Задача ранжирования синонимов с учётом контекста

Манохин Антон Павлович

Санкт-Петербургский государственный университет
Прикладная математика и информатика
Вычислительная стохастика и статистические модели

Научный руководитель: к. ф.-м. н., доцент П. В. Шпилёв
Рецензент: к. ф.-м. н., руководитель ML-сервисов Yandex Cloud
В. А. Ершов, «Яндекс.Технологии»

Санкт-Петербург
2024 г.

Постановка задачи

Дано предложение на английском языке, в нём выделено слово. Для этого слова дан неупорядоченный набор синонимов-кандидатов.

Из набора необходимо определить 5 самых подходящих по контексту синонимов и **упорядочить их по релевантности**.

Быстрая коричневая лиса прыгает через ленивую собаку.
The quick brown fox jumps over the lazy dog.

1. sluggish (вялую)
2. supine (лежащую на спине)
3. slothful (нерадивую)
4. idle (праздную)
5. torpid (оцепенелую)

shiftless

...

Необходимые обозначения

Дано: (C, w_t, S_t) — запрос

- $C = (w_1, \ldots, w_m)$ — контекст (1 предложение)
- $w_t \in C$ — целевое слово для замены
- $S_t = \{s_1, \ldots, s_n\}$ — **неупорядоченный набор** синонимов-кандидатов для слова w_t

Хотим получить **упорядоченный набор** $\widehat{S}_t = (\widehat{s}_1, \ldots, \widehat{s}_k)$ элементов из S_t, где $\widehat{s}_1$ — самый релевантный синоним, $\widehat{s}_2$ — второй по релевантности и т. д.

Необходимые обозначения

Введём $rel_{C,w_t}(s_i) \in [0,1]$ — истинную релевантность синонима s_i (исходя из разметки аннотаторов) и отношение $s_i \succcurlyeq s_j$ — «синоним s_i истинно лучше, чем s_j»:

$$s_i \succcurlyeq s_j \;\Leftrightarrow\; rel_{C,w_t}(s_i) \geq rel_{C,w_t}(s_j)$$

Цель — восстановить порядок, установленный $rel_{C,w_t}(s_i)$.

Пусть $f : (C, w_t, s_i) \longmapsto r_{s_i} \in \mathbb{R}$ — ранжирующий алгоритм.

Зафиксируем C, w_t, S_t и введём обозначения:

- $\widehat{s}_j$ — j-ый синоним по убыванию $f(C, w_t, s_i),\ \forall s_i \in S_t$;
- s_j^* — j-ый синоним по убыванию $rel_{C,w_t}(s_i),\ \forall s_i \in S_t$.

Метрики оценки ранжирования

Фиксируем $k = \min\{5, |S_t|\}$ — то есть, необходимо не более 5 самых лучших синонимов.

Индикаторная функция попадания $\widehat{s}_i$ в топ-k:

$$\mathbf{1}_k(\widehat{s}_i) = \begin{cases} 1, & \widehat{s}_i \in (s_1^*, \ldots, s_k^*), \text{ где } s_1^* \succcurlyeq \cdots \succcurlyeq s_k^*; \\ 0, & \text{иначе.} \end{cases}$$

- *Precision@k* — доля релевантных в полученной выборке:

$$P_k = \frac{1}{k} \sum_{i=1}^{k} \mathbf{1}_k(\widehat{s}_i)$$

- *Average Precision@k* — среднее P_i по позициям i релевантных $\widehat{s}_i$:

$$AP_k = \frac{1}{k} \sum_{i=1}^{k} \mathbf{1}_k(\widehat{s}_i) P_i$$

Метрики оценки ранжирования

- *nDCG@k* — нормированная дисконтированная сумма выигрышей (Järvelin and Kekäläinen, 2002):

$$DCG_k = \sum_{i=1}^{k} \frac{2^{\left(rel_{C,w_t}(\widehat{s}_i)\right)} - 1}{\log_2(i+1)}$$

$$nDCG_k = \frac{DCG_k}{IDCG_k},$$

где $IDCG_k$ — значение DCG_k при ранжировании по истинным значениям (для $s_1^* \succcurlyeq \cdots \succcurlyeq s_k^*$)

Получение векторных представлений слов

Для тройки (C, w_t, s_i) получаем 2 предложения:

1. исходное предложение (контекст C),
2. исходное предложение с замененным словом w_t на s_i.

Предложения после токенизации независимо подаются на вход предобученной модели $\text{BERT}_{\text{BASE}}$ (Devlin et al., 2019).

