Софийски Университет "Св. Климент Охридски"

Факултет по математика и информатика
Катедра "Софтуерни технологии"

ДИПЛОМНА РАБОТА

на тема

Автоматично откриване на събития в текст

Дипломант: **Елена Георгиева Тупарова**
Специалност: **Информатика**
Магистърска програма: **Извличане на информация и откриване на знания**
Факултетен номер: **4MI3400132**

Научни ръководители:
проф. д-р Иван Койчев
доц. д-р Светла Бойчева

София, октомври 2023

Съдържание

Списък с фигури **5**

Списък с таблици **7**

1 Увод **8**

1.1 Актуалност на проблема и мотивация 8

1.2 Цел и задачи на дипломната работа 9

1.3 Очаквани ползи от реализацията 9

1.4 Структура на дипломната работа 10

2 Обзор на подходи за откриване на събития в текст **12**

2.1 Основни понятия 12

2.1.1 Понятия, свързани със събития 12

2.1.2 Общи понятия 14

2.1.3 Метрики за оценка 15

2.2 Стандартни корпуси от данни 16

2.2.1 *ACE2005* корпус от данни 16

2.2.2 *MAVEN* корпус от данни 17

2.2.3 *FedSemcor* корпус от данни 18

2.2.4 *WikiEvents* корпус от данни 19

2.3 Подходи за автоматично откриване на събития в текст 19

2.3.1 Парадигми за решаването на подзадачите 20

2.3.2 Начини за извличане на активиращата дума и аргументите . . . 22

2.3.3 Техники за извличане на събития 24

2.3.4 Обобщение на подходите за автоматично откриване и извличане на събития 29

3 Събиране на данните и подготвяне на анотиран корпус **31**

3.1 Събиране и анализ на данните от база от данни с опровержения на фалшиви новини 31

3.2 Дефиниране на нови типове събития в данните 35
3.2.1 Методология за дефиниране на нови типове събития 35
3.2.2 Тип събитие *Cure-Claim* 35
3.2.3 Тип събитие *Severe-Weather* 36
3.2.4 Тип събитие *Rule-Change* 37
3.3 Анотиране на корпус от данни за новите типове събития 38
3.3.1 Избор на система за ръчно анотиране на данните 39
3.3.2 Подбор на документите за анотиране 40
3.3.3 Разработване на указания за анотиране 40
3.3.4 Ръчно анотиране на данните 41
3.4 Характеристики на новия корпус от данни 41

4 Обучение на модели за откриване на събития в текст 43
4.1 Методи за допълнително обучение (*fine-tuning*) на модели за автоматично откриване на събития в текст 43
4.1.1 Избор на модел за допълнително обучение за автоматично откриване на новите типове събития от *EXTEND* 43
4.1.2 Особености на избрания за допълнително обучение модел 44
4.1.3 Подход за допълнително обучение на модела 45
4.2 Методи без предварително обучение, използващи големи езикови модели 46
4.2.1 Избор на голям езиков модел 47
4.2.2 Подход без предварително обучение 47

5 Проведени експерименти и анализ на резултатите от тях 49
5.1 Експерименти с допълнително обучените *Text2Event* модели 49
5.1.1 Първоначално проучване за осъществимост върху типа *Cure-Claim* 49
5.1.2 Експерименти върху всички типове събития от *EXTEND* 55
5.2 Експерименти с *LLaMA-2* 62
5.3 Обобщение на проведените експерименти 64

6 Прототип на система за откриване на събития в текст 66

6.1 Проектиране на софтуерен прототип на система за автоматично откриване на събития в текст 66

6.1.1 Функционални изисквания 66

6.1.2 Нефункционални изисквания 66

6.1.3 Обща архитектура на системата 66

6.2 Реализация на софтуерен прототип на система за автоматично откриване на събития в текст 67

6.2.1 Използвани технологии 67

6.2.2 Детайли от имплементацията на компонентите на системата . . 68

7 Заключение 71

7.1 Обобщение на изпълнението на началните цели 71

7.2 Насоки за бъдещо развитие 72

Благодарности 73

Публикации по темата на дипломната работа 74

Литература 75

Приложения 83

Приложение 1. Речник на чуждите термини 83

Приложение 2. Списък със съкращенията 86

Приложение 3. Указания за анотиране на събития 88

Приложение 4. Ръководство на потребителя 97

Списък с фигури

1 Пример за изречение, в което се среща събитие 13
2 Пример за изречение, в което се срещат повече от едно събитие [1] . . . 14
3 Класификация на типове събития спрямо *ACE2005* 17
4 Честота на срещане на отделни типове събития в *ACE2005* [2] 18
5 Класификация на подходите за извличане на събития спрямо различни критерии 20
6 Стъпки при последователната парадигма за извличане на събития [1] . 21
7 Стъпки при паралелната парадигма за извличане на събития [1] 22
8 Начини за извличане на активиращата дума и аргументите [1] 23
9 Относителни дялове на десетте най-често срещани езика в извлечените от *DBKF* твърдения 32
10 Модел на данните в мрежата от знания 34
11 Честота на срещане на десетте най-срещани събития от *ACE2005* в данните от *DBKF* 34
12 Пример за изречение, в което се среща събитие от тип *Cure-Claim* . . . 36
13 Пример за изречение, в което се среща събитие от тип *Severe-Weather* . 37
14 Пример за изречение, в което се среща събитие от тип *Rule-Change* . . 38
15 Честота на срещане на типовете аргументи по типове събития 42
16 Структура на събития, извлечени чрез *Text2Event* модел 44
17 Примерна схема на събитията за *Text2Event* 45
18 Инструкции към *LLaMA-2-70b-chat* за разпознаване на събития от тип *Severe-Weather* 48
19 Изпращане на заявка към *LLaMA-2-70b-chat* чрез *replicate* 48
20 Брой документи, за които базовият модел, *NEXT100* и *NEXT500* са предсказали наличие на събитие от дадения тип 54
21 Брой документи, за които базовият модел е предсказал наличие на събитие от дадения тип, и брой документи в сечението на предсказаните от базовия модел и *NEXT100* и *NEXT500* съответно документи 55
22 Резултати от експериментите с *LLaMA-2-70b-chat* 63

23 Сравнение на най-добрите постигнати макро-*F1* оценки от допълно обучените *Text2Event* модели и *LLaMA-2* без предварително обучение . 65
24 Компонентна диаграма на софтуерен прототип на система за автоматично откриване на събития в текст 67
25 Пример за тяло на *POST* заявка към */extract-events* 69
26 Пример за тяло на отговор на *POST* заявка към */extract-events* 69
27 Изглед от потребителския интерфейс *SwaggerUI* на системата за автоматично анотиране на събития 70
28 Екран от системата за автоматично откриване на събития в текст (1) . 97
29 Екран от системата за автоматично откриване на събития в текст (2) . 98
30 Екран от системата за автоматично откриване на събития в текст (3) . 99
31 Екран от системата за автоматично откриване на събития в текст (4) . 100

Списък с таблици

1 Описателни статистики за *ACE2005* корпуса от данни 18
2 Брой документи с разпознати и неразпознати събития от модела *Text2Event dyiepp_ace2005_en_t_large* върху данните от *DBKF* 33
3 Данни за подмножествата на кандидатите за всеки нов тип и избраните документи за анотация 40
4 Описателни статистики за новия корпус от данни *EXTEND* 41
5 Сравнение по описателни статистики между *ACE2005* корпуса от данни и новия корпус от данни *EXTEND* 42
6 Осреднени резултати от първоначална крос-валидация - прецизност (P), връщане (R) и *F1* оценка за събития от тип *Cure-Claim* [3] 52
7 Осреднени резултати от крос-валидация - прецизност (P), връщане (R) и *F1* оценка на модели, обучени върху документи от *EXTEND-CC* . . 56
8 Осреднени резултати от крос-валидация - прецизност (P), връщане (R) и *F1* оценка на модели, обучени върху документи от *EXTEND-SW* . . 57
9 Осреднени резултати от крос-валидация - прецизност (P), връщане (R) и *F1* оценка на модели, обучени върху документи от *EXTEND-RC* . . 57
10 Осреднени макро-стойности за прецизност (P), връщане (R) и *F1* оценка за събития от всички типове от *EXTEND* 59
11 Осреднени макро-стойности за прецизност (P), връщане (R) и *F1* оценка за събития от типове *Cure-Claim* и *Severe-Weather* 59
12 Микро-оценки на новите модели върху десетте най-често срещани събития от *ACE2005* 62

1 Увод

1.1 Актуалност на проблема и мотивация

Откриването и извличането на събития от текст са актуални задачи от областите на извличането на информация и обработката на естествен език. Задачата за откриване на събития има за цел да определи какво събитие се споменава в даден текст. Задачата за извличане на събития допълнително се стреми да даде отговори и на въпросите *Кой?*, *Какво?*, *Кога?*, *Къде?*, *Защо?*, *Как?* относно събитието.

Задачите са първоначално дефинирани в края на осемдесетте години на 20. век, когато американската агенция за изследвания в областта на отбраната *DARPA (Defense Advanced Research Projects Agency)* [1] иска да автоматизира идентифицирането на събития, свързани с тероризъм. [4] От тогава насам областите на приложение на извличането на събития от текст се увеличават. Освен в областта на отбраната и сигурността [5], задачата намира приложения в областите на финансите [6, 7], биомедицината [8, 9], социалните медии [10–12], както и правната област [13].

Още едно приложение на извличането на събития от текст можем да разпознаем и в борбата срещу дезинформацията. Автоматичното откриване и извличане на събития от текст би спомогнало работата на журналисти и проверители на факти. Конкретен случай на употреба, който може да бъде посочен в тази сфера, е търсене на новини или опровержения на такива, които споменават даден тип събитие (например протест, атака и др.). Разширение на този случай на употреба би било търсене на база аргументите на събитието – кой е участвал в него, къде се е случило и т.н., в зависимост от спецификата на съответното събитие. Настоящата дипломна работа се фокусира именно върху откриването на събития, които са потенциално често срещани в дезинформационно съдържание.

Една от особеностите на конкретната разглеждана сфера на приложение се състои в това, че множеството на събитията, които се срещат в новинарските потоци и социалните медии, е много динамично, което е предпоставка за възникване на нови типове събития. Това обуславя необходимостта решенията в областта да са адаптивни

[1] `https://www.darpa.mil/`

и лесно скалируеми за новопоявяващи се събития.

1.2 Цел и задачи на дипломната работа

Целта на дипломната работа е да се разработят метод и софтуерен прототип, който разпознава в текст събития от предварително зададено множество, както и да се предложи методология за разширяване на предложеното множество с нови събития.

Задачите, произтичащи от целта, са:

1. Обзор на разработките в областта
2. Избор на подход за първоначално автоматично откриване на събития в текст
3. Разработване на методология за разширяване на множеството от разпознавани събития
4. Подготовка на анотиран корпус от данни
5. Развитие на избрания подход с цел разпознаване на разширеното множество от събития
6. Анализ и оценка на получените резултати
7. Разработване на софтуерен прототип на система за автоматично разпознаване на събития в текст

1.3 Очаквани ползи от реализацията

Като очаквани ползи и резултати от изследването в настоящата дипломна работа могат да бъдат посочени:

1. Методология за разширяване на множество от разпознавани събития, т.е. за идентифициране и дефиниране на нови типове събития;
2. Указания за анотация на събития от новодефинираните типове;
3. Анотиран корпус от данни за новите типове събития;

4. Обучени модели, които разпознават събития от новите типове;
5. Софтуерен прототип на система за автоматично разпознаване на събития в текст.

1.4 Структура на дипломната работа

Дипломната работа се състои от седем глави, както следва:

1. Глава 1 „Увод“ представя актуалността и мотивацията на поставената задача, дефинира целта на дипломната работа и задачите, произтичащи от нея, както и очакваните ползи от реализацията.
2. Глава 2 „Обзор на подходи за откриване на събития в текст“ въвежда основните понятия в областта, посочва някои стандартни корпуси от данни, които се използват за решаване на задачата, и разглежда различните съществуващи подходи за автоматично откриване на събития в текст, групирайки ги в няколко категории по различни критерии.
3. Глава 3 „Събиране на данните и подготовка на анотиран корпус от данни“ дефинира процесите по събирането и анализа на данните, които ще се използват, разработването на методология за разширяване на множество от събития (т.е. за дефиниране на нови типове събития, които да бъдат добавени към вече съществуващото множество), както и по подготовката на анотиран корпус от данни.
4. Глава 4 „Обучение на модели за откриване на събития в текст“ описва избраните методи за решаване на поставената задача, тяхната архитектура и приложените стратегии за обучение.
5. Глава 5 „Проведени експерименти и анализ на резултатите от тях“ представя проведените експерименти за автоматично откриване на събития в текст. Описан е дизайнът на експериментите, представени са резултатите от тях и е направен подробен анализ.
6. Глава 6 „Прототип на система за автоматично откриване на събития в текст“ описва процесите по проектирането и реализацията на софтуерен прототип на

система за автоматично откриване на събития в текст. Дефинират се функционалните и нефункционалните изискванията към системата, избраната архитектура, средствата за реализация и процесите по имплементацията на отделните модули.

7. Глава 7 „Заключение“ обобщава изпълнението на поставените задачи, като посочва и посоки за бъдещо развитие.

В допълнение към дипломната работа са представени четири приложения:

1. Приложение 1 "Речник на чуждите термини"съдържа всички използвани в дипломната работа термини на чужд език и преводите им.

2. Приложение 2 "Списък със съкращенията"съдържа всички използвани в дипломната работа съкращения.

3. Приложение 3 "Указания за анотиране на събития"съдържа създадените в рамките на дипломната работа указания за анотиране на данни за новодефинираните типове събития.

4. Приложение 4 "Ръководство на потребителя"съдържа указания за стартиране и работа с разработения софтуерен прототип на система за автоматично откриване на текст.

Забележка: Всички хипервръзки към уеб страници, реферирани в дипломната работа, са последно посетени на 21.10.2023.

2 Обзор на подходи за откриване на събития в текст

2.1 Основни понятия

Задачата за откриване на събития (*event detection*) е свързана с идентифицирането на **активиращата дума** на **събитието** и класифицирането му спрямо неговия тип.

Задачата за извличане на събития (*event extraction*) включва в себе си задачата за откриване на събития, както и идентифицирането и класифицирането на **аргументите** на даденото събитие.

Откриването на събития може да се разглежда по два основни начина - в **затворена** и в **отворена област**.

Откриването на събития от затворена област (*closed-domain*) цели да идентифицира събития от предварително зададена схема, която се състои от краен брой предефинирани типове събития. Настоящото изследване се фокусира именно върху откриване на събития от затворена област и всички следващи определения, свързани със събития, са в този контекст.

Откриването на събития от отворена област (*open-domain*) се случва без предварително известна схема от типове събития.

2.1.1 Понятия, свързани със събития

Събитие (*event*) се определя като конкретно случване на нещо в определено време и на определено място, включващо един или повече участници. Събитието често може да се определи и като промяна на дадено състояние.[2]

Всяко събитие се състои от следните задължителни или незадължителни компоненти:

[2] `https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-events-guidelines-v5.4.3.pdf`

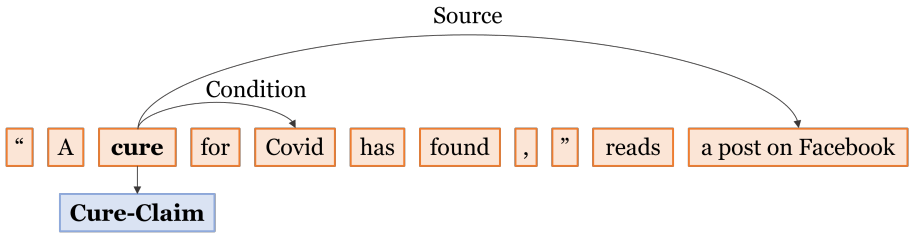
- **обхват на събитието** (задължителен);
- **активатор на събитието** (задължителен, точно един);
- **аргументи на събитието** (незадължителни).

Обхват на събитието (*event extent*) е изречението, в което се намира събитието.

Активатор на събитието (или активираща дума, *trigger word*) е думата, която най-точно сигнализира за срещането на събитието.

Аргументи на събитието (*event arguments*) са участници в събитието или други негови характеристики. Аргументите за всяко събитие са различни спрямо неговата специфика.

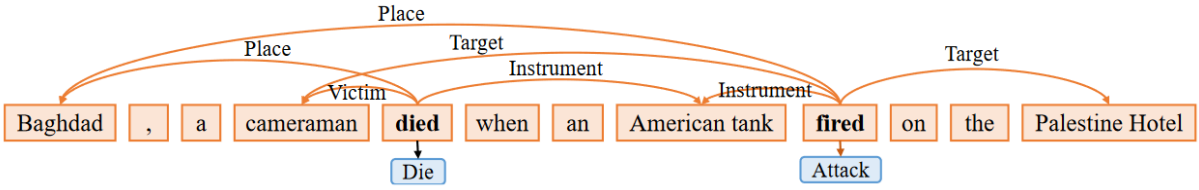
Фигура 1 показва пример за изречение, в което се среща едно събитие от тип *Cure-Claim* с активираща дума "*cure*"и два аргумента.[3]



Фигура 1: Пример за изречение, в което се среща събитие

Допустимо е в дадено изречение да се срещат повече от едно събитие от един и същи или от различни типове. Пример за такова изречение е показан на Фигура 2. Отбелязани са събитие от тип *Die* с активираща дума "*died*"и събитие от тип *Attack* с активираща дума "*fire*". За всяко от двете събития са отбелязани и съответстващите им аргументи. Връзката между събитие и аргумент е илюстрирана с помощта на стрелка, излизаща от активиращата дума и насочена към съответния аргумент.

[3]Фигурата е авторска и е адаптирана по стила на фигурите от [1]



Фигура 2: Пример за изречение, в което се срещат повече от едно събитие [1]

2.1.2 Общи понятия

Анотирани данни (*Annotated/Labeled data*) са данни, които имат поставени етикети за определени обекти в тях спрямо дадена класификация.

Машинно самообучение с учител (*Supervised Machine Learning*) е вид машинно самообучение, при което се използват набори от предварително анотирани данни за обучение на моделите.

Частично контролирано машинно самообучение (*Semi-supervised Machine Learning*) е вид машинно самообучение, при което се използва малко количество анотирани данни за обучение с учител и допълнително се анотират автоматично още данни.

Отдалечено контролирано машинно самообучение (*Distant Supervised Machine Learning*) е вид машинно самообучение, при което се използват автоматично анотирани данни за обучение с учител.

Трансферно обучение (*Transfer learning*) е техника в машинното самообучение, при която се използва предварително обучен модел за дадена задача за решението на нова сходна задача.

Подходи с малко предварително обучение (*Few-shot setting*) са подходи, целящи да класифицират нови данни с много малко налични анотирани примери за обучение.

Подходи без предварително обучение (*Zero-shot setting*) са подходи, целящи да класифицират нови данни без налични анотирани примери за обучение.

2.1.3 Метрики за оценка

Стандартно метриките, които се използват за оценка на подходите за откриване и извличане на събития, са прецизност, връщане и *F1* оценка.

Прецизност (*precision*) е метрика за оценка, която показва каква част от положително предсказаните документи са положително анотирани. Изчислява се по формулата:

$$precision = \frac{TP}{TP + FP}$$

където TP (*true positive*) е броят на документите, които са правилно предсказани като положителни, а FP (*false positive*) е броят на документите, които са неправилно предсказани като положителни.

Връщане (*recall*) е метрика за оценка, която показва каква част от положително анотираните документи са предсказани от модела. Изчислява се по формулата:

$$recall = \frac{TP}{TP + FN}$$

където TP (*true positive*) е броят на документите, които са правилно предсказани като положителни, а FN (*false negative*) е броят на документите, които са неправилно предсказани като отрицателни.

***F1* оценка (*F1 score*)** е метрика за оценка, която представлява средното хармонично на прецизност и връщане. Тъй като сами по себе си прецизността и връщането могат да си противоречат, *F1* оценката е по-показателна за цялостното представяне на модела. Изчислява се по формулата:

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

2.2 Стандартни корпуси от данни

Съществуват различни корпуси от данни, които се използват от изследователската общност при решаване на задачата за автоматично откриване и извличане на събития. Някои от тях могат да се използват и при решаване на други актуални задачи в областта на обработката на естествен език като например разпознаване на именовани единици, извличане на релации и др. В настоящата секция ще бъдат описани четири корпуса от данни, които съдържат документи от общ характер и от областта на новините, с техните характеристики и ограничения.

2.2.1 *ACE2005* корпус от данни

Един от най-често цитираните и използвани корпуси от данни в областта е корпусът от данни *ACE2005* [14], разработен от *Linguistic Data Consortium* (*LDC*)[4]. Той е многоезичен корпус, като данните в него са текстове от различен характер на английски, арабски и китайски (мандарин) език. Общият брой на документите в корпуса на трите езика е около 1800. Документите в корпуса са анотирани за именовани единици, релации и събития.[5]

Документите в *ACE2005* корпуса от данни са анотирани за 33 типа събития, групирани в осем основни категории, и за техните съответстващи аргументи. Фигура 3 представя типовете събития, отразени в *ACE2005*, по групи. Броят на типовете аргументи в *ACE2005* е 36.

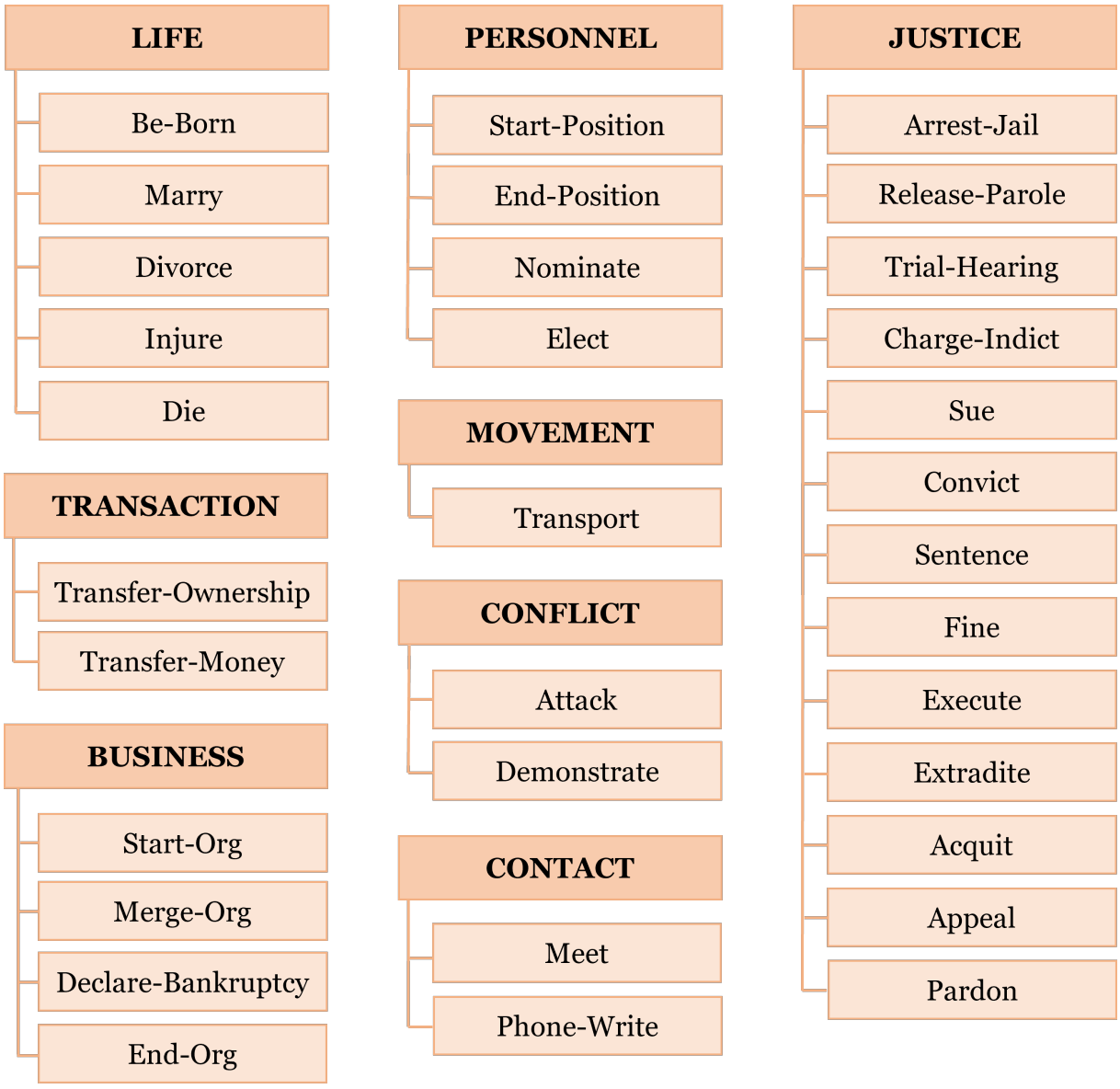
Таблица 1 показва основни статистики за частта от корпуса на английски език, както са представени в [15].

