Софийски университет „Св. Климент Охридски“

Факултет по математика и информатика

Катедра „Математическа логика и приложенията ѝ“

ДИПЛОМНА РАБОТА

на тема

Автоматично обобщаване на текстове на български език

Димитър Христов Христов

Магистърска програма „Компютърна лингвистика“

Факултетен номер: М-24904

Ръководител:

проф. д-р Светла Пенева Коева

Институт за български език „Проф. Любомир Андрейчин“

София, 2017 г.

Съдържание

Резюме

Тази дипломна работа разглежда няколко различни подхода за автоматично обобщаване на единични текстове – задача, наложена от постоянно нарастващото количество на дигитална информация. Основно внимание е отделено на алгоритъма за екстрактно обобщаване LexRank и възможността за неговото прилагане към текстове на български език. Предложена е замяна на мярката за близост на изречения, използвана от алгоритъма, с такава, базирана на най-дълги общи подредици. Към системата са добавени и методи за предварителна лингвистична обработка. Алгоритъмът LexRank и предложените промени са имплементирани и тяхната работа е оценена автоматично с инструмента ROUGE. Резултатите от експеримента отхвърлят общия принос на предложените промени.

1 Въведение в задачата

Автоматичното обобщаване на документи има широко приложение в множество сфери – от създаване на резюмета на научни статии, обобщения на уеб страници, социални постове, новини и други до създаване на семантично описание на големи корпуси и продължаващи потоци от информация. В тази дипломна работа се фокусираме върху обобщаването на единични писмени документи на български език с оглед на операцията търсене в корпус.

1.1 Търсене на документи

Търсенето е една от стандартните операции в документи. Операцията се осъществява чрез превръщане на т. нар. „информационна нужда“ в заявка, спрямо която съответната система за търсене предлага набор от документи – кандидати, които съдържат търсената информация [1]. Често ограничения на формата на заявката означават невъзможност за точно съответствие на информационната нужда, при което е необходима допълнителна оценка на резултата. За тази цел системи за търсене (търсачки) като Гугъл и Бинг добавят към заглавието и адреса на резултата и контекста в документа, в който е открита заявката. В момента на писането на тази дипломна работа, тези системи представят като контекст отрязък, който съдържа заявката [2], [3].

Представените отрязъци имат за цел представяне на контекста, за да бъде допълнително оценен от търсещия. Често се случва отрязъкът да е само малка част от документ, който има друга тема. Поради това отрязъкът не представя добре съдържанието на предложения документ. Едно решение на този проблем е предоставянето на обобщение на документа към предложения резултат, което да осигури по-ясна представа за неговото съдържание.

Намираме се в ерата на информацията, в която количеството на дигитално записаните данни стремглаво нараства. Според доклада Дигитална вселена (EMC Digital Universe) от 2014 г. количеството на дигитално записаната информация през 2013 г. е била около 4,4 ZB[1], като е описана прогноза за експоненциално увеличение до 16 ZB през 2017 г. и 44 ZB през 2020 г. [4]. Заедно с цялата дигитална информация расте и Интернет – през 2016 г. Гугъл оценява размера на световната мрежа на 130 трилиона страници [5], като от тях до днес от Гугъл са индексирани 47.2 милиарда [6].

[1] 1 ZB (зетабайт) = 10^{12} GB (гигабайт) = 10^{21} B (байт)

Създаване на обобщение на всеки един от тези 47.2 милиарда възможни резултати е задача, която не може да бъде извършена ръчно. Именно тук се появява и компютърно-лингвистичния проблем за генериране на автоматични обобщения на единични документи.

