



Софийски университет „Св. Кл. Охридски”

Факултет по математика и информатика

ДИПЛОМНА РАБОТА

на тема

„Анонимизация на клинична
информация за пациенти”

Дипломант: **Марин Маринов Ножчев**

Специалност: **Био- и медицинска информатика**

Факултетен номер: **M22089**

Научен ръководител:

Калин Георгиев

София, 2013 г.

Съдържание

I.	Въведение	4
II.	Полза от наличието на анонимизирани данни	5
III.	Теоретични аспекти на медицинската анонимизация.....	7
1	Дефиниция на лично-идентифицираща информация	7
2	Аспекти на анонимизацията.....	7
3	Дефиниция на задача на дипломната работа	9
4	Критерии за качество на анонимизацията	11
5	Обзор на публикуваните научни разработки в медицинската анонимизация.....	13
5.1	Анонимизация базирана на машинно самообучение.	13
5.2	Анонимизация на медицински текстове базирана на правила	18
IV.	Цели и задачи на дипломната работа.....	22
V.	Изследователска част	23
1	Архитектура на софтуера за анонимизация	23
2	Спецификация.....	25
2.1	Вход и изход	25
2.2	Изпълнение и дистрибуция на приложението.....	26
2.3	Входни точки за конфигурация.....	26
2.4	Възможности за интеграция	28
3	Компоненти на анонимизацията в детайли	28
3.1	Разпознаване на именувани обекти	28
3.2	Разпознаване на възраст и дати.....	30
3.3	Разпознаване на телефони и други идентифициращи номера	34
4	Резултати от оценката на точността на алгоритъма.....	36

	5	Бъдещо развитие на приложението	36
VI.		Заключение	38
VII.		Библиография	40
VIII.		Приложение 1	44

I. Въведение

Наличието на биомедицински данни за пациенти е критично за изследванията в медицината. Основен източник на такива данни са болниците, но здравните заведения нямат право по закон да ги разпространяват без анонимизиране [1]. Неразпространението на лични данни на пациентите е също част от Хипократовата клетва [2]. Въпреки тези ограничения, медицинските данни на пациенти могат да се използват за изследвания ако се трансформират по начин, който не позволява променените данни да се свържат с конкретни пациенти. Този процес се нарича „медицинска анонимизация“.

Анонимизацията често се извършва тривиално чрез заменяне на имената на пациентите с уникални идентификатори. Така се запазва цялата релевантна за клинични изследвания информация като възраст и местопребиваване на пациентите. За съжаление, с разпространението на и полу-публични бази данни с лична информация на населението като Facebook, все повече се улеснява свързването на частично идентифицираща информация като година на раждане, местоживееене и пол с записи в клинични бази данни, позволявайки де-анонимизирането им. Разбира се, този риск може да се избегне чрез премахването на всяка потенциално идентифицираща информация, но по този начин се намалява пригодността на данните за клинични изследвания.

Основна пречка за пълната анонимизация са немаркираните лични данни, особено в неструктурирани медицински текстове като анамнези и диагнози в амбулаторни листи. В тези документи често се споменават данни за пациентите дори извън съответните полета на формуляра. Например възрастта на пациента често се споменава в диагнозата, ако е релевантна към нея.

Целта на дипломната работа е създаване на софтуер за автоматично анонимизиране на свободен медицински текст с минимална загуба на не-идентифициращи данни.

II. Полза от наличието на анонимизирани данни

Модерното здравеопазване създава огромно количество данни и затова болници и лекари все по-често използват информационни технологии за да ги управляват [3]. Технологиите позволяват извличане на медицински знания от агрегираните данни за лечението на пациенти. Това може да подобри разбирането за протичането на различни болести, да ускори тяхната диагностика и да даде експериментално потвърдена информация за най-добрите подходи за управление и лечение на техните симптоми [4].

Извличане на знания – data mining – наричаме автоматичния или полу-автоматичния анализ на големи количества от данни с цел да се открият неизвестни зависимости между параметри, необичайни записи или групи от подобни записи. На базата на това могат да се стигне до нови знания в областта, в която оригиналните данни са създадени [5]. В медицината този подход има голям потенциал, но в близкото минало извличането на медицински знания се е концентрирало върху публично достъпни административни и държавни бази от данни относно разпространение на заболявания и епидемии [4]. Тези източници, обаче, не дават пълната картина, която намираме в документите, които лекарите създават всеки ден.

Един пример за използване на клинични данни за откриване на медицинско знание е опита на учени от Baylor College of Medicine, Хюстън. Те са използвали извличане на знания, за да оценят риска от повторно появяване на тумори, в рамките на пет години, при пациенти, преминали лечение на рак на простатата. При експеримент с данните на 983 пациенти, екипът е постигнал 79% точност на прогнозата [6]. Примерът показва, че анализът на големи количества данни подобрява оценяването на ефекта от различни медицински процедури.

Факторът, който най-много затруднява развитието на анализа на медицински данни, е трудността на тяхното споделяне. Медицинската тайна [2] е основата на доверието между доктори и пациенти. Свързването на конкретни хора с информация за техните заболявания не може да се допусне, тъй като влияе отрицателно на живота на пациентите като част от обществото.

Този конфликт може да се разреши чрез изолиране на лично идентифициращата информация и анонимизиране на медицинските документи.

III. Теоретични аспекти на медицинската анонимизация

1 Дефиниция на лично-идентифицираща информация

За правилно дефиниране на задачата за анонимизация на медицински текстове първо ще да изброим видовете данни, които биха свързали пациента с информацията за тяхното заболяване. Американското законодателство дава един от най-добре подбраните списъци от видове лично-идентифицираща информация. Правилникът за прилагане на закона за управление на сигурността на информацията (Federal Information Security Management Act, FISMA 2002) по отношение на медицински данни [7] дефинира следните видове лична информация:

- Името на пациента
- Имена на техни близки и пълномощници
- Имената на докторите
- Различни видове идентифициращи номера, които включват както номера на документи, така и номера, които свързват пациента с документацията, която болницата пази за него
- Телефонен номер
- Адрес и други географска информация, например местоживеене
- Името на болницата
- Точни дати

2 Аспекти на анонимизацията

Както всяка организация, лечебните заведения съхраняват както структурирани, така и неструктурирани данни за своите пациенти. Структурирани наричаме данни, които благодарение на позната си схема са годни за машинна обработка. В това число влизат XML документи с известна схема и релационни бази данни. Всички останали данни са неподходящи за директна машинна обработка и се включват в понятието не-структурирани данни [8].

Търсенето на лична информация в структурирани данни се свежда до откриването на концепции, дефинирани в схемата на данните, които са лична информация. Намирането на съответствие между видовете обекти с техните атрибути в медицинската база данни и обектите и атрибутите, които образуват личната информация, се свежда до задачата за ontology alignment [9]. Тази задача е трудна за пълно автоматизиране [9], но изисква малко ресурси при частично ръчно решение. Достатъчно е ръчно да се маркират кои понятия в схемата на данните са лична информация и след това тривиален алгоритъм може да обработи всички полета, които отговарят на маркираните понятия.

