Санкт-Петербургский государственный университет

Математика и компьютерные науки

Отчёт по научно-исследовательской работе (7 семестр)

Прогнозирование будущих заболеваний человека с использованием нейронных сетей типа трансформеры

Выполнила:

Майстер Анастасия Владимировна 20.Б13-мм

Научный Руководитель:

кандидат физико-математических наук

доцент кафедры теоретической кибернетики

Липкович Михаил Маркович

Кафедра теоретической кибернетики

Санкт-Петербург

2024

Постановка цели и задач

Данная работа является частью выпускной квалификационной работы.

Её **цель** заключается в обучении модели машинного обучения для прогнозирования будущих заболеваний пациента по его медицинской истории, в связи с чем на текущий семестр были поставлены следующие **задачи**:

1. выбрать датасет, подходящий для решения исходной задачи;
2. обучить языковую модель типа трансформер на выбранном датасете;
3. оценить результаты.

1 Введение

В последнее время всё больше информации переносится с бумажных носителей на электронные, и эта тенденция не могла не затронуть область медицины. Уже в 2017-м году более 79% больниц США использовали электронные отчеты лабораторных исследований [1]. Концепция электронных медицинских записей (Electronic Health Record, **EHR**) начала развиваться в 1970-х годах и с тех пор позволяет лучше отслеживать прогресс пациента и более эффективно назначать лечение [2].

EHR содержит информацию о поставленных диагнозах, выписанных лекарствах, датах вакцинации, аллергиях, медицинских снимках, результатах анализов, счетах за оказанные услуги и всём остальном, что касается пребывания пациента в медицинском учреждении.

Ввиду обилия разнородных данных и повсеместного распространения EHR стали предметом интереса исследователей в сфере прецизионной медицины. Ранние работы по применению глубокого обучения к EHR показали, что глубокие нейронные сети могут превосходить метод опорных векторов (*SVM*) и деревья решений в сочетании с ручным извлечением признаков в ряде задач прогнозирования на различных наборах данных [3]. Однако эти работы не учитывали тонкости данных EHR (например, то, что последовательность посещений имеет временной порядок и интервалы между посещениями могут быть неравномерными). В дальнейшем были предложены последовательные модели, основанные на рекуррентных [4] и сверточных [5] нейронных сетях.

В 2017-м году была предложена новая архитектура, основанная исключительно на механизме внимания и получившая название трансформер (*Transformer*). Она показала лучшие результаты в задаче машинного перевода при меньших затратах времени на обучение [6]. Архитектура состоит из энкодера и декодера. Энкодер представляет собой несколько слоёв с прямой связью и механизмом самовнимания (*self-attention mechanism*) и служит для получения эмбеддинга — векторного представления входных данных. Декодер подобным образом преобразует полученные эмбеддинги в выходные последовательности. Энкодер позволяет извлекать важные признаки и паттерны из данных, что играет большую роль при составлении прогнозов, а также находить связи между разными частями входных данных при помощи механизма самовнимания.

Возможности энкодера привели к появлению модели языковых представлений **BERT** (***B****idirectional* ***E****ncoder* ***R****epresentations from* ***T****ransformers*), которая оказалась эффективной для многих задач обработки естественного языка [7].

2 BERT

Языковая модель BERT была предложена в 2018-м году. Работа с ней состоит из двух этапов: предобучение (***pre-training***) и дообучение (или точная настройка, ***fine-tuning***). На первом этапе происходит обучение модели на непомеченных данных под разные задачи, а на втором — настройка на выбранных помеченных данных под конкретную задачу, для чего используется заранее предобученная модель.

С точки зрения архитектуры, модель является многослойным (12 слоёв) двунаправленным энкодером, то есть учитывает как левый, так и правый контекст.