1.2 Автоматично обобщаване на текстове

Автоматичното обобщаване на текст е операция, създаващ нов, съществено по-кратък текст от вече съществуващ, с цел точно предаване на основното съдържание на оригиналния текст. Задачата се разделя в няколко групи спрямо конкретната цел и методика. Може да бъде създадено обобщение на единичен документ (single document) или на множество документи (multi-document). Обобщаването на единични документи подпомага търсенето в корпуси чрез бързо оценяване на съдържанието на отделните документи. Обобщаването на множество документи представя общата информация представена от тези документи, като заедно с това може да филтрира информацията според нейната истинност (определена чрез честотата ѝ в документите) и централност (т.е. колко близка е до основната обща тема на обобщаваните документи). В тази работа разглеждаме обобщаване на единични документи.

Автоматичното обобщаване на текстове се разделя и според методиката, използвана за извличане на основната информация от документа и генериране на текста на обобщението. Създаване на обобщение чрез избор на отделни думи, фрази или цели изречения се нарича екстрактно обобщаване (extractive summarisation). При него текстът на обобщението е сглобен от отделни лексикални части на основния документ. Различна е идеологията на абстрактното обобщаване (abstractive summarisation), при което семантичната информация в документа е представена чрез междинна структура, от която се извличат основните за текста понятия и се създава напълно нов текст чрез текстов генератор.

1.3 Цели и задачи на разработката

В рамките на дипломната работа са поставени следните цели и задачи:

1. Описание на проблема за автоматично обобщаване на текст;
2. Преглед на съвременното състояние на науката;
3. Избор на алгоритъм за автоматично обобщаване на текстове на български език;
4. Предлагане на промени в избрания алгоритъм с цел подобряване на резултата;
5. Имплементация на избрания алгоритъм – създаване на система за автоматично обобщаване на текстове на български език;
6. Имплементация на предложените промени;
7. Създаване на корпус с ръчни обобщения с цел използването му за оценяване на системата на автоматично обобщаване на текстове на български език;
8. Подробен преглед на инструментът ROUGE [7] за автоматично оценяване на обобщения на текстове;
9. Оценяване чрез ROUGE на обобщенията, автоматично създадени от системата за автоматично обобщаване на текстове на български език, с оглед на предложените промени в избрания алгоритъм;
10. Анализ на получените оценки.

Дипломната работа цели да реши поставената задача, като се фокусира върху корпуси от новини и статии от Уикипедия – публицистични и енциклопедични текстове, чието основно съдържание е фактологична информация.

1.4 Структура на дипломната работа

В рамките на дипломната работа са разгледани проблемът за автоматично обобщаване на текст и възможни негови решения с основен фокус върху алгоритъма на Еркан и Радев LexRank [8]. Увод и описание на проблема за автоматично обобщаване на текстове са изложени в Глава 1. В Глава 2 са описани няколко алгоритъма, които имат за цел решаване на тази задача, използващи различни подходи. Тук е направена и мотивация за избора на LexRank с оглед на наличните ресурси и програми за обработка на български език и възможността за прилагане на разгледаните алгоритми за решаване на задачата за автоматично обобщаване на текстове на български език. Глава 3 представя алгоритъма LexRank в дълбочина. В Глава 4 са предложени промени с цел подобряване на качеството на получените от LexRank автоматични обобщения. Имплементацията на алгоритъма LexRank и създаването на система за автоматично обобщаване на текстове на български език са описани в Глава 5. В Глава 6 е представен подробен преглед и описание на начина на работа на инструмента за автоматично оценяване на обобщения ROUGE [7]. Глава 7 описва използваните за оценяване на работата на системата за автоматично обобщаване на текстове на български език корпуси с обобщения – Българския корпус с обобщения (БКО) и многоезиковите корпуси с обобщения от семинарите Мултилинг (MultiLing) от 2015 г. и 2017 г [9], [10]. Резултатите от автоматичното оценяване на обобщенията, получени чрез системата за автоматично обобщаване на текстове на български език, са представени и анализирани в Глава 8. На базата на тези изследвания е дадена обща оценка на възможността за автоматично обобщаване на единични текстове на български език, като също така са описани и възможности за продължаване на проучването в тази сфера, както и разширяване на обхвата на темата в Глава 9.