Откриването на лични данни в неструктурирана информация – като различни видове записки на лекари – е по-трудна задача. При нея нямаме схема, които описват смисъла на данните, във формат, който е удобен за компютърен анализ. Изследванията в медицинската анонимизация се концентрират върху този проблем, защото ръчното маркиране на лични данни в такъв текст изисква големи ресурси [10] [11]. Това е и задачата на дипломната работа. В следващата точка ще бъде разгледана подробно дефиницията на този проблем.

За да се завърши анонимизацията, след като се маркират личните данни, те се премахват или заменят с друг текст. Заменянето се предпочита пред премахването, защото запазва отчасти смисъла на данните и четливостта им [12]. За целта, например, в медицинска диагноза името на пациента не се изтрива, а се заменя с идентификатор като <име_на_пациент_1>. Всяко споменаване на същото име се заменя със същия идентификатор. За да се запази текста четлив, вместо идентификатор като горния може да използва произволно име на човек от даден речник [13]. По аналогичен начин се заменят дати, имена на институции и други лични данни.

3 Дефиниция на задача на дипломната работа

Основна задача на дипломната работа е разработката на софтуер за анонимизация на неструктурирани медицински текстове като диагнози и доклади за протичане на лечение на пациент. Предвид горния анализ на аспектите на анонимизацията се открояват следните изисквания към софтуера:

- да работи със свободен текст без предварително маркирани лично-идентифициращи данни
- да маркира изредените в т. III.1 типове лични данни и да може да бъде разширяван за да поддържа други видове данни
- да заменя маркирани лични данни с подходящ за типа им заместител

Приложението обработва текстове на английски език и може да бъде разширявано да поддържа други езици. Английският език беше избран заради наличието на голямо количество достъпни примерни медицински доклади [10].

Софтуерът е нова реализация на известен подход за анонимизация на текстове – прилагане на разпознаване на именувани обекти (NER) за откриване на лични данни. Той е обединение от Java [14] и JAPE компоненти [15] и е базиран на известната платформа с отворен код за анализ на текст – GATE [16]. Разпознаването на именувани обекти е базирано на ANNIE – A Nearly New Information Extraction [17].

Пример за текст, с който софтуерът работи¹:

¹ Примерният текст е анонимизиран.

ADMISSION DATE : 11/19/94

DISCHARGE DATE : 11/28/94

ADMISSION DIAGNOSIS : Aspiration pneumonia , esophageal laceration .

HISTORY OF PRESENT ILLNESS : Mr. Blind is a 79-year-old white male (sic) with a history of diabetes mellitus , inferior myocardial infarction , who underwent open repair of his increased diverticulum November 13th at Sephsandpot Center . The patient developed hematemesis November 15th and was intubated for respiratory distress . He was transferred to the Valtawnprinceel Community Memorial Hospital for endoscopy and esophagoscopy on the 16th of November which showed a 2 cm linear tear of the esophagus at 30 to 32 cm . The patient 's hematocrit was stable and he was given no further intervention . The patient attempted a gastrografen swallow on the 21st , but was unable to cooperate with probable aspiration . The patient also had been receiving generous intravenous hydration during the period for which he was NPO for his esophageal tear and intravenous Lasix for a question of pulmonary congestion . On the morning of the 22nd the patient developed tachypnea with a chest X-ray showing a question of congestive heart failure

Фигура 1

Пример за изходен анонимизиран текст:

ADMISSION DATE : [[DATE]]

DISCHARGE DATE : [[DATE]]

ADMISSION DIAGNOSIS : Aspiration pneumonia , esophageal laceration .

HISTORY OF PRESENT ILLNESS : Mr. Smith is a 79-year-old white white male (sic) with a history of diabetes mellitus , inferior myocardial infarction , who underwent open repair of his increased diverticulum [[DATE]] at Sephsandpot Center . The patient developed hematemesis [[DATE]] and was intubated for respiratory distress . He was transferred to the Valtawnprinceel Community Memorial Hospital for endoscopy and esophagoscopy on the [[DATE]] which showed a 2 cm linear tear of the esophagus at 30 to 32 cm . The patient's hematocrit was stable and he was given no further intervention . The patient attempted a gastrografin swallow on the 21st , but was unable to cooperate with probable aspiration . The patient also had been receiving generous intravenous hydration during the period for which he was NPO for his esophageal tear and intravenous Lasix for a question of pulmonary congestion . On the morning of the [[DATE]] the patient developed tachypnea with a chest X-ray showing a question of congestive heart failure

Фигура 2

4 Критерии за качество на анонимизацията

По подобие на откриването на именувани обекти в текст, качеството на анонимизацията се оценява на базата на това каква част от единиците на текста са правилно маркирани като лична информация [10]. (Тук и по-нататък в изложението, под „единица на текста“ разбираме понятието token в англоезичната литература [18] – лингвистично значима поредица от знаци, отделена с интервали или пунктуация.) В частност, задачата се разглежда като бинарен въпрос: дадената текстова единица принадлежи ли на лични данни от тип А?. Така резултатите на алгоритъма върху корпус от текстове могат да се опишат от следните 4 стойности:

- Верни положителни отговори на бинарния въпрос – true positives
– TP

- Верни отрицателни отговори – true negatives – TN
- Грешни положителни отговори – false positives – FP
- Грешни отрицателни отговори – false negatives – FN

За да нормализираме резултатите спрямо броя на текстови единици в корпуса от текстове, не можем да използваме тривиална мярка като $\frac{\text{верни отговори}}{\text{общ брой отговори}}$, защото много по-голямата част от единиците не принадлежат на лична информация. Един алгоритъм, който винаги отговаря „не“, би получил висока оценка, ако се използва тривиалната мярка. Вместо това ще използваме стандартните статистически мерки – „точност“ (в английската литература позната като precision, specificity, positive predictive value) и „чувствителност“ (recall, sensitivity, true positive rate). Тъй като преводите на precision и recall не са разпространени в българската литература в останалата част от дипломната работа ще използвам английските имена. Тези мерки се дефинират по следния начин:

$$\text{precision: } P = \frac{TP}{TP + FP}$$

$$\text{recall: } R = \frac{TP}{TP + FN}$$

$$F - \text{score: } F_{\beta} = \frac{(1 + \beta^2)P \cdot R}{\beta^2 P + R}$$

F-мярката (F-score) е число между 0 и 1, което представлява нормализирана мярка за качеството на анонимизационния алгоритъм, разглеждан като бинарен класификатор. β е коефициент за предпочитанието върху recall срещу precision. При $\beta = 2$, recall се приема за два пъти по-важен от precision. Тъй като recall измерва каква част от личните данни са маркирани от алгоритъма, $\beta > 1$ отразява изискването към автоматичната анонимизация да поставя нуждите на пациентите пред нуждите на лекарите.