Входными данными для модели являются последовательности — одно предложение или два предложения (вопрос и ответ). В начало последовательности всегда помещается специальный токен — [CLS], предложения между собой разделяются сепараторами [SEP]. При помощи алгоритма WordPiece с размером словаря 30000 токенов предложения преобразуются в числовые последовательности — эмбеддинги — заменой слова на номер соответствующего токена в словаре (*token embedding*).

WordPiece — один из популярных методов токенизации на подслова, состоящий из следующих шагов [8]:

1. Создаётся словарь, состоящий из всех символов в тексте (на котором обучается токенайзер).

2. В словарь добавляется новый токен, являющийся объединением двух токенов из словаря и обеспечивающий максимальное значение правдоподобия среди всех пар.

3. Предыдущий шаг повторяется, пока не будет достигнуто ограничение на максимальный размер словаря или значение правдоподобия не окажется меньше некоторого порогового.

Кроме того, ко входной последовательности добавляется обучаемый эмбеддинг для каждого токена, который показывает, к какому предложению принадлежит этот токен (*segment embedding*), а также позиционный эмбеддинг, соотвествующий номеру слова в предложении (*position embedding*). Затем три эмбеддинга складываются для получения одной последовательности.

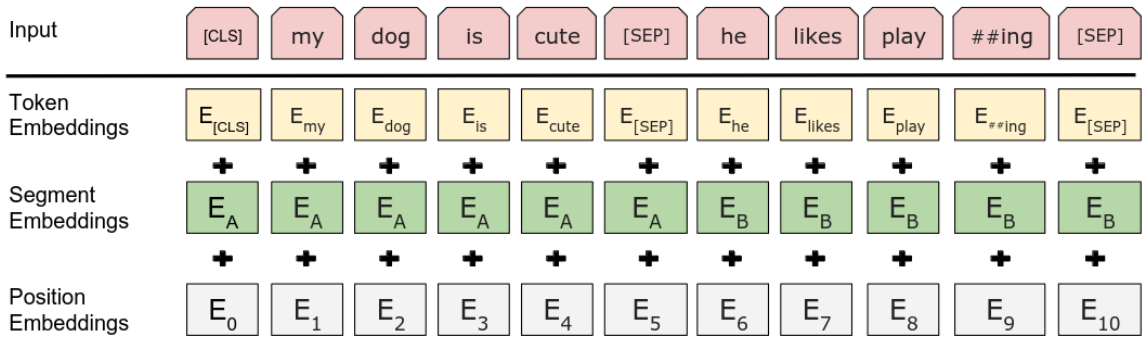

Рис. 1: Пример получения входной последовательности [7]

Предобучение модели осуществлялось на двух задачах — задаче прогнозирования замаскированного слова (*Masked Language Modeling*, ***MLM***) и задаче предсказания

следующего предложения (*Next Sentence Prediction*). Для обеих задач использовался BooksCorpus (800M слов) и тексты из английской Википедии (2500M слов).

Для обучения задаче MLM случайным образом выбирались 15% токенов из всего тренировочного датасета и каждый токен

- с вероятностью 80% заменялся на специальный токен [MASK];
- с вероятностью 10% заменялся на новый случайно выбранный токен;
- с вероятностью 10% оставался без изменений.

В качестве функции потерь использовалась кросс-энтропия (*Cross-entropy Loss*).

3 Обзор датасетов

Большой проблемой при использовании электронных медицинских записей исследователями являются ограничения, накладываемые на распространение, ввиду наличия персональной информации в данных. В связи с этим все датасеты, подходящие под поставленную задачу, можно разделить на два типа: синтетические (данные, сгенерированные по определённому закону) и содержащие реальные данные.

3.1 EHR-RelB [9]

Этот датасет 2020-го года состоит из 3630-ти пар понятий (каждое понятие — концепция — диагноз, симптом или жалоба со своим SNOMED ID[1]). Сначала были собраны сочетания по 2 концепции из того, что известно про пациента (из общего списка диагнозов, жалоб и прочего для каждого пациента), затем пары по всем пациентам были объединены в один список, из которого были выбраны наиболее частые пары, по 6 присутствий каждой концепции на весь датасет для большего охвата.