2 Съвременно състояние на науката

В следващите раздели са описани различни разработки за обобщаване на единични документи, разгледани при подготовката на дипломната работа. Обзорът е разделен в зависимост от метода, използван за извличане на основната информация от текста и създаването на текст на обобщение – екстрактни, абстрактни и абстрактно ориентирани методи. За всеки метод е представено обобщение на предимствата и проблемите, както и оценка на възможността за използването му за обобщаване на текстове на български език.

2.1 Екстрактно обобщаване

Широко разпространени са алгоритмите за екстрактно обобщаване на текстове, които използват широко-популярни статистически методи за определяне на относителната важност на изреченията и генериране на обобщения, базирани на тази информация. Тези алгоритми имат предимството, че са езиково независими – те не използват езиково специфични ресурси, което позволява прилагането им към текстове на български език. Основен недостатък на екстрактното обобщаване е възможната липса на свързаност между изреченията в обобщението поради неразрешени анафори и неанализирани семантични връзки.

Михалчеа и Тарау [11] адаптират алгоритъма PageRank [12], използван от системата за търсене на Гугъл за определяне на относителната важност на уеб страница, за извличане на информация от текстови документи в техния алгоритъм TextRank. Използването на PageRank като основа на алгоритъма за обобщаване на текстове на Михалчеа и Тарау е подбудено от приликата на

семантичните връзки между езиковите единици – термове[2], словосъчетания, прости изречения в състава на сложното, самостоятелни прости или сложни изречения и пр., с хипервръзките, свързващи страниците в Интернет и определящи уеб пространството като граф. Двата подхода са описани като системи за „гласуване“ между върховете в граф – лексикални единици или уеб страници, гласът на всеки връх се раделя между неговите съседи, а тежестта му се определя от относителната важност на върха. Михалчеа [13] предлага и имплементация, използваща алгоритъма HITS [14], който, подобно на PageRank, изчислява относителната важност на страници в уеб пространството на базата на препратките между тях. Адаптирането на алгоритъма HITS следва същата методика като това на PageRank, като описаните от Михалчеа резултати представят подобно качество на обобщенията генерирани при използването на PageRank и HITS. Предложената от Михалчеа мярка за близост е броят на общите термове, разделен на дължината на двете изречения. Бариос и съавт. [15] описват няколко вариации на мярката за близост на изречения, които могат да се използват в рамките на алгоритъма TextRank – най-дълга обща подредица (longest common subsequence, LCS), косинусово разстояние между векторно представяне на изреченията и BM25 – вероятностен модел, базиран на tf-idf [16], [17]. Резултатите от експеримента показват подобно качество при използването на различните мерки, с най-добро постижение при използването на BM25, но също и подобрения при използването на най-дълга обща подредица и косинусово разстояние.

Еркан и Радев [8] представят подобна адаптация на PageRank алгоритъма. В техния алгоритъм LexRank е заложена идеята за разглеждането на изреченията като „торби от думи“ (bag of words), в които редът на термовете не е от значение. Всяко изречение от текста е представено чрез вектор, измеренията на който определят честотата на срещане на различните термове и тяхната значимост, представени чрез общата оценка за термове tf-idf [16], [17]. Мярката, която Еркан и Радев използват за определяне на семантичната близост на изреченията и съответно наличието на ребро между върховете на тези изречения в графа е косинусово разстояние между техните векторни представяния.