Корпусът *ACE2005* не е балансиран откъм честота на срещане на отделните типове събития, както може да се види на Фигура 4. Прекъснатата линия показва прага от 100 срещания. Видно е, че около 70% от типовете събития се срещат под 100 пъти. Най-често срещаното събитие се среща около два пъти по-често от второто по честота.

ACE2005 корпусът от данни е достъпен под платен лиценз, което е и основното

[4] `https://www.ldc.upenn.edu/`
[5] `https://catalog.ldc.upenn.edu/LDC2006T06`

ограничение за използването му.



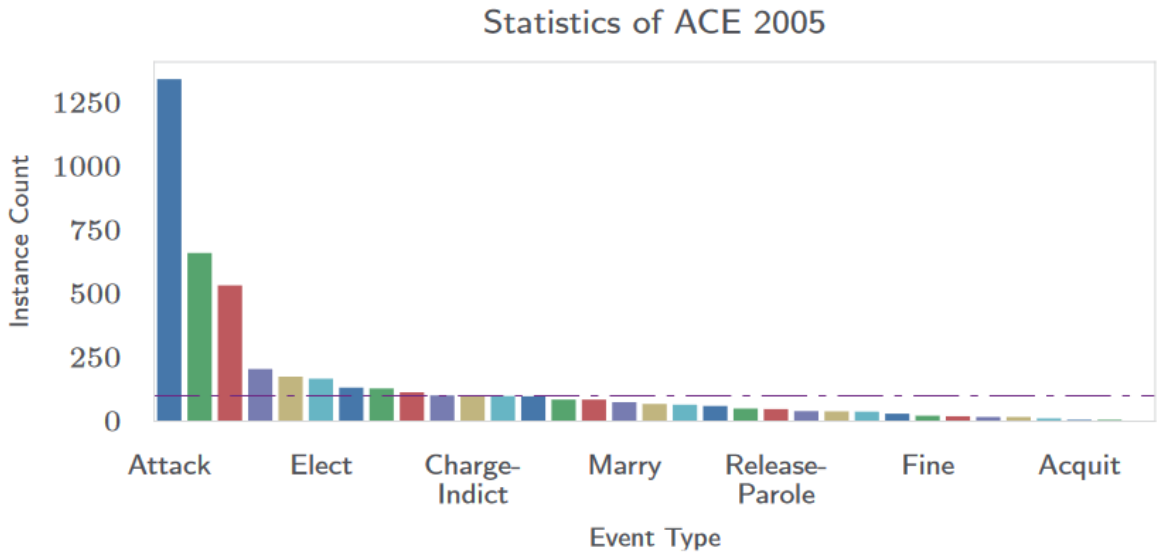
Фигура 3: Класификация на типове събития спрямо *ACE2005*

2.2.2 *MAVEN* корпус от данни

Корпусът от данни *MAVEN* (*MAssive eVENt detection dataset*) е представен от Wang et al. [16] Той е анотиран само за типове събития и активиращи думи, което го прави подходящ за използване само при решаването на задачата за откриване на събития.

Таблица 1: Описателни статистики за *ACE2005* корпуса от данни

	Обучителни данни	**Валидационни данни**	**Тестови данни**
Документи	529	40	30
Изречения	14 837	863	672
Активиращи думи	4 337	497	438
Аргументи	7 768	933	911



Фигура 4: Честота на срещане на отделни типове събития в *ACE2005* [2]

Документите в него са от общ характер и са на английски език. Авторите на *MAVEN* корпуса от данни го определят като най-големия корпус от данни за откриване на събития към момента на създаването му.

MAVEN корпусът от данни се състои от 4 480 документа и 49 873 изречения. Типовете събития, за които е анотиран, са 168, а самите събития наброяват 111 611 - над 20 пъти повече от тези в *ACE2005*.

2.2.3 *FedSemcor* корпус от данни

Le and Nguyen [17] предлагат корпус от данни *FedSemcor* с изключително фина грануларност на типовете събития - общо 449. Като една от причините за необходимостта от такъв корпус авторите посочват невъзможността на съществуващите корпуси с по-малко на брой и по-общи типове събития да отразят различните нюанси на съби-

тията.

FedSemcor е базиран на *Semcor* корпуса от данни [18] за разграничаване на смисъла на думите (*word sense disambiguation*). Подобно на *MAVEN* корпуса от данни, *FedSemcor* е анотиран само за типове събития и активиращите им думи. Той съдържа общо 34 666 анотирани събития.

2.2.4 *WikiEvents* корпус от данни

Споменатите до момента корпуси от данни разглеждат задачите за откриване и извличане на събития като задачи на ниво изречение. Li et al. [19] виждат това като ограничение и предлагат нов корпус от данни - *WikiEvents*, в който събитията са на ниво цял документ. Това означава, че обхватът на дадено събитие не е ограничен до едно изречение, а е допустимо в различни изречения от документа да се срещат едно и също събитие (представено чрез активиращата си дума и аргументите си).

Документите в *WikiEvents* корпуса от данни са събирани от статии в Wikipedia и са анотирани за типове събития, активиращи думи и аргументи. Следвана е онтологията от проекта на *DARPA KAIROS*[6], която включва 67 типа събития и 57 типа аргументи. Общо документите в корпуса наброяват 206, отделните изречения 5262, а събитията - 3241.

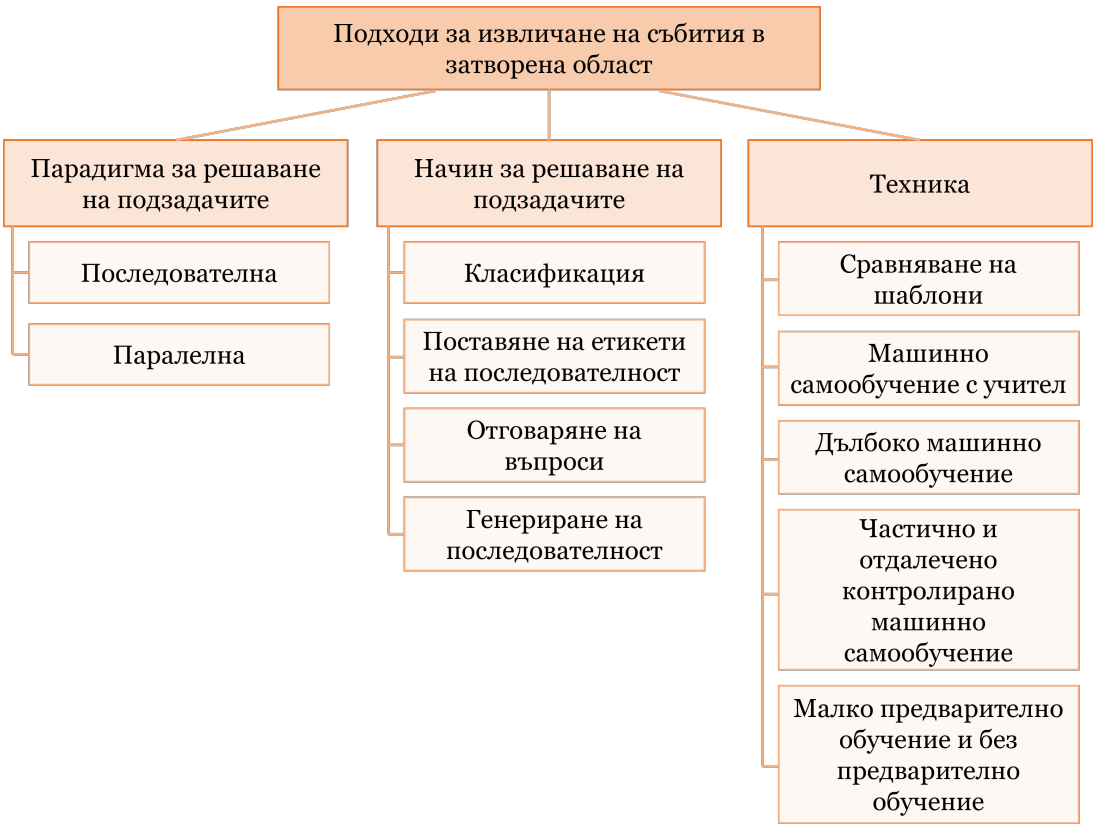
2.3 Подходи за автоматично откриване на събития в текст

Подходите за откриване и извличане на събития в текст могат да се класифицират по различни признаци и критерии. Съгласно Liu et al. [20] двете най-основни групи подходи подходите за откриване и извличане на събития в затворена и в отворена област.

Затворените подходи могат допълнително да се класифицират в различни групи на база няколко критерия - парадигма за решаване на подзадачите, начин на решаване на подзадачите и прилагана техника, както е показано на Фигура 5.

При тази група подходи задачата се разглежда като съвкупност от четири подзадачи, а именно:

[6] `https://www.ldc.upenn.edu/collaborations/current-projects`



Фигура 5: Класификация на подходите за извличане на събития спрямо различни критерии

- идентифициране на активиращата дума;
- класификация на типа на събитието;
- идентифициране на аргументите на събитието;
- класифициране на ролите на аргументите.

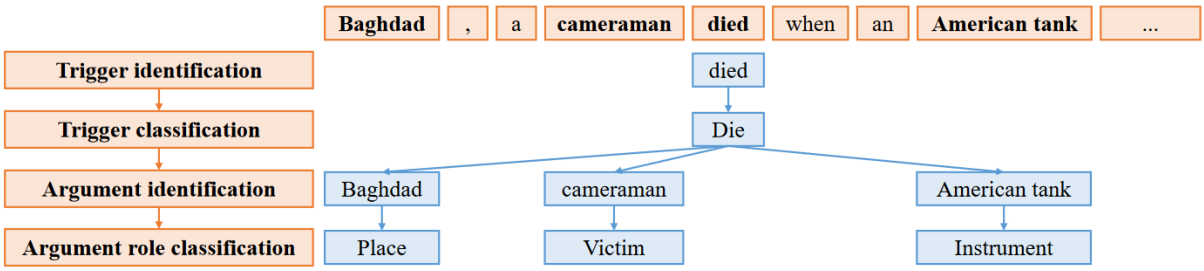
2.3.1 Парадигми за решаването на подзадачите

В зависимост от стратегията за организиране на решаването на четирите подзадачи можем да идентифицираме две основни парадигми за затворено извличане на събития - последователна (*pipeline*) и паралелна (*joint*).

2.3.1.1 Последователна парадигма При последователната парадигма четирите подзадачи се решават една след друга. Фигура 6 онагледява съответните стъпки.

Всяка стъпка отговаря на една от подзадачите и представлява отделен класификатор. Като първа стъпка се идентифицира активиращата дума на събитието, след което то се класифицира спрямо типа му. В зависимост от резултата от тази класификация се извличат и аргументите на даденото събитие - за всяка дума от документа се определя дали тя е аргумент на вече разпознатото събитие и след това разпознатите аргументи се класифицират спрямо ролите им.

Примери за подходи, базирани на последователната парадигма, са разработките на Zhao et. al [21], Chen et. al [22], Li et. al [23].

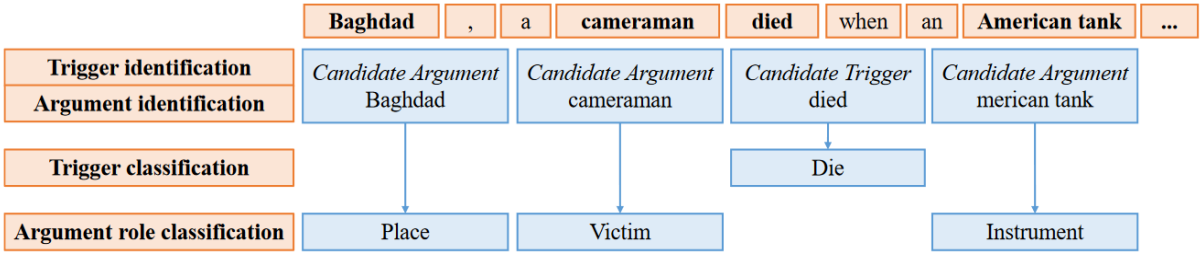


Фигура 6: Стъпки при последователната парадигма за извличане на събития [1]

Основен недостатък на последователната парадигма е възможността за разпространение на грешката към следващи подзадачи. Например, ако активиращата дума се идентифицира грешно или типът на събитието се класифицира грешно, това ще има отношение към класификацията на аргументите, тъй като те зависят от схемата на всеки отделен тип събитие.

2.3.1.2 Паралелна парадигма За да се преодолее основният недостатък на последвователната парадигма, е предложена паралелната (*joint*) парадигма. Фигура 7 онагледява съответните стъпки. При нея активиращата дума и аргументите се идентифицират едновременно, след което се класифицират също едновременно.

Примери за подходи, имплементиращи паралелната парадигма, са предложени от Sheng et. al [24], Wadden et. al [25], Yang et. al [26].



Фигура 7: Стъпки при паралелната парадигма за извличане на събития [1]

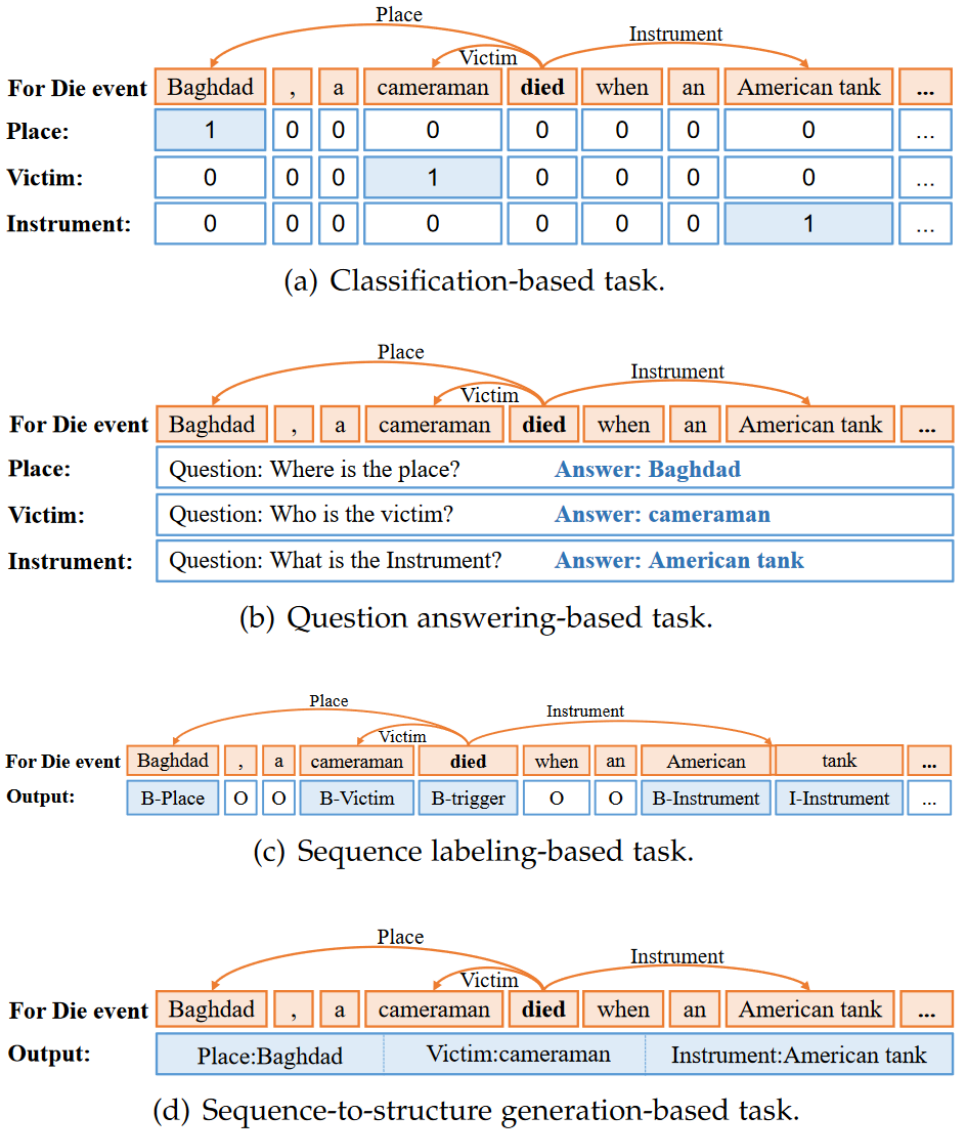
2.3.2 Начини за извличане на активиращата дума и аргументите

Li et al. [1] посочват четири начина за извличане на активиращата дума и аргументите и съответно разглеждат задачата за извличане на събития като:

- класификационна задача;
- задача за отговаряне на въпроси (*question answering* или още *machine-reading comprehension*);
- задача за поставяне на етикети на последователност (*sequence labeling*);
- генеративна (*sequence-to-structure*) задача.

2.3.2.1 Класификационна задача Най-просто задачата за извличане на събития може да се разглежда като мулти-клас класификационна, където всяка единица в документа се класифицира като инстанция на даден клас (активираща дума или даден тип аргумент). Този начин за извличане залага на разпознаването на именувани единици, което е предпоставка за разпространяване на грешки. Механизмът на работа е представен на Фигура 8(a). Zhao et. al [21] и Chen et. al [22] предлагат подходи, третиращи задачата за извличане на събития като класификационна.

2.3.2.2 Задача за отговаряне на въпроси Задачата за отговаряне на въпроси предполага задаване на въпроси за всеки аргумент на събитието и за активиращата му дума. Тъй като отделните типове събития имат различни аргументи, правилното класифициране на типа събитие е от ключово значение за извличането на аргументите му. Пример за извличане на аргументи чрез подходящи въпроси е представен на



Фигура 8: Начини за извличане на активиращата дума и аргументите [1]

Фигура 8(b). Подходи, разглеждащи задачата за извличане на събития като задача за отговаряне на въпроси, са предложени от Li et. al [23], Zhou et. al [27] и Lu et. al [28].

2.3.2.3 Задача за поставяне на етикети на последователност Когато разглеждаме задачата за извличане на събития като задача за поставяне на етикети на последователност, за всеки аргумент се отчита началната и крайната позиция в тек-

ста, като за целта се използва *BIO* нотацията[7]. В нея първата дума на всяка единица се отбелязва с *B* (*beginning* - начало), а всяка следваща с *I* (*inside* - вътре). Думите, които не са част от нито една единица, се отбелязват с *O* (*outside* - извън). Например, на Фигура 8(c) фразата *"American tank"* трябва да бъде разпозната като аргумент от тип *Instrument* и на *American* се поставя етикет *B-Instrument*, а на *tank* съответно *I-Instrument*. Примери за подходи, използващи тази стратегия, са разработените от Sheng et. al [24] и Wadden et. al [25].

2.3.2.4 Генеративна задача Когато задачата за извличане на събития се разглежда като генеративна задача, активиращата дума на събитието и аргументите му се генерират директно заедно със съответните им типове, както е илюстрирано на Фигура 8(d). Такъв генеративен модел за извличане на събития е *Text2Event*, предложен от Lu et al. [29].

2.3.3 Техники за извличане на събития

В зависимост от прилаганите техники Liu et al. [20] групират подходите за затворено извличане на събития в следните четири категории:

- Подходи, базирани на сравнение на шаблони (*pattern matching*);
- Подходи, базирани на машинно самообучение с учител;
- Подходи, базирани на дълбоки невронни мрежи;
- Подходи, базирани на частично контролирано и отдалечено контролирано машинно самообучение.

Допълнително можем да разгледаме и група подходи с малко предварително обучение и без предварително обучение.

2.3.3.1 Подходи, базирани на сравнение на шаблони (*pattern matching*) Хронологично погледнато, подходите, базирани на сравнение на шаблони, са едни от първите подходи за извличане на събития. В основата им стоят шаблони на събития, които са специфични за конкретната област на приложение. Често се прилагат

[7] `https://paperswithcode.com/task/named-entity-recognition-ner`

синтактични дървета, регулярни изрази, използват се лексикални и морфологични свойства на входните данни. [20]

Основните предимства на тези подходи са липсата на необходимост от големи обеми анотирани данни, както и възможността за постигане на висока точност в конкретни области на приложение. Недостатъците са свързани с факта, че създаването и поддържането на шаблони е времеемък и трудоемък процес. Освен това невинаги е лесно и/или подходящо да се преизползват шаблони на събития от една област в друга. [20]

Един от класическите примери за подход, базиран на сравнение на шаблони, е системата на Riloff - *AutoSlog* [30]. В нея на базата на лингвистични шаблони се построяват речници, които се използват за откриване на събития.

2.3.3.2 Подходи, базирани на машинно самообучение с учител В сравнение с подходите, базирани на сравнение на шаблони, подходите, базирани на класическо машинно самообучение с учител, могат да бъдат генерализирани и преизползвани по-успешно. [20] Подходите от тази група разчитат на предварително анотирани данни. От голямо значение при прилагането им е подборът на лексикалните и контекстните свойства на данните, които ще се използват. Те могат да бъдат селектирани на база съставни анализатори (*consituent parsers*), анализатори на зависимостите (*dependency parsers*), механизми за разпознаване на частите на речта (*POS taggers*) и др. [31]

Примери за подходи, базирани на машинно самообучение, които се прилагат най-често в решаването на задачата за извличане на събития, са машини с поддържащи вектори (*support vector machines*, *SVM*), метод на максималната ентропия (*maximum entropy*, *ME*), наивен Бейсов класификатор (*Naive Bayes*, *NB*), случайни условни полета (*conditional random field*, *CRF*), йерархично агломеративно клъстеризиране (*hierarchical agglomerative clustering*, *HAC*). [20]

Liao and Grishman [32] предлагат подход, прилагащ метода на максималната ентропия за обучение на модел над *ACE2005* корпуса. Постигнати са *F1* оценки от 68.8% за класифициране на типа събитие и 44.6% за класифициране на аргументите.

Основно предимство на подходите, базирани на машинно самообучение, пред под-

ходите, базирани на сравняване на шаблони, е по-добрата генерализация и възможност за преизползване. Недостатъците на тази група подходи могат да се обобщят в няколко насоки. От една страна е налице необходимостта от повече анотирани данни за обучение. От друга страна при тези подходи подборът на свойствата на входните данни, които ще се използват, (т.нар. *feature engineering*) е задача, изискваща много време и специфична експертиза. Още повече некоректно подбраните и извлечени свойства могат да повлияят негативно на представянето на приложените в последствие модели. [31] Не на последно място, подходите, базирани на класическо машинно самообучение, имат известни ограничения, що се отнася до научаването на по-комплексни нелинейни връзки. [20]

2.3.3.3 Подходи, базирани на дълбоки невронни мрежи Подходите, базирани на дълбоки невронни мрежи, преодоляват някои от недостатъците на подходите, базирани на класическо машинно самообучение, като автоматизират подбора на свойствата на данните и предлагат възможности за научаване на нелинейни връзки. [20] Това, заедно с добрите резултати, които се постигат чрез използването им, е предпоставка подходите от тази група да стават все по-предпочитани за решаване на редица задачи, в т.ч. и извличането на събития.