Для каждой пары даны 3 оценки наличия связи понятий по шкале 0-3, где 0 соответствует ситуации совершенно несвязанных концепций, а 3 — ситуации, при которой две концепции, с точки зрения медицины, всегда возникают вместе (например, алкогольная болезнь печени и цирроз печени). Степень связанности оценивали опытные доктора, по 3 человека из 5-ти привлечённых, коэффициент согласия оценщиков составил 0.63. Датасет можно считать сбалансированным в смысле распределения оценок связи понятий. Также приведены UMLS CUI для понятий в паре с целью сравнимости с уже существующими датасетами.

У этого датасета можно выделить следующие **достоинства**:

1. Собран из реальных анонимизированных EHR, охватывающих 5% населения Великобритании (*IQVIA Medical Research Data, IMRD*)
2. Уже собраны пары сопутствующих друг другу событий, не нужно делать это вручную
3. Свободный доступ

И **недостатки**:

1. Не все UMLS CUI заполнены (есть SNOMED ID, специфичные для Великобритании, не входящие в международный вариант)
2. Имеющийся уровень согласия оценщиков и наличие только трёх оценок не гарантируют правдивость полученных данных
3. Малый размер датасета (3630 строк)

	snomed_id_1	snomed_label_1	snomed_id_2	snomed_label_2	rater_A	rater_B	rater_C	rater_D	rater_E	mean_rating	CUI_1	CUI_2
0	13645005	Chronic obstructive lung disease	79955004	Chronic cor pulmonale	NaN	NaN	3.0	1.0	2.0	2.000000	C0024117	C0238074
1	102549009	Cramp in lower leg associated with rest	400047006	Peripheral vascular disease	NaN	1.0	NaN	2.0	2.0	1.666667	C0262578	C0085096
2	72866009	Varicose veins of lower extremity	274023003	Varicose vein operation	3.0	NaN	3.0	2.0	NaN	2.666667	C0155778	C0521235
3	109894007	Retained placenta	47821001	Postpartum haemorrhage	NaN	3.0	NaN	2.0	1.0	2.000000	C0242669	C0032797
4	161656000	H/O: regular medication	232346004	Allergy to cat dander	1.0	0.0	0.0	NaN	NaN	0.333333	C0455633	C0339805

Рис. 2: Датасет EHR-RelB [9]

[1]SNOMED ID — идентификатор диагноза/симптома/ситуации в медицинской номенклатуре SNOMED CT.

3.2 EMRBots [10]

EMRBots представляет собой базу данных **EMR** (*Electronic Medical Record*; в отличие от EHR, содержит информацию только из той медицинской организации, которой принадлежит, и не передаётся другим) на 100 (1.5MB), 10000 (140MB) и 100000 пациентов (1.4GB). Она содержит те же характеристики, что и реальные медицинские БД: сведения о поступлении пациентов, демографические и социально-экономические данные, лабораторные измерения, однако значения для каждого пациента генерируются случайно внутри заданного дипазона.

База данных состоит из 4 файлов в формате txt:

1. AdmissionsDiagnosesCorePopulatedTable.txt (диагноз, поставленный во время посещения),
2. AdmissionsCorePopulatedTable.txt (начало и конец посещения),
3. LabsCorePopulatedTable.txt (результаты лабораторных измерений),
4. PatientCorePopulatedTable.txt (пол, дата рождения, раса и прочее).

Достоинством этой базы данных является возможность её настройки: исходный код, написанный на C#, позволяет создать популяцию с любым распределением в плане пола, расы, значений лабораторных измерений (даны допустимые диапазоны). Однако сам автор пишет, что получаемые данные непригодны для оценки сценариев реальных исходов (например, прогноза заболевания), так как они не учитывают взаимодействие факторов во времени.

Поэтому, ввиду описанных особенностей, в том числе невозможности выявления причинно-следственных связей, датасет подходит для использования только в учебных целях.

