Исследование подходов для построения векторных представлений текстов в задаче сопоставления вакансий и резюме

Петрова Александра, 417 группа

Происхождение задачи

Задача **сопоставления вакансий и резюме** связана с необходимостью выбора работодателями кандидатов на основе большого количества резюме.

Актуальность задачи

Задача стала особенно актуальной с развитием интернет-платформ для поиска работы, где количество резюме может быть значительным. Работодатели сталкиваются с проблемой переполнения информации и неэффективного использования времени на поиск подходящих кандидатов.

Формальная постановка задачи

Пусть есть выборка вакансий V и резюме R, множество $\{\mathrm{R}_v\}_{v=1}^{l}$,
где R_v — множество релевантных резюме для вакансии v
Необходимо построить отображение $\mathrm{S}\colon V \to L^n$, L^n — множество упорядоченных списков резюме из R длины n

Такое что: $MAP@n = \frac{1}{l}\sum_{v=1}^{l} AP_v@n = \frac{1}{l}\sum_{v=1}^{l}\frac{1}{|R_v|}\sum_{k=1}^{n} I[L_v^n[k] \in R_v]\, P_v@k \to \max_S,$

где $P_v@k = \frac{\left|L_v^k \cap R_v\right|}{k}$, L_v^n — упорядоченный список резюме для вакансии v длины n

Обзор существующих методов

Job Recommendation Systems (JRS):

- Контентно-ориентированные системы (Content-Based JRS)
- Системы коллаборативной фильтрации (Collaborative Filtering JRS)
- Гибридные системы (Hybrid JRS)
- Системы на основе знаний (Knowledge-Based JRS)

В докладе будут рассматриваться методы, которые используют семантическое понимание и общий контекст, а не полагается на исторические данные или взаимодействия

Данные

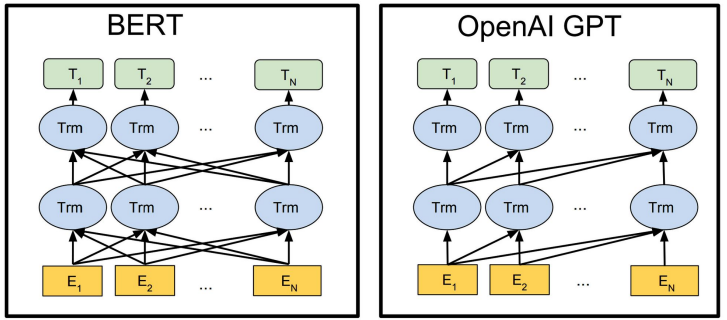
- Данные: набор вакансий и резюме на русском языке, предоставленный HR-отделом компании ACD/Labs.
- Для каждой вакансии известен перечень резюме кандидатов, приглашенных на собеседование. В выборке отсутствуют резюме, не связанные с конкретными вакансиями, и каждое резюме привязано к единственной вакансии
- Синтетические данные включают в себя пары соответствующих друг другу вакансий и резюме

Количество резюме	Количество вакансий	Описание
1	1	1 вакансия - 1 резюме
2	2	1 вакансия - 2 резюме
3	1	1 вакансия - 3 резюме
4	2	1 вакансия - 4 резюме
5	1	1 вакансия - 5 резюме
8	1	1 вакансия - 8 резюме
9	1	1 вакансия - 9 резюме
11	2	1 вакансия - 11 резюме
13	2	1 вакансия - 13 резюме
17	1	1 вакансия - 17 резюме
90	13	Total

Таблица 1: Распределение вакансий по количеству резюме

Метод решения

1. Предобработка текста, токенизация
2. Получение векторных представлений текстов с помощью моделей на архитектуре Transformer
3. Ранжирование на основе cosine similarity
4. Оценка качества: MAP@K



$$AP_v@K = \frac{1}{|R_v|} \sum_{k=1}^{K} 1\left[L_v^K[k] \in R_v\right] P_v@k$$

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Выбор моделей

Model	Parameters	Layers	Languages
bert-base-multilingual-cased	110M	12	104
bert-base-ru-cased	110M	12	Russian
paraphrase-MiniLM-L6-v2	22M	6	Multilingual
rugpt2large	774M	48	Russian
rugpt3large	760M	96	Russian
text-embedding-ada-002	-	-	Multilingual

