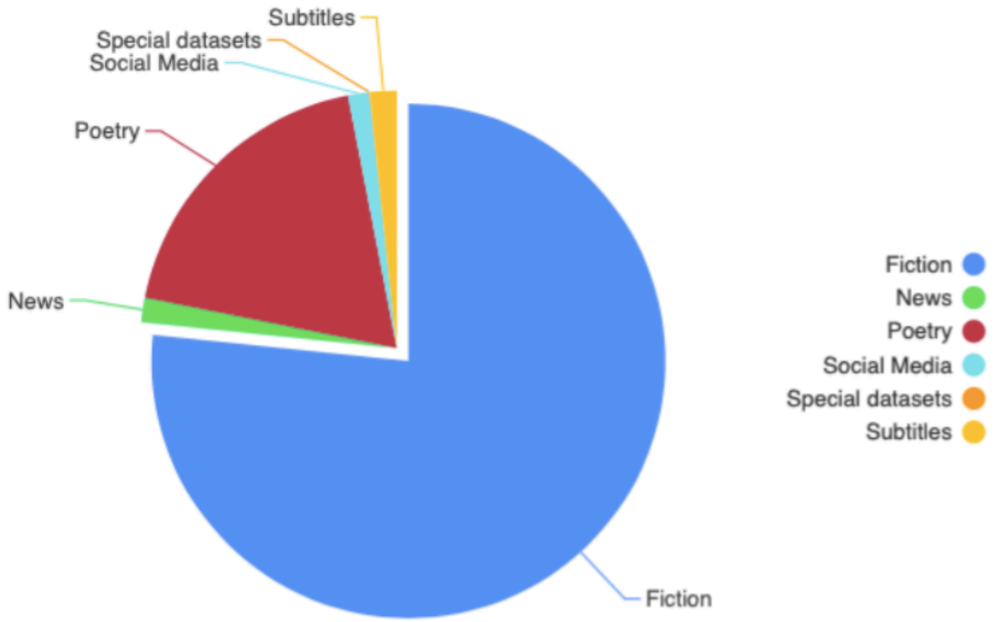
- SynTagRus (конвертированный)
- GSD (Русская Википедия)
- Тайга (Корпус Татьяны Шавриной и Ольги Ляшевской)
- ГИКРЯ - в скором времени.

Taiga UD

Корпус для машинного обучения

Размечен UDPipe

Genres	Tokens, millions	%
News	92	1.5
Literary Texts	4605	76
Special datasets	2.5	0.5
Social media	80	1.5
Subtitles	101	1.5
Poems	1130	19



Тагсет UD для русского

Universal Dependencies

POS Tags

ADJ – ADP – ADV – AUX – CCONJ – DET – INTJ – NOUN – NUM – PART – PRON – PROPN – PUNCT – SCONJ – SYM – VERB – X

Features

Animacy – Aspect – Case – Degree – Foreign – Gender – Mood – Number – Person – Polarity – Tense – Variant – VerbForm – Voice

Идем на https://github.com/dialogue-evaluation/GramEval2020/blob/master/UDtagset/UD-Russian_tagset.md смотреть тагсет.

RNNMorph

Morphological analyzer (POS tagger) for Russian and English languages based on neural networks and dictionary-lookup systems (pymorphy2, nltk).

Показал лучший результат на соревновании MorphoRuEval2017

https://github.com/IlyaGusev/rnnmorph

Domain	Full tag	PoS tag	F.t. + lemma
Lenta (news)	96.31%	98.01%	92.96%
VK (social)	95.20%	98.04%	92.06%
JZ (lit.)	95.87%	98.71%	90.45%
All	**95.81%**	**98.26%**	N/A

RNNMorph

Pip install rnnmorph (у меня еще надо указывать версию 0.4.0)

- работает из коробки
- медленный
- высокая точность, но можно лучше
- не учитывает синтаксис
- почти UD, но тагсет поменяли, а rnnmorph остался

Парсер Анастасьева

GramEval2020

- Сложная архитектура, есть на разных моделях (BERT, ELMo)
- SOTA
- Одновременно лемматизирует и делаем морфологический и синтаксический парсинг
- Работает долго
- Ошибки бредогенерации в леммах (нет словаря)
- Омонимию разрешает очень хорошо