3.3 MIMIC-IV v2.2 [11]

MIMIC-IV — реляционная база данных, состоящая из пяти модулей: hosp (данные из больничных EHR), icu (отделение реанимации), ed (отделение неотложной помощи), cxr (таблицы поиска и метаданные из MIMIC-CXR, позволяющие подключаться к MIMIC-IV) и note (клинические записи в свободной форме), управление которой осуществляется через PostgreSQL.

Модуль hosp содержит персональную информацию пациентов, результаты наблюдений, поставленные диагнозы, списки принимаемых лекарств и необходимых дозировок, выписанные рецепты, назначенные процедуры, информацию о переводе пациентов между разными отделениями. В icu содержатся данные об инъекциях, капельницах, анализах, настройках устройств и аппаратов и жизненных показателях пациентов, находящихся в отделении реанимации. Каждой из вышеперечисленных характеристик соответствует своя таблица (общая для всех пациентов).

Все данные являются анонимизированными, собраны с 40000 пациентов медицинского центра в США (Beth Israel Deaconess Medical Center) и относятся ко временному промежутку 2008-2019-го годов. При этом все имеющиеся даты сдвинуты на некоторое число дней вперёд (своё для каждого пациента), они согласованы по пациенту, но не согласованы по пациентам. Это обстоятельство вносит сложности в процесс использования базы данных, потому что возраст пациента на момент события определяется с точностью до трёх лет (для каждого пациента предоставляется трёхлетний диапазон, содержащий его реальный год рождения).

Кроме того, для получения доступа к базе необходимо пройти курс по защите

данных участников исследований на людях, подписать соглашение об использовании и получить одобрение.

Однако большое количество информации в разных форматах (текстовые, числовые, временные) и источник происхождения данных выгодно отличают эту БД от остальных.

3.4 Synthea [12]

Synthea — симулятор популяции, написанный на java. Он позволяет сгенерировать синтетическую популяцию с заданными параметрами, среди которых есть размер, минимальный/максимальный возраст, страна и город.

Симулятор основан на статистике и демографии (по умолчанию используются данные переписи населения штата Массачусетс, США), диагнозы и симптомы генерируются с помощью специальных модулей, которые описывают состояния и переходы между ними (пример можно увидеть в приложении — Рис. 6). Каждый пациент моделируется независимо от остальных, с рождения и до смерти.

4 Сборка датасета

Для данной работы с помощью симулятора Synthea была сгенерирована популяция из 10-ти тысяч человек с параметрами по умолчанию следующей командой (в папке с jar-файлом)

```
java -jar synthea-with-dependencies.jar –exporter.csv.export true -p 10000 -s 43
```

В результате было создано 18 файлов в формате csv:

File	Description
allergies.csv	Patient allergy data.
careplans.csv	Patient care plan data, including goals.
claims.csv	Patient claim data.
claims_transactions.csv	Transactions per line item per claim.
conditions.csv	Patient conditions or diagnoses.
devices.csv	Patient-affixed permanent and semi-permanent devices.
encounters.csv	Patient encounter data.
imaging_studies.csv	Patient imaging metadata.
immunizations.csv	Patient immunization data.
medications.csv	Patient medication data.
observations.csv	Patient observations including vital signs and lab reports.
organizations.csv	Provider organizations including hospitals.
patients.csv	Patient demographic data.
payer_transitions.csv	Payer Transition data (i.e. changes in health insurance).
payers.csv	Payer organization data.
procedures.csv	Patient procedure data including surgeries.
providers.csv	Clinicians that provide patient care.
supplies.csv	Supplies used in the provision of care.

Рис. 3: Результат работы симулятора [13]

Далее использовались только файлы 'conditions.csv' с результатами посещения врачей (диагнозы, временные метки) и 'patients.csv' с информацией о пациентах (дата рождения).