Голяма част от най-новите разработки в областта прилагат подходи, базирани на векторни представяния и дълбоки невронни мрежи като конволюционни невронни мрежи, рекурентни невронни мрежи, трансформатори.

Векторните представяния на думите са от ключово значение за функционирането на подходите, базирани на дълбоки невронни мрежи. [31] Те съдържат в себе си широк набор от синтактични и семантични свойства на думите, научени от големи обеми текстови данни. [33]

Съществуват два основни вида векторни представяния - безконтекстни (или още статични) и контекстно-базирани (или още динамични).

Безконтекстните векторни представяния на думите не взимат предвид контекста, в който те са употребени. Това означава, че думи, приемащи различни значения, винаги ще имат едно и също векторно представяне, което е и основното ограничение на

този вид векторни представяния. Сред най-добре познатите безконтекстни векторни представяния, които се използват в областта на извличането на събития (и не само), са *Word2Vec* [33], *GloVe* [34], *fastText* [35].

Контекстно-базираните векторни представяния преодоляват най-съществения ограничение на безконтекстните такива, като динамично кодират думите в зависимост от контекста на използването им в дадено изречение. Базирани на трансформатори и езикови модели като *BERT* [36], *RoBERTa* [37], *ELMo* [38] и др., тези векторни представяния са широко използвани в редица задачи от областта на обработката на естествен език [20], в това число и в задачата за извличане на събития.

Например Wadden et al. [25] предлагат подход, базиран на *BERT*, наречен *DYGIE++* (*DYnamic Graph Information Extraction*), който цели да решава едновременно задачите за разпознаване на именувани единици, извличане на релации и извличане на събития. Моделът е обучаван върху *ACE2005* и постига *F1* оценки от 73.6% за класифициране на типа събитие и 52.5% за класифициране на аргументите.

Конволюционните невронни мрежи (*CNN*) се справят с автоматичното селектиране на локалните (на ниво изречение) свойства на входните данни, които ще бъдат разглеждани.

Chen et al. [22] предлагат подход, базиран на динамични конволюционни невронни мрежи с *multi-pooling* - *DMCNN*. Този модел подбира автоматично лексикалните свойства и свойствата на ниво изречение и прилага последователната парадигма за решаване на подзадачите за откриване и класифициране на активиращата дума и аргументите на събитието. Постигнати са *F1* оценки от 69.1% за класифициране на типа събитие и 53.5% за класифициране на аргументите върху *ACE2005*.

Основен недостатък на подходите, използващи конволюционни невронни мрежи, е невъзможността им да засичат връзки между по-отдалечени думи. [31]

Рекурентните невронни мрежи (*RNN*) като *GRU* (*Gated Recurrent Unit*) и *LSTM* (*Long Short-Term Memory*) се справят по-добре от конволюционните с връзките между по-отдалечените думи в текста и са подходящи за задачи за поставяне на етикети на последователност.

Nguyen et al. [39] първи предлагат модел за извличане на събития, базиран на рекурентни невронни мрежи и по-точно на *GRU - JRNN*. Представеният подход използва две рекурентни невронни мрежи, които решават задачата съгласно паралелната парадигма. Обученият модел постига *F1* оценки от 69.3% за класифициране на типа събитие и 55.4% за класифициране на аргументите.

Трансформаторите представляват архитектура на невронна мрежа, базирана на механизъм с внимание [40]. Механизмите за внимание позволяват на обучавания модел да се "фокусира"върху по-важната информация. В последно време много подходи за откриване на събития са базирани именно на трансформатори.

Lu et al. [29] прилагат архитектура, базирана на трансформатори, за *Text2Event* - генеративен подход, използващ като основа езиковия модел *T5* [41]. *Text2Event* генерира събитието заедно с аргументите му от край до край и показва добри възможности за трансферно обучение. Върху *ACE2005* показва най-добри *F1* оценки от 71.8% за класифициране на типа на събитието и 54.4% за класифициране на аргументите.

2.3.3.4 Подходи, базирани на частично контролирано и отдалечено контролирано машинно самообучение

Подходите, базирани на частично контролирано и отдалечено контролирано машинно самообучение, обикновено се прилагат с цел генериране на допълнително количество данни за обучение на моделите.

В областта на откриването на събития Zhou et al. [27] предлагат частично контролиран подход, наречен *DualQA*, който моделира извличането на аргументите на събитието като задача за отговаряне на въпроси. За тази подзадача предложеният подход постига *F1* оценка 45.4% върху *ACE2005*.

Пример за отдалечено контролиран подход е предложеният от Yang et al. [42]. Този подход разглежда задачата за извличане на финансови събития на китайски език на ниво цял документ, като използва автоматично генерирани анотирани данни.

2.3.3.5 Подходи с малко предварително обучение и без предварително обучение

Подходите с малко предварително обучение и без предварително обу-

чение също се прилагат в сценарии с малко налични ресурси за обучение. Те са сравнително нова посока за изследване в областта.

Lai et al. [43] предлагат подход с малко предварително обучение, който прилага трансферно обучение от подходи за разграничаване на значението на думите. Авторите твърдят, че предложеният от тях подход подобрява генерализацията в сравнение с други подходи с малко предварително обучение.

Lyu et al. [44] предлагат подход за извличане на събитя без предварително обучение, формулирайки задачата като задача за правене на извод от текст (*textual entailment*) и за отговаряне на въпроси. Авторите прилагат предварително обучени за тези задачи модели, но отбелязват, че тези модели трудно се справят с трансферното обучение.

2.3.4 Обобщение на подходите за автоматично откриване и извличане на събития

Задачите за откриване и извличане на събития са комлексни и предизвикателни. Съществуват множество различни подходи за решаването им, които се групират в категории по различни признаци за сравнение.

Тенденциите в областта на откриването на събития, а и на обработката на естествен език като цяло, сочат, че подходите, базирани на дълбоки невронни мрежи са силно предпочитани поради възможността им да научават по-сложни връзки в данните и липсата от необходимост от ръчен подбор на входните свойства на данните. Недостатъкът на тези подходи обаче се състои в това, че изискват голямо количество анотирани данни за обучение.

За задачи, в които анотираните данни са оскъдни, могат да се прилагат частично или отдалечено контролирани подходи, както и подходи с малко или без никакво предварително обучение. Изследванията до момента обаче показват, че тези групи подходи не успяват да достигнат резултатите, постигнати с подходи, базирани на дълбоки невронни мрежи, и имат още поле за развитие.

Подходите, демонстриращи добри възможности за трансферно обучение, също могат да се използват при наличието на малко количество анотирани данни, тъй като могат да използват знанията си, научени за една задача (или тип данни) върху

друга. Пример за такъв подход от разгледаните в настоящия обзор, който показва много добри резултати върху данните от *ACE2005*, е *Text2Event* [29]. Постигнатите добри резултати заедно с докладваните възможности за трансферно обучение правят *Text2Event* подхода подходящ за разширяване с цел разпознаване на допълнителни типове събития.

3 Събиране на данните и подготвяне на анотиран корпус

3.1 Събиране и анализ на данните от база от данни с опровержения на фалшиви новини

Предметът на изследване на настоящата дипломна работа е откриване на събития в областта на фалшивите новини и дезинформацията. Поради специфичната област на приложение е важно данните, с които боравим, да бъдат също от съответната област.

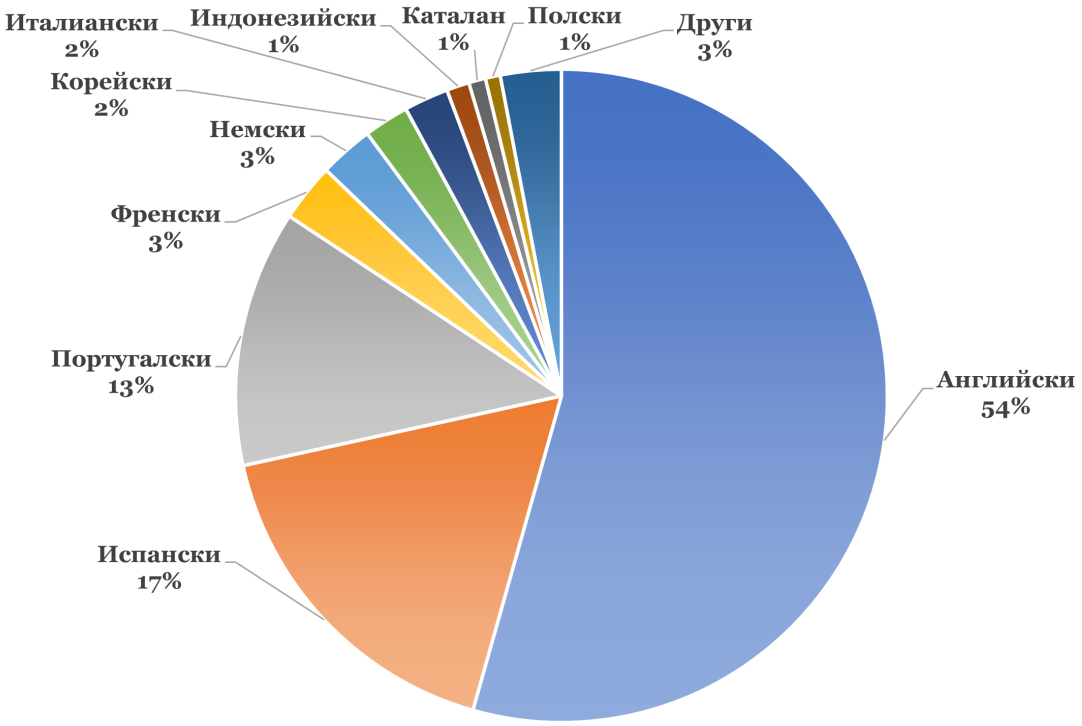
Данните, които ще се използват в изследването, са извлечени от *Database of Known Fakes* (*DBKF*) [45]. Това е нерелационна (и по-точно графова) база от данни, съдържаща опровержения на фалшиви новини от реномирани световни организации за проверка на факти, които са обогатени с допълнителни метаданни. [45]

Основните обекти от модела на данните на *DBKF*, върху които се фокусира изследването, са т.нар. твърдения (*claims*). Твърденията представляват кратки текстове, които са знакови за съответното опровержение. Тяхната дължина най-често е едно-две изречения, но в по-редки случаи може да достигне няколко параграфа.

Извлечени са общо 78 246 твърдения на различни езици от *DBKF*. За автоматично разпознаване на езика на всяко твърдение е използван модул за разпознаване на език[8] от *SpaCy*[9]. В данните са разпознати общо 46 езика, като относителните дялове на десетте най-често срещани езици са представени на Фигура 9. Видно е, че английският език е доминиращ - малко над половината твърдения са именно на този език. Също така си струва да се отбележи, че твърденията на английски, испански и португалски език представляват около 84% от всички твърдения.

Настоящото изследване се фокусира върху английски език и затова за бъдеща работа са подбрани твърденията на английски език, които са 42 555 на брой. Допълнително тези твърдения са разделени на отделни изречения с помощта на разделител

[8] `https://pypi.org/project/spacy-language-detection/`
[9] `https://spacy.io/`



Фигура 9: Относителни дялове на десетте най-често срещани езика в извлечените от *DBKF* твърдения

на изречения от *NLTK*[10]. Резултатът от тази операция са 54 280 изречения.

Към този момент на анализа на събраните данни не е известно какви и колко събития от познатите класификации се срещат в тях, тъй като данните не са анотирани относно събития. Поради тази причина е използван предварително обученият върху данни на английски език от *ACE2005* модел на *Text2Event* *dyiepp_ace2005_en_t_large*[11] върху извлечените твърдения и изречения. Целта е да се установи кои събития от *ACE2005* класификацията се срещат в документите[12] и с каква честота.

Броят документи, в които избраният модел е открил или съответно не е открил събития, е представен в Таблица 2. Видно е, че както при целите твърдения, така и при индивидуалните изречения, моделът открива събития само в около 20% от документите.

С цел по-удобен анализ на типовете събития, аргументите им и т.н., получени-

[10] `https://www.nltk.org/api/nltk.tokenize.html`
[11] `https://github.com/luyaojie/Text2Event`
[12] Под документ се разбира както цяло твърдение, така и изречение.

Таблица 2: Брой документи с разпознати и неразпознати събития от модела *Text2Event dyiepp_ace2005_en_t_large* върху данните от *DBKF*

	Цели твърдения	**Изречения**
Нито едно събитие	32 967	43 259
Поне едно събитие	9 588	11 021
Точно едно събитие	6 509	8 602
Няколко събития	3 079	2 419
Общ брой документи	**42 555**	**54 280**

те данни са трансформирани в *RDF*[13] мрежа от знания. *RDF* (*Resource Description Framework*) е графов модел за описание на ресурси, с помощта на който лесно могат да се представят семантиката им и връзките между тях. Информацията в *RDF* се представя чрез наредени тройки от субект, предикат и обект. Семантиката на наредената тройка е, че субектът стои в определена от предиката релация с обекта.

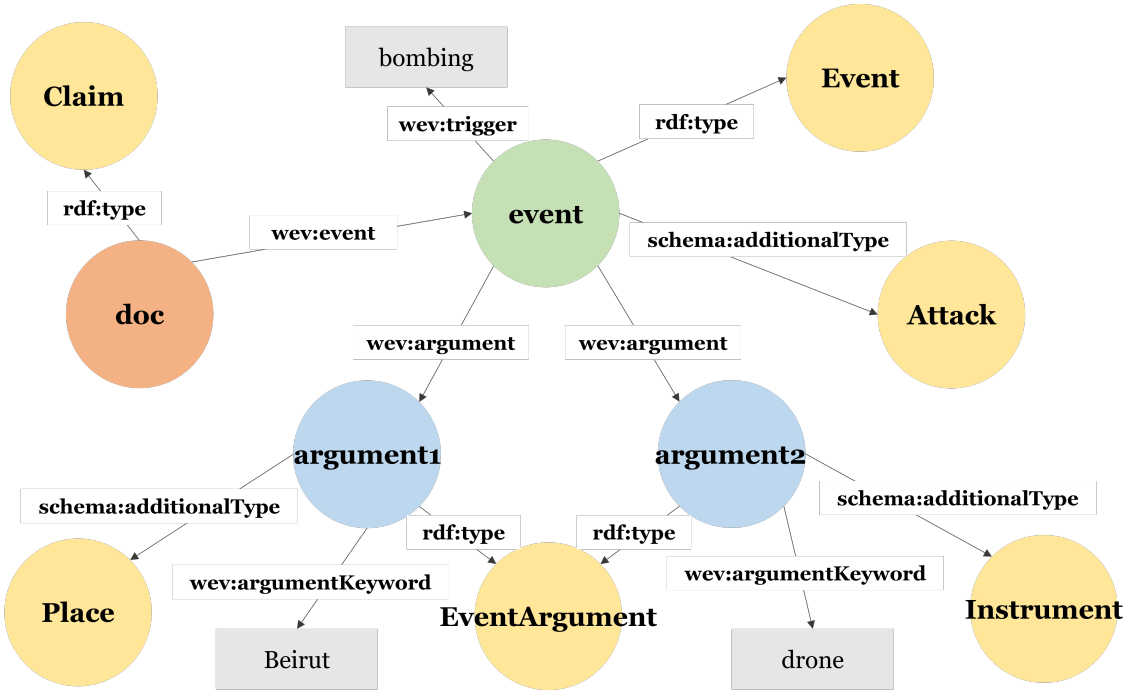
Трансформацията на данните е постигната с помощта на *OntoRefine*[14] - инструмент за преобразуване на данни от табличен, *XML*[15] и *JSON*[16] формат в мрежа от знания.

Фигура 10 илюстрира връзките между данните в мрежата от знания. Кръговете на фигурата показват ресурсите, а правоъгълниците - литералите. Посочен е пример за документ, в който е разпознато събитие от тип *Attack* с активираща дума "*bombing*"и два аргумента - *Place* ("*Beirut*") и *Instrument* ("*drone*").

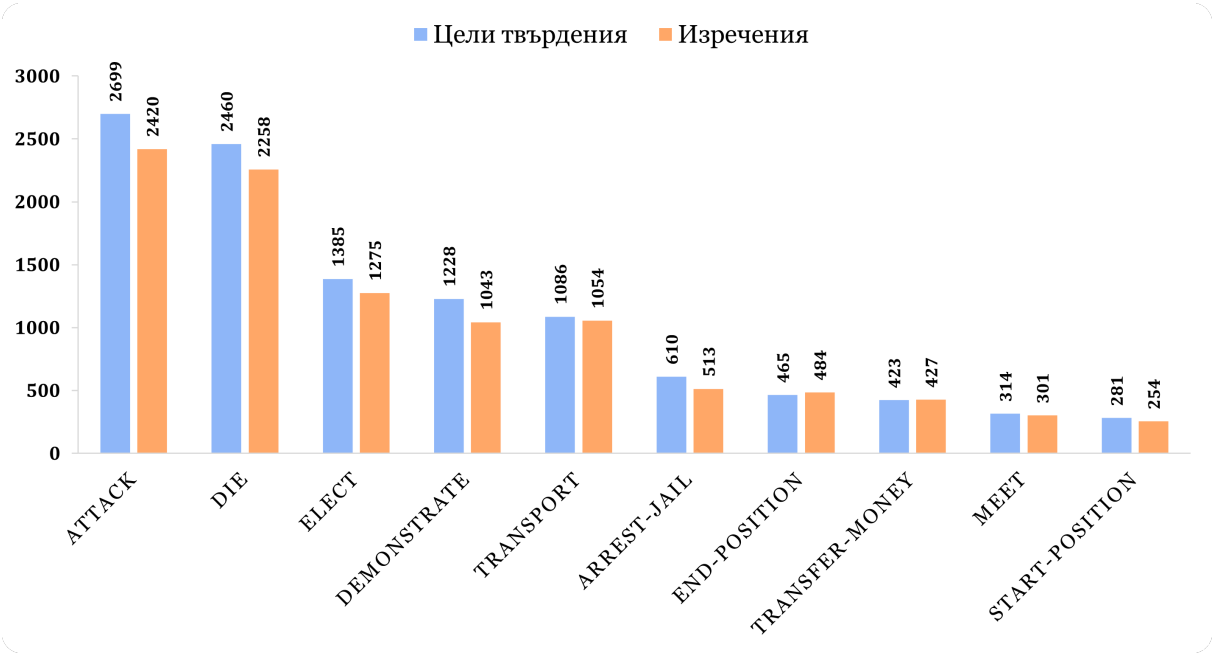
Представените в *RDF* данни са вмъкнати в хранилище в в *GraphDB*[17] - вид графова база от данни, след което са анализирани с помощта на заявки на езика *SPARQL* (*SPARQL Protocol and RDF Query Language*)[18] - семантичен език за заявки към бази от данни, съхраняващи данни в *RDF* формат.

Фигура 11 показва честотата на срещане на десетте най-срещани събития в изследваните данни според избрания модел.

[13] `https://www.w3.org/RDF/`
[14] `https://www.ontotext.com/products/ontotext-refine/`
[15] `https://www.w3.org/XML/`
[16] `https://www.json.org/json-en.html`
[17] `https://www.ontotext.com/products/graphdb/`
[18] `https://www.w3.org/TR/rdf-sparql-query/`



Фигура 10: Модел на данните в мрежата от знания



Фигура 11: Честота на срещане на десетте най-срещани събития от *ACE2005* в данните от *DBKF*

3.2 Дефиниране на нови типове събития в данните

3.2.1 Методология за дефиниране на нови типове събития

Изискванията към новите типове събития, които ще бъдат дефинирани и разглеждани в следващите стъпки, са те да бъдат релевантни към областта на фалшивите новини и дезинформацията, да бъдат от интерес за потенциалните заинтересовани лица и да бъдат достатъчно често срещани в наличните данни.

Като първа стъпка от процеса се генерират предложения за нови типове събития, представляващи потенциален интерес. След това за тези събития се изготвят списъци с подходящи кандидат-активиращи думи и на тяхна база се извършва търсене по ключови думи в мрежата от знания.

Въз основа на количествените резултати от изпълнените заявки, както и оценка на приоритетността на и интереса към всеки кандидат-тип, са подбрани три нови типа събития - *Cure-Claim* (Твърдение-за-лечение, *Severe-Weather* (Опасно-време) и *Rule-Change* (Промяна-на-правилата), които ще бъдат разгледани в следващите подсекции и ще бъдат изследвани оттук нататък.

3.2.2 Тип събитие *Cure-Claim*

Дефиниция на тип събитие *Cure-Claim* Събитие от тип *Cure-Claim* настъпва, когато някой или нещо (напр. статия, публикация в социалните медии и др.) твърди, че нещо е лек за дадено медицинско състояние.

Аргументи Идентифицирани са следните шест типа аргументи към типа събитие *Cure-Claim*:

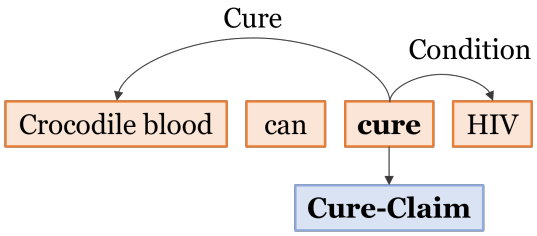
- ***Source*** - източникът на твърдението;
- ***Cure*** - лекът, за който се твърди, че лекува съответното състояние;
- ***Condition*** - състоянието, което се твърди, че се лекува;
- ***Patient*** - лекуваният;
- ***CureCreator*** - създателят на лека;

- ***CureAdministrator*** - този, който прилага лека.

Потенциални активиращи думи Следните думи (и техните производни) са потенциални активиращи думи за събития от типа *Cure-Claim* и са използвани при подготовката на корпуса от данни:

boost, cure, heal, relieve, remedy, treat, treatment

Примери Фигура 12 показва изречение, в което се среща събитие от тип *Cure-Claim* с активираща дума "*cure*". Илюстрираното събитие има два аргумента - "*Crocodile blood*"от тип *Cure* и "*HIV*"от тип *Condition*.



Фигура 12: Пример за изречение, в което се среща събитие от тип *Cure-Claim*

3.2.3 Тип събитие *Severe-Weather*

Дефиниция на тип събитие *Severe-Weather* Събитие от тип *Severe-Weather* настъпва, когато се споменава настъпването на тежки метеорологични феномени, които могат да нанесат материални щети и да бъдат опасни за живота, като например торнада, гръмотевични бури, порои и наводнения, екстремни суши и горещини и др.[19]

Аргументи Идентифицирани са следните пет типа аргументи към типа събитие *Severe-Weather*:

- ***Source*** - източникът на твърдението за настъпването на метеорологичния феномен;
- ***Place*** - мястото, където е настъпил феноменът;

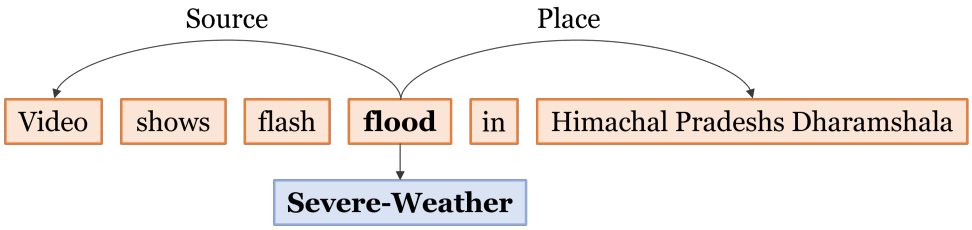
[19] `https://web.archive.org/web/20170103094209/https://www.wmo.int/pages/prog/www/DPS/Meetings/Wshop-SEEF_Toulouse2004/Doc3-1%281%29.doc`

- ***Time*** - кога е настъпил феноменът;
- ***Target*** - потърпевшият от настъпването на феномена (може да е както живо същество, така и неодушевен обект, например сграда);
- ***Cause*** - причината за настъпването на феномена;
- ***NamedPhenomenon*** - името на настъпилия феномен, в случай, че има такова.

