

## РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ АВТОМАТИЧЕСКОЙ ОЦЕНКИ СВЯЗНОСТИ ПЕРЕВОДА

**Аннотация.** В статье рассматривается разработка приложения для оценки связности перевода. В работе описывается процесс сбора и разметки данных, выявление наилучшего по метрикам на тестовой выборке алгоритма машинного обучения для построения прогнозной модели связности перевода. В результате было разработано веб-приложение, в которое была встроена модель, реализованная на базе нейронной сети BERT.

**Ключевые слова:** обработка естественного языка, связность текста, языковая модель BERT, sentence-pair regression, ruwordnet.

**Введение.** Связность текста — это признак текста, при котором следующее предложение строится на базе предыдущего с помощью языковых средств [1]. Связность также является важным содержательным критерием, от которого зависит качество его реферирования [2] и сложность восприятия читателем [3]. Инструменты для оценки данного критерия могут найти применение в области обработки естественного языка, в частности при анализе сложности восприятия текста читателем. Существующие сервисы для английского языка используют различные подходы, включающие семантические сети (WordNet), дистрибутивную семантику (LSA, word2vec), эвристические подходы подсчета повторяющихся слов [4-5]. При этом сервисы для русского языка с открытым описанием их принципа работы используют только последний подход, не учитывая словосочетания производных слов, родовидовые и ссылочные отношения, которые также являются средствами связности [6].

В связи с этим требуется разработать приложение для оценки связности текста на русском языке, которое будет учитывать, кроме повторов слов, еще и словосочетания производных слов, родовидовые и ссылочные отношения.

**Материалы и методы.** Для анализа связности был сформирован собственный датасет на основе 3503 текстов публицистической и информационной направленности корпуса несовершенных переводов [7]. Тексты представляют собой студенческие переводы с английского на русский язык со средней длиной в предложениях 26,02, в словах — 437,30. Исходные тексты с помощью токенизатора библиотеки NLTK [8] были разбиты на предложения, затем перемешаны в случайном порядке, т. к. предложения в исходном порядке считаются связными, потому что формируют текст. На основе полученного набора предложений был составлен список пар соседних предложений объемом 91 169 элементов.

Для получения столбца целевой переменной были реализованы 5 методов поиска отношений связности между словами внутри одной пары предложений на языке Python. Каждый метод возвращает 1 при успешном поиске и 0 в случае неуспешного. Признаками наличия отношений [6] являются случаи, когда одно из слов для другого является:

- 1) видовым (гипонимом) —  $x_1$ ;
- 2) родовым (гиперонимом) —  $x_2$ ;
- 3) производным —  $x_3$ ;
- 4) словоформой —  $x_4$ ;
- 5) отсылкой для другого —  $x_5$ .

Общим этапом в работе реализованных методов является разбиение предложений на токены. Методы для нахождения признаков 1–3 используют библиотеку *ruwordnet* [9], предоставляющую интерфейс для работы с одноименным тезаурусом. Он хранит информацию о синсетах — группах синонимичных слов, бинарно соответствующих другим группам посредством разметки: «гипонимы–гиперонимы», «производные–производящие» и т. д. Для поиска синсетов токены были приведены к нормальным формам, исключены знаки пунктуации. Условием окончания работы методов являлось наличие нормальной формы конкретного токена в плоском списке синсетов соседнего предложения.

Признаки 4 — 5 рассчитываются с использованием модуля *stem* библиотеки *NLTK* и морфологического анализатора *rumorphy2* [10]. Для токенов соседних предложений с помощью алгоритма стемминга *Snowball* создаются 2 списка стемов (основ слов)  $t\_stem1$ ,  $t\_stem2$ . Если  $t\_stem1 \cap t\_stem2 \neq \emptyset$ , то метод возвращает 1, в противном случае — 0. Метод поиска возможной анафорической связи с помощью экземпляра морфологического анализатора *MorphAnalyzer* находит тэги частей речи, числа и рода для токенов без знаков пунктуации. Если в списке токенов второго предложения есть местоимение, совпадающее по роду и числу с существительным из первого списка, метод возвращает 1, в противном случае 0.

На основе признаков по формуле:  $\max(x_1, x_2, x_3, x_4, x_5)$  формируется значение целевой переменной связности. 2 предложения являются связными, если целевая переменная равна 1, и не являются связными, если — 0.