Датасет собирался следующим образом:

1. Были исключены все концепции, не являющиеся диагнозами, — ситуации (например, 'ожидает пересадки почки') и обнаруженные факты ('безработный', 'вовлечён в деятельность, связанную с риском').

2. Были исключены пациенты, на которых имелось менее пяти диагнозов во всём датасете (файл 'conditions.csv'), с целью выявления более сложных зависимостей.

3. Диагнозы по каждому пациенту были отсортированы по времени постановки (от старых к новым), а затем был сформирован файл в формате txt с последовательностями диагнозов, разделённых сепаратором [SEP], в котором каждая строка соответствует какому-то пациенту (в качестве диагнозов использовались их SNOMED ID).

Сначала сепаратором разделялись диагнозы в рамках одного визита, но, так как часто в один визит пациенту выставлялся только один диагноз, было решено разделять диагнозы по годам (то есть диагнозы, выставленные в разные визиты, имевшие место в один год, не разделяются).

Пример получившейся последовательности:

307426000 [SEP] 271737000 [SEP] 195662009 195662009 [SEP] 195662009 444814009 [SEP] 444814009

Также был сформирован аналогичный файл, но без сепараторов.

Выбор использования данного симулятора объясняется его гибкостью, разнообразием данных (которые можно будет использовать в дальнейшем) и свободным доступом. Несмотря на то что полученный датасет относится к категории синтетических, он является достаточным на начальном этапе. После получения рабочей модели целесообразно переходить на датасет, содержащий реальные анонимизированные данные.

5 Реализация модели

Для решения поставленной задачи использовалась предобученная модель BERT по той причине, что последовательности диагнозов имеют большое сходство с предложениями на естественных языках: как в языке перестановка слов меняет смысл предложения, так в последовательности перестановка диагнозов меняет причинно-следственные связи. Кроме того, некоторые болезни чаще возникают в детском/взрослом возрасте и, соответственно, оказываются в начале/конце последовательностей аналогично словам в предложении.

Хотя для каждой концепции со своим SNOMED ID существует словесное описание, было решено использовать именно коды, чтобы не получить несуществующие диагнозы (когда соединяются слова из описаний разных диагнозов).

По этой же причине в качестве алгоритма токенизации использовался не WordPiece, а WordLevel — последовательности разделялись на токены по пробелам, а затем все токены помещались в словарь с определённым номером. Также были добавлены специальные токены: [UNK] (вставляется на месте токена, не входящего в словарь, если такой впоследствии возникает), [PAD] (паддинг), [CLS] (обозначает начало последовательности), [SEP] (разделитель), [MASK] (токен-маска). Общий размер словаря составил 123 токена.

Собранный датасет был разделён на две части: 80% случайно выбранных последовательностей попали в тренировочный датасет (5165 штук), а остальные 20% — в тестовый. Затем, после токенизации, по каждому полученному датасету все последовательности были собраны в одну и разбиты на блоки размером 32 токена с целью сокращения времени, затрачиваемого на дообучение.

По правилу, описанному выше для задачи MLM, некоторые токены были заменены на токен [MASK]. Версия 'bert-base-uncased' (не различающая заглавные и строчные буквы) модели BERT из библиотеки transformers сообщества Hugging Face была обучена на 20-ти эпохах со значением параметра скорости обучения (*learning rate*), равным $5 \cdot 10^{-5}$, согласно рекомендациям авторов модели [7].

Результаты оценивались с помощью двух метрик — accuracy (отношение правильно угаданных токенов к числу всех замаскированных) и $F1$:

$$F1 = 2\frac{precision \cdot recall}{precision + recall}$$

$$precision = \frac{TP}{TP + FP}, \qquad recall = \frac{TP}{TP + FN},$$

где значение $F1$ считается по каждому токену, а затем усредняется с учётом весов, отвечающих количеству присутствий токена в целевом векторе.

TP (*True Positive*) — число случаев, когда объект был верно классифицирован, как принадлежащий положительному классу.

