HW1

20 февраля 2019 г.

In [1]:
```
import pymorphy2
import pymorphy2_dicts_ru
import xml.etree.ElementTree as etree
```

In [2]:
```
# морфологический анализатор
morph = pymorphy2.MorphAnalyzer()
```

In [3]:
```
# размеченный корпус OpenCorpora со снятой омонимией
file = etree.parse('annot.opcorpora.no_ambig_strict.xml').getroot()
```

In [4]:
```
# точность снятия омонимии по лемме
lemma_score = 0

# точность снятия омонимии по части речи
pos_score   = 0

# точность снятия омонимии по всем морфологическим характеристикам
all_score   = 0

#количество слов (токенов) с возможной омонимией в корпусе
words_amount = 0

# парсинг размеченного корпуса OpenCorpora
for text in file:
    paragraphs = text[1]
    for paragraph in paragraphs:
        for sentence in paragraph:
            tokens = sentence[1]
            for token in tokens:
                # текущее слово для обработки
                word = token.attrib['text']

                # результат обработки слова морфологическим анализатором
                parse_result = morph.parse(word)

                # если разбор слова анализатором однозначен, то его не обрабатываем
                # так как в этом случае нет омонимии
                if (len(parse_result) < 2):
```

```
                    continue

                # увеличение общего числа слов с возможной омонимией
                words_amount += 1

                # лемма текущего слова из корпуса
                lemma = token[0][0][0].attrib['t']

                # множетсво характеристик текущего слова из корпуса
                grammemes = set()
                gs = token[0][0][0]
                for g in gs:
                    grammemes.add(g.attrib['v'])

                # разрешение омонимии (бесконтекстное):
                # выбор наиболее вероятного набора морфологических характеристик
                params  = parse_result[0]

                # совпадают ли все морфологические характеристики, выданные анализатором
                all_eq = True

                # если омонимии по лемме не обнаружено, то увеличиваем соответствующий с
                if params.normal_form == lemma:
                    lemma_score += 1
                # иначе не совпадает лемма, выданная анализатором с леммой из корпуса
                else:
                    all_eq = False

                # если омонимии по части речи не обнаружено, то увеличиваем соответствую
                if params.tag.POS in grammemes:
                    pos_score += 1
                # иначе не совпадает часть речи, выданная анализатором с частью речи из
                else:
                    all_eq = False

                # поиск характеристик, выданных анализатором и не совпадающих с эталонны
                for g in grammemes:
                    if g not in params.tag.grammemes:
                        all_eq = False

                # если омонимии по всем характеристикам не обнаружено, то увеличиваем со
                if all_eq:
                    all_score += 1

# нормировка по числу слов (токенов)
lemma_score /= words_amount
pos_score   /= words_amount
all_score   /= words_amount
```

```
        # печать результатов
        print('Точность разрешения омонимии по лемме:', lemma_score)
        print('Точность разрешения омонимии по части речи:', pos_score)
        print('Точность разрешения омонимии по всем морфологическим характеристикам:', all_score
```

```
Точность разрешения омонимии по лемме: 0.9229637960119382
Точность разрешения омонимии по части речи: 0.9584757126173278
Точность разрешения омонимии по всем морфологическим характеристикам: 0.7088541891950344
```