МГУ им. М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра алгоритмических языков

Свойства метрик синтактико-семантического сходства предложений русского языка

Выполнил: Авагян Давид, гр. 624

Научный руководитель: к.ф.-м.н., доцент
Волкова Ирина Анатольевна

Москва, 2022

Постановка задачи

- Цель работы – построение и анализ *метрики* синтактико-семантического сходства предложений русского языка
- Подзадачи:
  - Выбор способа синтактико-семантического **представления** предложений русского языка, являющегося входом метрики
  - Построение и анализ **алгоритма** сопоставления двух предложений русского языка на основе выбранного синтактико-семантического представления
  - Анализ **свойств** предложенной метрики сходства предложений
  - **Реализация** предложенного алгоритма на языке Python 3

Актуальность задачи

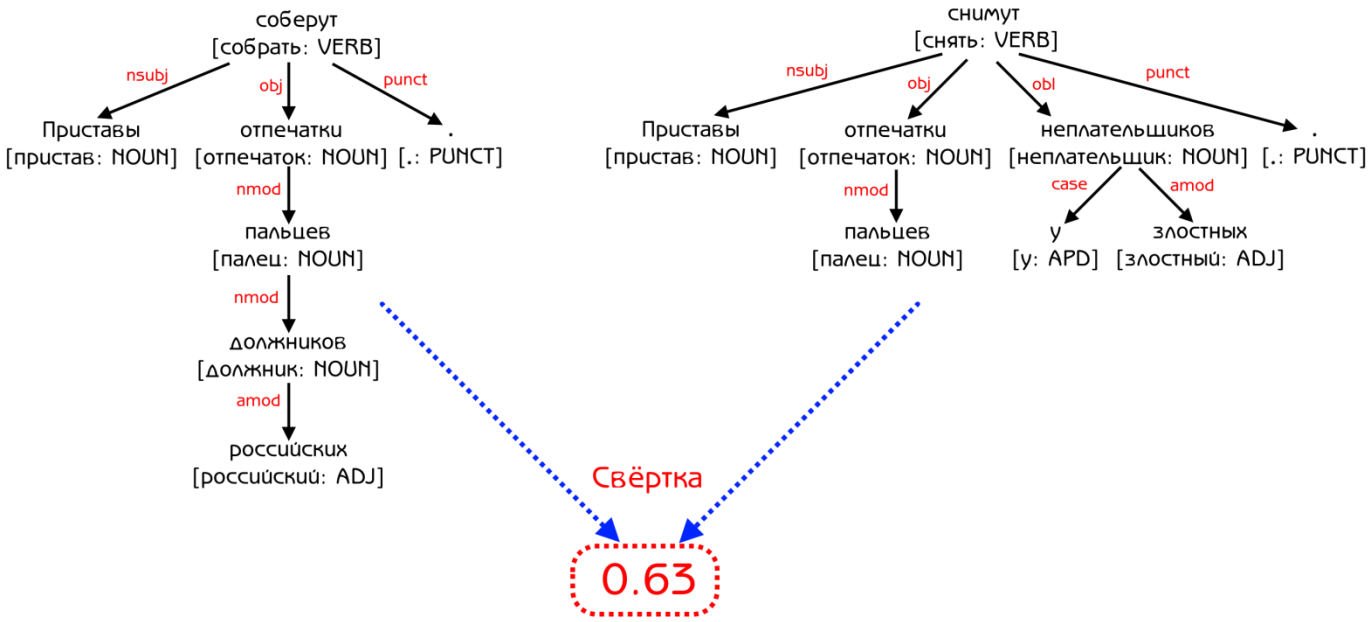
- Многие задачи обработки ЕЯ **сводятся** к определению сходства текстов
  - Дедупликация текстов, поиск плагиата, информационный поиск, обнаружение и генерация перифраз
- Значение метрики сходства может служить простым **признаком** для модели, решающей произвольную задачу КЛ
- Метрика может стать ядром алгоритмов *поиска плагиата* и перифраз
  - Замена тривиальных алгоритмов поиска n–грамм

- **Гипотеза** исследования: произвести *сопоставление* предложений в задаче поиска перифраз можно с высокой точностью, оперируя синтактико-семантической информацией об *устройстве* и *смысле* предложения

Сходство предложений

Приставы соберут отпечатки пальцев российских должников.

Приставы снимут отпечатки пальцев у злостных неплательщиков.



Простое ядро сопоставления (SABK)

$$SABK(T_1, T_2) = \frac{\sum_{e \in T_1} \sum_{\hat{e} \in T_2} \mathrm{sim}(e, \hat{e})}{|E_1| \cdot |E_2|}$$

$$\mathrm{sim}(e, \hat{e}) = \frac{s(e_d, \hat{e}_d) + s(e_h, \hat{e}_h)}{2} \times q(e_t, \hat{e}_t)$$

- $e = (e_h, e_d)$ – синтаксическая биграмма (ребро из e_h в e_d)
- sim – функция сходства биграмм
- s – функция сходства вершин
  - При наличии эмбеддингов – сходство векторов
  - Иначе мера Жаккара множеств граммем при совпадении лемм
- q – функция сходства синтаксических отношений
  - Матрица вида $E + \lambda I, \lambda \geq 0$