FP (*False Positive*) — число случаев, когда объект был классифицирован, как принадлежащий положительному классу, хотя принадлежит отрицательному.

FN (*False Negative*) — число случаев, когда объект был классифицирован, как принадлежащий отрицательному классу, хотя принадлежит положительному.

При этом для каждого токена положительным классом считаем этот токен, а отрицательным — все остальные.

Step	Training Loss	Validation Loss	Accuracy	F1	Randombaseline accuracy	Baseline accuracy
400	3.064000	2.674225	0.338095	0.267885	0.014286	0.150000
800	2.428800	2.313627	0.402778	0.346803	0.009838	0.148727
1200	2.234100	2.121341	0.437780	0.392809	0.005045	0.145179
1600	2.089800	1.996113	0.460202	0.414149	0.012332	0.142377
2000	1.971100	1.936542	0.484447	0.438685	0.009217	0.139401
2400	1.935700	1.850466	0.491525	0.450351	0.005085	0.146893
2800	1.836100	1.904641	0.497490	0.453483	0.008924	0.149470
3200	1.812700	1.879042	0.489784	0.440053	0.008173	0.156451

Рис. 4: Результаты дообучения модели

Значение accuracy для полученной модели составляет 0.49, что в несколько раз превышает соответствующее значение для модели, выдающей в качестве прогноза для всех пациентов наиболее часто встречающийся диагноз ('Baseline accuracy' в таблице), и тем более для модели, выдающей каждый раз случайный диагноз ('Randombaseline accuracy'). Значение F1 — 0.44.

Как было выявлено впоследствии, собранный датасет не является сбалансированным: по распределению диагнозов на Рис. 5 можно заметить, что болезни с номерами 5, 6 и 7 встречаются гораздо чаще, чем остальные (составляют более 30%).