5 Обзор на публикуваните научни разработки в медицинската анонимизация

Маркирането на лично-идентифициращи данни в медицински текстове се изследва активно в последните години и в резултат са изпробвани различни подходи за решаване на задачата. В литературата се откриват три основни групи решения:

- Алгоритми, базирани на статистически класификатори – техника, позната като машинно самообучение. Използваните класификатори включват линейни класификатори като support vector machines (SVM), контекстни класификатори като скрити Маркови модели (Hidden Markov Models, HMM) и Conditional Random Fields.
- Алгоритми, базирани на лексикални и граматически правила
- Комбинация от статистически класификатор и правила.

5.1 Анонимизация базирана на машинно самообучение.

При машинното самообучение на статистически класификатори, голямо количество от текстове с предварително маркирани лични данни се използват за обучение на класификатора. Ефективното обучение на класификатора зависи от ръчния или полу-автоматичен подбор на отличаващи атрибути на текстовите единици. Например отличаващия атрибут на телефонния номер 1-800-5555 не са цифрите 1, 8 и 5, а поредицата <цифра>-<три цифри>-<четири цифри> [19].

Добър пример за прилагане на машинно самообучение за анонимизация е участието на екипът на Wellner et al. [11] в състезанието на организацията „Informatics for Integrating Biology and the Bedside“ (i2b2) за анонимизация на медицински текст. Те свеждат задачата до скрит Марковски модел, използвайки реализацията в LingPipe за обучение на модела. Въпреки липсата на подробности в статията, използвайки документацията на LingPipe за HMM [20] може да се стигне до извода, че екипът е представил задачата за анонимизация като конкретизация на скрит Марковски модел, наречен Марковски модел с максимална ентропия (Maximum Entropy Markov Model,

MEMM) [21] . За да се изясни техният подход, нека разгледаме дефинициите на скрит Марковски модел и MEMM.

Нека имаме процес, който на всяка стъпка избира състояние от крайно множество S . На всяка стъпка по случаен начин също се избира „наблюдение“ на базата на текущото състояние от крайно множество наблюдения O . Изборът на наблюдение зависи само от текущото състояние (долу s_n, o_n са съответно състоянието и наблюдението на стъпка n) :

$$P(o_n | s_1 \dots s_n) = P(o_n | s_n)$$

Процесът се нарича Марковски, ако изборът на състояние зависи само от състоянието на предишната стъпка, а не зависи директно нито от предишните наблюдения, нито от останалите предишни състояния:

$$P(s_n | s_1 \dots s_{n-1}) = P(s_n | s_{n-1})$$

Така един скрит Марковски модел може изцяло да се опише от множествата S, O и условните вероятности $P(s|s'), P(o|s)$, където s' е състоянието преди s . Две основни задачи произтичат от дефиницията на HMM:

- Обучение на модела – по дадена поредица от n състояния и наблюдения, да се намерят вероятностите $P(s|s'), P(o|s), \forall s \in S, \forall o \in O$, така че $P(o_1 \dots o_n | s_1 \dots s_n)$ е максимална.
- Прилагане на модела – по дадена поредица от n наблюдения и даден Марковски модел, да се намерят състоянията, съответстващи на всяко наблюдение така че $P(o_1 \dots o_n | s_1 \dots s_n)$ е максимална. Колкото по-висока е намерената максимална вероятност P толкова по-предсказуем е Марковския процес. Целта при моделирането на естествени процеси - като редуването на лични и не-лични данни в медицински текст - със скрити Марковски процеси е такъв подбор на състояния и наблюдения, че полученият Марковски процес да бъде максимално предсказуем.

Дотук дефинирахме обикновен скрит Марковски модел. Използването на такъв модел за решаването на задача за анонимизация е вид машинно самообучение. Например нека изберем множество на състоянията на процеса с два елемента - $S: \{true, false\}$ – дали текущата текстова единица е част от лични данни или не. Нека наблюденията от процеса да бъдат множеството от всички текстови единици. Тогава задачата за обучение на Марковски модел отговаря на обучаване на машината как да разпознава лични данни в текст. Този подход има някои ограничения:

- Очевидно фактът дали предишната текстова единица е част от лични данни не е достатъчна информация дали текущата е, особено ако самата текуща дума е двусмислена. Например Huntington е както име на човек, така и име на болест. В изречението „Пациентът Хънтингтън е болен от болестта на Хънтингтън.“, думите „Пациентът“ и „на“ предшестват две споменавания на „Хънтингтън“. Едното от тях е част от лични данни, а другото не. И двете предшестващи думи не са лични данни.
- Скрытият Марковски модел предполага, че множеството от наблюдения е крайно и известно. Множеството от всички възможни текстови единици е крайно, но не е известно.

За да заобиколим тези проблеми правим следните промени в модела:

- Множеството от състояния е подмножество на наредените тройки: $\{(PII(n-1), PII(n), PII(n+1))\}$. Тук $PII(i)$ е функция, която за дадена единица на позиция i в текста, показва дали единицата е част от лични данни от определен тип.
- При прилагане на модела, $P(o|s)$ за o , което не е било срещано при обучение, се приема, че е $1/\text{брой известни наблюдения}$.

Приносът на Maximum Entropy Markov Model е включването в модела на множество от функции, дефинирани върху множеството от наблюдения. В общата дефиниция на MEMM множеството от резултати на всяка от тези функции може да бъде както \mathbb{R} , така и всяко крайно множество. В случая,

Wellner et al. използват булеви функции и кардинални функции. Тези функции позволяват да се опишат представените по-горе отличаващи атрибути (features) на наблюдението.

Така скритият Марковски модел се конкретизира до Марковски модел с максимална ентропия:

$$P(s|s', o) = \frac{1}{Z(o, s')} \exp \left(\sum_a \lambda_a f_a(o) \right)$$

Тук f_a са функциите на атрибутите, λ_a са параметри, които определят относителна тежест на всеки атрибут към получената вероятност, а Z е функция, която е нужно да се избере така, че дясната част да бъде валидна вероятност, т.е. да бъде в интервала 0 .. 1. Предимството на тази дефиниция на вероятността на преход е, че разпределението на вероятността е с максимална ентропия [22] [23].