Разработката на Парвийн и Щрубе [18] добавя към горните предложения мярка за локална свързаност на текста, чрез която да се осигури свързаността при екстрактно обобщаване. Те използват алгоритъма HITS в комбинация с граф на обекти (entity graph) – разширение на мрежата от обекти (entity grid) на Барзилай и Лапата [19]. Мрежата от обекти представя присъствието на определени обекти (предмети, абстрактни понятия, личности, организации и др.) в отделните изречения от текста, като отбелязва тяхната синтактична функция: подлог (subject – S), допълнение (object – O) или нито едното от двете (neither – X). За да бъде отбелязано правилно присъствието на обекти, е необходимо предварителното разрешаване на анафори. Разширена с двустранен граф, в който върховете, представящи изречения, са свързани с върховете, представящи съдържаните в изреченията обекти, структурата дава възможност за откриване на семантични връзки между изреченията според общите им обекти.

2.2 Абстрактно обобщаване

За разлика от екстрактното обобщаване, което създава обобщение от непроменени части на изходния текст, абстрактното обобщаване изисква семантично разбиране на текста и генериране на нов текст, съдържащ най-централната абстрактна информация. Различни разработки предлагат различно междинно представяне на семантичната информация от

[2] В дипломната работа терм се използва като еквивалентен термин на токън: последователност от символи между празни символи.

оригиналния текст, алгоритми за извличане на основните понятия и методики за генерирането на граматически правилен и свързан текст от това представяне.

Методът, описан от Хан и съавт. [20], е създаден с цел обобщаване на множество от документи, но е добър пример за разработка в сферата на абстрактното обобщаване на документи. Предложено е представяне на информацията като семантичен граф на понятия. Методът използва маркиране на семантични роли (semantic role labelling) [21] за извличане на предикатно-аргументни структури (predicate argument structures), които служат за семантично представяне на документа. За маркиране на семантични роли, както и за определянето на части на речта и откриването на именувани обекти (named entities), се използва инструментът SENNA [22]. На базата на тази информация се създават предикатно-аргументни структури, които представят семантичното съдържание на текста. Изчислява се семантичната близост на предикатно-аргументните структури чрез сравнение на съответните им елементи – сказуемо, подлог, допълнение, локация и др. В случаите, в които това е възможно, при сравнението на определени елементи от различни предикатно-аргументни структури се използват и връзките в WordNet [23], [24]. Предикатно-аргументните структури след това се разделят на групи (клъстеризират) според изчисленото подобие между тях. Това е изпълнено чрез събирателно йерархично клъстеризиране (agglomerative hierarchical clustering) [25]. Хан и съавт. използват множество от 10 текстови свойства на предикатно-аргументните структури [20, Раздел 2.5.1] и прилагат генетичен алгоритъм за определяне на оптималните коефициенти на тези 10 свойства. Спрямо избраните свойства, техните стойности и коефициенти се определя ранк на изреченията във всяка група. Спрямо този ранк се избират основните структури от групите. Избраните предикатно-аргументни структури се сливат на местата, където подлозите съвпадат, и на тази база се генерира абстрактното обобщение чрез инструмента SimpleNLG (simple natural language generator) [26].

Друг вид семантично представяне използват Лю и съавт. [27] – представяне на абстрактно значение (abstract meaning representation) [28]. Абстрактното представяне на значение се извършва под формата на семантично анотирани, лесно четими от човек графи, които са изчистени от синтактична информация. Абстрактното представяне на значение използва PropBank – анотиран корпус на семантични роли [29], [30], като това определя насочеността на алгоритъма изцяло към текстове на английски език. За представянето на документа в тази структура се използва семантичният парсер JAMR [31]. За всяко изречение се създава абстрактно представяне на значението. Общите понятия в отделните графи се сливат и всички корени се свързват с един общ нов централен връх, като това създава свързан граф, представящ семантично целия документ. От получения граф се избира подграф, който удовлетворява условията за съдържание на основна информация, истинност, краткост и възможност за създаване на коректен текст. Дохаре и съавт. [32] разширяват тази разработка, като също така използват и вече създадените инструменти за генериране на текст от представяне на абстрактно значение JAMR-Generator [33] и Neural AMR [34].