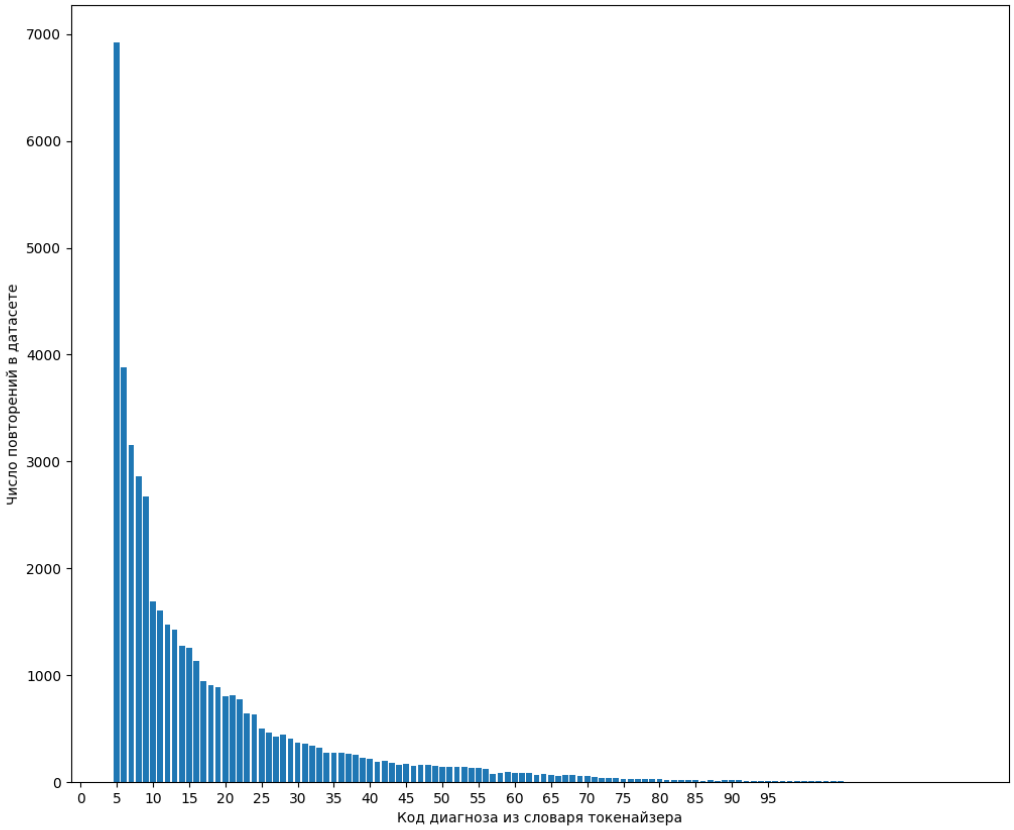

Рис. 5: Распределение диагнозов в тренировочном датасете

Это обстоятельство можно использовать для улучшения качества прогнозирования — например, учитывать веса токенов при подсчёте значения функции потерь (веса обратно пропорциональны частотам). Другим направлением развития является усложнение модели. Например, можно учитывать возраст пациента на момент постановки диагноза, а не только порядок диагнозов в медицинской истории.

Результаты

В рамках данной работы в 7-м семестре

1. был проведён обзор подходящих под тематику наборов данных, а также собран датасет непосредственно под задачу;
2. была обучена языковая модель на основе BERT, которая показала accuracy 0.49 на тестовом датасете и превзошла более простые решения.

Код доступен на GitHub.