С тези дефиниции вече можем да опишем решението на задачата анонимизация. Използваме MEMM със следните свойства [20]:

- Състояния на Марковския процес
 - B_X – начална текстова единица на лични данни от тип X
 - M_X – междинна текстова единица на лични данни
 - E_X – крайна текстова единица на лични данни
 - W_X – самостоятелна текстова единица на лични данни от тип X
 - BB_O_X – текстова единица, която не е част от лични данни, но предишната е край на данни от тип X
 - EE_O_X - текстова единица, която не е част от лични данни, нито е от предходния тип, но следващата започва лични данни от тип X
 - WW_O_X - текстова единица, който не е част от лични данни, но предишната е край на лични данни от някакъв тип, а следващата е начало на данни от тип X

- MM_O – текстова единица, която не е част от лични данни и нито предишната, нито следващата са част от такива
- „Наблюдение“ на Марковския процес – самата текстова единица
- Атрибути на наблюдението (features)
 - Съкращение – по-къса от 4 букви и завършва на точка
 - Ортография на текстовата единица – дали започва с главна буква или е изписан изцяло с главни букви, изцяло с малки или смесен
 - Съдържа ли цифри – не съдържа цифри; започва с цифра, завършва с цифра; съдържа само цифри
 - Съдържа ли пунктуация - като ., -
 - Отговаря ли на формата на телефонен номер – да; не
 - Отговаря ли на формата на дата – да; не;
 - Корен на думата

Създаденият на тази база скрит Марков модел с реализацията на LingPipe за обучение и прилагане е постигнал следните резултати на i2b2 Medical De-Identification Challenge [10] :

Precision	0,9212
Recall	0,9430
F_1	0,9320
F_2	0,9386

Основният недостатък на машинното самообучение при анонимизацията на медицински текстове и въобще при анализа на текст е нуждата от голямо количество обучаващи данни. При повечето алгоритми няма лесен и гъвкав начин за повишаване на точността чрез кодиране на експертно знание. Например, професионалист в медицината може да подобри резултатите на горе-описания алгоритъм само по следните начини:

1. Като подбере атрибути, базирани на наблюденията – По дефиниция обаче атрибутите могат да зависят само от даденото наблюдение. За да се изразят по-сложни правила е нужно множеството от наблюдения да се базира на уникални поредици над две текстови единици, което прави Марковски модел по-сложен и по-бавно сходим.
2. Като създаде речници от ключови думи, които подсказват смисъла на текстовите единици. Например дума, която е първо име на човек, е почти сигурен белег за лични данни и списък на първи имена може да подобри значително работата на класификатора. В МЕММ, това е частен случай на горният подход – принадлежността на текстова единица към даден списък е булев атрибут на единицата. В обикновените скрити Марковски модели речниците могат да се използват да се замени наблюдението с вектор от булеви атрибути, всеки от които показва принадлежността към даден речник. По този начин може да се намали ентропията на случайната величина $P(o|s)$. Формално търсим следната замяна

$f: o \rightarrow o' | \forall s \in S, H(P(o'|s)) < H(P(o|s))$, където

$$H(X) = - \sum_i P(x_i) \ln P(x_i)$$

3. Като ръчно маркира текст, увеличавайки корпуса за обучение. Това е обаче трудоемка задача, която е сравнително по-малко ефективна от създаването на алгоритъм базиран на правила.

5.2 Анонимизация на медицински текстове, базирана на правила

Класическият подход за анализ на текст е създаването на алгоритъм, базиран на множество правила, които изразяват знанията на експерти в областта [24]. Например, експерти могат да кажат, че съкращението Д-р. никога не се слага пред болест и съответно в израза „болестта на Ножчев“ фамилното име никога не реферира пациента. Тези прости правила решават описания по-горе пример за двусмисленост между лични данни на

пациент и име на болест. Подходът е особено полезен при по-тясно специализирани области, където наличието на експертни знания е по-голямо от обема на достъпни данни за машинно самообучение. В медицината има много малко сборници от текстове за обучение на класификатори на лични данни. Показателно е, че голяма част от разработките се базират на едни и същи документи - сборника на (Американски) Национален център за биомедицинска информатика [10].

Добър пример за прилагане на правила за премахване на лично-идентифициращи данни в медицински текст е работата на Ishna Neamatullah et. al [25]. Техният алгоритъм използва речници от думи, регулярни изрази и някои евристики. Специфично в техния подход е настройването на алгоритъма към документите на конкретна болница. Използваните речници са следните:

- База данни от имената на пациентите и лекарите в болницата
- Речник от често срещани английски имена и често срещани префикси и суфикси на имена
- Речник от английски думи и речникът от медицински термини Unified Medical Language System (UMLS) [26]
- Списък от индикатори на лични данни като титли - Гн., Д-р,
- Индикатори на имена – майка, син, отговорно лице,
- Индикатори на места като – болница, град, улица
- Ключови думи за възраст – „пациентът е на ... години, .. е на възраст“.

Важно е да се отбележи че тези речници имат общи членове, които при този подход са били специално отбелязани като „неясни“. Авторите са положили специални усилия за да отделят речниците от реализацията на алгоритъма, което позволява решението да се пренастройва към други типове документи.

Следните лични данни са откривани чрез регулярни изрази, реализирани на езикът за програмиране Perl (приложенията към [25] съдържат пълен списък):

- Адреси на улици, номера на пощенски кутии
- Дати
- Телефонни номера

Накрая, авторите са използвали няколко евристични правила, които допълват алгоритъма. Специално внимание е обърнато на маркирането на имена на пациенти, които авторите наричат „най-ценната категория лични данни“. Алгоритъмът маркира като лични данни всички текстови единици, които са в речниците от имена, но не са в речника с медицински термини. Думите, които са в двата речника, се анонимизират, ако в съседство има други текстови единици, които са имена. Поредиците <първо име> <фамилия>; <фамилия>, <първо име>; <първо име> <презиме> <фамилия> и <първо име> <инициал> <фамилия> винаги се маркират като лични данни, дори някоя от текстовите единици да е в речника на медицински термини.

За да оценят качеството на подхода си, авторите са изпитали алгоритъма върху тестов корпус от 1836 записки на медицински сестри с общо 296 400 текстови единици. Авторите са дали данни само за recall:

Тип лични данни	Пропуснати лични данни от този тип	Пропуснати за 100 000 думи	Recall
-----------------	------------------------------------	----------------------------	---------------

Име на пациента	49	17	
Място (без номер на улица)	7	2	
Пълна дата	2	1	
Частична дата	9	3	
Година	8	3	
Възраст ²	3	1	
Общо	78	27	0,94

² Алгоритъмът анонимизира само възрасти над 89 години.

IV. Цели и задачи на дипломната работа

Целта на дипломната работа е подобряване на достъпа на изследователите в областта на биологията и медицината до анонимизирани медицински текстове чрез създаването на гъвкав автоматичен метод за премахване на лични данни.

За да бъде осъществена целта на дипломната работа, са изпълнени следните задачи:

- Направен е обзор на съществуващите разработки в медицинската анонимизация и са сравнени реализираните алгоритми за премахване на лични данни.
- Разработен е метод за анонимизация, който е по-гъвкав и лесен за конфигуриране спрямо съществуващите. Така методът е по-удобен за прилагане върху голямото разнообразие от неструктурирани медицински данни. Гъвкавостта също му позволява да бъде адаптиран за други езици като български.
- Реализиран е софтуер, базиран на широко разпространена платформа за анализ на текст. Използваните технологии позволяват лесно интегриране на софтуера в по-големи системи.

V. Изследователска част

1 Архитектура на софтуера за анонимизация

От анализа на съществуващите решения се вижда, че изискването за гъвкавост на софтуера най-добре може да бъде изпълнено от решение, базирано на правила. Решението следва осем стъпки:

1. Конвертира форматирането на текста до XML подходящ за избраната платформа за анализ на текст.
2. Разделя текста на текстови единици и изречения.
3. Маркира части на речта чрез „плитък“ парсер [24] ³.
4. На базата на правила, които използват предишният анализ и външни речници, се маркират именуваните обекти.
5. На базата на правила се маркират личните данни, които не са именувани обекти (възраст).
6. Личните данни се премахват.
7. Анонимизираният текст се записва в текстов файл.