Ядро на основе TF-IDF (TABK)

$$TABK(T_1, T_2) = \frac{\sum_{e \in T_1} \sum_{\hat{e} \in T_2} \mathrm{sim}_t(e, \hat{e})}{N(T_1, T_2)}$$

$$\mathrm{sim}_t(e, \hat{e}) = \frac{\mathrm{weight}(e_d, \hat{e}_d) + \mathrm{weight}(e_h, \hat{e}_h)}{2} \times q(e_t, \hat{e}_t)$$

$$\mathrm{weight}(u, v) = \mathrm{tfidf}(u) \cdot \mathrm{tfidf}(v) \cdot s(u, v)$$

$$\mathrm{tfidf}(u) = \mathrm{tf}(u) \cdot \mathrm{idf}(u)$$

$$N(T_1, T_2) = \frac{S_1 \cdot S_2 + \left(\widehat{S_1} + \mathrm{tfidf}\left(\mathrm{root}(T_1)\right) \cdot \deg \mathrm{root}(T_1)\right) \cdot \left(\widehat{S_2} + \mathrm{tfidf}\left(\mathrm{root}(T_2)\right) \cdot \deg \mathrm{root}(T_2)\right)}{2}$$

Ядро сопоставления поддеревьев (MSK)

$$MSK(T_1, T_2) = \frac{\sum_{e \in T_1} \sum_{\hat{e} \in T_2} \widehat{\mathrm{sim}}(e, \hat{e})}{|E_1| \cdot |E_2|}$$

$$\widehat{\mathrm{sim}}(e, \hat{e}) = K_c(e_h, \hat{e}_h)$$

$$K_c(u, v) = \alpha \cdot s(u, v) + \nu \cdot \frac{\sum_{\hat{u} \in C_{T_1}(u)} \sum_{\hat{v} \in C_{T_2}(v)} K_c(\hat{u}, \hat{v}) \cdot q\left((u, \hat{u})_t, (v, \hat{v})_t\right)}{\left|C_{T_1}(u)\right| \cdot \left|C_{T_2}(v)\right|}$$

- $C_T(u)$ – множество потомков вершины u в дереве T
- $\alpha + \nu = 1$ – параметры баланса сходства вершин и поддеревьев

$$CK(T_1, T_2) = \beta \cdot TABK(T_1, T_2) + \delta \cdot MSK(T_1, T_2)$$

$$\beta + \delta = 1$$

Аксиомы метрики сходства

$s: X^2 \mapsto \mathbb{R}$ – метрика сходства на множестве X

- $s(x,y) = s(y,x) \ \ \forall x, y \in X$ – симметричность
- $s(x,x) \geq 0 \ \ \forall x \in X$ – неотрицательность
- $s(x,y) \leq s(x,x) \ \ \forall x, y \in X$
- $s(x,x) = s(y,y) = s(x,y) \Leftrightarrow x = y \ \ \forall x, y \in X$
- $s(x,y) + s(y,z) \leq s(x,z) + s(y,y) \ \ \forall x, y, z \in X$ – неравенство треугольника

- Если потребовать $SABK(T,T) \equiv 1 \ \ \forall T$, то ядро SABK – *метрика сходства* деревьев
- Необходимо также, чтобы s и q были метриками сходства
  - Тогда и функция sim окажется метрикой сходства
  - *Стандартные* реализации s и q удовлетворяют аксиомам метрики сходства

Сложность алгоритмов обработки деревьев

- Построение деревьев
  - Запуск анализаторов pymorphy2 и UDPipe + загрузка/вычисление эмбеддингов
  - Один обход дерева в глубину: конструирование и разметка
  - $O(|T|)$ времени и памяти на дерево
- Вычисление ядер свёртки деревьев
  - $O(|T_1| \cdot |T_2|)$ времени для всех ядер
  - $O(|T_1| + |T_2|)$ дополнительной памяти для SABK и TABK
  - $O(|T_1| \cdot |T_2|)$ дополнительной памяти для MSK и CK
- Эвристика отбора пар предложений для сопоставления двух текстов
  - Отбор k пар предложений из n за время $O\big((n+k)\log k\big)$ времени и $O(k)$ дополнительной памяти с помощью бинарной кучи

Недостатки и перспективы метрик

- Из-за требования $SABK(T,T) \equiv 1$ необходима **калибровка** значений SABK
  - Для немного отличающихся деревьев они склонны быть сильно *меньше* единицы
  - Замена функции суммирования функцией max может исправить положение
  - Нормировка метрики важна для *интерпретируемости* её значений
- Информация о сходстве **поддеревьев** доступна лишь в ядре MSK
  - Возможно сведéние к задаче поиска *изоморфизма* с максимальным сходством
    - Приём похож на замену суммирования максимизацией
  - Число синтаксических валентностей *ограничено*, поэтому сложность не возрастёт
  - Использование сходства *рёбер* будет затруднено при поиске изоморфизма
- Ядра **штрафуют** деревья с большим числом вершин
- Возможно расширение **семантической** составляющей представления

Заключение

- Построена *метрика* синтактико-семантического сходства предложений русского языка
  - Выбран способ синтактико-семантического **представления** предложений на основе *размеченного дерева зависимостей*
  - Предложены **алгоритмы** вычисления метрики сходства двух предложений
  - Проанализированы **свойства** предложенной метрики сходства, а также построенного алгоритма её вычисления
  - Алгоритмы **реализованы** на языке Python 3