Разработките в сферата на абстрактното обобщаване са обещаващи, тъй като се доближават до идеята за създаване на семантично обобщение, създадено чрез генериране на нов текст. Това, от своя страна, доближава човешкото създаване на обобщения. Основен проблем в тази насока е голямата зависимост от езикови ресурси и инструменти – онтологии, семантични парсери, структури за семантично представяне на информацията, семантично анотирани корпуси, генератори на естествен език и др. Представените разработки използват инструменти като SENNA, SimpleNLG, PropBank, JAMR, JAMR-Generator и Neural AMR, които са налични за английски

език, но не за български, поради което не е възможно те да бъдат приложени към текстове на български език на този етап.

2.3 Абстрактно ориентирано обобщаване

Съществуват няколко разработки, които идеологически попадат между екстрактните и абстрактните подходи за обобщаване. Лорет и съавт. [35] наричат тези подходи абстрактно ориентирано обобщаване. Тези подходи използват екстрактно създадени обобщения, които впоследствие биват преструктурирани в свързан текст със същото съдържание.

Предложената система от Лорет и съавт. е разделена на две части изпълняващи тези две задачи – обобщаваща и преструктурираща. Екстрактното обобщение е създадено в няколко стъпки. Първо се извършва предварителна лингвистична обработка, включваща токънизация, разделяне на изречения, анотиране на части на речта и идентификация на стоп думи. След това се премахват повторенията – изречения, които повтарят информация, включена в други изречения, се идентифицират чрез инструмент за текстов извод (textual entailment). Метод за текстов извод, в който се използва лексикална и синтактична информация. е предложен от Ферандез и съавт. [36]. Идентифицират се темите, представени от термовете с голяма честота. Оценката на важността на изреченията е ръководена от принципа на кодовото количество (code quantity principle): оценката на всяко изречение се изчислява според броя и дължината на словосъчетания, съдържащи термове с голяма честота, като за целта са словосъчетания избрани словосъчетания на съществителните. Изреченията с най-висока оценка за важност се извличат и от тях се генерира екстрактно обобщение, като редът на изреченията е същият като в оригиналния документ.

Втората част на предложената от Лорет и съавт. система за автоматично обобщаване на текстове е преструктурирането на извлеченото екстрактно обобщение. Това се извършва чрез междинна структура – претеглен насочен граф от думи. В този граф върхове са термовете, срещащи се в екстрактното обобщение. Дъгите между тях са поставени според последователността на термовете в изреченията – дъга $a \rightarrow b$ между термовете a и b има тогава и само тогава, когато те се срещат един след друг в някое изречение. Теглото на дъгата се определя от обратната честота на едновременно срещане на термовете a и b в документа. От получения граф се извличат нови изречения, които са определени от най-късите пътища в графа. Тъй като тези изречения могат да бъдат граматически неправилни или недовършени, са възложени ограничения върху съдържанието и структурата на изреченията – те трябва да са дълги поне 3 терма, да съдържат глагол и да не завършват на частица, предлог, въпросителна дума или съюз. Генерираните по този начин изречения се сравняват с тези от екстрактното обобщение чрез изчисляване на косинусовото разстояние между векторните им представяния. От тях се избират тези, които са близки до някое от изреченията от екстрактното обобщение. Изреченията от екстрактното обобщение се заместват с ново изречение, ако съществува достатъчно близко такова. В обратен случай изреченията от екстрактното обобщение се използват директно. Абстрактно ориентираното обобщение се композира от приетите за подходящи извлечени и новосъздадени изречения.

Липсата на необходимост от специализирани инструменти за семантична обработка, каквито използват алгоритмите за абстрактно обобщаване, позволява абстрактно ориентираните методи да бъдат приложени за езици, за каквито такива ресурси липсват. Тъй като този подход всъщност е разширение на методите за екстрактно обобщаване, неговото приложение е включено в описанието на бъдещите разработки (Глава 9).