Изборът на подходяща платформа е критичен за качеството на архитектурата на решението. Най-широко използваните Java-базирани платформи за анализ на текст са [27]:

- GATE (General Architecture for Text Engineering), разработван от Университета на Шефилд, Великобритания [17].
- UIMA (Unstructured Information Management Applications), система за анализ на текст, спонсорирана от IBM и разработвана от фондация Apache [28]
- LingPipe, разработван от Alias-I, Inc [29]

В Таблица 1 са сравнени трите платформи по характеристиките, които имат отношение към задачите на дипломната работа:

³ Тук понятието „плитък“ парсер отговаря на shallow parser в англоезичната литература. Това е алгоритъм, който маркира частите на изречението - подлог, определение, допълнение и др. – но не определя връзките между тези части. Например, „плитък“ парсер няма да покаже дали дадено определение се отнася към подлога или допълнението в едно изречение.

Характеристика	GATE	UIMA	LingPipe
Интегрирана графична среда за разработка на анализ на текст	Да	Да	Не
Лесна за интегриране библиотека от алгоритми	Да	Не	Да
Разнообразие от готови компоненти	Да	Да	Не
Архитектурна поддръжка на други езици	Да	Да	Слаба
Поддръжка на компоненти от другите платформи	Да (UIMA, LingPipe)	Да (GATE)	Не
Версия, използвана при сравнението ⁴	7.1	2.4	4.1

GATE 7.1 има видимо предимство пред алтернативите и е най-добрият избор за платформа.

Решения, базирани на GATE, се създават на базата на три вида компоненти:

1. Анализиращи ресурси (Processing Resources, PR) – софтуерни компоненти, написани на езика за програмиране Java, пакетирани като един или няколко jar архива. Входът за всеки PR е множество от именувани анотации върху документ
2. JAPE анализатори – специален вид анализиращ ресурс, който маркира текст, базиран на дадена граматика от тип JAPE (Java Annotation Patterns Engine) [30]
3. Corpus Pipeline (CP) – обединение от анализиращи ресурси, които се изпълняват в определен ред, като изходът от един ресурс е вход на следващия. CP се описват декларативно чрез XML-

⁴ За всички платформи, това е най-новата версия към момента.

базиран език. За улеснение, те също могат да се дефинират чрез графичният интерфейс на GATE Developer.

CP могат да се влагат един в друг. Ако разглеждаме PR като функции, то CP е композиция на PR. Тъй като дефиниционно множество на PR съвпада с множеството на резултатите, PR и CP могат да се композират по произволен начин. От гледна точка на софтуерното инженерство имаме интерфейс ProcessingResource, който се реализира от JAPE анализатор, Corpus Pipeline или от произволен Java component. Софтуерът за анонимизация се възползва от възможността за композиция. Това се вижда на схемата на използваните анализираци ресурси в Приложение 1.

2 Спецификация

2.1 Вход и изход

Приложението приема на вход текстови файлове или XML файлове, които съдържат допълнителни метаданни. Един XML файл може да съдържа един или няколко медицински доклада. Примерен XML файл:

```
<ROOT>
  <RECORD ID="502">
    <TEXT>
      ... Тук се съдържа текст от типа на примера във Фигура 1. ...
    </TEXT>
  <RECORD ID="503">
    <TEXT>
      ...
    </TEXT>
  </RECORD>
</ROOT>
```

Фигура 3

Изходът от програмата е един или няколко текстови файла, в зависимост от това дали входът е единичен текстов файл или XML с множество записи. Съдържанието им е текст от типа на примера на Фигура 2 на стр. 9. Схемата на XML файловете е избрана за съвместимост с формата на примерните медицински данни от колекцията на I2B2 [10].

2.2 Изпълнение и дистрибуция на приложението

Приложението е пакетирено като стандартна Java апликация. Стартира се с команда с формат:

```
anon.cmd <входен файл> <директория, където ще се запише изхода>
```

При операционна система Linux, скриптът, който стартира приложението, е anon.sh вместо anon.cmd. Софтуерът се дистрибутира като zip със следната вътрешна структура:

/bin	Скриптове за стартиране на приложението
/lib	Библиотеки от тип jar
/classes	Класове и ресурси
/jape	JAPE граматики
/conf	Конфигурационни файлове, вкл. Corpus Pipeline

Външните изисквания на приложението са runtime environment на Java 7 (JRE) и инсталация на GATE Developer или Embedded, версия 7.1 или по-нова.


2.3 Входни точки за конфигурация

Основната начин за конфигуриране на приложението е променянето на параметри в XML конфигурацията на GATE Corpus Pipeline, записана в /conf/anon_v1.gapp. Пример за схемата на конфигурацията има на Фигура 4. За




удобство може да се използва графичната среда GATE Developer. Фигура 5 показва как се променят параметри на приложението в графичната среда.

```
<runtimeParams class="gate.util.persistence.MapPersistence">
  <mapType>gate.util.SimpleFeatureMapImpl</mapType>
  <localMap>
    <entry>
      <string>wholeWordsOnly</string>
      <boolean>true</boolean>
    </entry>
    <entry>
      <string>longestMatchOnly</string>
      <boolean>true</boolean>
    </entry>
    <entry>
      <string>document</string>
      <null/>
    </entry>
    <entry>
      <string>corpus</string>
      <null/>
    </entry>
    <entry>
      <string>annotationSetName</string>
      <null/>
    </entry>
  </localMap>
</runtimeParams>
<resourceType>gate.creole.gazetteer.DefaultGazetteer</resourceType>
<resourceName>ANNIE Gazetteer</resourceName>
```

Фигура 4

Corpus:  Corpus for uk-wales-south-east-wales-21864741@print=true.html_00061

Runtime Parameters for the "ANNIE Gazetteer" ANNIE Gazetteer:

Name	Type	Required	Value
 annotationSetName	String		
 longestMatchOnly	Boolean	✓	true
 wholeWordsOnly	Boolean	✓	true

Фигура 5

2.4 Възможности за интеграция

Приложението за анонимизация би било най-ценно като компонент на по-голяма система за управление на медицински данни или болнична ERP⁵ система. За тази цел, софтуерът предоставя няколко възможности за интеграция.

Java приложения могат да използват софтуера за анонимизация като обикновена библиотека. Класът `medanon.AnonymizerFactory` е входна точка за библиотеката. Тъй като софтуерът и GATE Embedded използват много външни библиотеки, могат да се получат конфликти, ако външният софтуер вече използва други версии на същите компоненти. За решаване на този проблем се препоръчва използване на система за изолация на компоненти като OSGi [31].

Приложения на други езици могат да използват Java Native Interface (JNI), за да използват анонимизацията като библиотека [32]. Скриптов езици като Python също биха могли да ползват command-line интерфейса като най-удобна входна точка.