Итоговый набор содержит 13 002 пары предложений ввиду того, что при отборе учитывались только пары, в которых представлен только один из признаков или ни одного. Таким образом, в датасете 6501 пара имеет средство связности, другая половина — нет. Данные обучающей и тестовой выборки делятся в пропорции 80:20, для избежания предвзятости модели в пользу одного из классов, обе выборки были стратифицированы.

**Проблема исследования.** Дана выборка  $D = \{(p_1, y_1), (p_2, y_2), \dots, (p_n, y_n)\}$ ,  $n = 13\,002$ , элементы которой представляют пары предложений  $p = \{(s_1, s_2)\}$ , где  $s_1, s_2$  — два неповторяющихся предложения и бинарные оценки  $y \in \{0, 1\}$ .  $D$  разделяется на  $P_{train}, P_{test}, y_{train}, y_{test}$ . На входе модель для обучения получает набор векторных представлений  $\{(v_1^T, v_2^T)\}$  для  $\forall p \in P_{train}$ ,  $\dim(v_i^T) = m$ ,  $i \in [1, 2]$ ,  $m \in \mathbb{N}$  и набор оценок  $y_{train}$ . На выходе модель выдает набор вещественных оценок  $y_{pred}$ , определяющих характер взаимосвязи между значениями целевых переменных  $y_{test}$  и векторных представлений  $P_{test}$ , в котором каждая оценка принадлежит отрезку от 0 до 1. Необходимо построить модель машинного обучения, находящую неизвестное отображение  $y^*: T_{train} \rightarrow y_{train}$ .

**Результаты.** Для проведения сравнительного анализа (см. табл. 1) рассматривались модели машинного обучения на базе архитектуры *transformer* *BERT* [11] (*rubert-base-cased* [12] и *sbert\_large\_nlu\_ru*), градиентного спуска (*CatBoostRegressor*), байесовского подхода (*Bayesian Ridge*), метода *LASSO* (*LassoLarsCV*) и сетей с долгой краткосрочной памятью *LSTM*. Для векторизации данных обучающей и тестовой выборки применялись разные виды представлений, такие как: эмбединги *sentence transformers* исходной модели *rubert-base-cased*, *tf-idf*, усредненные по эмбедингам слов векторы *elmo*, *fasttext* и *word2vec*.

Сравнение качества обучения моделей на тестовой выборке

Название модели	Метрика MSE	Метрика MAE
rubert_base_cased	<b>0.167</b>	0.335
ai-forever/sbert_large_nlu_ru	0.171	<b>0.333</b>
BayesianRidge на эмбедингах elmo длиной 1024	0.216	0.431
LassoLarsCV на эмбедингах elmo длиной 1024	0.219	0.435
LassoLarsCV на эмбедингах sentence transformers модели rubert_base_cased	0.219	0.435
BayesianRidge на эмбедингах sentence transformers модели rubert_base_cased	0.217	0.435
CatBoostRegressor на эмбедингах tf-idf	0.227	0.439
CatBoostRegressor на эмбедингах fasttext	0.22	0.43
LSTM на эмбедингах word2vec	0.23	0.455

Наилучший результат по метрикам показала rubert\_base\_cased со следующими гиперпараметрами: число эпох — 1, скорость обучения — , размер батча — 8. Дообучение происходило всего на 1 эпохе, т. к. на последующих эпохах качество снижалось из-за переобучения.

Для демонстрации работы модели было разработано веб-приложение с использованием фреймворка Flask.

Фронтенд написан с использованием HTML, CSS и JS. Модель интегрируется в приложение с помощью библиотеки simpletransformers [13]. Взаимодействие с пользователем осуществляется через точки API посредством HTTP GET и POST запросов. Для получения предсказаний пользователь выбирает текстовый файл посредством диалогового окна, по кнопке “Загрузить файл” файл сохраняется в хранилище для его дальнейшего использования моделью. После нажатия “Получить предсказание” начинается процесс обработки данных моделью, после завершения на экран также выводятся средние значения признаков. Пользователь также может скачать отчет в формате csv. Интерфейс приложения представлен на рис. 1.