3 LexRank

Системата, която е създадена за целта на тази дипломна работа, е базирана на алгоритъма LexRank на Еркан и Радев [8]. LexRank представлява адаптация на идеята на алгоритъма PageRank [12], използван от Гугъл за оценяване на относителната важност на уеб страници. PageRank представя уеб пространството като граф, чийто върхове са уебстраници, а ребрата са определени от хипервръзки. LexRank също използва граф, като чрез него представя документ, поставяйки отделните изречения във върховете, а реброто между два върха е определено от семантичната близост на съответните изречения. Важността на дадено изречение се определя от важността на изреченията, с чиито върхове в графа е свързано, като следва метода за „гласуване“, аналогичен на „случайната разходка“ в PageRank.

За да може PageRank да работи с текстови данни, LexRank заменя идеята на PageRank за хипервръзки, определящи насочени ребра (дъби) между върховете в графа, с идеята за семантична връзка, определена от семантична близост, която няма насоченост. Тази липса на насоченост не променя работата на PageRank, тъй като подобен уеб граф може да съществува в случаите, когато за всяка хипервръзка има обратна такава.

3.1 Мярка за семантична близост

LexRank разглежда изреченията като торби от думи (bag of words). Използва се векторно представяне на изреченията с измерения, определени от множество от термове. Мярката на всяко измерение е броят на срещанията на определен терм в изречението, като се използва и коефициент за важност на съответния терм, в случая tf-isf. Това е модификация на tf-idf (term frequensy – inverted document frequency) [16], [17], като за корпус се смята самият документ, а за документи – изреченията.

Нека множеството от термове е $T = \{t_1, t_2, \dots, t_{|T|}\}$, където $|T|$ е размерът на множеството T, а $t_1, \dots, t_{|T|}$ са термовете. Нека $D = \{s_1, s_2, \dots, s_{|D|}\}$ е документът, а $s_1, \dots, s_{|D|}$ са изреченията в него.

Нека $\mathrm{tf}_s(t)$ е броят на срещанията на терма t в изречението s и

$$\mathrm{isf}(t) = \log\left(\frac{|D|}{|\{s \in D | t \in s\}|}\right) \tag{3.1}$$

Тогава изречението $s \in D$ ще бъде представено от вектора:

$$\vec{s} = \left[\mathrm{tf}_s(t_1) \times \mathrm{isf}(t_1) \quad \mathrm{tf}_s(t_2) \times \mathrm{isf}(t_2) \quad \dots \quad \mathrm{tf}_s\left(t_{|T|}\right) \times \mathrm{isf}\left(t_{|T|}\right)\right] \tag{3.2}$$

Като мярка за семантична близост LexRank използва косинуса на ъгъла между векторите на изреченията $x, y \in D$:

$$\mathrm{sim}(x, y) = \cos(\vec{x}, \vec{y}) = \frac{\sum_{i=1}^{|T|} \mathrm{tf}_x(t_i) \times \mathrm{tf}_y(t_i) \times \mathrm{isf}^2(t_i)}{\sqrt{\sum_{i=1}^{|T|} \mathrm{tf}_x^2(t_i) \times \mathrm{isf}^2(t_i)} \sqrt{\sum_{i=1}^{|T|} \mathrm{tf}_y^2(t_i) \times \mathrm{isf}^2(t_i)}} \tag{3.3}$$

Тъй като векторите винаги са положителни ($\mathrm{tf}_s(t)$ и $\mathrm{isf}(t)$ са положителни), то $\cos(\vec{x}, \vec{y}) \in [0; 1]$, където 0 означава крайно далечни изречения (ортогонални вектори), а 1 – крайно близки или съвпадащи.

3.2 Представяне на документа като граф

Еркан и Радев [8] разглеждат две възможни представяния като граф. Първото представяне е непретеглен граф, в който две изречения са определени като семантично близки или семантично далечни чрез наличието или отсъствието на ребро между съответните върхове. Това представяне е близко до дискретните връзки между страници в уеб пространството, с които работи PageRank. Второто представяне е чрез пълен претеглен граф, където всяко ребро е с тегло, равно на мярката за близост на изреченията във върховете на реброто – косинуса на ъгъла между двата вектора. Този вид граф е допълнение към алгоритъма PageRank, налагащо се поради характера на различните данни, с които LexRank работи.