Результаты экспериментов

	Random	BERT multilingual	BERT Russian	RuGPT2 Large	RuGPT3 Large	MiniLM	text-embedding-ada-002
MAP@10	0.03	0.1	0.11	0.11	0.13	0.41	0.58
MAP@20	0.04	0.13	0.15	0.14	0.15	0.44	0.66

Линейное преобразование

- Эмбеддер: text-embedding-ada-002
- Вектора вакансий и резюме попадают в разные области общего векторного пространства
- Для проверки наличия разрыва: максимальное среднее расхождение (MMD)
- Для сокращения разрыва: полносвязный линейный слой: матрица W размера nxn, где n - размерность векторов вакансий и резюме (нулевое смещение)

$$X, Y \text{ such that } k(X,Y) = \langle \phi(X), \phi(Y) \rangle_{\mathcal{F}}$$

$$\mu_p\left(\phi(X)\right) = [E[\phi(X_1], \cdots, E[\phi(X_m]]^T$$

$$MMD^2(P,Q) = \|\mu_P - \mu_Q\|_{\mathcal{F}}^2$$

$$MMD^2(P,Q) = E_P\left[k(X,X)\right] - 2E_{P,Q}\left[k(X,Y)\right] + E_Q\left[k(Y,Y)\right]$$

	MMD
(вакансии, резюме)	**0.087**
(вакансии, вакансии)	0.003
(резюме, резюме)	0.002

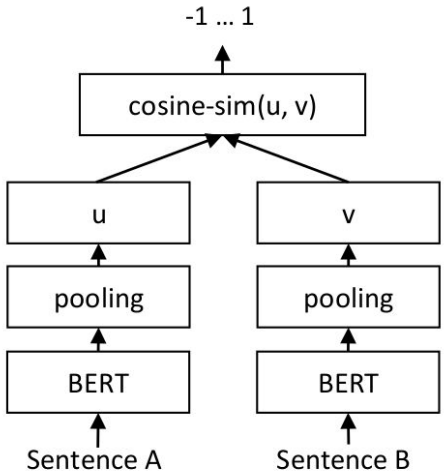
	MMD
(вакансии, резюме)	**0.003**
(вакансии, вакансии)	0.003
(резюме, резюме)	0.002

Fine-tuning

Fine-tuning Sentence-Transformer:

1. Сразу получаем вектор всего текста при помощи Sentence-Transformer
2. Сравниваем эмбеддинги (косинусное сходство).
3. Обучение через MultipleNegativesRankingLoss

$$\text{Loss} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(\text{sim}(\mathbf{u}_i, \mathbf{v}_i))}{\sum_{j=1}^{N}\exp(\text{sim}(\mathbf{u}_i, \mathbf{v}_j))}$$



Результаты экспериментов

	MiniLM	MiniLM (fine-tuning)	text-embedding-ada-002	text-embedding-ada-002 (linear transformation)
MAP@10	0.24	0.91	0.9	0.95
MAP@20	0.25	0.91	0.9	0.95

Выводы

- Были рассмотрены разные архитектуры моделей из семейства Transformer в качестве эмбеддеров и предложены методы для улучшения качества сопоставления.
- Базовые BERT и GPT продемонстрировали схожие показатели качества, однако BERT содержит меньше параметров и более эффективен для использования.
- Sentence Transformer превзошел базовые трансформеры, обеспечив более высокие результаты. Это объясняется его адаптацией для задач сравнения текстов, таких как семантическое сходство, благодаря обучению на парах предложений.
- Модель text-embedding-ada-002 продемонстрировала наилучший результат на экспертных данных.
- Линейное преобразование дало улучшение в качестве за счет уменьшения разрыва между областями векторов вакансий и резюме.
- Fine-tuning модели MiniLm из семейства Sentence-Transformer показал улучшение в качестве.
- В дальнейшем, можно рассматривать разные комбинации архитектур и методов.