Потенциални активиращи думи Следните думи (и техните производни) са потенциални активиращи думи за събития от типа *Severe-Weather* и са използвани при подготовката на корпуса от данни:

avalanche, cold wave, cyclone, drought, firenado, flood, fog, heat wave, hurricane, lightning, thunderstorm, tornado, whirlpool, wildfire

Примери Фигура 13 показва изречение, в което се среща събитие от тип *Severe-Weather* с активираща дума "*flood*". Илюстрираното събитие има два аргумента - "*Video*"от тип *Source* и "*Himachal Pradeshs Dharamshala*"от тип *Place*.



Фигура 13: Пример за изречение, в което се среща събитие от тип *Severe-Weather*

3.2.4 Тип събитие *Rule-Change*

Дефиниция на тип събитие *Rule-Change* Събитие от тип *Rule-Change* настъпва, когато някакъв закон, правило, нормативна уредба и т.н., е въведен, променен или премахнат.

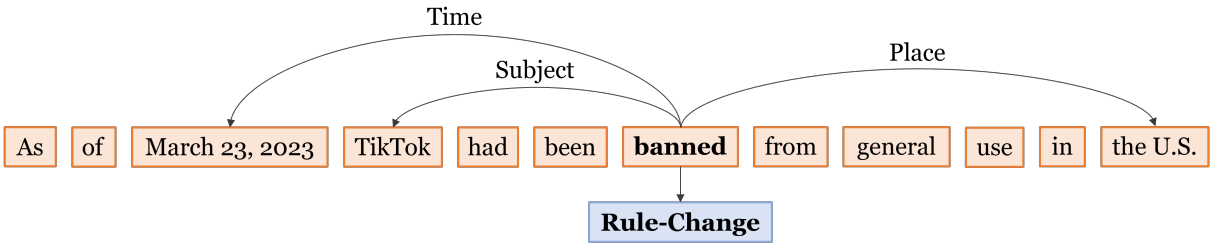
Аргументи Идентифицирани са следните пет типа аргументи към типа събитие *Rule-Change*:

- ***Source*** - източникът на твърдението за промяната на закона, правилото и т.н.;
- ***Place*** - мястото, където е въведена промяната (напр. държава, град) или което ще бъде повлияно от въведената промяна;
- ***Time*** - кога е въведена промяната;
- ***Agent*** - извършителят на промяната (напр. правителство, организация, личност и др.);
- ***Subject*** - предметът на въведената промяната.

Потенциални активиращи думи Следните думи (и техните производни) са потенциални активиращи думи за събития от типа *Rule-Change* и са използвани при подготовката на корпуса от данни:

abolish, adopt, amend, approve, authorize, ban, bar, bill, enact, decree, decriminalize, disallow, law, legalize, legislation, outlaw, overrule, pass, permit, prohibit, ratify, repeal, regulate, sign, uphold, veto

Примери Фигура 14 показва изречение, в което се среща събитие от тип *Rule-Change* с активираща дума "*banned*". Илюстрираното събитие има три аргумента - "*March 23, 2023*"от тип *Time*, "*TikTok*"от тип *Subject* и "*the U.S.*"от тип *Place*.



Фигура 14: Пример за изречение, в което се среща събитие от тип *Rule-Change*

3.3 Анотиране на корпус от данни за новите типове събития

След като новите типове събития са идентифицирани и дефинирани, следва да бъде подготвен анотиран корпус от данни за тях, на базата на който ще бъдат обучени модели за автоматичното им откриване.

Анотирането на данни представлява процеса по поставяне на етикети на данните, класифициращи ги в различни категории.

В случая всеки документ ще бъде анотиран за типа събитие, който споменава, неговата активираща дума, както и всички споменати аргументи, принадлежащи към съответния тип събитие. По този начин полученият анотиран корпус от данни ще може да се използва както при обучение на модели за откриване на събития, така и на модели за цялостно извличане на събития.

Целият процес по анотиране на данните включва следните стъпки, които ще бъдат описани по-подробно в отделни подсекции:

1. Избор на система за ръчно анотиране на данните
2. Подбор на документите за анотиране
3. Разработване на указания за анотиране
4. Ръчно анотиране на данните от трима анотатори

Стъпки 2, 3 и 4 са извършени последователно за всеки един от новодефинираните типове събития.

3.3.1 Избор на система за ръчно анотиране на данните

Системата за ръчно анотиране на данните трябва да предоставя удобен графичен интерфейс за маркиране и анотиране на отделните части на документите, както и възможност за независимо анотиране от няколко анотатори.

За целите на процеса е избрана системата *Ontotext Metadata Studio* (*OMDS*)[20]. Тя предоставя набор от функционалности, позволяващи както ръчно анотиране на данните, така и допълнителното им обогатяване от различни услуги за извличане на информация от текст. В случая са използвани само възможностите за ръчно анотиране на текст.

В *OMDS* анотациите могат да бъдат на ниво документ (*document annotations*) или на ниво подниз от документа (*inline annotations*). Използвани са и двата вида анотации, като:

[20] `https://www.ontotext.com/products/ontotext-metadata-studio/`

- за типа събитие, който се среща в текста, се използва анотация на ниво документ;
- за активиращата дума, аргументите и обхвата на събитието се използват анотации на ниво подниз от документа.

3.3.2 Подбор на документите за анотиране

Документите за ръчно анотиране за всеки тип са избрани от подмножество на документите, извлечени от *DBKF*, което ще наричаме подмножество на кандидатите за този тип. Всяко подмножество на кандидатите е получено чрез търсене в мрежата от знания, използващо потенциалните активиращи думи за съответния тип като ключови думи.

Данни за размерите на подмножествата на кандидатите за всеки нов тип събития и броя избрани от тях документи за анотация са представени в Таблица 3.

Таблица 3: Данни за подмножествата на кандидатите за всеки нов тип и избраните документи за анотация

	Cure-Claim	***Severe-Weather***	***Rule-Change***
Брой документи в подмножеството на кандидатите	639	259	3038
Брой документи, избрани за анотация	65	67	67
Процент на избраните за анотация документи	10%	26%	2%

Ръчното анотиране на данни е бавен и трудоемък процес. Поради тази причина за всеки от трите типа събития са подбрани само по 65-67 документа за анотация, като е съблюдавано да има както позитивни, така и негативни примери, и също да има добро представяне на различните възможни активиращи думи.

3.3.3 Разработване на указания за анотиране

Разработени са подробни указания за анотиране, базирани на официалните указания за анотиране на корпуса от данни *ACE2005*[21]. Освен синтезирана версия на общова-

[21] `https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-events-guidelines-v5.4.3.pdf`

лидните правила за анотиране на събития от официалните указания, новосъздадените включват подробни насоки за анотиране на всеки един от новите типове събития с примери, илюстриращи по-особени случаи, характерни за всеки един отделен тип. Разработените указания за анотация са предоставени в Приложение 3.

3.3.4 Ръчно анотиране на данните

Данните са анотирани независимо от трима анотатори (единият от които е авторът на дипломната работа), поради което полученият корпус от данни може да се смята за т.нар. златен стандарт.

Различията в анотациите на анотаторите са разрешени посредством дискусии, като за финални се взимат доминиращите анотации и/или анотациите, които найточно следват указанията за анотиране.

3.4 Характеристики на новия корпус от данни

След края на процеса по анотиране на данните разполагаме с набор от анотирани данни за трите типа нови събития. Таблица 4 показва основни описателни статистики за новия корпус от данни, който ще наричаме *EXTEND* (*EXTended EveNts Dataset*).

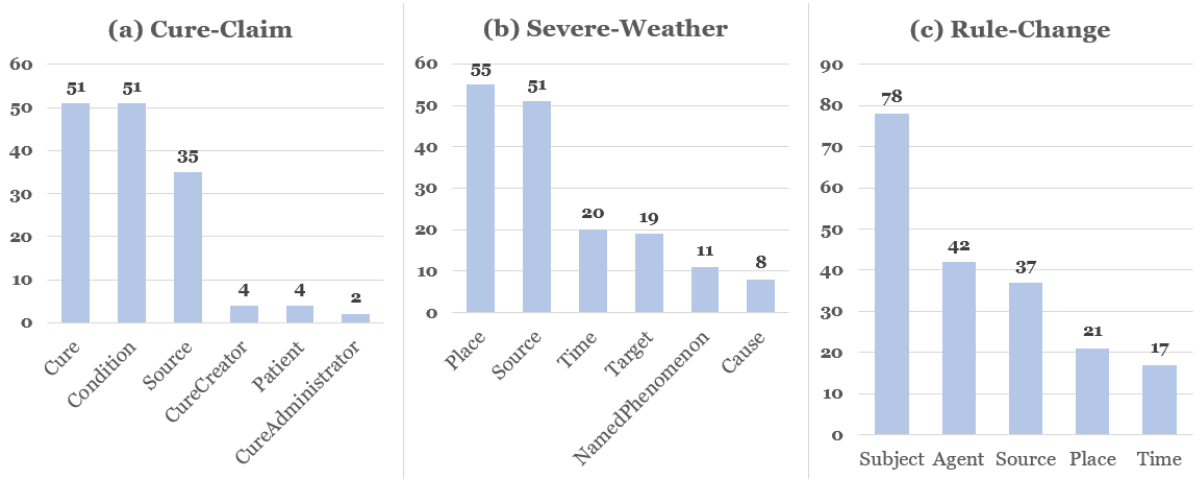
Тъй като процесите по подбор на документи за анотация и анотирането им се извършват последователно за отделните типове събития, в общия корпус могат да бъдат разграничени отделни части, които ще наричаме съответно *EXTEND-CC* за *Cure-Claim*, *EXTEND-SW* за *Severe-Weather* и *EXTEND-RC* за *Rule-Change*. Всеки един от тези под-корпуси може да бъде използван за обучение на модели, които разпознават само конкретния тип събитие, за който се отнася даденият под-корпус.

Таблица 4: Описателни статистики за новия корпус от данни *EXTEND*

	EXTEND-CC	***EXTEND-SW***	***EXTEND-RC***	***EXTEND***
Документи	65	67	67	199
Изречения	81	75	92	248
Активиращи думи	54	52	52	158
Аргументи	147	164	195	506

Фигура 15 показва честотата на срещане на типовете аргументи по типове съ-

бития, съответно (*a*) за тип *Cure-Claim*, (*b*) за тип *Severe-Weather*, (*c*) за тип *Rule-Change*.



Фигура 15: Честота на срещане на типовете аргументи по типове събития

В Таблица 5 е представено сравнение по описателни статистики между *ACE2005* корпуса от данни и новия корпус от данни *EXTEND*. Трябва да се отбележи, че три от тринадесетте аргумента от *EXTEND* присъстват и в *ACE2005*. Това са *Place*, *Time* и *Target*.

Таблица 5: Сравнение по описателни статистики между *ACE2005* корпуса от данни и новия корпус от данни *EXTEND*

	ACE2005	***EXTEND***
Документи	599	199
Изречения	16 372	248
Типове събития	33	3
Типове аргументи	36	13
Активиращи думи	5 272	158
Брой аргументи	9 612	506
Среден брой събития за тип	159,75	52,67
Среден брой аргументи за тип	267	38,93
Среден брой изречения на документ	27,33	1,25

4 Обучение на модели за откриване на събития в текст

4.1 Методи за допълнително обучение (*fine-tuning*) на модели за автоматично откриване на събития в текст

Допълнителното обучение (*fine-tuning*) е подход за прилагане на трансферно обучение, при който предварително обучен върху дадено изходно множество данни модел се дообучава върху друго целево множество от данни с цел да преизползва натрупаните вече знания. В случаите, в които целевото множество е много по-малко по размер от изходното, процесите по допълнително обучение спомагат за по-добрата генерализация на модела.[22].

Тъй като разполагаме с корпус от данни, който съдържа около 30 пъти по-малко събития в сравнение с *ACE2005* (както беше показано в Таблица 5), е предприета стратегия за допълнително обучение на модел, предварително обучен върху данните от *ACE2005*.

4.1.1 Избор на модел за допълнително обучение за автоматично откриване на новите типове събития от *EXTEND*

На база направения обзор на съществуващите методи за автоматично откриване на събития в текст и заключението от него, представено в подсекция 2.3.4, е избран подходът *Text2Event* [29] за допълнително обучение за автоматично откриване на новите типове събития от *EXTEND*. В частност е избран моделът *dyiepp_ace2005_en_t_large*[23], който е предварително обучен върху данните на английски език за всички 33 типа събития от *ACE2005*.

[22] `https://d2l.ai/chapter_computer-vision/fine-tuning.html`
[23] `https://github.com/luyaojie/Text2Event`

4.1.2 Особености на избрания за допълнително обучение модел

Както бе посочено в обзора, Lu et al. [29] разглеждат задачата за извличане на събития като генеративна и използват архитектура, базирана на трансформатори [40], в частност *T5* [41].

Text2Event извлича събитията от текста от край до край (*end-to-end*), като генерира пълната им структура в дървовиден формат, представен на Фигура 16. Таговете *<extra_id_0>* и *<extra_id_1>* са респективно отварящи и затварящи. В представения пример има две отделни събития, като всяко от тях е представено чрез отделен под-елемент от двойка *(тип събитие, активираща дума)*. Аргументите към всяко събитие са представени чрез вложени под-елементи от двойки *(тип аргумент, дума/фраза за аргумент)*.

```
<extra_id_0>
    <extra_id_0> Meet meeting
        <extra_id_0> Entity EU <extra_id_1>
    <extra_id_1>
    <extra_id_0> Start-Position hire
        <extra_id_0> Person Fischer<extra_id_1>
    <extra_id_1>
<extra_id_1>
```

Фигура 16: Структура на събития, извлечени чрез *Text2Event* модел

Ако в даден текст няма споменато събитие, се генерира низът "*<extra_id_0> <extra_id_1>*".

При обучение на модела се подава т.нар. схема на събитията (*event schema*). Под формата на списъци и речници от езика *Python* тя съдържа типовете събития, за които моделът ще се обучава, типовете аргументи и връзките между събития и аргументи. Фигура 17 представя примерна схема на събитията, съдържаща два типа събития и техните аргументи. Първият списък съдържа типовете събития, вторият - типовете аргументи, а речникът показва кои аргументи към кои типове събития се отнасят.

За обучение на моделите авторите прилагат две различни стратегии - обучение

```
["Cure-Claim", "Severe-Weather"]

["Condition", "Cure", "CureAdministrator", "CureCreator",
      "Patient", "Source", "Place", "Time", "Target",
      "NamedPhenomenon", "Cause"]

{"Cure-Claim": ["Condition", "Cure", "CureAdministrator",
      "CureCreator", "Patient", "Source"],
 "Severe-Weather": ["Source", "Place", "Time", "Target",
      "NamedPhenomenon", "Cause"]}
```

Фигура 17: Примерна схема на събитията за *Text2Event*

за пълни структури (*full structure learning*) и смесено обучение (*curriculum learning*). Смесеното обучение включва в себе си обучение за пълни структури, предшествано от обучение за подструктури (*substructure learning*).

Обучението за подструктури цели да обучи модела да генерира прости подструктури на събитието като двойките *(тип събитие, активираща дума)*, *(тип аргумент, дума/фраза за аргумент)*. Обучението за пълни структури цели да обучи модела да генерира пълната структура на събитието.

Също така в [29] са описани възможностите на *Text2Event* моделите да се обучават, като се прилагат методи за трансферно обучение. Авторите описват експерименти, при които използват данните за десетте най-често срещани типове събития от *ACE2005* за предварително обучение на модели, които след това обучават допълнително за останалите 23 типа събития. По този начин са постигнати *F1* оценки за класификация на активиращата дума и аргументите по-високи с респективно 3.7% и 3.2%.

4.1.3 Подход за допълнително обучение на модела

За допълнителното обучение на избрания модел е приложено само обучение за пълни структури, тъй като моделът вече е предварително обучен за подструктури.

Обучени са отделни модели, разпознаващи всеки един от новите типове събития, както и общи модели, разпознаващи два или всички нови типове събития.

Изследвани са различни количествени разпределения на данните в тренировъчно и тестово множество, както и различен брой епохи за обучение, за да се определят

оптималните им стойности. Тъй като количеството анотирани данни за новите типове събития от *EXTEND*, е много малко, е приложена *5-fold* крос-валидация. Дизайнът и резултатите от експериментите са подробно описани в секция 5.1.

На база резултатите от експериментите с крос-валидация са обучени модели, разпознаващи по отделно всеки един от новите типове събития или два или всички типове наведнъж, като са използвани 90% от наличните данни за обучение и 10% за валидация. Моделите са обучавани за различен брой епохи с цел да се открие оптимален модел, който разпознава новите типове събития и същевременно запазва възможностите си да разпознава събитията от *ACE2005*.

За целите на допълнителното обучение на *Text2Event* модела за новите типове събития е разширен кодът от официалното *GitHub* хранилище на *Text2Event*.[24]

Обучението на моделите изисква графичен процесор (*GPU*). Всички модели са обучавани на *NVIDIA RTX A5000 GPU*. Използвани са *learning rate* 1e-4 и *batch size* 16.

Резултатите от всички експерименти са описани в секция 5.1.

4.2 Методи без предварително обучение, използващи големи езикови модели

Големите езикови модели (*large language models*) са езикови модели, които обработват и генерират текст на естесвен език. Обикновено големите езикови модели са обучавани върху огромни количества текстови данни. [46] Архитектурата им най-често е базирана на трансформатори и механизми с внимание.

В последните няколко години големите езикови модели набират все по-голяма популярност. Това се дължи до голяма степен на възможностите им с помощта на подходящи инструкции (*prompts*) да постигат сходни резултати със *state-of-the-art* модели за конкретни задачи, но без необходимостта от предварително обучение или само с малко предварително обучение. [47]

В рамките на дипломната работа прилагаме и методи без предварително обучение, използващи големи езикови модели. Целите са да установим доколко един голям

[24] `https://github.com/luyaojie/Text2Event`

езиков модел се справя със задачите за откриване и извличане на събития без предварително обучение и да проверим дали подобен подход може да се използва като автоматичен анотатор на данните за по-нататъшно обучение на други модели.

4.2.1 Избор на голям езиков модел

За целите на изследването е избран езиковият модел на *Meta* - *LLaMA* (*Large Language Model Meta AI*), и по-точно най-новата му версия *LLaMA-2*, издадена през юли 2023. [48]

LLaMA-2 е с отворен код. Достъпни са модели с различен брой параметри - 7, 13 и 70 милиарда. В случая е избран моделът със 70 милиарда параметъра поради по-добрите му възможности за справяне с логически и фактологически въпроси.[25]

4.2.2 Подход без предварително обучение

Приложен е подход без предварително обучение за всеки тип от *EXTEND* по отделно. *LLaMA-2* моделът със 70 милиарда параметъра (*LLaMA-2-70b-chat*[26]) е използван чрез *replicate*[27] програмен интерфейс (*API*) за програмния език *Python*. Моделът се изпълнява на отдалечен графичен процесор - *Nvidia A100 (80GB) GPU*.

Зададени са инструкции към модела, които обясняват естеството на задачата, типовете събития и техните аргументи. Фигура 18 показва инструкциите, които се изпращат на модела за събитията от тип *Severe-Weather* (за останалите типове инструкциите са аналогични). На мястото на *text* се поставя всеки документ, който трябва да бъде обработен от модела.

В допълнение, под формата на системни инструкции (*system prompt*) се задава ролята, от гледна точка на която моделът трябва да даде своите отговори - в този случай проверител на факти. За да бъдат отговорите възможно най-детерминистични, температурата на модела е зададена като минималната възможна стойност. Фигура 19 представя сегмент от кода, който се използва за изпращане на заявка към модела с описаните параметри.

[25] `https://replicate.com/blog/how-to-prompt-llama#7b-v-13b-v-70b`
[26] `https://huggingface.co/meta-llama/Llama-2-70b-chat`
[27] `https://replicate.com/`

```
prompt_template_sw = """Is there an event of type Severe-Weather in TEXT?
       An Event is a specific occurrence involving participants.
       An event of type Severe-Weather occurs when the occurrence of
       severe weather phenomena is mentioned.
       An event of type Severe-Weather has arguments of types: Place,
       Time, Target, Cause, Source.
       Answer with True/False to the question: Is there an event of type
       Severe-Weather in TEXT?
       If the answer to the previous question is True, extract all
       arguments of the event, mentioned in TEXT. Answer in the format
       ["argumentType", "argument"] without any explanation. If no
       arguments of a given type are mentioned, do not give an answer for
       this type. If no arguments at all are mentioned, then only answer
       "[]".
       TEXT: ```{text}.```"""
```

Фигура 18: Инструкции към *LLaMA-2-70b-chat* за разпознаване на събития от тип *Severe-Weather*

```
output = replicate.run(
       llama2_70b,
       input={
              "prompt": prompt,
              "system_prompt": """You are a fact-checker. Your task is to
              answer to the questions asked in the exact format requested.
              Do not add any additional words and explanations.""",
              "temperature": 0.01
        }
)
```

Фигура 19: Изпращане на заявка към *LLaMA-2-70b-chat* чрез *replicate*

Резултатите от експериментите с *LLaMA-2* са описани в секция 5.2.

5 Проведени експерименти и анализ на резултатите от тях

5.1 Експерименти с допълнително обучените *Text2Event* модели

5.1.1 Първоначално проучване за осъществимост върху типа *Cure-Claim*

Преди да се пристъпи към обучение на модели за всички нови типове събития, са проведени експерименти само с първия от тях - *Cure-Claim*, използвайки документи само от *EXTEND-CC*, за да се установи до каква степен избраният подход за допълнително обучение е подходящ и постига добри резултати.

Проведените експерименти са описани в [3]. Първоначално са проведени експерименти с крос-валидация с различни разпределения на данните в обучително и тестово множество, след това са обучени модели върху всички анотирани данни от *EXTEND-CC* и за тези модели е направена оценка доколко запазват възможностите си за разпознаване на типовете събития от *ACE2005*.

5.1.1.1 Експерименти с приложена крос-валидация Експериментите с приложена крос-валидация са проведени с цел изследване на представянето на обучаваните модели с различно количество обучителни данни и различен брой епохи за обучение.

Дизайн на експериментите Изследвани са четири различни количествени разпределения на данните в обучителното и тестовото множество от вида *X:Y*, където X е процентът от данните, използвани за обучителното множество, а Y - процентът от данните, използвани за тестовото множество. Четирите изследвани разпределения са съответно 20:80, 40:60, 60:40 и 80:20.

За всяко от разпределенията е извършена *5-fold* крос-валидация, като процедурата включва следните стъпки:

1. Отделят се $Y\%$ от анотираните данни за тестово множество.

2. Оставащите $X\%$ от анотираните данни се използват за обучителни и валидационни цели. Тъй като се прилага *5-fold* крос-валидация, пет пъти се заделят по 20% от X като валидационно множество, така че всеки един документ (без тези в тестовото множество) да участва във валидационно множество точно веднъж.

Моделите са обучавани съответно за 30, 100, 300 и 500 епохи.

Метрики Докладват се осреднените метрики прецизност, връщане и *F1* оценка, както са дефинирани в 2.1.3, за следните три задачи:

- класификация на типа събитие;
- класификация на типа събитие + разпознаване на активиращата дума;
- класификация на аргументите.

За отделните задачи вярно и грешно положителните и съответно отрицателните примери се дефинират по различен начин.