Накрая, приложения за анализ на текст, които вече са базирани на GATE, могат да използват анонимизацията като GATE компонент от тип Corpus Pipeline (CP) по описаната на стр. 23 схема.

3 Компоненти на анонимизацията в детайли

3.1 Разпознаване на именувани обекти

Понятието „именувани обекти“ включва [17]:

- имена на хора; В медицински текстове това са имена на пациенти и доктори
- организации; В случая това са имена на болници и други медицински заведения

⁵ ERP – Enterprise Resource Planning – са софтуерни продукти, които покриват всички аспекти на работата на дадена компания.

- места и географски обекти; В случая това са пълни или частични адреси на пациенти и местоположения на болници.

Разпознаването на именувани обекти – named entity recognition (NER) - е базирано на системата ANNIE [17]. ANNIE беше избрана след експеримент, който сравни точността (в смисъл на F-мярка) на разпознаване на ANNIE, OpenNLP [33] и LingPipe NER върху примерните медицински доклади. Резултатите са на Таблица 2.

<i>Характеристика</i>	<i>ANNIE</i>	<i>LingPipe</i>	<i>OpenNLP</i>
<i>True Positives</i>	2225	1064	65
<i>False Negatives</i>	1142	2301	3300
<i>False Positives</i>	3351	6960	364
<i>Precision</i>	0.40	0.13	0.15
<i>Recall</i>	0.66	0.32	0.02
<i>F₁ score</i>	0.50	0.19	0.03
<i>Време за изпълнение за 669 документа в секунди.</i>	331.95	527.71	524.30

Таблица 2

Експериментът показва, че ANNIE работи значително по-добре върху медицински доклади. Резултатите не са изненадващи защото алгоритмите, базирани на машинно самообучение, като OpenNLP и LingPipe, често дават добри резултати само на текстове от същия тип като този, на който са обучени. LingPipe прави няколко вида сериозни грешки, които намаляват точността на системата – например, маркира “She” като първо име на човек. Направен беше и опит да се комбинира ANNIE с LingPipe, но той също не доведе до по-добър резултат.

Анализът на ANNIE не може да се използва за медицинска анонимизация без корекции. Бяха нужни промени, аналогични на разгледаната по-горе разработка на Ishna Neamatullah et. al [25]. Бяха махнати от речниците с имена на компании и маркери за имена всички стандартни медицински термини. Например, без тази промяна, ANNIE винаги маркира „ADMISSION” като име на институция. Бяха направени и подобрения в ANNIE, които не са специално свързани с медицинските текстове. Правилата за разпознаване на имена на институции не отчитаха, че новите редове прекъсват контекстната връзка между някои текстови единици. На Фигура 6 е едно променено правило от ANNIE. (Промените са с **удебелен шрифт**.) То гласи, че ако име на обект от неизвестен тип (Unknown.kind = PN) е предшестван от име на място (Location) то тогава двойката е име на организация⁶. Например „Пирин Голф“ е име на фирма от този тип. След промяната, правилото изключва двойки от място и име, разделени от нов ред.

3.2 Разпознаване на възраст и дати

Разпознаване на споменавания на възраст е много подходящ проблем за решаване с правила, защото има малък брой типове изрази за възраст на човек, особено за възраст на пациент в медицински текст. Подобно на [22] бяха разгледани начините на изписване на възраст в документите на болницата и две правила на Фигура 7.

Разпознаването на дати е базирано на ANNIE с някои подобрения. В примерните медицинските доклади се ползват често кратки дати от типа 7/9, което отговаря на девети юли, защото документите са от американска болница. Такива дати не се маркират от ANNIE, вероятно защото не е бил намерен начин да се разграничат от числа, разделени с наклонена черта. За да се подобри откриването на дати беше добавено правило, показано на Фигура 8, което се възползва от инструментите, вградени в Java. Добавянето на това правило увеличи false positives поради двусмислието на поредици като 1/2 (втори януари или ½). Въпреки това, precision беше намален с по-малко от 1%,

⁶ В медицински документи, имената на организации са имена на болници или други здравни институции

а recall беше увеличен с около 2%. Като цяло F₂-мярката беше увеличена с 1%.

```
/*
 * Based on org_context.jape
 *
 * Copyright (c) 1998-2004, The University of Sheffield.
 */
Phase:    Med_Org_Context
Input:    Token Lookup Organization Unknown Location Person SpaceToken
Options:  control = appelt
Rule:LocOrg
Priority: 20
// guess that Unknown preceded by Loc is an Org
(
  {Location}
  {{SpaceToken.kind != control}}?
  {Unknown.kind == PN}
):org
-->
{
  ... дясната страна не е показана за краткост
}
```

Фигура 6

```

Phase: Age
Input: Person Token
Options: control = appelt debug = true

Rule: PersonAge
({Person}{Token.kind == "punctuation"})
({Token.kind == "number"})
:age
-->
:age.Age = {rule="PersonAge"}

Rule: AgeYearsOld
(
    {Token.kind == "number"}
    ({Token.kind == "punctuation"})?
    {Token.string ==~ "year.*"}
):age
-->
:age.Age = {rule="AgeYearsOld"}

```

Фигура 7

```

Phase:      Med_Date
Input: Token Date
Options: control = appelt

Rule:PartialDate
Priority: 20

(
    {Token.kind == "number"}
    {Token.string == "/" }
    {Token.kind == "number"}
):date
-->

```

Фигура 8 (продължава на следващата страница)


```

{
    AnnotationSet bindingDate = (AnnotationSet) bindings.get("date");
    Long start = bindingDate.firstNode().getOffset();
    Long end = bindingDate.lastNode().getOffset();
    String dateString;
    try {
        dateString = doc.getContent().getContent(start, end).toString();
    } catch (InvalidOffsetException e) {
        throw new RuntimeException(e);
    }
    boolean match = false;
    for (String format : new String[] { "MM/dd", "M/d" }) {
        try {
            java.text.DateFormat df = new
java.text.SimpleDateFormat(format);
            Date d = df.parse(dateString);
            if (dateString.equals(df.format(d))) {
                match = true;
                break;
            }
        } catch (java.text.ParseException e) {
            // not a date
        }
    }
    if (match) {
        gate.FeatureMap features = Factory.newFeatureMap();
        features.put("rule", "PartialDate");
        outputAS.add(bindingDate.firstNode(), bindingDate.lastNode(),
        "Date", features);
    }
}
}

```

Фигура 8 (продължение)

3.3 Разпознаване на телефони и други идентифициращи номера

С реализацията на разпознаването на телефонни номера и други идентификатори, алгоритъмът за откриване на лични данни по дефиницията от точка III.3 е завършен. Телефонните номера се маркират чрез на прост JAPE шаблон, показан на Фигура 9. Шаблонът игнорира интервалите, така че въпреки кратката си дефиниция, постига пълно покритие (recall) на различните видове изписване на номера.

```
Phase: Phones
Input: Token ID
Options: control = appelt

Rule: Phone
(
  ({Token.kind == "number", Token notWithin ID})
  ({Token.string == "-"}{Token.kind == "number"})[2,10]
):phone
-->
:phone.PHI = {type="PHONE", rule="Phone"}
```