Контекстуальные векторные представления слова w (Alammar, 2018):

$$B_C(w) = \frac{1}{m} \sum_{j:\, t_j \subset w} \sum_{i=9}^{12} E_{ij} \;\in\; \mathbb{R}^{768},$$

где $E_{ij} \in \mathbb{R}^{768}$ — значение i-го скрытого слоя модели в позиции j, соответствующей входному токену t_j, m — количество токенов в разбиении слова w.

Базовые подходы к ранжированию

- ***Random*** — случайное упорядочивание синонимов
- **CosSim** — сортировка по значениям *косинусного расстояния* между векторами контекстных представлений целевого слова и предлагаемого синонима

$$CosSim(a, b) = \cos(\widehat{a, b}) = \frac{a^T b}{\|a\| \cdot \|b\|}$$

Алгоритм:

1. находятся $B_C(w_t)$, $B_C(s_1)$, $\ldots$, $B_C(s_n)$;
2. вычисляется $r_{s_i} = CosSim\big(B_C(w_t), B_C(s_i)\big)$ для каждого $s_i \in S_t$;
3. получаем упорядоченное множество $\widehat{S}_t$ путём сортировки S_t по убыванию r_{s_i}.

Подход к ранжированию RankNet (Burges et al., 2005)

Определим $g_\theta : \mathbb{R}^{768} \to \mathbb{R}$ — нейросеть с параметрами (весами) θ.

Для каждого s_i определим вектор $\mathbf{x_i} = B_C(w_t) - B_C(s_i) \in \mathbb{R}^{768}$.

Используется для ранжирования вероятностная модель Bradley–Terry (1952):

$$\mathbb{P}(s_i \text{ лучше } s_j) = \frac{e^{g_\theta(\mathbf{x_i})}}{e^{g_\theta(\mathbf{x_i})} + e^{g_\theta(\mathbf{x_j})}} = \frac{1}{1 + e^{-(g_\theta(\mathbf{x_i}) - g_\theta(\mathbf{x_j}))}}$$

Подход к ранжированию RankNet (Burges et al., 2005)

Задача ранжирования сводится к минимизации следующего функционала:

$$Q(g_\theta) = \sum_{\substack{i,j: \\ s_i \succ s_j}} L\big(g_\theta(\mathbf{x_i}) - g_\theta(\mathbf{x_j})\big) \longrightarrow \min_\theta,$$

где $L(x) = \log(1 + e^{-x})$.

Функционал оптимизируется методом градиентного спуска:

$$\theta_r = \theta_r - \eta \sum_{\substack{i,j: \\ s_i \succ s_j}} \lambda_{ij} \Big(\frac{\partial g_\theta(\mathbf{x_i})}{\partial \theta_r} - \frac{\partial g_\theta(\mathbf{x_j})}{\partial \theta_r}\Big),$$

$$\text{где } \lambda_{ij} = -\frac{1}{1 + e^{g_\theta(\mathbf{x_i}) - g_\theta(\mathbf{x_j})}}$$

Затем синонимы сортируются по убыванию значений $g_\theta(\mathbf{x_i})$.

Подход к ранжированию LambdaRank (Burges, 2010)

Оптимизация достигается за счёт умножения λ_{ij} на модуль разности значения nDCG ($|\Delta nDCG_{ij}|$), полученный при перестановке $s_i \rightleftarrows s_j$.

Шаг градиентного спуска для оптимизации параметров θ будет выглядеть так:

$$\theta_r = \theta_r - \eta \sum_{\substack{i,j: \\ s_i \succ s_j}} \lambda_{ij} \left(\frac{\partial g_\theta(\mathbf{x_i})}{\partial \theta_r} - \frac{\partial g_\theta(\mathbf{x_j})}{\partial \theta_r} \right),$$

$$\text{где } \lambda_{ij} = -\frac{|\Delta nDCG_{ij}|}{1 + e^{g_\theta(\mathbf{x_i}) - g_\theta(\mathbf{x_j})}}.$$

Данные

- CoInCo (Kremer et al., 2014) — 15 629 слов, для которых синонимы предложены аннотаторами

  Оценки: количество аннотаторов, которые предложили данный синоним

- Swords (Lee et al., 2021) — 1132 слов, примеры взяты из CoInCo с неконтекстными синонимами из тезауруса

  Оценки: количество аннотаторов, которые отметили данный синоним как подходящий по контексту

Датасет	Кол-во контекстов	Кол-во целевых слов	Кол-во синонимов	Среднее кол-во синонимов на целевое слово
CoInCo	2474	15 629	112 742	**7.21**
Swords	1132	1132	68 683	**60.67**

Таблица: Сводные данные по датасетам

Результаты

Метод	mean P_k	mean AP_k	mean $nDCG_k$
Random	0.6801	0.3054	0.6398
CosSim	0.7220	0.3782	0.7712
RankNet	0.7244	0.3849	0.7954
LambdaRank	**0.7304**	**0.3951**	**0.8115**

Таблица: Средние значения целевых метрик при тестировании на CoInCo (test) для моделей, обученных на наборе данных CoInCo (dev).