Класификация на типа събитие Предсказаните от модела примери (на ниво събитие в документ) се класифицират по следния начин:

- Вярно положителни (*True Positive*, *TP*) - има анотирано събитие от дадения тип в документа и моделът е предсказал типа му правилно;
- Вярно отрицателни (*True Negative*, *TN*) - няма анотирано събитие от дадения тип в документа и моделът не е предсказал такова;
- Грешно положителни (*False Positive*, *FP*) - няма анотирано събитие от дадения тип в документа, но моделът е предсказал такова;
- Грешно отрицателни (*False Negative*, *FN*) - има анотирано събитие от дадения тип в документа, но моделът не е предсказал такова.

Класификация на типа събитие + разпознаване на активиращата дума Предсказаните от модела примери (на ниво събитие и активираща дума) се класифицират по следния начин:

- Вярно положителни (*True Positive*, *TP*) - има анотирано събитие от дадения тип и моделът е предсказал типа и активиращата дума правилно;
- Вярно отрицателни (*True Negative*, *TN*) - няма анотирано събитие от дадения тип и моделът не е предсказал такова;
- Грешно положителни (*False Positive*, *FP*) - няма анотирано събитие от дадения тип, но моделът е предсказал такова ИЛИ има анотирано събитие от дадения тип и моделът е предсказал такова, но с грешна активираща дума;
- Грешно отрицателни (*False Negative*, *FN*) - има анотирано събитие от дадения тип, но моделът не е предсказал такова.

Класификация на аргументите Разглеждат се следните четири сценария:

1. Анотирано събитие е предсказано от модела с правилна активираща дума.
2. Анотирано събитие е предсказано от модела с грешна активираща дума.
3. Има анотирано събитие от дадения тип, но моделът не е предсказал такова. В този случай считаме събитието и всичките му анотирани аргументи като грешно отрицателни.
4. Няма анотирано събитие от дадения тип, но моделът е предсказал такова. В този случай считаме събитието и всичките му предсказани от модела аргументи като грешно положителни.

За първите два случая предсказаните от модела примери (на ниво аргументи) се класифицират по следния начин[28]:

- Вярно положителни (*True Positive*, *TP*) - предсказаният аргумент съвпада по тип и обхват с анотиран аргумент;
- Грешно положителни (*False Positive*, *FP*) - предсказаният аргумент съвпада по тип, но не и по обхват с анотиран аргумент ИЛИ предсказаният аргумент не съвпада с анотиран аргумент нито по тип, нито по обхват;

[28]За предсказаните аргументи не се разглежда категорията вярно отрицателни.

- Грешно отрицателни (*False Negative*, *FN*) - за дадения анотиран аргумент няма предсказан такъв, който да съвпада с него по тип и/или обхват.

Трябва да се отбележи, че сравнението на аргументите е строго, т.е. за да приемем, че предсказан аргумент съвпада с анотиран по обхват, е необходимо да има пълно съвпадение на низовете.

За изчислението на всички описани метрики са написани функции на програмния език *Python*.

Резултати и анализ Таблица 6 показва получените осреднени резултати (прецизност, връщане и *F1* оценка) от първоначалните експерименти с крос-валидация за типа *Cure-Claim*. Резултати са представени за класификация на типа събитие (колона *Събитие*), класификация на типа събитие и разпознаване на активиращата дума (колона *Събитие + Акт.*), класификация на аргументите при правилно разпозната активираща дума (колона *Аргументи+*) и при неправилно разпозната активираща дума (колона *Аргументи-*).

Резултатите от експериментите с допълнително обучение за 30 епохи не са включени в таблицата, т.к. за всички разпределения на данните обучените модели постигат резултат 0 за всички докладвани метрики.

Таблица 6: Осреднени резултати от първоначална крос-валидация - прецизност (P), връщане (R) и *F1* оценка за събития от тип *Cure-Claim* [3]

Модел		**Събитие**			**Съб. + Акт.**			**Аргументи+**			**Аргументи-**		
	X:Y	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
100 епохи	20:80	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	40:60	0.60	0.02	0.04	0.60	0.02	0.04	0.30	0.60	0.40	0.00	0.00	0.00
	60:40	0.83	0.84	0.82	0.76	0.83	0.78	0.54	0.75	0.63	0.75	0.83	0.79
	80:20	0.84	0.83	0.83	0.78	0.82	0.80	0.52	0.72	0.59	0.67	0.70	0.68
300 епохи	20:80	0.87	0.65	0.73	0.75	0.62	0.66	0.38	0.50	0.43	0.24	0.44	0.31
	40:60	0.80	0.78	0.79	0.71	0.76	0.73	**0.60**	0.74	**0.64**	0.43	0.76	0.55
	60:40	0.86	0.75	0.80	0.80	0.74	0.76	0.53	0.70	0.60	0.83	0.88	0.85
	80:20	0.84	0.83	0.84	0.80	0.82	0.81	0.53	0.73	0.61	0.60	0.60	0.60
500 епохи	20:80	0.85	0.69	0.75	0.72	0.66	0.68	0.48	0.60	0.53	0.32	0.54	0.40
	40:60	0.82	0.81	0.81	0.71	0.78	0.74	0.56	0.71	0.62	0.48	0.79	0.59
	60:40	**0.88**	0.83	0.85	**0.83**	0.82	0.82	0.55	0.74	0.63	**1.00**	**1.00**	**1.00**
	80:20	0.86	**0.92**	**0.89**	0.77	**0.92**	**0.84**	0.57	**0.76**	**0.64**	0.93	**1.00**	0.96

Трябва да се отбележи, че стандартните отклонения на докладваните метрики за класификация на събитие, класификация на събитие и активираща дума и класификация на аргументи при правилно разпозната активираща дума са в интервала от 0.006 до 0.15 с единствено отклонение (*outlier*) 0.48. За класификация на аргументи при неправилно разпозната активираща дума стандартните отклонения са в интервала от 0.06 до 0.48. Това може да се обясни с факта, че много малко примери попадат в тази група.

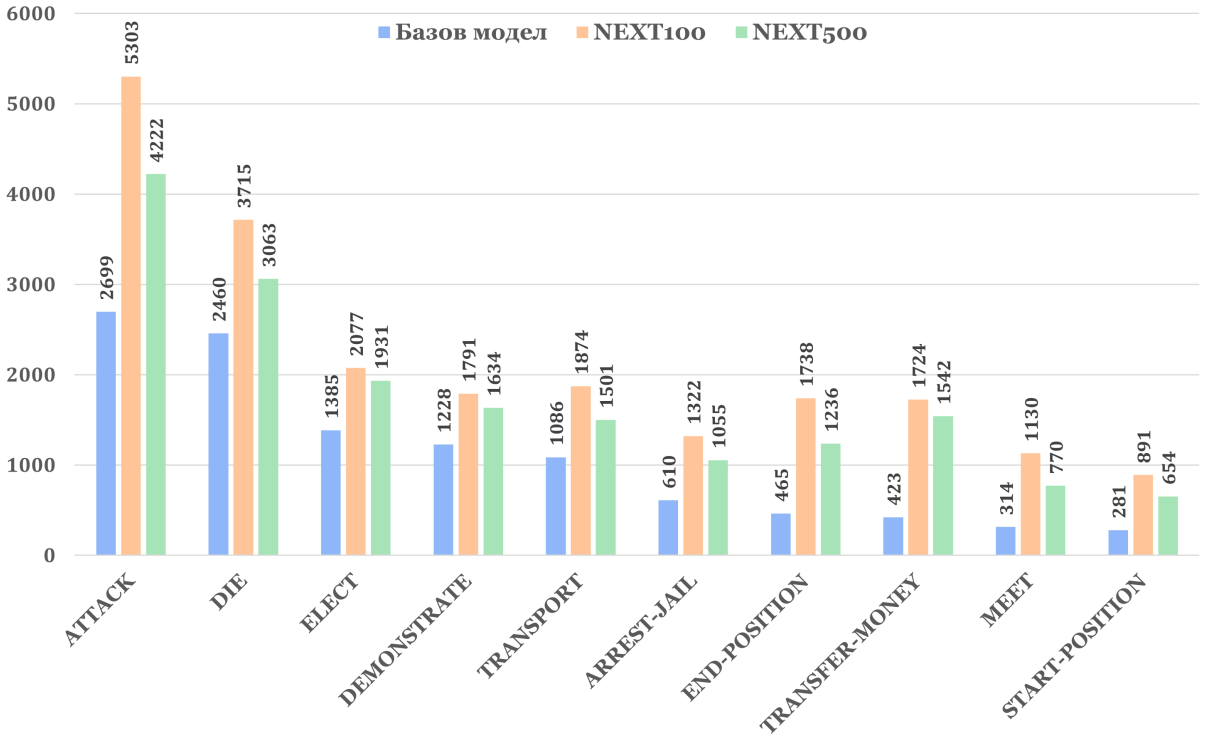
За представените резултати се наблюдава тенденцията те да се подобряват с увеличаване на броя документи в обучителното множество и на броя епохи за обучение. Моделите, които са обучавани за 300 и 500 епохи, успяват да постигнат високи резултати дори и при най-малкото обучително множество.

5.1.1.2 Експерименти върху типове събития от *ACE2005*

Дизайн на експериментите След проведените експерименти с крос-валидация са обучени допълнително два нови модела, като са използвани 90% от документите в *EXTEND-CC* за обучение и 10% за валидация, съответно за 100 и 500 епохи. Двата модела се наричат респективно *NEXT100* и *NEXT500* (*New Event eXTraction*).

При допълнителното обучение на модели е важно да се направи оценка на това доколко моделът запазва представянето си върху класовете, за които е бил предварително обучаван. В случая това са типовете събития от *ACE2005*. Както бе посочено в 2.2.1, корпусът от данни е достъпен под платен лиценз. Поради тази причина нямаме достъп до него и не е възможно да се направи директна оценка на *NEXT* моделите върху данните от него.

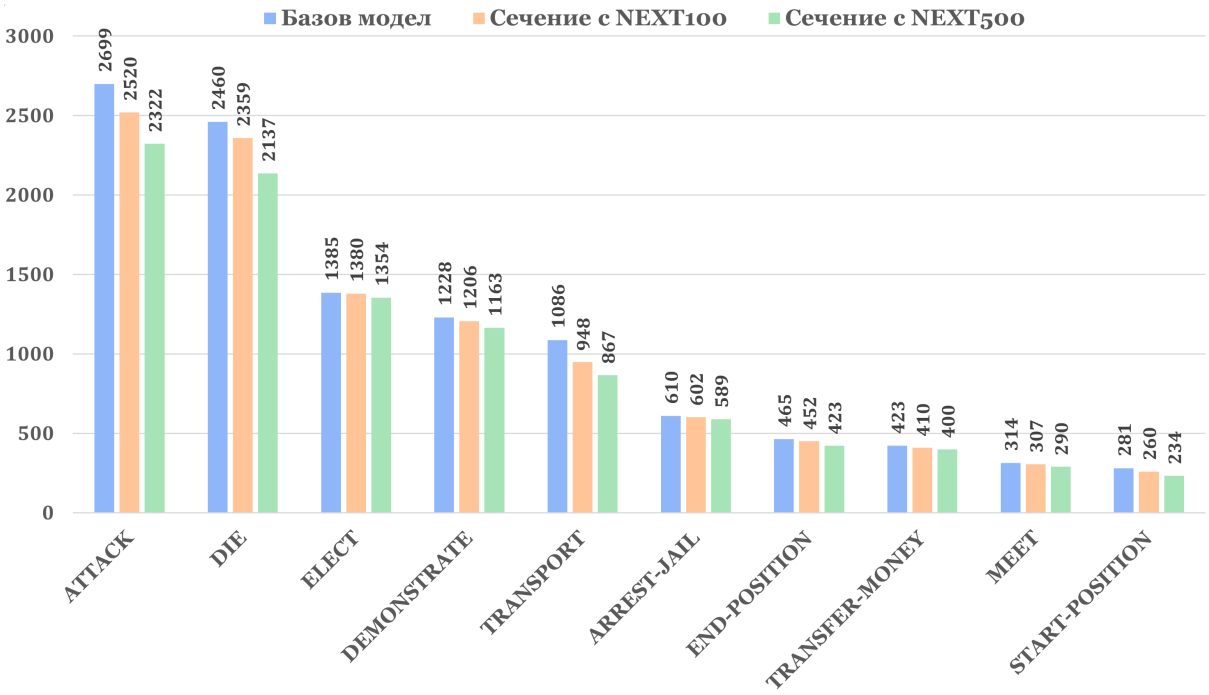
За да се сравнят представяниятa на *NEXT* модели с това на предварително обучения базов *Text2Event* модел върху типовете събития от *ACE2005*, са извършени сравнения върху пълния корпус с извлечени твърдения от *DBKF* (който не е анотиран спрямо въпросните типове събития), като са разгледани десетте най-често срещани в него типове събития от *ACE2005*.



Фигура 20: Брой документи, за които базовият модел, *NEXT100* и *NEXT500* са предсказали наличие на събитие от дадения тип

Резултати и анализ Фигура 20 представя броя документи (изречения), за които базовият модел, *NEXT100* и *NEXT500* са предсказали наличие на събитие от дадения тип. Видно е, че новите модели предсказват повече събития от базовия модел за всеки един от разглежданите типове, като *NEXT100* предсказва повече събития от *NEXT500*.

Допълнително се разглеждат сеченията на множествата от документите, съдържащи събитие от даден тип, предсказани съответно от базовия модел и *NEXT100* и базовия модел и *NEXT500*. Фигура 21 представя броя на документите, за които базовият модел е предсказал наличие на събитие от даден тип, и размерите на съответните сечения. От графиката става ясно, че почти всички прогнози, направени от базовия модел, са направени и от *NEXT* моделите, като сечението на множествата на прогнозите на базовия модел и *NEXT100*, е по-голямо от това на базовия модел и *NEXT500*.



Фигура 21: Брой документи, за които базовият модел е предсказал наличие на събитие от дадения тип, и брой документи в сечението на предсказаните от базовия модел и *NEXT100* и *NEXT500* съответно документи

5.1.2 Експерименти върху всички типове събития от *EXTEND*

След описаните в секция 5.1.1 експерименти може да се приеме, че предложената методология за допълнително обучение на *Text2Event* модел е успешна. Следва да бъдат повторени експериментите и за останалите типове събития, за да се докаже ефективността ѝ и върху по-широк набор от типове.

В допълнение към експериментите за типовете *Severe-Weather* и *Rule-Change* са повторени и експериментите за типа *Cure-Claim*. Разликата с експериментите, описани в секция 5.1.1, се състои в подредбата на аргументите - в първоначалните експерименти те се подават в разбъркан ред, а не в реда на срещане в изречението. При всички експерименти, описани в настоящата секция, аргументите на събитията са подредени в реда на срещане.

5.1.2.1 Експерименти с приложена крос-валидация Проведени са експерименти с крос-валидация за всеки от трите типа събития от *EXTEND* по отделно (т.е.

използвайки *EXTEND-CC*, *EXTEND-SW* и *EXTEND-RC* отделно), за трите типа заедно, както и за комбинацията от типове *Cure-Claim* и *Severe-Weather* (т.е. използвайки *EXTEND-CC* и *EXTEND-SW*).

Дизайн на експериментите и метрики Дизайнът на експериментите и метриките за оценка са същите като описаните в секция 5.1.1.1.

Разглеждаме резултатите от експериментите с модели, обучавани само за един тип събитие, и от експериментите с модели, обучавани за повече от един тип събития, последователно.

Резултати и анализ на експерименти с по един тип събитие Таблици 7, 8 и 9 показват получените осреднени резултати (прецизност, връщане и *F1* оценка) за моделите, обучавани отделно върху *EXTEND-CC*, *EXTEND-SW* и *EXTEND-RC*. Резултати са представени за класификация на типа събитие (колона *Събитие*), класификация на типа събитие и разпознаване на активиращата дума (колона *Събитие + Актив.*) и класификация на аргументите при правилно разпозната акти- вираща дума (колона *Аргументи+*).

Таблица 7: Осреднени резултати от крос-валидация - прецизност (P), връщане (R) и *F1* оценка на модели, обучени върху документи от *EXTEND-CC*

Модел		**Събитие**			**Събитие + Актив.**			**Аргументи+**		
	X:Y	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
100 епохи	20:80	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	40:60	0.75	0.17	0.27	0.60	0.22	0.32	0.40	0.63	0.49
	60:40	0.85	0.58	0.69	0.67	0.54	0.60	0.54	0.71	0.61
	80:20	0.81	0.84	0.82	0.54	0.80	0.65	0.59	0.79	0.67
300 епохи	20:80	0.73	0.48	0.58	0.61	0.47	0.53	0.57	0.70	0.63
	40:60	0.77	0.68	0.72	0.66	0.65	0.65	0.61	0.74	0.67
	60:40	0.85	0.67	0.75	**0.69**	0.64	0.66	0.54	0.71	0.61
	80:20	0.82	0.90	0.86	0.53	0.87	0.66	0.61	0.85	0.71
500 епохи	20:80	0.74	0.55	0.63	0.62	0.53	0.57	0.56	0.68	0.61
	40:60	0.78	0.79	0.79	0.65	0.77	**0.70**	**0.63**	0.77	0.69
	60:40	**0.87**	0.75	0.80	0.67	0.72	0.69	0.60	0.76	0.67
	80:20	0.82	**0.92**	**0.87**	0.52	**0.89**	0.66	0.65	**0.87**	**0.74**

Резултатите от експериментите с допълнително обучение за 30 епохи не са включени в таблиците, т.к. както в първоначалните експерименти за типа *Cure-Claim*, за

всички разпределения на данните обучените модели постигат резултат 0 за всички докладвани метрики.

Таблица 8: Осреднени резултати от крос-валидация - прецизност (P), връщане (R) и *F1* оценка на модели, обучени върху документи от *EXTEND-SW*

Модел		**Събитие**			**Съб. + Акт.**			**Аргументи+**		
	X:Y	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
100 епохи	20:80	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	40:60	0.20	0.01	0.01	0.01	0.01	0.01	0.20	0.20	0.20
	60:40	0.88	0.90	**0.89**	0.66	0.89	0.76	0.43	0.55	0.48
	80:20	0.79	0.98	0.87	0.74	0.98	0.84	0.52	0.65	0.58
300 епохи	20:80	**0.93**	0.73	0.81	0.65	0.70	0.67	0.44	0.52	0.47
	40:60	0.82	0.83	0.83	0.61	0.82	0.70	0.41	0.51	0.45
	60:40	0.87	0.91	**0.89**	0.65	0.90	0.75	0.51	0.62	0.56
	80:20	0.78	**1.00**	0.88	0.71	**1.00**	0.83	0.64	**0.74**	**0.68**
500 епохи	20:80	0.89	0.79	0.84	0.64	0.77	0.70	0.47	0.52	0.49
	40:60	0.83	0.83	0.83	0.63	0.82	0.71	0.40	0.49	0.44
	60:40	0.85	0.93	**0.89**	0.67	0.93	0.78	0.51	0.63	0.56
	80:20	0.81	**1.00**	**0.89**	**0.75**	**1.00**	**0.86**	**0.65**	0.72	**0.68**

Таблица 9: Осреднени резултати от крос-валидация - прецизност (P), връщане (R) и *F1* оценка на модели, обучени върху документи от *EXTEND-RC*

Модел		**Събитие**			**Съб. + Акт.**			**Аргументи+**		
	X:Y	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
100 епохи	20:80	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	40:60	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	60:40	0.66	**0.86**	0.75	0.36	**0.81**	0.50	0.32	0.42	0.36
	80:20	0.67	0.84	0.74	0.25	0.73	0.37	0.38	0.40	0.38
300 епохи	20:80	0.66	0.29	0.40	0.30	0.24	0.27	0.27	0.34	0.30
	40:60	0.71	0.71	0.71	0.36	0.62	0.45	0.33	0.40	0.36
	60:40	0.70	0.79	0.74	**0.42**	0.73	**0.53**	0.34	0.43	0.38
	80:20	**0.73**	0.74	0.73	0.33	0.61	0.42	0.38	0.40	0.38
500 епохи	20:80	0.67	0.33	0.44	0.34	0.28	0.31	0.25	0.33	0.28
	40:60	0.67	0.74	0.70	0.35	0.64	0.45	0.33	0.40	0.36
	60:40	0.65	0.74	0.69	0.39	0.66	0.49	0.36	0.43	0.39
	80:20	**0.73**	0.82	**0.77**	0.39	0.73	0.51	**0.39**	**0.44**	**0.41**

Стандартните отклонения на метриките за моделите, обучавани върху данни от *EXTEND-CC*, са в интервала от 0.01 до 0.12. За моделите, обучавани върху данни от *EXTEND-SW*, те са в интервала от 0.01 до 0.07 с едно единствено отклонение (*outlier*) от 0.40. За моделите, обучавани върху данни от *EXTEND-RC*, стандартните отклонения варират от 0.01 до 0.14 с едно отклонение от 0.22.

С малки изключения отново се наблюдава тенденция успеваемостта на изследваните модели да се повишава с увеличаването на броя документи в обучителното множество и броя епохи за обучение.

За задачата за класификация на типа събитие моделите, обучавани върху *EXTEND-CC* и *EXTEND-SW*, демонстрират близки най-високи *F1* оценки - съответно 0.87 и 0.89. Моделите, обучавани върху *EXTEND-RC*, постигат най-висока *F1* оценка от 0.77 за тази задача.

За задачата за класифициране на типа събитие и активиращата му дума моделите, обучавани върху *EXTEND-CC*, *EXTEND-SW* и *EXTEND-RC*, постигат най-високи *F1* оценки съответно от 0.70, 0.86 и 0.53. За задачата за класифициране на аргументите най-високите постигнати *F1* оценки от трите типа модели са съответно 0.74, 0.68 и 0.41.

Забелязва се, че моделите, обучавани върху *EXTEND-RC*, се представят най-зле за всички задачи в сравнение с моделите, обучавани върху *EXTEND-CC* или *EXTEND-SW*.

Резултати и анализ на експерименти с повече от един тип събития Таблици 10 и 11 показват получените осреднени макро-резултати (прецизност, връщане и *F1* оценка) за моделите, обучавани съответно върху целия корпус *EXTEND* и върху комбинация от *EXTEND-CC* и *EXTEND-SW*. Резултати са представени за класификация на типа събитие (колона *Събитие*), класификация на типа събитие и разпознаване на активиращата дума (колона *Събитие + Акт.*) и класификация на аргументите при правилно разпозната активираща дума (колона *Аргументи+*).

В случая се докладват макро-резултати, т.к. корпусите са балансирани откъм брой събития от всеки тип.

Стандартните отклонения на метриките за моделите, обучавани върху всички данни от *EXTEND*, са в интервала от 0.01 до 0.14. За моделите, обучавани само върху данни от *EXTEND-CC* и *EXTEND-SW*, те са в интервала от 0 до 0.10 с едно отклонение (*outlier*) от 0.20.