Фигура 9

Разпознаването на идентификатори е базирано на наблюдението, че всеки идентификатор е низ от букви и цифри, които не образуват дума и се споменава в началото на текста на отделен ред. Описанието на това правило на езика JAPE е на Фигура 10. В остатъкът от текста, идентификаторът може да се споменава отново и затова се премахват всички споменавания на низове, които са разпознати като лични идентификатори.

```
Phase: OneLiners
Input: Token SpaceToken
Options: control = all

Macro: NL
({SpaceToken.kind == "control"})

Rule: OonlineID
(NL)
(
  ({Token.kind == "number", Token.length > 1})
  ({Token.kind == "punctuation"}{Token.kind == "number"})*
):id
(NL)
-->
: id.PHI = {type="ID", rule="OonlineID"}

Rule: OonlineHospital
(NL)
(
  {Token.orth == "allCaps", Token.length > 1, Token.length < 4}
):id
(NL)
-->
: id.PHI = {type="HOSPITAL", rule="OonlineHospital"}
```

Фигура 10

4 Резултати от оценката на точността на алгоритъма

Беше направена оценка на качеството на алгоритъма върху обучаващия корпус от i2b2 Medical De-Identification Challenge [10]. В него личните данни вече са маркирани с тагове PHI в XML форматирането. Метриките са същите като използваните по-горе в секция III.4 – precision, recall и F-measure.

<i>True Positives</i>	10978
<i>False Negatives</i>	3275
<i>False Positives</i>	8010
<i>True Negatives</i>	not relevant
<i>Precision</i>	0.58
<i>Recall</i>	0.77
<i>F₁ score</i>	0.66
<i>F₂ score</i>	0.72

5 Бъдещо развитие на приложението

Модулният дизайн на приложението оставя много възможности на бъдещо развитие. На първо място, анонимизацията може да се разшири с поддръжка на български и други езици.

За пълната поддръжка на нов език са нужни няколко компонента: разделител на текстови елементи (tokenizer), разделител на изречения, идентификатор на части на речта и компонент за откриване на именувани обекти. Всички компоненти, освен последния, са относително лесни за създаване на базата на съществуващата платформа в GATE 7.1. Разделител на текстови елементи за български език може да се направи само с конфигуриране на класа `gate.creole.tokeniser.DefaultTokeniser` [34]. Адаптирането към български е лесно, защото почти няма разлики в

разделителите между думи в английски и български (за разлика от езици като китайски и арабски). Аналогично, разделител на изречения може да се направи на базата на `gate.creole.splitter.RegexSentenceSplitter`. Беше проведен прост експеримент, който потвърди, че стандартното разделяне на изречения работи добре, дори без промяна, за медицински текст на български език⁷.

Идентификатор за части на речта на български език е задача, която е решена от екипа на д-р Кирил Симов в проекта BulTreeBank-DP [35]. Платформата GATE може да използва идентификатор, базиран на BulTreeBank чрез интеграцията си с LingPipe. За целта е нужно да се направи инстанция на Processing Resource от тип „LingPipe POS Tagger PR“ и да се зададе модел `bulgarian-full.model` [36]. Накрая, за разпознаване на именувани обекти на български език може да се използва решението на Георги Георгиев, Преслав Наков и Кузман Ганчев базирано на Conditional Random Fields [37].

След интегрирането на основните компонентите за анализ на естествен език, описани по-горе, за пълна поддръжка на медицински текстове на български език остава само да се обновят речниците с медицински понятия и ключови думи по методологията описана в [25]. Една посока за развитието на дипломата работа е автоматизиране на някои от тези методи. Като цяло, архитектурата на приложението позволява лесно надграждане на алгоритмите за анализ на текст.

Друга посока за развитие на приложението е подобряване на компонентният му интерфейс с цел улесняване интеграцията в медицински системи. Освен точките за интеграция, описани в точка V.2.4, приложението може да се адаптира към употреба като Уеб услуга (web service) и като OSGi услуга (service bundle). С тези добавки, софтуерът ще може да се включи като компонент във всички основни софтуерни архитектури.

⁷ За експеримента беше използвана листовка на лекарство Тайлол Хот от <http://apteka.framar.bg/>.

VI. Заключение

Анонимизацията на свободен медицински текст е проблем, който може да бъде решен с инструментите за обработка на естествен език. Решаването на тази задача освобождава огромно количество информация от затворените бази данни на болниците и я предоставя на университетите и научните институти, където тази информация ще бъде използвана за откриване нови начини за лечение и за подобряване на сегашните. Софтуерът, направен за тази дипломна работа, е още една възможност за автоматизиране на анонимизация, която се отличава с модулен дизайн и добри възможности за пренастройване към различни езици и видове текстове.

В първата и втората част на дипломната работа видяхме, че значителна част от медицинското знание се пази в доклади за протичане на заболяване и лечение на различни пациенти. Тези доклади биха били много полезни на начинаещи лекари. Текстовете също могат да бъдат анализирани с методи за насочено търсене на информация (data mining). Болниците обаче не са подготвени за такива дейности – информационните технологии не са фокус за никое лечебно заведение и затова данните се налага да стигнат до научните институти. Законът, обаче, с право не разрешава личната медицинска информация да се използва по този начин. Автоматичното разделяне на медицинското знание от личните данни е единственото решение, което едновременно защитава пациентите и дава възможност опита от тяхното лечение да бъде използван в бъдеще.

В третата част на дипломната работа бяха разгледани съществуващите решения за анонимизация, теорията зад тях и техните ограничения. Двата вида подходи – базирани на машинно самообучение и базирани на правила – имат своите предимства и недостатъци. При статистическите подходи със скрити Марковски модели и Conditional Random Fields се постига по-голяма точност с относително по-малко усилия, защото, при прилагането на тези методи върху нов вид текст, не се изисква да се допълнително програмира разпознаване на всички шаблони, специфични за новия вид. Тези методи обаче изискват значително по-голям ръчно маркиран тренировъчен корпус и са по-малко гъвкави, когато се налага да се направи малка промяна в

поведението им. Методите, базирани на правила, като този, който е предмет на дипломната работа, са по-трудоемки за реализация, но веднъж направени са по-лесни за фино настройване.

Накрая, самият предмет на дипломната работа е софтуер за анонимизация, базиран на GATE 7.1. Приложението се отличава с модулна архитектура, лесно конфигуриране чрез графичния интерфейс на GATE и добри възможности за интеграция като компонент в по-големи системи. Чрез възможностите за развитие, като добавяне на поддръжка на български език, софтуерът може да подобри медицинските системи и качеството на здравеопазването в България.