Результаты

Проверка статистической значимости различия значений целевых метрик между LambdaRank и другими подходами с использованием парного критерия знаковых рангов Уилкоксона, уровень значимости $\mathbf{p = 0.05}$:

Метод	p-value для P_k	p-value для AP_k	p-value для $nDCG_k$
Random	$< \mathbf{0.01}$	$< \mathbf{0.01}$	$< \mathbf{0.01}$
CosSim	0.0536	$< \mathbf{0.01}$	$< \mathbf{0.01}$
RankNet	0.0813	**0.0165**	$< \mathbf{0.01}$

Таблица: Значения p-value для статистики Уилкоксона при сравнении значений метрик с LambdaRank (обучение и тест на CoInCo)

Результаты

Метод	mean P_k	mean AP_k	mean $nDCG_k$
Random	0.1209	0.0029	0.2273
CosSim	0.4047	0.1704	0.5521
RankNet	0.4108	0.1938	0.5798
LambdaRank	**0.4202**	**0.2076**	**0.6340**

Таблица: Средние значения целевых метрик при тестировании на Swords (test) для моделей, обученных на наборе данных Swords (dev).

Результаты

Проверка значимости различия значений целевых метрик между LambdaRank и другими подходами:

Метод	p-value для P_k	p-value для AP_k	p-value для $nDCG_k$
Random	$< \mathbf{0.01}$	$< \mathbf{0.01}$	$< \mathbf{0.01}$
CosSim	$< \mathbf{0.01}$	$< \mathbf{0.01}$	$< \mathbf{0.01}$
RankNet	$< \mathbf{0.01}$	$\mathbf{0.0448}$	$< \mathbf{0.01}$

Таблица: Значения p-value для статистики Уилкоксона при сравнении значений метрик с LambdaRank (обучение и тест на Swords)

Заключение

- Были реализованы подходы к ранжированию синонимов с учётом контекста на основе методов LambdaRank, RankNet, и CosSim
- Показано, что подход на основе LambdaRank значительно превосходит результаты случайной сортировки и статистически значимо улучшает результаты метрик в сравнении с CosSim и подходом на основе RankNet
- Исходный код предложенных алгоритмов представлен на платформе Zenodo (`DOI 10.5281/zenodo.11238430`).

Заключение

Спасибо за внимание!

Список литературы I

- Järvelin, Kalervo and Jaana Kekäläinen (окт. 2002). “Cumulated Gain-Based Evaluation of IR Techniques”. В: *ACM Trans. Inf. Syst.* 20, с. 422—446. DOI: 10.1145/582415.582418.
- Devlin, Jacob et al. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv: 1810.04805 [cs.CL].
- Alammar, Jay (2018). *The Illustrated Transformer [Blog post]*. URL: https://jalammar.github.io/illustrated-transformer/.
- Burges, Chris et al. (2005). “Learning to rank using gradient descent”. In: *Proceedings of the 22nd International Conference on Machine Learning*. ICML ’05. Bonn, Germany: Association for Computing Machinery, pp. 89–96. ISBN: 1595931805. DOI: 10.1145/1102351.1102363.
- Burges, Chris (июнь 2010). *From RankNet to LambdaRank to LambdaMART: An Overview*. Тех. отч. MSR-TR-2010-82. URL: https://www.microsoft.com/en-us/research/publication/from-ranknet-to-lambdarank-to-lambdamart-an-overview/.

Список литературы II

 Kremer, Gerhard et al. (2014). “What Substitutes Tell Us – Analysis of an 'All-Words' Lexical Substitution Corpus”. In: *Proceedings of EACL*. Gothenburg, Sweden, pp. 540–549. URL: http://www.aclweb.org/anthology/E14-1057.pdf.

 Lee, Mina et al. (June 2021). “Swords: A Benchmark for Lexical Substitution with Improved Data Coverage and Quality”. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pp. 4362–4379. URL: https://arxiv.org/abs/2106.04102.

 Schuster, Mike and Kaisuke Nakajima (2012). “Japanese and Korean Voice Search”. В: *International Conference on Acoustics, Speech and Signal Processing*, с. 5149—5152.