Както при експериментите, описани до момента, и тук наблюдаваме повишаване на резултатите с увеличение на обема на обучителното множество и броя епохи за

Таблица 10: Осреднени макро-стойности за прецизност (P), връщане (R) и *F1* оценка за събития от всички типове от *EXTEND*

Модел		**Събитие**			**Съб. + Акт.**			**Аргументи+**		
	X:Y	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
30 епохи	20:80	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	40:60	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	60:40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	80:20	0.72	0.61	0.63	0.31	0.51	0.38	0.41	0.67	0.49
100 епохи	20:80	0.75	0.58	0.64	0.40	0.52	0.44	0.39	0.53	0.45
	40:60	0.77	0.64	0.69	0.55	0.60	0.57	0.47	0.55	0.51
	60:40	0.74	0.81	0.77	0.52	**0.81**	**0.65**	0.46	0.57	0.51
	80:20	0.75	0.75	0.74	0.46	0.69	0.54	0.54	0.69	0.60
300 епохи	20:80	0.80	0.65	0.71	0.49	0.60	0.53	0.46	0.55	0.50
	40:60	0.79	0.74	0.76	0.58	0.71	0.63	0.54	0.62	0.58
	60:40	0.77	**0.84**	**0.80**	0.55	**0.81**	**0.65**	0.50	0.57	0.53
	80:20	0.77	0.78	0.76	0.47	0.71	0.55	0.58	0.73	0.64
500 епохи	20:80	0.78	0.74	0.76	0.54	0.70	0.60	0.45	0.56	0.50
	40:60	**0.81**	0.74	0.77	**0.60**	0.71	**0.65**	0.54	0.61	0.57
	60:40	0.76	0.83	0.79	0.53	0.79	0.63	0.54	0.61	0.57
	80:20	0.77	0.81	0.78	0.50	0.75	0.58	**0.59**	**0.74**	**0.65**

Таблица 11: Осреднени макро-стойности за прецизност (P), връщане (R) и *F1* оценка за събития от типове *Cure-Claim* и *Severe-Weather*

Модел		**Събитие**			**Съб. + Акт.**			**Аргументи+**		
	X:Y	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
30 епохи	20:80	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	40:60	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	60:40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	80:20	0.10	0.01	0.02	0.10	0.01	0.02	0.00	0.00	0.00
100 епохи	20:80	0.38	0.02	0.03	0.08	0.01	0.02	0.18	0.35	0.23
	40:60	0.85	0.73	0.78	0.59	0.70	0.63	0.51	0.66	0.57
	60:40	0.86	0.88	0.87	0.67	0.86	0.74	0.55	0.70	0.62
	80:20	0.86	0.91	0.88	0.70	0.90	0.78	0.57	0.74	0.64
300 епохи	20:80	0.82	0.68	0.74	0.53	0.63	0.56	0.48	0.61	0.54
	40:60	0.85	0.80	0.82	0.62	0.78	0.68	0.53	0.66	0.59
	60:40	0.85	0.93	0.89	0.70	0.92	0.78	0.59	0.72	0.65
	80:20	0.84	0.91	0.87	0.74	0.90	0.81	0.64	0.82	0.72
500 епохи	20:80	0.82	0.73	0.76	0.56	0.69	0.60	0.50	0.61	0.54
	40:60	0.84	0.90	0.87	0.68	0.88	0.76	0.58	0.70	0.63
	60:40	0.83	0.92	0.87	0.74	0.92	0.81	0.58	0.72	0.64
	80:20	**0.89**	**0.95**	**0.91**	**0.75**	**0.94**	**0.83**	**0.70**	**0.84**	**0.76**

обучение.

За разлика от експериментите, при които моделите се обучават само върху един тип събитие, тук забелязваме, че и при обучение за 30 епохи се постигат резултати, различни от 0, когато се използва най-голямото обучително множество. За моделите, обучавани върху документи от целия корпус *EXTEND*, за 30 епохи при разпределение на данните 80:20 се постигат макро-*F1* оценки от 0.63, 0.38 и 0.49 съответно за задачите за класификация на типа събитие, на типа събитие и активиращата дума и на аргументите. За моделите, обучавани само върху документите от *EXTEND-CC* и *EXTEND-SW* тези резултати са съответно 0.02, 0.02 и 0.00. Това най-вероятно се дължи на разликата в общия брой документи в използваните корпуси и подсказва, че при наличие на повече обучителни данни като цяло биха се постигнали по-добри резултати при обучение за по-малко на брой епохи.

От таблиците става ясно, че с изключение на моделите, обучавани за 30 епохи, моделите, обучавани само върху данните от *EXTEND-CC* и *EXTEND-SW* за всеки брой епохи и всички разпределения на данните постигат по-добри резултати от моделите, обучавани върху всички данни от *EXTEND*.

За задачите за класификация на типа събитие, на типа събитие и активиращата дума и на аргументите, моделите, обучавани върху всички данни от *EXTEND*, постигат най-високи макро-*F1* оценки съответно 0.80, 0.65 и 0.65. За сравнение моделите, обучавани само върху данни от *EXTEND-CC* и *EXTEND-SW*, постигат съответно 0.91, 0.83 и 0.76.

Трябва да се отбележи също, че най-високите макро-*F1* оценки, постигнати от моделите, обучавани върху данни от *EXTEND-CC* и *EXTEND-SW* едновременно, са по-високи от най-високите *F1* оценки, постигнати от моделите, обучавани само върху *EXTEND-CC* и само върху *EXTEND-SW*.

5.1.2.2 Експерименти върху типове събития от *ACE2005*

Дизайн на експериментите Обучени са модели върху *EXTEND-CC*, *EXTEND-SW* и *EXTEND-RC* по отделно за 100, 300 и 500 епохи, като са използвани 90% от данните за всеки тип като обучителни и 10% като валидационни. По същата

стратегия са обучени модели върху EXTEND и върху *EXTEND-CC* и *EXTEND-SW* - за 30, 100, 300 и 500 епохи.

Следва да се направи оценка на представянията на обучените модели за десетте най-често срещани типове събития от *ACE2005*.

За целта са използвани множества от документи от първоначално извлечените от *DBKF*, за които базовият модел и *NEXT* моделите, описани в секция 5.1.1.2, са предсказали събитие от даден тип. Разгледани са по седем множества от документи за всеки тип, за които наличие на събитие от дадения тип е предсказано от:

- само базовия модел, само *NEXT100* или само *NEXT500*;
- само базовия модел и *NEXT100*, само базовия модел и *NEXT500* или само *NEXT100* и *NEXT500*;
- всички модели.

От всяко множество за всеки от разглежданите типове събития са избрани на случаен принцип по до 20 документа. За всички документи е валидирано ръчно дали наистина съдържат събитие от предсказания според извадката тип. Също така за всеки документ допълнително са анотирани ръчно още срещания на събития от разглежданите 10 типа. Това е необходимо, т.к. даден документ може да съдържа събития от два типа, но да е попаднал в извадка само за единия от тези типове.

Полученото множество от анотирани документи се нарича *DBKF-ACE-Top10* и се състои от 989 документа, в които се срещат общо 1319 събития.

Метрики Върху *DBKF-ACE-Top10* се оценяват обучените модели за прецизност, връщане и *F1* оценка на ниво правилна класификация на типа събитие, споменато в дадения документ (без да се разглеждат активиращите думи и аргументите). Докладват се микро-оценки, т.к. множеството не е балансирано относно отделните типове.

За изчислението на описаните метрики е използвана *classification_report* функцията от *sklearn.metrics*[29].

[29] `https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics`

Резултати и анализ Таблица 12 представя микро-осреднените оценки за прецизност, връщане и *F1* на всички споменати модели върху *DBKF-ACE-Top10*.

Таблица 12: Микро-оценки на новите модели върху десетте най-често срещани събития от *ACE2005*

	30 епохи			**100 епохи**			**300 епохи**			**500 епохи**		
Данни за обучение	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
EXTEND-CC	-	-	-	0.76	0.79	0.77	0.78	0.73	0.75	0.79	0.68	0.73
EXTEND-SW	-	-	-	0.80	0.64	0.71	0.80	0.65	0.72	0.80	0.44	0.57
EXTEND-RC	-	-	-	0.82	0.23	0.35	0.86	0.07	0.13	0.86	0.09	0.16
EXTEND	0.84	0.25	0.38	0.87	0.17	0.29	0.89	0.13	0.22	0.88	0.12	0.21
EXTEND-CC+EXTEND-SW	0.78	0.75	0.77	0.80	0.61	0.69	0.83	0.45	0.58	0.82	0.41	0.55

За всички изследвани корпуси, използвани за обучение, се забелязва, че с нарастване на броя епохи за обучение прецизността остава същата или дори се повишава, но връщането спада, което се отразява и в спад на *F1* оценката. При моделите, за чието обучение са използвани документи от *EXTEND-RC*, спадът в стойностите на връщането и *F1* оценката е особено осезаем. Това наблюдение може да се дължи на факта, че голямо количество от типовете събития в *ACE2005* са от правната област, от която е и *Rule-Change* типа.

Също така може да се отбележи, че като цяло най-добре върху тези типове събития се справят моделите, обучавани само върху данни от *EXTEND-CC*.

От моделите, обучавани за повече от един тип събития, значително по-добре се представят моделите, обучавани само върху *EXTEND-CC* и *EXTEND-SW*.

5.2 Експерименти с *LLaMA-2*

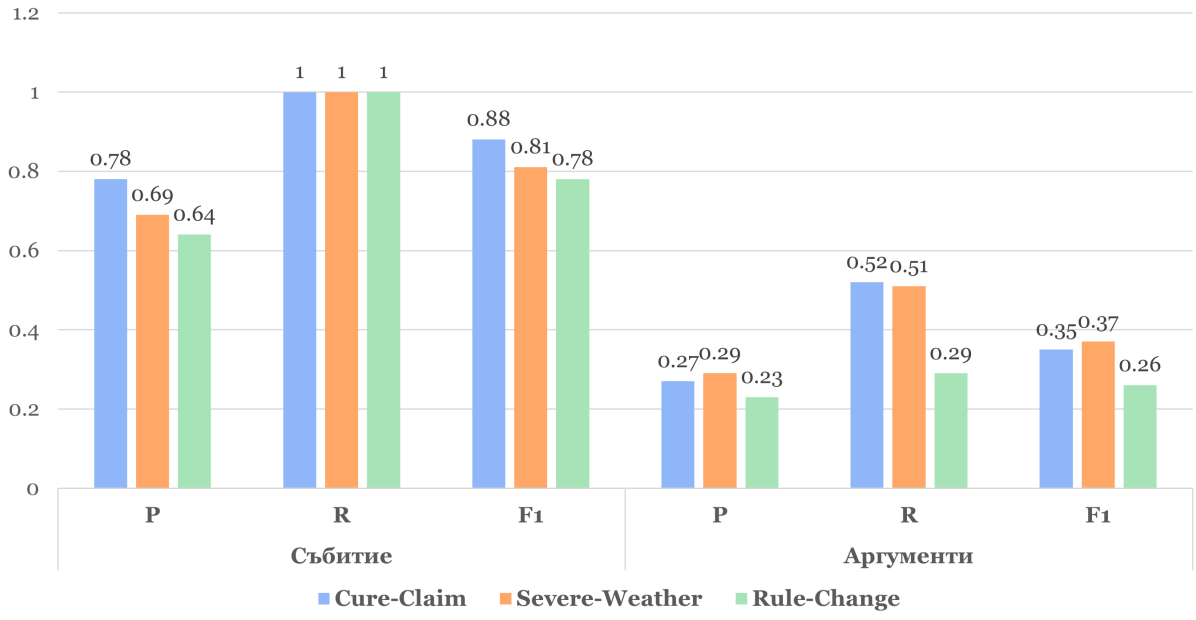
Дизайн на експериментите Направени са експерименти с *LLaMA-2* модела със 70 милиарда параметъра (*LLaMA-2-70b-chat*[30]) без предварително обучение. Използвани са инструкциите, описани в секция 4.2.2. За всеки един от типовете събития в *EXTEND* е проведен отделен експеримент върху документите от *EXTEND-CC*, *EXTEND-SW* и *EXTEND-RC*.

Метрики Докладват се прецизност, връщане и *F1* оценка на ниво класификация на типа събитие и класификация на аргументите за всеки тип събитие по отделно.

[30] `https://huggingface.co/meta-llama/Llama-2-70b-chat`

За изчислението на описаните метрики са написани функции на програмния език *Python*.

Резултати и анализ Фигура 22 показва получените резултати от експериментите с *LLaMA-2-70b-chat* за отделните типове събития и нива на класификация.



Фигура 22: Резултати от експериментите с *LLaMA-2-70b-chat*

Веднага се забелязва, че на ниво класификация на типа събитие е постигнато максимално връщане със стойност 1 за всички разглеждани типове. Също така на това ниво моделът се представя най-добре за събития от тип *Cure-Claim*, а най-зле - за събития от тип *Rule-Change*.

Наблюдава се драстична разлика между постигнатите резултати на ниво класификация на типа събитие и на ниво класификация на аргументите за всички изследвани типове.

От резултатите може да се направи извод, че с така зададените инструкции *LLaMA-2-70b-chat* се справя добре със задачата за класификация на типа събитие без предварително обучение, което се споменава в текста, но не чак толкова добре със задачата за класификация на аргументите.

5.3 Обобщение на проведените експерименти

Описаните експерименти с допълнително обучените *Text2Event* модели са проведени с цел да се установи успеваемостта на избрания подход и да се открие моделът, който едновременно разпознава добре новите типове събития от *EXTEND* и запазва възможностите си да разпознава типовете събития от *ACE2005*, върху които е бил предварително обучаван.

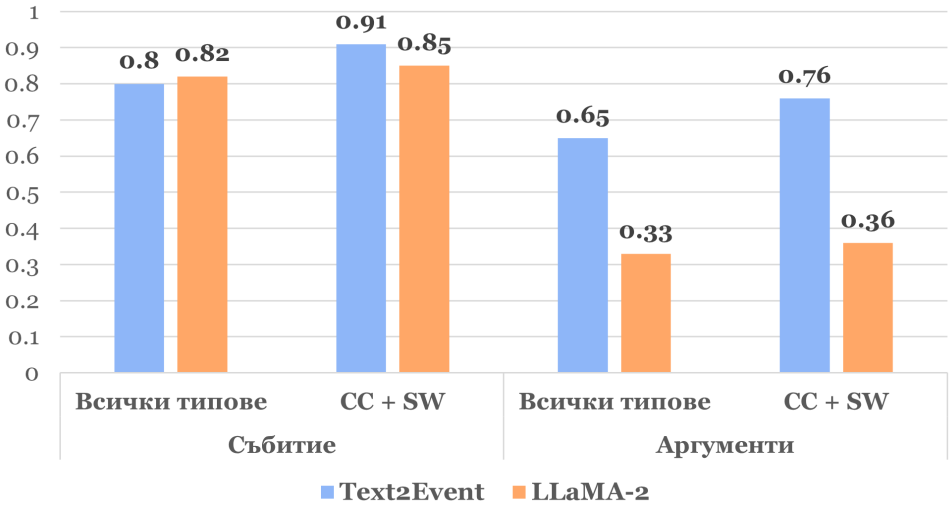
На база представените в секция 5.1 резултати може да се направи извод, че методологията за допълнително обучение на *Text2Event* модел е успешна за целите на изследването. Като модел, който постига най-добър баланс между разпознаването на новите и старите събития, може да се посочи модела, който е обучаван за 100 епохи върху данните от *EXTEND-CC* и *EXTEND-SW*. От всички направени експерименти става ясно, че моделите, за чието обучение са използвани данни от *EXTEND-RC*, показват значително по-ниски резултати от останалите, както за новите, така и за старите типове събития.

Описаните в секция 5.2 експерименти без предварително обучение с големия езиков модел *LLaMA-2* са проведени с цел да се установи доколко подобен модел се справя с поставените задачи без предварително обучение и дали би могъл да се използва като инструмент за автоматично анотиране на данни.

На база представените резултати може да се твърди, че моделът би могъл да се използва без предварително обучение за автоматично анотиране само на типа на събитията, които се срещат в даден текст. В допълнение биха могли да се направят експерименти с малко предварително обучение, за да се установи дали това ще подобри представянето на модела за задачата за класификация на аргументите.

Фигура 23 представя сравнение между най-добрите постигнати макро-*F1* оценки от допълнително обучените *Text2Event* модели и *LLaMA-2* без предварително обучение с представените в 4.2.2 инструкции. Показани са макро-оценките за всички типове събития, както и за комбинацията от типовете *Cure-Claim* и *Severe-Weather*.

От представените данни отново се вижда, че резултатите само върху *Cure-Claim* и *Severe-Weather* типовете са по-добри от тези върху всички типове нови събития. Също така може да се отбележи, че резултатите на ниво класификация на типа



Фигура 23: Сравнение на най-добрите постигнати макро-*F1* оценки от допълно обучените *Text2Event* модели и *LLaMA-2* без предварително обучение

събитие са съизмерими и за двата приложени подхода, но се забелязва, че допълнително обучените *Text2Event* модели постигат значително по-добри макро-*F1* оценки на ниво класификация на аргументите.

6 Прототип на система за откриване на събития в текст

Като част от дипломната работа е проектиран и реализиран софтуерен прототип на система за автоматично откриване на събития в текст. Целта на прототипа е да демонстрира функционалността, която може да се постигне с помощта на обучените модели за откриване на събития. На по-късен етап прототипът може да бъде интегриран в по-сложни системи, които да се използват от крайни потребители журналисти и проверители на факти.

6.1 Проектиране на софтуерен прототип на система за автоматично откриване на събития в текст

6.1.1 Функционални изисквания

Изискване 1. Системата трябва да предоставя функционалност за анотация на текст спрямо типовете събития, които се срещат в него, и съответните им аргументи. (Висок приоритет)

6.1.2 Нефункционални изисквания

Изискване 2. Системата трябва да предоставя възможности за интеграция с други системи и услуги. (Висок приоритет)

Изискване 3. Системата трябва да може да бъде инсталирана на различни платформи и инфраструктури. (Висок приоритет)

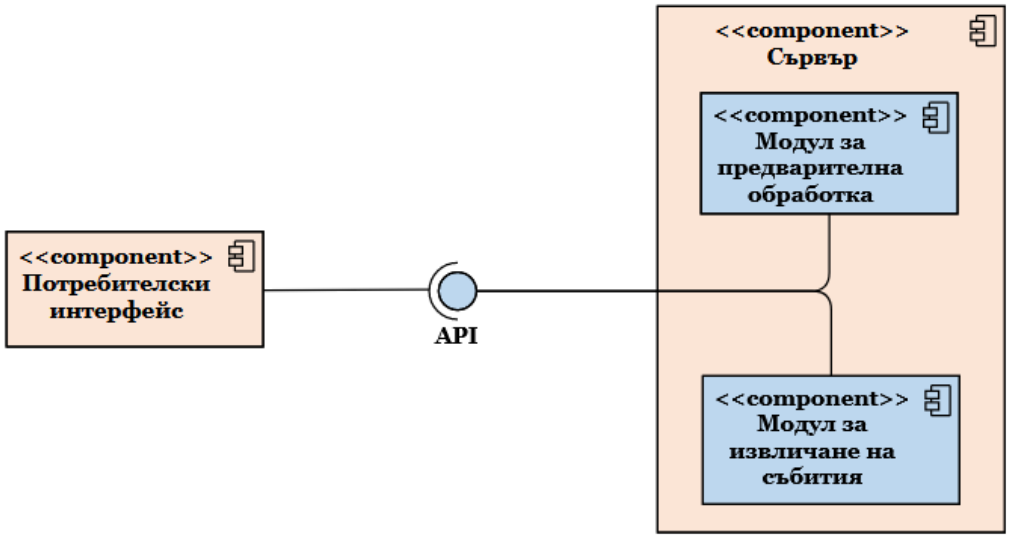
6.1.3 Обща архитектура на системата

Проектираната система има два основни компонента - сървър и потребителски интерфейс, които са показани на фигура 24.

Бизнес логиката на системата се реализира в сървърния компонент. В него се разграничават модул за предварителна обработка и модул за извличане на събития.

Модулът за предварителна обработка се грижи подаденият текст да бъде разделен на отделни изречения, преди да бъде изпратен към модула за извличане на събития.

Потребителският интерфейс комуникира със сървъра чрез програмен интерфейс (*API*). Той позволява на потребителя да изпраща заявки за откриване на събития в текст и да получава съответните резултати.



Фигура 24: Компонентна диаграма на софтуерен прототип на система за автоматично откриване на събития в текст

6.2 Реализация на софтуерен прототип на система за автоматично откриване на събития в текст

6.2.1 Използвани технологии

Език за разработка и библиотеки Системата за автоматично откриване на събития в текст трябва да съдържа в себе си модул за извличане на събития. Тъй като моделът, който е обучен за тези цели, е реализиран на езика *Python*, той ще бъде най-удобно интегриран в система, която също е реализирана на този език.

Системата е реализирана под формата на уеб приложение, като е използвана библиотеката *Flask*[31]. *Flask* е лека библиотека за разработка на уеб приложения,

[31] `https://flask.palletsprojects.com/en/3.0.x/`

която е особено подходяща за малки проекти. Това е основната причина да бъде предпочетена пред други библиотеки за разработка на уеб приложения на езика *Python* като *Django*[32], която е по-подходяща за по-големи и сложни проекти.

Използвана е също библиотеката *flask-swagger-ui*[33] за интегриране на сървърната част на приложението с прост потребителски интерфейс - *SwaggerUI*[34]. *SwaggerUI* е инструмент, който генерира интерактивна документация на програмния интерфейс. По този начин се предоставя удобен и интуитивен потребителски интерфейс, който демонстрира функционалностите на системата.

Програмни интерфейси Комуникацията със системата се извършва посредством програмни интерфейси. Използва се *HTTP*[35] като протокол за комуникация. Данните се изпращат в заявка, представени като прост текст, и се връщат обработени в *JSON*[36] формат. По този начин се удовлетворява **Изискване 2** към системата.

Дистрибуция Системата е пакетирана в *Docker*[37] изображение и се дистрибутира чрез него. Това улеснява инсталацията и стартирането ѝ на различни платформи и инфраструктури. По този начин се удовлетворява **Изискване 3** към системата.

Указания за стартиране на системата и работа с нея са предоставени в Приложение 4.