6 Приложение

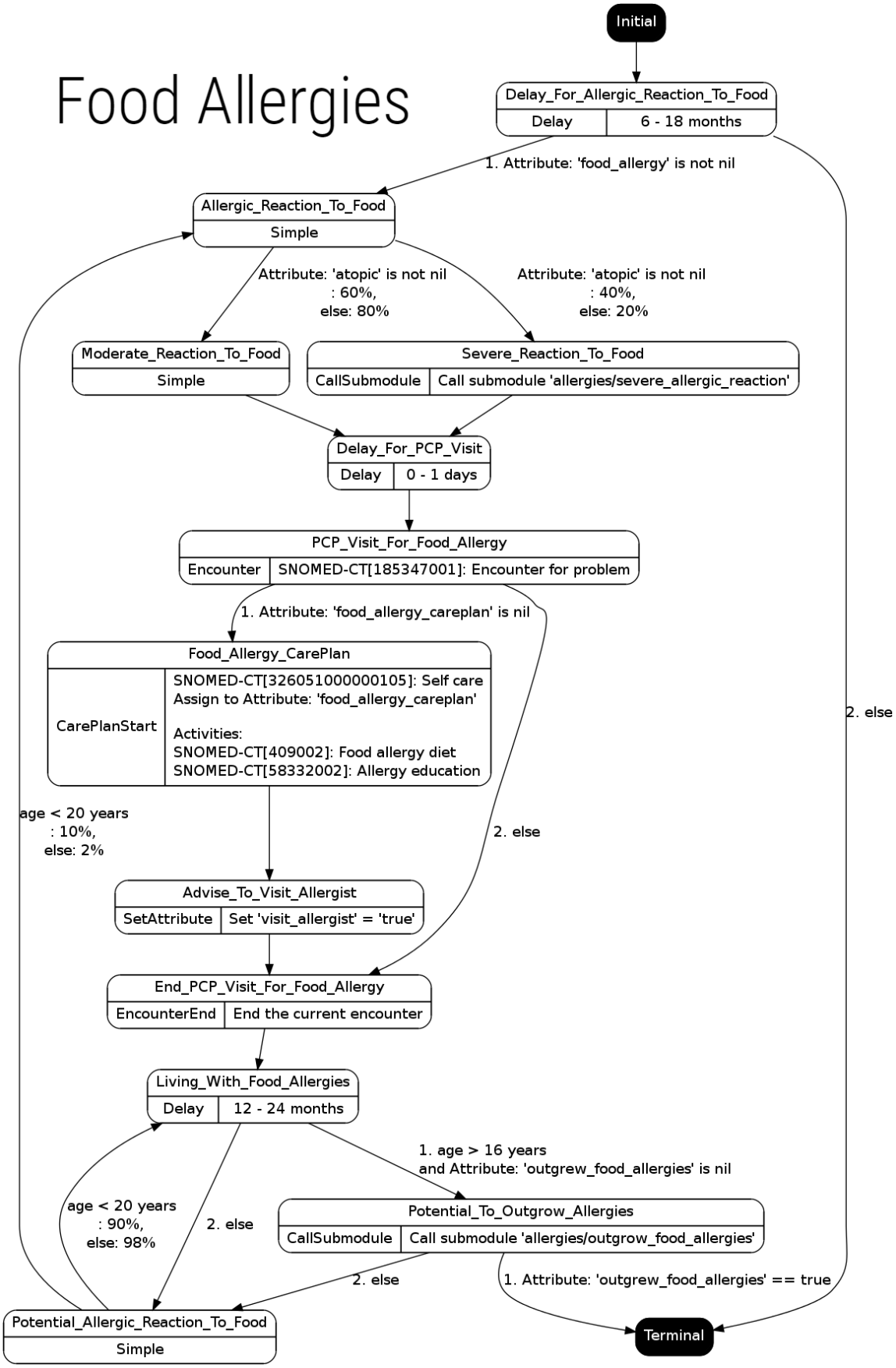

Рис. 6: Пример модуля, отвечающего за пищевые аллергии [12]

Список литературы

[1] Smith L. ELR in Virginia. — 2018. — Access mode: https://www.healthit.gov/sites/default/files/2018-12/ElectronicPublicHealthReporting.pdf.

[2] Boyles O. The History of Electronic Health Records. — 2019. — Access mode: https://www.icanotes.com/2019/04/16/a-history-of-ehr-through-the-years/#1970.

[3] BEHRT: Transformer for Electronic Health Records / Li Y., Rao S., Solares J. R. A., Hassaine A., Canoy D., Zhu Y., Rahimi K., and Salimi-Khorshidi G. — 2019. — Access mode: https://arxiv.org/abs/1907.09538.

[4] Doctor AI: Predicting Clinical Events via Recurrent Neural Networks / Choi E., Bahadori M. T., Schuetz A., Stewart W. F., and Sun J. — 2015. — Access mode: https://arxiv.org/abs/1511.05942.

[5] Deepr: A Convolutional Net for Medical Records / Nguyen P., Tran T., Wickramasinghe N., and Venkatesh S. — 2016. — Access mode: https://arxiv.org/abs/1607.07519.

[6] Attention Is All You Need / Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser L., and Polosukhin I. — 2017. — Access mode: https://arxiv.org/abs/1706.03762.

[7] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding / Devlin J., Chang M.-W., Lee K., and Toutanova K. — 2018. — Access mode: https://arxiv.org/abs/1810.04805.

[8] Schuster M., Nakajima K. Japanese and Korean Voice Search. — Access mode: https://static.googleusercontent.com/media/research.google.com/ru//pubs/archive/37842.pdf.

[9] Biomedical Concept Relatedness – A large EHR-based benchmark / Schulz C., Levy-Kramer J., Assel C. V., Kepes M., and Hammerla N. — 2020. — Access mode: https://arxiv.org/abs/2010.16218.

[10] Kartoun U. Advancing informatics with electronic medical records bots (EMRBots). — 2019. — Access mode: https://www.researchgate.net/publication/336079387_Advancing_informatics_with_electronic_medical_records_bots_EMRBots.

[11] Medical Information Mart for Intensive Care. — Access mode: https://physionet.org/content/mimiciv/2.2/.

[12] Synthetic Patient Population Simulator. — Access mode: https://synthetichealth.github.io/synthea/.

[13] Synthea — GitHub Wiki. — Access mode: https://github.com/synthetichealth/synthea/wiki/CSV-File-Data-Dictionary.