Данные

Используемые наборы данных имеют следующие пропорции разделения на обучающие и тестовые выборки:

Датасет	Контекстов	Целевых слов	Синонимов	Оценок аннотаторов
CoInCo (dev)	1577	10 179	67 814	98 950
CoInCo (test)	897	5450	44 928	68 496
Swords (dev)	370	370	22 978	121 938
Swords (test)	762	762	45 705	253 917

Таблица: Сводные данные по датасетам (размеры обучающих и тестовых выборок)

Токенизация

Предложение перед подачей в BERT разбивается на последовательность токенов методом WordPiece (Schuster and Nakajima, 2012).
В модели BERT используется словарь из 30 522 токенов.

Пример:

We use tokenizer before getting embeddings

↓

`[we, use, token, ##izer, before, getting, em, ##bed, ##ding, ##s]`

Результаты

Метод	mean P_k	mean AP_k	mean $nDCG_k$
CosSim	0.4047	0.1704	0.5521
L-Rank (CoInCo-trained)	0.3956	0.1618	0.5645
L-Rank (Swords-trained)	0.4202	0.2076	0.6340

Таблица: Средние значения целевых метрик при тестировании на Swords (test) для моделей LambdaRank, обученных на наборах данных CoInCo (dev) и Swords (dev).

Результаты

Распределение значений метрики nDCG для модели, обученной на основе подхода LambdaRank:

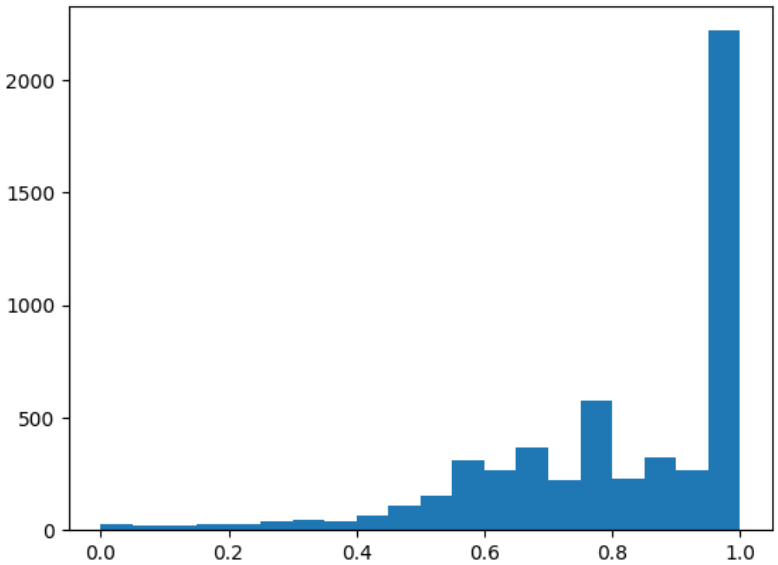

Рис.: обучение на CoInCo (dev), тестирование на CoInCo (test)

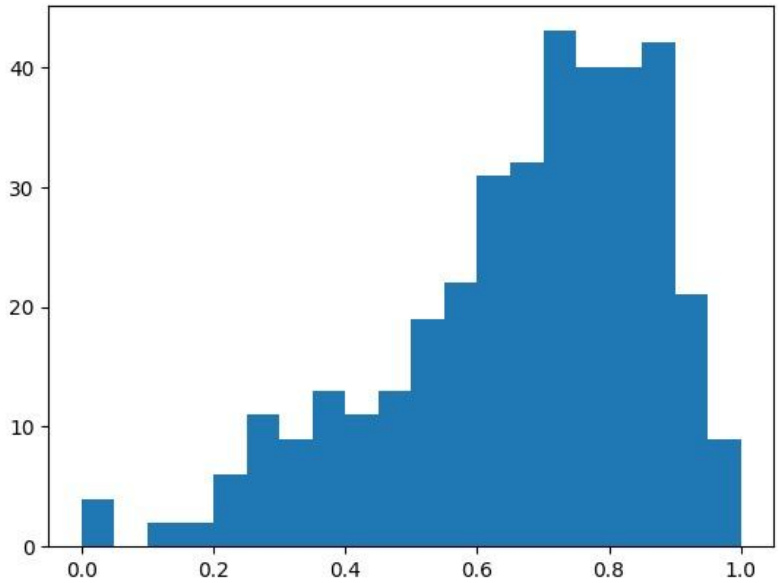

Рис.: обучение на Swords (dev), тестирование на Swords (test)