6.2.2 Детайли от имплементацията на компонентите на системата

6.2.2.1 Сървър Сървърната част на системата енкапсулира бизнес логиката ѝ. За комуникация с нея се използва *API*, чийто основен *endpoint* е */extract-events*. Той използва *HTTP* метод *POST* за обработка на заявките и има следната спецификация, илюстрирана чрез конкретен пример, представен от фигури 25 и 26:

HTTP POST */extract-events*

Content-Type: *text/plain; charset=utf-8*

[32] `https://www.djangoproject.com/`
[33] `https://pypi.org/project/flask-swagger-ui/`
[34] `https://swagger.io/tools/swagger-ui/`
[35] `https://developer.mozilla.org/en-US/docs/Web/HTTP`
[36] `https://www.json.org/json-en.html`
[37] `https://www.docker.com/`

Request Body:

```
Crocodile blood cures Covid-19
```

Фигура 25: Пример за тяло на *POST* заявка към */extract-events*

Accept: *application/json*

Response Body:

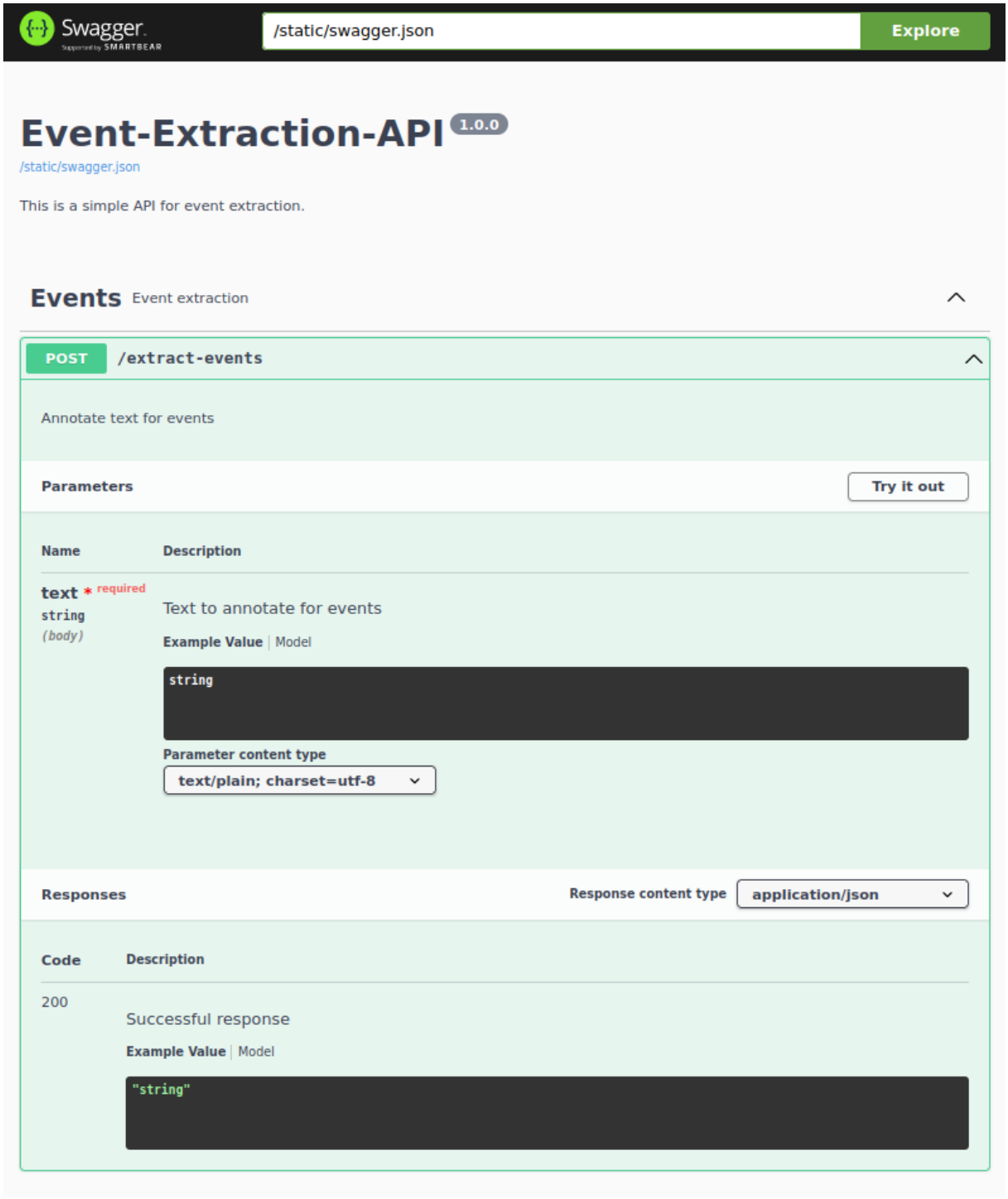

```
[
   {
      "event":  [
         {
            "arguments":  {
               "Condition": [ "Covid-19" ],
               "Cure": [ "Crocodile blood" ]
             },
            "trigger": "cures",
            "type": "Cure-Claim"
          }
       ],
       "text": "Crocodile blood cures Covid-19"
   }
]
```


Фигура 26: Пример за тяло на отговор на *POST* заявка към */extract-events*

При изпращане на текст като тяло на заявка към системата, най-напред той се разделя на изречения от модула за предварителна обработка. За целта е използван токенизатор от библиотеката *nltk*[38]. След това отделните изречения се подават за анотиране на модула за откриване на събития. Той използва *Text2Event* модела, допълнително обучен върху *EXTEND-CC* и *EXTEND-SW* за 100 епохи. За зареждане на модела е адаптиран код от хранилище в *GitHub*[39].

6.2.2.2 Потребителски интерфейс Системата разполага с прост потребителски интерфейс, реализиран като *SwaggerUI*. Фигура 27 показва изглед от него.

[38] `https://www.nltk.org/api/nltk.tokenize.html`
[39] `https://github.com/anonymoussubmissions94/CIKM2022_submission`



Фигура 27: Изглед от потребителския интерфейс *SwaggerUI* на системата за автоматично анотиране на събития

7 Заключение

7.1 Обобщение на изпълнението на началните цели

Предметът на изследване на дипломната работа са задачите за откриване и извличане на събития - предизвикателни задачи от областта на обработката на естествен език, в случая разгледани в контекста на борбата срещу дезинформацията.

В рамките на дипломната работа са изпълнени целта и задачите, произтичащи от нея, както са дефинирани в секция 1.2.

Направен е подробен обзор на съществуващите подходи, които се прилагат в областта на откриването и извличането на събития. Въз основа на направените от него изводи е избран подходът *Text2Event* за по-нататъшно изследване. Предварително обученият модел е приложен върху данни, извлечени от нерелационна база от данни с опровержения на фалшиви новини - *DBKF*, с цел да се установи какви събития от *ACE2005* се срещат в тях.

Разработена е методология за разширяване на множеството от разпознавани събития, като се дефинирани три нови типа събития - *Cure-Claim*, *Severe-Weather* и *Rule-Change*. Създадени са указания за анотиране на данни за новите типове събития, на базата на които са анотирани данните от новия корпус *EXTEND*.

Предварително обученият върху данните от *ACE2005* модел *Text2Event* е обучен допълнително за данните от новия корпус *EXTEND*. Приложено е машинно самообучение с учител. Проведени са експерименти с цел да се изследва значението на количеството използвани данни за обучение и броя епохи за обучение. Резултатите показват ефективността на подхода относно представянето на допълнително обучените модели както върху новите типове събития от *EXTEND*, така и върху типовете събития от *ACE2005*. Също така са проведени и експерименти с големия езиков модел *LLaMA-2* (конкретно *LLaMA-2-70b-chat*, който има 70 милиарда параметъра). Приложен е подход без предварително обучение. Резултатите показват, че че с приложените инструкции към *LLaMA-2-70b-chat* за класифициране на типа събитие, което се среща в даден текст, двата изследвани подхода се справят съизмеримо добре. За класифицирането на аргументите на събитието обаче допълнително обучените

Text2Event модели демонстрират по-добро представяне.

Въз основа на резултатите от експериментите и направения анализ е избран този допълнително обучен *Text2Event* модел, който се справя едновременно най-добре с разпознаването на събития от новите типове от *EXTEND* и от старите типове от *ACE2005*. Моделът е използван като основна част на софтуерен прототип на система за автоматично откриване на събития в текст.

7.2 Насоки за бъдещо развитие

Една от възможностите за бъдещо развитие на разработките от настоящата дипломна работа е да бъде разширено множеството от типове събития в *EXTEND* с нови типове, които са от интерес за заинтересованите лица. Също така може да бъде създаден по-голям корпус с анотирани данни за новите типове събития от *EXTEND*. Тъй като ръчната анотация на данни е бавен и трудоемък процес, могат да се изследват възможности за отдалечено контролирано машинно самообучение, т.е. да бъдат автоматично анотирани данни, които след това да бъдат само валидирани ръчно. Една възможност за автоматично анотиране на данни е да се използват допълнително обучените в рамките на дипломната работа *Text2Event* модели. Като друга възможност може да се разглеждат големите езикови модели като *LLaMA-2*. От направените експерименти става ясно, че със зададените инструкции *LLaMA-2* се справя добре със задачата за класификация на типа събитие в текст. За задачата за класификация на аргументите може да се търси подобрение на резултатите, като се експериментира с различни инструкции към модела.

Друга насока за бъдещо развитие засяга създадения прототип на софтуерна система за автоматично откриване на събития в текст. Той може да бъде интегриран в съществуваща система или да бъде използван като основа за създаване на нова по-комплексна система. Тази система ще позволява на журналисти и проверители на факти освен да изпращат заявки за откриване на събития в текст, да запазват резултатите в база от данни и да търсят в нея по критетии като тип на събитие или аргументи.

Благодарности

Изказвам благодарности на Онтотекст АД за предоставените данни и ресурси като част от работа по проекти, частично финансирани от Европейската комисия - *vera.ai*[40] (договор No:101070093) и *VIGILANT*[41] (договор No:101073921).

Също така благодаря на колегите от Онтотекст АД, които взеха участие в процесите по анотиране на данните от корпуса *EXTEND*.

[40] `https://www.veraai.eu/home`
[41] `https://www.vigilantproject.eu/`

Публикации по темата на дипломната работа

1. E. Tuparova, P. Ivanov, A. Tagarev, S. Boytcheva, and I. Koychev, “Next: An event schema extension approach for closed-domain event extraction models”, The 6th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text, RANLP, 2023 - под печат

Приносът на дипломанта е свързан с началния анализ на събраните данни, процесите по дефиниране на новия тип събитие, създаване на анотиран корпус от данни (подбор на данни за анотация, създаване на указания за анотиране и анотиране на данните), допълнително обучение на *Text2Event* модела и описаните в секция 5.1.1 експерименти.

Литература

[1] Q. Li, J. Li, J. Sheng, S. Cui, J. Wu, Y. Hei, H. Peng, S. Guo, L. Wang, A. Beheshti *et al.*, "A survey on deep learning event extraction: Approaches and applications," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[2] J. Liu, Y. Chen, K. Liu, and J. Zhao, "Event detection via gated multilingual attention mechanism," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Apr. 2018. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/11919

[3] E. Tuparova, P. Ivanov, A. Tagarev, S. Boytcheva, and I. Koychev, "Next: An event schema extension approach for closed-domain event extraction models," 2023.

[4] F. Hogenboom, F. Frasincar, U. Kaymak, F. de Jong, and E. Caron, "A survey of event extraction methods from text for decision support systems," *Decision Support Systems*, vol. 85, pp. 12–22, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167923616300173

[5] H. Tanev, J. Piskorski, and M. Atkinson, "Real-time news event extraction for global crisis monitoring," in *Natural Language and Information Systems: 13th International Conference on Applications of Natural Language to Information Systems, NLDB 2008 London, UK, June 24-27, 2008 Proceedings 13*. Springer, 2008, pp. 207–218.

[6] J. Sheng, S. Guo, B. Yu, Q. Li, Y. Hei, L. Wang, T. Liu, and H. Xu, "Casee: A joint learning framework with cascade decoding for overlapping event extraction," *arXiv preprint arXiv:2107.01583*, 2021.

[7] S. Zheng, W. Cao, W. Xu, and J. Bian, "Doc2EDAG: An end-to-end document-level framework for Chinese financial event extraction," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 337–346. [Online]. Available: https://aclanthology.org/D19-1032

[8] Q. Wei, Z. Ji, Z. Li, J. Du, J. Wang, J. Xu, Y. Xiang, F. Tiryaki, S. Wu, Y. Zhang, C. Tao, and W. Qi, "A study of deep learning approaches for medication and adverse drug event extraction from clinical text," *Journal of the American Medical Informatics Association : JAMIA*, vol. 27, 05 2019.

[9] Y. Peng, M. Moh, and T.-S. Moh, "Efficient adverse drug event extraction using twitter sentiment analysis," in *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2016, pp. 1011–1018.

[10] A. Ritter, Mausam, O. Etzioni, and S. Clark, "Open domain event extraction from twitter," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 1104–1112.

[11] D. Zhou, L. Chen, and Y. He, "A simple Bayesian modelling approach to event extraction from Twitter," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Baltimore, Maryland: Association for Computational Linguistics, Jun. 2014, pp. 700–705. [Online]. Available: https://aclanthology.org/P14-2114

[12] F. Kunneman and A. Van Den Bosch, "Open-domain extraction of future events from twitter," *Natural Language Engineering*, vol. 22, no. 5, pp. 655–686, 2016.

[13] E. Filtz, M. Navas-Loro, C. Santos, A. Polleres, and S. Kirrane, *Events Matter: Extraction of Events from Court Decisions*, 12 2020.

[14] G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel, "The automatic content extraction (ACE) program – tasks, data, and evaluation," in *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*. Lisbon, Portugal: European Language Resources Association (ELRA), May 2004. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2004/pdf/5.pdf

[15] B. Yang and T. M. Mitchell, "Joint extraction of events and entities within a document context," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics:*

Human Language Technologies. San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 289–299. [Online]. Available: https://aclanthology.org/N16-1033

[16] X. Wang, Z. Wang, X. Han, W. Jiang, R. Han, Z. Liu, J. Li, P. Li, Y. Lin, and J. Zhou, "MAVEN: A Massive General Domain Event Detection Dataset," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 1652–1671. [Online]. Available: https://aclanthology.org/2020.emnlp-main.129

[17] D. Le and T. H. Nguyen, "Fine-grained event trigger detection," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, Apr. 2021, pp. 2745–2752. [Online]. Available: https://aclanthology.org/2021.eacl-main.237

[18] G. A. Miller, M. Chodorow, S. Landes, C. Leacock, and R. G. Thomas, "Using a semantic concordance for sense identification," in *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, 1994. [Online]. Available: https://aclanthology.org/H94-1046

[19] S. Li, H. Ji, and J. Han, "Document-level event argument extraction by conditional generation," 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, Jun. 2021, pp. 894–908. [Online]. Available: https://aclanthology.org/2021.naacl-main.69

[20] J. Liu, L. Min, and X. Huang, "An overview of event extraction and its applications," *arXiv preprint arXiv:2111.03212*, 2021.

[21] Y. Zhao, X. Jin, Y. Wang, and X. Cheng, "Document embedding enhanced event detection with hierarchical and supervised attention," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short*

Papers). Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 414–419. [Online]. Available: https://aclanthology.org/P18-2066

[22] Y. Chen, L. Xu, K. Liu, D. Zeng, and J. Zhao, "Event extraction via dynamic multi-pooling convolutional neural networks," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, Jul. 2015, pp. 167–176. [Online]. Available: https://aclanthology.org/P15-1017

[23] F. Li, W. Peng, Y. Chen, Q. Wang, L. Pan, Y. Lyu, and Y. Zhu, "Event extraction as multi-turn question answering," in *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 829–838. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.73

[24] J. Sheng, S. Guo, B. Yu, Q. Li, Y. Hei, L. Wang, T. Liu, and H. Xu, "CasEE: A joint learning framework with cascade decoding for overlapping event extraction," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, Aug. 2021, pp. 164–174. [Online]. Available: https://aclanthology.org/2021.findings-acl.14

[25] D. Wadden, U. Wennberg, Y. Luan, and H. Hajishirzi, "Entity, relation, and event extraction with contextualized span representations," *arXiv preprint arXiv:1909.03546*, 2019.

[26] S. Yang, D. Feng, L. Qiao, Z. Kan, and D. Li, "Exploring pre-trained language models for event extraction and generation," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 5284–5294. [Online]. Available: https://aclanthology.org/P19-1522

[27] Y. Zhou, Y. Chen, J. Zhao, Y. Wu, J. Xu, and J. Li, "What the role is vs. what plays the role: Semi-supervised event argument extraction

via dual question answering," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 16, pp. 14 638–14 646, May 2021. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/17720

[28] D. Lu, S. Ran, J. Tetreault, and A. Jaimes, "Event extraction as question generation and answering," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 1666–1688. [Online]. Available: https://aclanthology.org/2023.acl-short.143

[29] Y. Lu, H. Lin, J. Xu, X. Han, J. Tang, A. Li, L. Sun, M. Liao, and S. Chen, "Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 2795–2806. [Online]. Available: https://aclanthology.org/2021.acl-long.217

[30] E. Riloff, "Automatically constructing a dictionary for information extraction tasks," 11 2000.

[31] V. D. Lai, "Event extraction: A survey," 2022.

[32] S. Liao and R. Grishman, "Using document level cross-event inference to improve event extraction," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden: Association for Computational Linguistics, Jul. 2010, pp. 789–797. [Online]. Available: https://aclanthology.org/P10-1081

[33] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," 2013.

[34] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association

for Computational Linguistics, Oct. 2014, pp. 1532–1543. [Online]. Available: https://aclanthology.org/D14-1162

[35] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," 2017.

[36] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[37] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019.

[38] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," 2018.

[39] T. H. Nguyen, K. Cho, and R. Grishman, "Joint event extraction via recurrent neural networks," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 300–309. [Online]. Available: https://aclanthology.org/N16-1034

[40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[41] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text

transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-074.html

[42] H. Yang, Y. Chen, K. Liu, Y. Xiao, and J. Zhao, "DCFEE: A document-level Chinese financial event extraction system based on automatically labeled training data," in *Proceedings of ACL 2018, System Demonstrations*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 50–55. [Online]. Available: https://aclanthology.org/P18-4009

[43] V. D. Lai, M. V. Nguyen, T. H. Nguyen, and F. Dernoncourt, "Graph learning regularization and transfer learning for few-shot event detection," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 2172–2176. [Online]. Available: https://doi.org/10.1145/3404835.3463054

[44] Q. Lyu, H. Zhang, E. Sulem, and D. Roth, "Zero-shot event extraction via transfer learning: Challenges and insights," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 322–332. [Online]. Available: https://aclanthology.org/2021.acl-short.42

[45] A. Tagarev, K. Bozhanova, I. Nikolova-Koleva, and I. Ivanov, "Tackling multilinguality and internationality in fake news," in *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*. Held Online: INCOMA Ltd., Sep. 2021, pp. 1380–1386. [Online]. Available: https://aclanthology.org/2021.ranlp-1.154

[46] M. U. Hadi, Q. Al-Tashi, R. Qureshi, A. Shah, A. Muneer, M. Irfan, A. Zafar, M. Shaikh, N. Akhtar, J. Wu, and S. Mirjalili, "Large language models: A comprehensive survey of its applications, challenges, limitations, and future prospects," 07 2023.

[47] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," *CoRR*, vol. abs/2005.14165, 2020. [Online]. Available: https://arxiv.org/abs/2005.14165

[48] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom, "Llama 2: Open foundation and fine-tuned chat models," 2023.

Приложения

Приложение 1. Речник на чуждите термини

- *Annotated (labeled) data* – анотирани данни
- *Claim* – твърдение
- *Closed-domain* – затворена област
- *Conditional random fields* – случайни условни полета
- *Constituent parser* – съставен анализатор
- *Convolution neural network* – конволюционна невронна мрежа
- *Curriculum learning* – смесено обучение
- *Dependency parser* – анализатор на зависимостите
- *Distant supervised machine learning* – отдалечено контролирано машинно самообучение
- *Document annotation* – анотация на ниво документ
- *Embedding* – векторно представяне
- *Event* – събитие
- *Event arguments* – аргументи на събитие
- *Event detection* – откриване на събития
- *Event extent* – обхват на събитие
- *Event extraction* – извличане на събития
- *Event schema* – схема на събитията
- *F1 score* – *F1* оценка
- *False negative* – грешно отрицателен

- *False positive* – грешно положителен
- *Feature engineering* – подбор на свойствата
- *Few-shot setting* – подходи с малко предварително обучение
- *Fine-tuning* – допълнително обучение
- *Full structure learning* – обучение за пълни структури
- *Hierarchical agglomerative clustering* – йерархично агломеративно клъстеризиране
- *Inline annotation* – анотация на ниво подниз от документ
- *Joint paradigm* – паралелна парадигма
- *Large language model* – голям езиков модел
- *Open-domain* – отворена област
- *Outlier* – отклонение
- *Pattern matching* – сравнение на шаблони
- *Pipeline paradigm* – последователна парадигма
- *POS tagger* – механизъм за разпознаване на частите на речта
- *Precision* – прецизност
- *Prompt* – инструкция
- *Question answering task* – задача за отговаряне на въпроси
- *Recall* – връщане
- *Recurrent neural network* – рекурентна невронна мрежа
- *Semi-supervised machine learning* – частично контролирано машинно самообучение
- *Sequence labeling task* – задача за поставяне на етикети на последователност
- *Sequence-to-structure task* – генеративна задача

- *Substructure learning* – обучение за подструктури
- *Supervised Machine Learning* – машинно самообучение с учител
- *Support Vector Machines* – машини с поддържащи вектори
- *System prompt* – системна инструкция
- *Textual entailment* – правене на извод от текст
- *Transfer learning* – трансферно обучение
- *Transformer* – трансформатор
- *Trigger word* – активираща дума; активатор
- *True negative* – вярно отрицателен
- *True positive* – вярно положителен
- *Word sense disambiguation* – разграничаване на смисъла на думите
- *Zero-shot setting* – подходи без предварително обучение

Приложение 2. Списък със съкращенията

- *ACE – Automatic Content Extraction*
- *API – Application Programming Interface*
- *BERT – Bidirectional Encoder Representations from Transformers*
- *CNN – Convolutional Neural Network*
- *CRF – Conditional Random Fields*
- *DARPA – Defense Advanced Research Projects Agency*
- *DBKF – Database of Known Fakes*
- *EXTEND – EXTended EveNts Dataset*
- *EXTEND-CC – EXTEND Cure-Claim*
- *EXTEND-RC – EXTEND Rule-Change*
- *EXTEND-SW – EXTEND Severe-Weather*
- *FN – False Negative*
- *FP – False Positive*
- *GPU – Graphical Processing Unit*
- *GRU – Gated Recurrent Unit*
- *HAC – Hierarchical Agglomerative Clustering*
- *HTTP – Hypertext Transfer Protocol*
- *JSON – JavaScript Object Notation*
- *LLaMA – Large Language Model Meta AI*
- *LDC – Linguistic Data Consortium*
- *LSTM – Long Short-Term Memory*
- *MAVEN – MAssive eVENt detection dataset*

- *ME – Maximum Entropy*
- *NEXT – New Event eXTraction*
- *NB – Naive Bayes*
- *OMDS – Ontotext Metadata Studio*
- *POS – Part of Speech*
- *RDF – Resource Description Framework*
- *RNN – Recurrent Neural Network*
- *SPARQL – SPARQL Protocol and RDF Query Language*
- *SVM – Support Vector Machine*
- *TN – True Negative*
- *TP – True Positive*
- *XML – eXtensible Markup Language*

Приложение 3. Указания за анотиране на събития

Общи указания за анотиране на събития

Следните общи указания за базирани на официалните указания за анотиране на събития на *ACE*[42] (за повече подробности може да се разгледат страници 5-17 от тях, които описват подробно правилата за анотиране на събития, техните активиращи думи и аргументи).

Частите **Основни понятия, свързани със събития**, **Анотиране на активиращи думи** и **Анотиране на активиращи думи при повече от един кандидати** са синтезирана версия на официалните *ACE* указания за анотиране. Секциите след това са авторски, създадени за целите на текущото изследване.

Основни понятия, свързани със събития

Съгласно указанията на *ACE* "Събитието е нещо, което се случва и включва участници. Едно събитие често може да бъде описано като промяна на състоянието."

Двете най-важни части на едно събитие са обхватът на събитието и активиращата дума на събитието. Обхватът на събитието е изречението, в което то се среща, докато активиращата дума на събитието е най-показателната дума за това събитие.

Указанията на *ACE* също дефинират участниците в събитието и атрибутите на събитието, наричани като цяло аргументи на събитието. Аргументите на събитието се намират винаги в рамките на обхвата му и могат да варират в зависимост от конкретния тип събитие. Обхватът на събитието и активиращата дума са задължителни компоненти за анотиране на събитие, докато аргументите на събитието са незадължителни.

Анотиране на събития

Анотиране на активиращи думи За да се анотира срещане на дадено събитие, трябва да се анотира неговата активираща дума. Активиращите думи могат да бъдат:

[42] `https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-events-guidelines-v5.4.3.pdf`

- **Основен глагол.** Често активиращата дума е основният глагол, който директно описва събитието, например:

  *He **died** yesterday of renal failure.*

  В някои случаи основният глагол може да е представен под формата на прилагателно име или причастие, например:

  *17 sailors were **killed**.*

- **Определение (прилагателно име или причастие).** Активиращата дума може да се срещне под формата на определение, например:

  *A **retired** congressman Gibbons gave a civics lesson in a portable classroom – another sign of growth too fast.*

- **Съществително име или местоимение**

  *The **attack** killed 7 and injured 20.*

  Забележка: В някои случаи в изречението има съществително име, което се отнася до участник в събитие и намеква за случането му, например:

  The attacker killed 7 and injured 20.