VII. Библиография

- [1] ЗАКОН ЗА ЗАЩИТА НА ЛИЧНИТЕ ДАННИ, Република България.
- [2] „Greek Medicine - The Hippocratic Oath,” [Онлайн]. Available: http://www.nlm.nih.gov/hmd/greek/greek_oath.html. [Отваряно на 2013].
- [3] The Congress of the United States, Congressional Budget Office, „Evidence on the Costs and Benefits of Health Information Technology,” May 2008. [Онлайн]. Available: <http://www.cbo.gov/sites/default/files/cbofiles/ftpdocs/91xx/doc9168/05-20-healthit.pdf>.
- [4] J. C. Prather, D. F. Lobach, L. K. Goodwin, J. W. Hales, M. L. Hage и W. E. Hammond, „Medical data mining: knowledge discovery in a clinical data warehouse.,” в *AMIA Annu Fall Symp.*, Durham, 1997.
- [5] U. Fayyad, G. Piatetsky-Shapiro и P. Smyth, „From data mining to knowledge discovery in databases,” *AI magazine*, том 17, № 3, p. 37, 1996.
- [6] M. Kattan, „A preoperative nomogram for disease recurrence following radical prostatectomy for prostate cancer.,” *Journal of the National Cancer Institute*, том 90, № 10, pp. 766-771, 1998.
- [7] NIST. U.S. Department of Commerce, „Guide to Protecting the Confidentiality of Personally Identifiable Information (PII),” Gaithersburg, MD, 2010.
- [8] „Unstructured data,” [Онлайн]. Available: http://en.wikipedia.org/wiki/Unstructured_data. [Отваряно на June 2013].
- [9] J. Berlin и M. Amihai, „Database Schema Matching Using Machine Learning with Feature Selection,” в *Proceedings of the 14th International Conference on Advanced Information Systems Engineering*, 2002.
- [10] O. Uzuner, Y. Luo и P. Szolovits, „Evaluating the state-of-the-art in automatic de-identification.,” *Journal of the American Medical Informatics Association*, том 5, № 14, 2007.
- [11] Wellner, „Rapidly retargetable approaches to de-identification in medical records.,” *Journal of the American Medical Informatics Association*, том 5, № 14, 2007.

- [12] R. Yeniterzi, J. Aberdeen, S. Bayer, B. Wellner, L. Hirschman и B. Malin, „Effects of personal identifier resynthesis on clinical text de-identification,“ *Journal of the American Medical Informatics Association*, том 17, № 2, pp. 159-168, 2010.
- [13] J. Aberdeen, „MIST documentation: Replacement, redaction and resynthesis,“ [Онлайн]. Available: http://mist-deid.sourceforge.net/current_docs/html/tasks/core/doc/general.html#Replacement_redaction_and_resynthesis. [Отваряно на May 2013].
- [14] „Java (software platform),“ [Онлайн]. Available: http://en.wikipedia.org/wiki/Java_%28software_platform%29.
- [15] D. Thaker, T. Osman и P. Lakin, „GATE JAPE Grammar Tutorial,“ [Онлайн]. Available: <http://gate.ac.uk/sale/thakker-jape-tutorial/GATE%20JAPE%20manual.pdf>.
- [16] V. Tablan, H. Cunningham, D. Maynard и K. Bontcheva, *Text Processing with GATE (Version 6)*, 2011.
- [17] V. Tablan, H. Cunningham, D. Maynard и K. Bontcheva, „GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications,“ в *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, 2002.
- [18] C. Trim, „The Art of Tokenization (IBM NLP DeveloperWorks),“ 23 January 2013. [Онлайн]. Available: <https://www.ibm.com/developerworks/community/blogs/nlp/entry/tokenization>. [Отваряно на 11 August 2013].
- [19] S. Tong и D. Koller, „Support Vector Machine Active Learning with Applications to Text Classification,“ *JOURNAL OF MACHINE LEARNING RESEARCH*, том 2, pp. 45-66, 2001.
- [20] Alias-I, „Javadoc of HmmChunker,“ [Онлайн]. Available: <http://alias-i.com/lingpipe/docs/api/com/aliasi/chunk/HmmChunker.html>. [Отваряно на 05 2013].
- [21] A. McCallum и D. Freitag, „Maximum entropy markov models for information extraction and segmentation,“ 2000.
- [22] „Principle of maximum entropy,“ [Онлайн]. Available: http://en.wikipedia.org/wiki/Principle_of_maximum_entropy. [Отваряно на May 2013].

- [23] „Entropy (information theory),“ [Онлайн]. Available: http://en.wikipedia.org/wiki/Entropy_%28information_theory%29. [Отваряно на May 2013].
- [24] E. Brill, „A simple rule-based part of speech tagger,“ в *Proceedings of the third conference on Applied natural language processing*, Trento, Italy, 1992.
- [25] I. Neamatullah, M. Douglass, L.-w. H Lehman и A. Reisner, „Automated De-Identification of Free-Text Medical Records,“ *BMC MEDICAL INFORMATICS AND DECISION MAKING*, том 8, № 32, 2008.
- [26] U.S. National Library of Medicine, Unified Medical Language System (UMLS), 2009.
- [27] „Wikipedia: Open Source Text Analysis Software,“ [Онлайн]. Available: http://en.wikipedia.org/wiki/Text_mining#Open_source. [Отваряно на June 2013].
- [28] D. Ferrucci и A. Lally, „UIMA: an architectural approach to unstructured information processing in the corporate research environment,“ *Natural Language Engineering*, том 10, № 3, pp. 327-348, 2004.
- [29] Alias-I Inc., „LingPipe 4.1.0,“ 2008. [Онлайн]. Available: <http://alias-i.com/lingpipe>.
- [30] H. Cunningham, „JAPE: a Java Annotation Patterns Engine,“ Department of Computer Science, University of Sheffield, 1999.
- [31] „OSGi framework,“ [Онлайн]. Available: <http://en.wikipedia.org/wiki/OSGi>. [Отваряно на June 2013].
- [32] R. Gordon, Essential JNI: Java Native Interface, Prentice-Hall, Inc., 1998.
- [33] J. Kottmann, „Apache OpenNLP,“ 2010. [Онлайн]. Available: <http://opennlp.apache.org/>. [Отваряно на June 2013].
- [34] H. Cunningham, „GATE ANNIE Tokenizer Configuration,“ [Онлайн]. Available: <http://gate.ac.uk/sale/tao/splitch6.html#sec:annie:tokeniser>. [Отваряно на 2013].
- [35] K. Simov, P. Osenova, A. Simov и M. Kouylevlov, „Design and implementation of the Bulgarian HPSG-based treebank,“ *Journal of*

Research on Language and Computation, том 2, № 4, p. 495–522, 2004.

- [36] K. Simov, P. Osenova и M. Slavcheva, „BulTreeBank morphosyntactic tagset,“ BulTreeBank Project, 2004.
- [37] G. Georgiev, P. Nakov, K. Ganchev, P. Osenova и K. Simov, „Feature-Rich Named Entity Recognition for Bulgarian Using Conditional Random Fields,“ в *Proceedings of the Conference on Recent Advances in Natural Language Processing (RANLP'2009)*, 2009.

VIII. Приложение 1

Схема на подреждането на анализиращи ресурси