Оценка связности текста

Загруженный текст

Даниэль К. Эделсон, доктор философии  
 Каждый день, осознаем мы это или нет, каждый член нашего современного общества принимает судьбоносные

Результаты

#	Среднее значение связности для пар предложений текста	Среднее количество пар предложений с повторяющимися словоформами суц.	Среднее количество пар предложений с производной лексикой	Среднее количество пар предложений с гипонимами	Среднее количество пар предложений с гиперонимами	Среднее количество пар предложений с анафорой
Значение	0.96064	1	0.5714285714285714	0.5714285714285714	0.8571428571428571	0

Choose File RU\_1\_148\_1.txt

Загрузить файл

Рис. 1. Интерфейс приложения

**Заключение.** В данной статье описан подход для автоматической оценки связности для текста на русском языке. В дальнейшем планируется добавление новых признаков и доработка существующих методов. Также планируется валидация работы модели на текстах переводов студентов ТюмГУ направления «Лингвистика» и нахождение корреляций по оценкам с сервисом «Антиплагиат».

## СПИСОК ЛИТЕРАТУРЫ

1. Канакина В.П., Горецкий В.Г. Русский язык. 2 класс: учебник для общеобразовательных организаций: в 2 ч. Ч. 1 (ФГОС). — М.: Просвещение, 2017. — С. 17.
2. Белогорская Д.В., Резанова З.И. Лингвистическая оценка автоматически сгенерированных рефератов новостных текстов // Язык и культура. — 2023. — № 61. — С. 15-28. doi: 10.17223/19996195/61/2.
3. Соловьев В.Д., Вольская Ю.А., Андреева М.И., Заикин А.А. Словарь русского языка с индексами конкретности/абстрактности // Вестник РУДН. Серия: Лингвистика. — 2022. — № 2. — С. 515-549. — URL: <https://cyberleninka.ru/article/n/slovar-russkogo-yazyka-s-indeksami-konkretnosti-abstraktnosti> (дата обращения: 31.03.2024).
4. Graesser A.C., McNamara D.S., Louwerse, M.M., Cai Z. Coh-Metrix: Analysis of text on cohesion and language // Behavior Research Methods, Instruments, & Computers. — 2004. — Vol. 36. — Pp. 193-202. — URL: <https://doi.org/10.3758/BF03195564> (дата обращения: 31.03.2024).
5. Crossley S.A., Kyle K., Dascalu M. The Tool for the Automatic Analysis of Cohesion 2.0: Integrating semantic similarity and text overlap // Behavior Research Methods. 2019. Vol. 51. Pp. 14–27. <https://doi.org/10.3758/s13428-018-1142-4>.
6. Иванова Р.А. Когезия как текстоорганизующий признак // Когнитивные исследования языка. 2012. Вып. 13. С. 732-743.
7. Kutuzov A.B., Kunilovskaya M.A., Chepurkova A.Y., Oshepkov A.Y. Russian Learner Parallel Corpus as a Tool for Translation Studies // Компьютерная лингвистика и интеллектуальные технологии: труды XVIII Международной конференции «Диалог 2012»: в 2-х томах, Бекасово, 30 мая — 3 июня 2012 года. — Вып. 11 (18). Том 1. — Бекасово: Российский государственный гуманитарный университет. — 2012. — Pp. 362-369.
8. Bird S. NLTK: the natural language toolkit // Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions. — 2006. — С. 69-72.
9. Лукашевич Н.В., Лашевич Г., Герасимова А.А., Иванов В.В., Добров Б.В. Порождение тезауруса типа WordNet для русского языка // Труды конференции по искусственному интеллекту КИИ-2016. — 2016. — Т. 2. — С. 89-97.
10. Korobov M. Morphological Analyzer and Generator for Russian and Ukrainian Languages // Analysis of Images, Social Networks and Texts. — 2015. — Pp. 320-332. — URL: <https://pymorphy2.readthedocs.io/en/stable/misc/citing.html#citing> (дата обращения: 31.03.2024).
11. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // Proceedings of NAACL-HLT. — 2019. — С. 4171-4186.
12. Kuratov Y., Arkhipov M. Adaptation of deep bidirectional multilingual transformers for Russian language // Komp'juternaja Lingvistika i Intellektual'nye Tehnologii. — 2019. — С. 333-339.
13. ThilinaRajapakse. simpletransformers. — URL: <https://github.com/ThilinaRajapakse/simpletransformers> (дата обращения: 31.03.2024).