  В тези случаи съществителното име (*attacker*) не се анотира като активираща дума, т.к. 1) не изразява събитието по същия начин като *attack*; 2) думата би била анотирана и като аргумент на събитието, а не е допустимо дадена дума да се анотира едновременно като активираща дума и като аргумент на събитието,

Анотиране на активиращи думи при повече от един кандидати В някои случаи само гореописаните правила не са достатъчни за анотиране на активиращата дума, т.к. са налице повече кандидати. Това могат да са следните случаи:

- **Съществително име + глагол.** В някои случаи като потенциални активиращи думи могат да бъдат разглеждани едновременно съществително име и глагол, например:

  The leaders held a meeting in Beijing.

Hamas launched an attack.

В тези случаи се прилага *правилото за самостоятелното съществително име*, което гласи:

"В случаи, в които има повече от една възможна активираща дума, избираме съществителното име винаги когато то може да се използва самостоятелно, за да изрази събитието."

Следователно горните примери ще бъдат анотирани, както следва:

*The leaders held a **meeting** in Beijing.*

*Hamas launched an **attack**.*

- **Глагол + прилагателно име.** В много случаи, в които събитието се описва като резултат от нещо, то се изразява едновременно с глагол и съществително име, например:

  The explosion left at least 30 dead.

  В тези случаи се прилага *правилото за самостоятелното прилагателно име*, което гласи:

  "В случаи, в които глагол и прилагателно име са използвани заедно, за да се изрази дадено събитие, избираме прилагателното име като активираща дума винаги когато то може да се използва самостоятелно, за да изрази резултантното състояние на събитието."

  Следователно горният пример ще бъдат анотиран, както следва:

  *The explosion left at least 30 **dead**.*

- **Няколко глагола.** В някои случаи няколко глагола се използват едновременно, за да се опише събитието, например:

  79 million people have been born since the war ended.

  В тези случаи основният глагол се избира като активираща дума на събитието, т.к. останалите изпълняват поддържаща функция, т.е. примерът трябва да бъде анотиран като:

*79 million people have been **born** since the war ended.*

В тази категория попадат и примери от вида:

*John tried to **kill** Mary.*

В случая *tried* е второстепенен глагол и активиращата дума е *kill*.

- **Глагол + частица** Когато глаголът се използва във фраза с частица, и двете думи се анотират като активиращи само тогава когато са последователни в изречението. В противен случай се анотира само основният глагол, например:

  *Jane was **laid off** by XYZ Corp.*

  *XYZ Corp **laid** Jane off.*

По-специфични правила за анотиране

Правило на най-малката единица Винаги анотираме най-малката единица, която изразява аргумента напълно, например:

A condition known as human parainfluenza can be treated with antiviral medication.

В случая *"a condition known as human parainfluenza"* не носи допълнителна информация в сравнение с *"human parainfluenza"*, но *"parainfluenza"* не носи достатъчно информация от своя страна за еднозначно определяне на аргумента.

Фрази, съдържащи определителни и неопределителни членове В случай, че фраза, подлежаща на анотиране, съдържа определителен или неопределителен член, той също трябва да бъде анотиран, например:

A post on Facebook claims that a cure for Covid has been found.

Анотиране на публикации от социални мрежи Поради спецификата на данните, които се анотират, често се срещат аргументи, които се отнасят до публикации в социалните мрежи. В тези случаи важат следните правила:

- Ако се споменава името на група или страница, то се анотира като част от аргумента, например:

A post shared in the Facebook group "Say no to vaccines claims that a cure for Covid has been found.

- Ако се споменава държава, най-често тя не се анотира като част от аргумента, например:

  A post shared on Facebook in Nigeria claims that a cure for Covid has been found.

- Изключение към горното правило е случай като следния:

  A post shared on the Nigeria-based Facebook page "Say no to vaccines claims that a cure for Covid has been found.

  В този случай се анотира цялата фраза.

Прекъснати документи В случай, че даден документ съдържа цитат и е прекъснат в края, като например:

"A cure for Covid has been found," reads a

целият документ се анотира като обхват на събитието.

Указания за анотиране на събития от тип *Cure-Claim*

Описание на типа *Cure-Claim*

Събитие от тип *Cure-Claim* настъпва, когато някой или нещо (напр. статия, публикация в социалните медии и др.) твърди, че нещо е лек за дадено медицинско състояние.

Събитията от тип *Cure-Claim* могат да имат следните аргументи:

- ***Source*** - източникът на твърдението;
- ***Cure*** - лекът, за който се твърди, че лекува съответното състояние;
- ***Condition*** - състоянието, което се твърди, че се лекува;
- ***Patient*** - лекуваният;
- ***CureCreator*** - създателят на лека;
- ***CureAdministrator*** - този, който прилага лека.

Забележка: Аргументите не са задължителни, но е задължително да се идентифицира активиращата дума, за да се анотира събитие от съответния тип.

Следните думи (и техните производни) са потенциални активиращи думи за събития от типа *Cure-Claim*:

boost, cure, heal, relieve, remedy, treat, treatment

Примери за събития от тип *Cure-Claim*

В следните примери обхватът на събитието е *наклонен*, активиращата дума е **удебелена** и аргументите са подчертани.

Документ 1: *Dr Li Wenliang said coffee* ***cures*** *COVID-19.* A WHO representative denied the claim.

Dr Li Wenliang, *coffee* и *COVID-19* са аргументи от типове *Source*, *Cure* и *Condition* респективно.

Документ 2: *A viral tweet claims doctors in Mumbai hospital are using a new drug discovered by a local medical student to treat older patients from COVID-19.* A hospital representative was not available for comment.

A viral tweet, *doctors in Mumbai hospital*, *a new drug*, *a local medical student*, *older patients* и *COVID-19* са аргументи от типове *Source*, *CureAdministrator*, *Cure*, *CureCreator*, *Patient* и *Condition* респективно.

Специфични правила за събития от тип *Cure-Claim*

Няколко инстанции на аргумент от тип *Cure* В случай, че в рамките на едно събитие са споменати няколко възможни лека за дадено състояние, т.е. няколко възможни инстанции на аргумент от тип *Cure*, се прилагат следните правила:

- Ако лековете са споменати независимо, например:

  X, Y and Z can be used as treatments for Covid.

  следва да се анотират като отделни аргументи.

- Ако са споменати заедно, например като комбинация, се анотират като един аргумент, например:

A mixture of X, Y and Z can treat Covid.

Аргументи от тип *Cure*, съдържащи причастия В случай, че даден лек е споменат с причастие, описващо начина му на приложение, то също следва да се анотира като част от аргумента от тип *Cure*, например:

Inhaling steam can relieve a stuffy nose.

Eating coconut oil improves digestion.

Rubbing coconut oil improves skin hydration.

Указания за анотиране на събития от тип *Severe-Weather*

Описание на типа *Severe-Weather*

Събитие от тип *Severe-Weather* настъпва, когато се споменава настъпването на тежки метеорологични феномени, които могат да нанесат материални щети и да бъдат опасни за живота, като например торнада, гръмотевични бури, порои и наводнения, екстремни суши и горещини и др. [43]

Събитията от тип *Severe-Weather* могат да имат следните аргументи:

- ***Source*** - източникът на твърдението за настъпването на метеорологичния феномен;
- ***Place*** - мястото, където е настъпил феноменът;
- ***Time*** - кога е настъпил феноменът;
- ***Target*** - потърпевшият от настъпването на феномена (може да е както живо същество, така и неодушевен обект, например сграда);
- ***Cause*** - причината за настъпването на феномена;
- ***NamedPhenomenon*** - името на настъпилия феномен, в случай, че има такова.

Забележка: Аргументите не са задължителни, но е задължително да се идентифицира активиращата дума, за да се анотира събитие от съответния тип.

[43] `https://web.archive.org/web/20170103094209/https://www.wmo.int/pages/prog/www/DPS/Meetings/Wshop-SEEF_Toulouse2004/Doc3-1%281%29.doc`

Следните думи (и техните производни) са потенциални активиращи думи за събития от типа *Severe-Weather*:

avalanche, cold wave, cyclone, drought, firenado, flood, fog, heat wave, hurricane, lightning, thunderstorm, tornado, whirlpool, wildfire

Примери за събития от тип *Severe-Weather*

В следните примери обхватът на събитието е *наклонен*, активиращата дума е **удебелена** и аргументите са подчертани.

Документ 1:

Video shows flash ***flood*** *in Himachal Pradeshs Dharamshala.*

Video и *Himachal Pradeshs Dharamshala* са аргументи от типове *Source* и *Place* съответно.

Указания за анотиране на събития от тип *Rule-Change*

Описание на типа *Rule-Change*

Събитие от тип *Rule-Change* настъпва, когато някакъв закон, правило, нормативна уредба и т.н., е въведен, променен или премахнат.

Събитията от тип *Rule-Change* могат да имат следните аргументи:

- ***Source*** - източникът на твърдението за промяната на закона, правилото и т.н.;
- ***Place*** - мястото, където е въведена промяната (напр. държава, град) или което ще бъде повлияно от въведената промяна;
- ***Time*** - кога е въведена промяната;
- ***Agent*** - извършителят на промяната (напр. правителство, организация, личност и др.);
- ***Subject*** - предметът на въведената промяната.

Забележка: Аргументите не са задължителни, но е задължително да се идентифицира активиращата дума, за да се анотира събитие от съответния тип.

Следните думи (и техните производни) са потенциални активиращи думи за събития от типа *Rule-Change*:

abolish, adopt, amend, approve, authorize, ban, bar, bill, enact, decree, decriminalize, disallow, law, legalize, legislation, outlaw, overrule, pass, permit, prohibit, ratify, repeal, regulate, sign, uphold, veto

Примери за събития от тип *Rule-Change*

В следните примери обхватът на събитието е *наклонен*, активиращата дума е **удебелена** и аргументите са подчертани.

Документ 1:

As of March 23, 2023, the video platform TikTok had been banned from general use in the U.S.

March 23, 2023, *TikTok* и *the U.S.* са аргументи от типове *Time*, *Subject* и *Place* съответно.

Приложение 4. Ръководство на потребителя

Създаденият прототип на система за автоматично откриване на събития в текст е достъпен под формата на уеб приложение. То е пакетирано и се дистрибутира чрез *Docker*[44] изображение. За да се стартира системата локално, са необходими инсталирани *Docker*[45] и *docker-compose*[46].

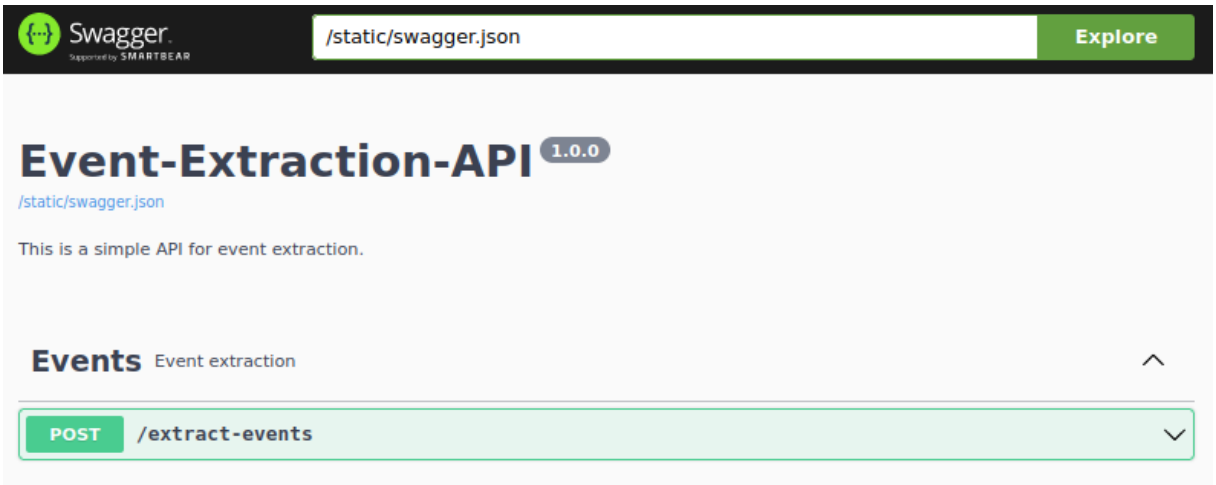
За да се създаде изображението и да се стартира системата, е необходимо в директорията *EventDetectionService* да бъдат изпълнени командите:

docker-compose build

docker-compose up

Тъй като моделът за автоматично откриване на събития е голям и се изтегля от външен ресурс, е възможно известно забавяне при стартирането.

След като системата се стартира окончателно, потребителският й интерфейс е достъпен на адрес *localhost:5000/swagger/*. Екранът, който се визуализира, е показан на фигура 28.



Фигура 28: Екран от системата за автоматично откриване на събития в текст (1)

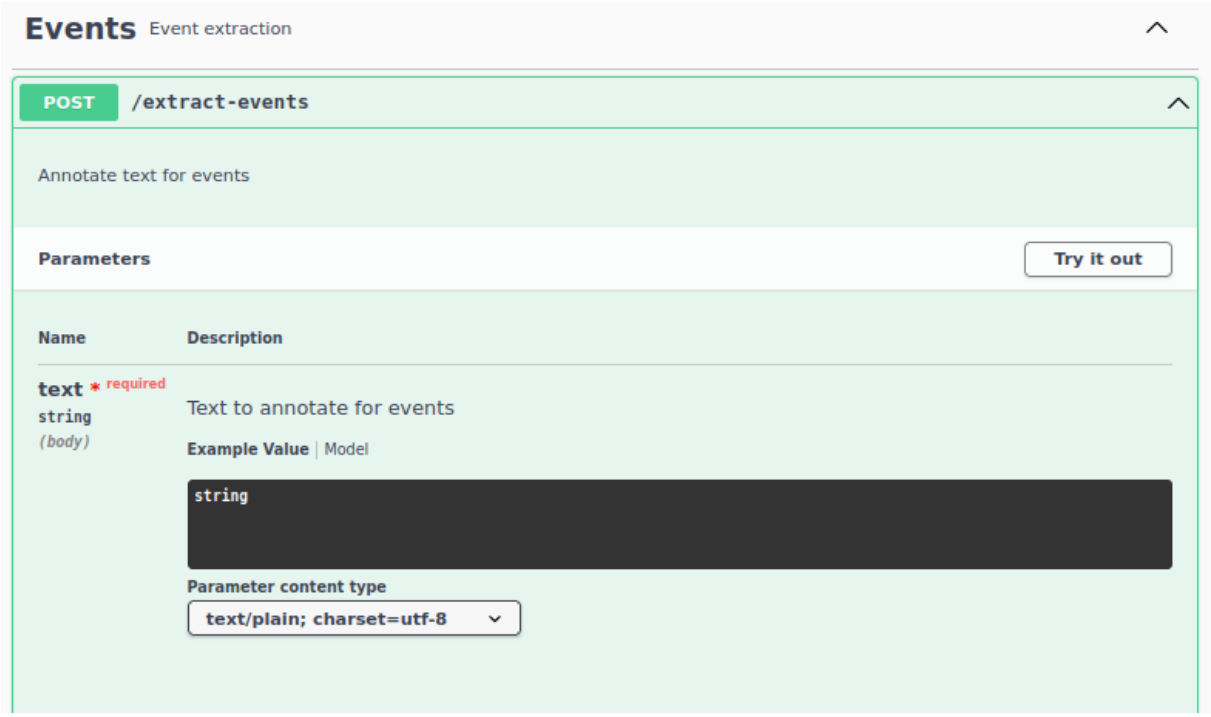
За да се пристъпи към функционалността за откриване на събития в текст, е необходимо да се щракне върху стрелката в десния край на зеления правоъгълник с надпис */extract-events*. Тогава се визуализира екрана от фигура 29. Използва се

[44] `https://www.docker.com/`

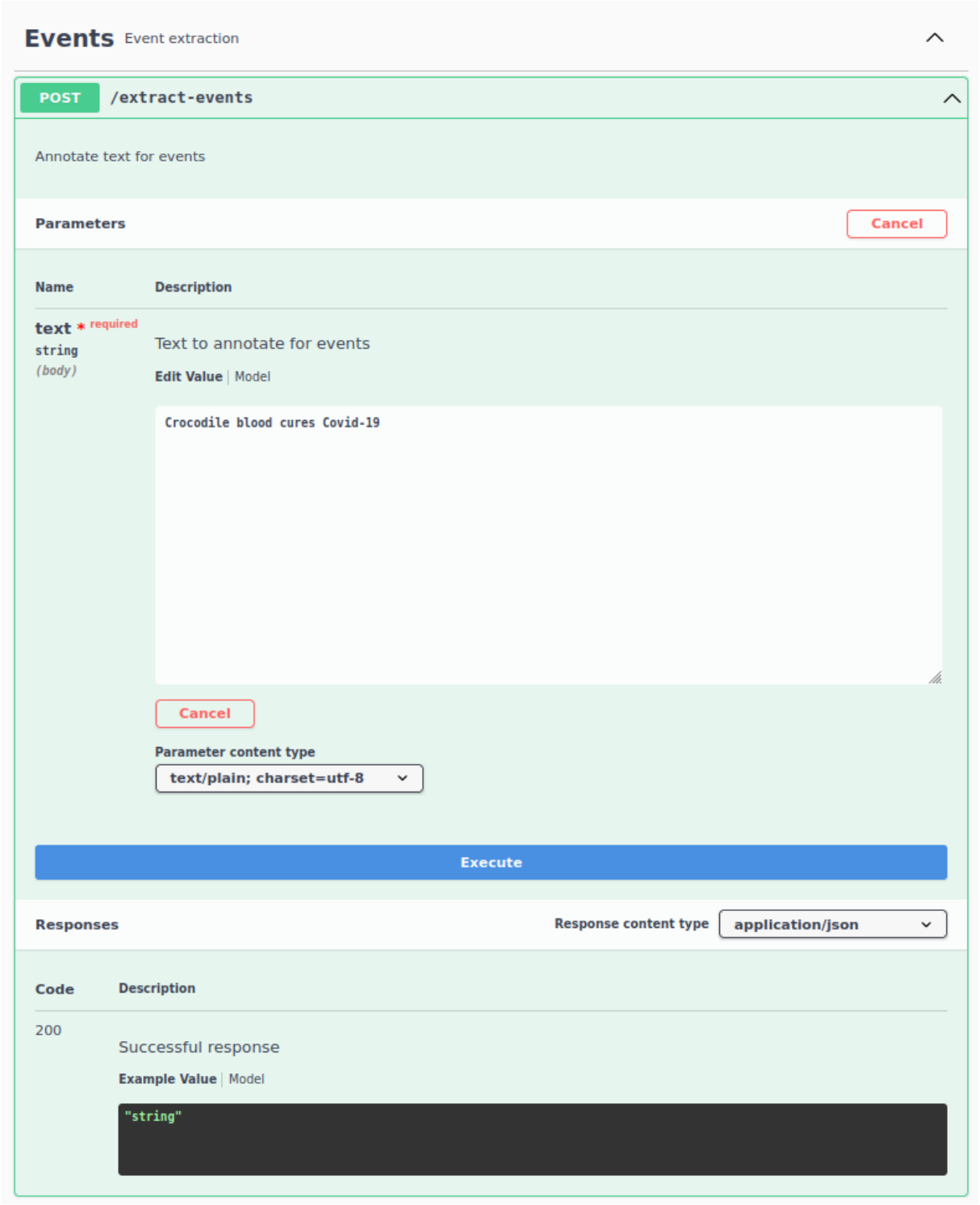
[45] Инструкции за инсталация са налични на `https://docs.docker.com/engine/install/`

[46] Инструкции за инсталация са налични на `https://docs.docker.com/compose/install/`

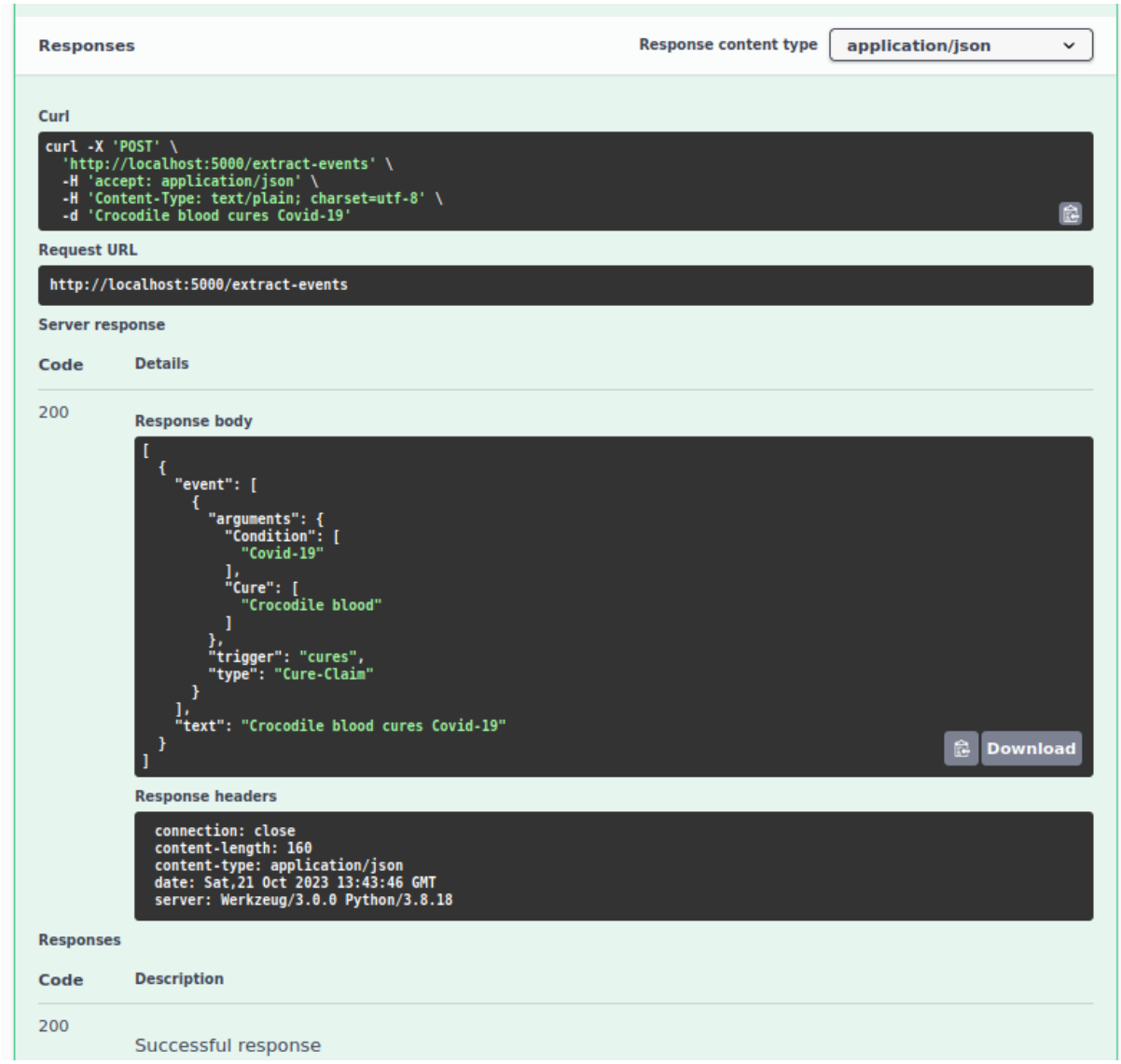
бутонът ***Try it out***, след което в бялото поле под *"Text to annotate for events"* се въвежда текста, в който ще се търсят събития, и се натиска бутонът ***Execute*** (фигура 30). Резултатът от изпълнението на заявката ще се визуализира в *JSON* формат под *Responses* и по-точно *Response body*. Резултатът може да бъде изтеглен чрез натискане на бутона ***Download*** (фигура 31).



Фигура 29: Екран от системата за автоматично откриване на събития в текст (2)



Фигура 30: Екран от системата за автоматично откриване на събития в текст (3)



Фигура 31: Екран от системата за автоматично откриване на събития в текст (4)