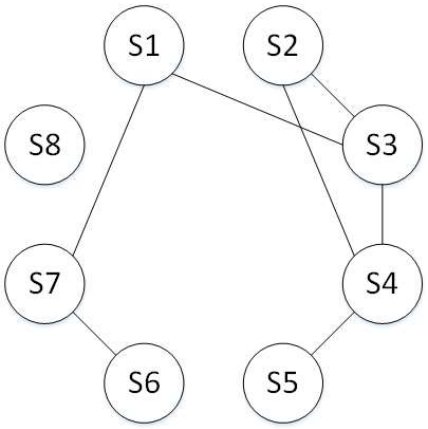
Тъй като избраната мярка за близост е реално число в интервала $[0; 1]$, е необходимо да бъде зададен праг θ, който определя наличието на ребро в непретегления граф. Таблица 3.1 и Таблица 3.2 съдържат примерни изречения от текст и оценката за семантична близост между всеки две от тях, изчислена чрез косинусово разстояние между векторните представяния на изреченията. Фигура 3.1 представя примерен непретеглен граф, съставен според оценките за семантична близост в Таблица 3.2 и праг $\theta = 0.05$.

ID	Изречение
s_1	*Адам Смит се счита за баща на икономикса главно заради неговата книга „Богатството на народите“.*
s_2	*Но той също е написал „Теория на моралните чувства“, която илюстрира моралните основи на жизнения и социално отговорен капитализъм.*
s_3	*Възгледите на Смит, от своя страна, могат да бъдат свързани с известното произведение на Макс Вебер „Протестантската етика и капитализма“, което проследява корелацията между стойностите, институциите и икономическото развитие.*
s_4	*Франсис Фукуяма, в „Краят на историята“, бе прав в основното - отхвърлянето на тоталитарния комунизъм затвори тази глава от историята.*
s_5	*Но болшинството от очакванията на Фукуяма бяха помрачени от събитията през последното десетилетие.*
s_6	*Напоследък въпросите за бизнес етиката и социалната отговорност между фирмите и обикновените хора излизат в центъра на обществения дебат.*
s_7	*Широко разпространената корупция и неетично поведение се разглеждат като присъщи за институционалната нестабилност и липсата на демократични права, характерни предимно за развиващите се страни.*
s_8	*Но множеството неотдавнашни корпоративни скандали зад океана и в Европа показват една по-сложна действителност.*

Таблица 3.1 Примерни изречения от текст на български език

Близост	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
s_1	**1.000**	0.043	**0.065**	0.042	0.001	0.023	**0.105**	0.000
s_2	0.043	**1.000**	**0.063**	**0.050**	0.022	0.010	0.016	0.025
s_3	**0.065**	**0.063**	**1.000**	**0.095**	0.036	0.043	0.039	0.008
s_4	0.042	**0.050**	**0.095**	**1.000**	**0.081**	0.036	0.016	0.039
s_5	0.001	0.022	0.036	**0.081**	**1.000**	0.001	0.000	0.027
s_6	0.023	0.010	0.043	0.036	0.001	**1.000**	**0.051**	0.034
s_7	**0.105**	0.016	0.039	0.016	0.000	**0.051**	**1.000**	0.009
s_8	0.000	0.025	0.008	0.039	0.027	0.034	0.009	**1.000**

Таблица 3.2 Оценки за близост между примерните изречения



Фигура 3.1 Граф, базиран на близост между изречения с $\theta = 0.05$

Второто представяне е чрез претеглен граф, който е пълен (има ребро между всеки два върха), като всяко ребро е с тегло близостта на изреченията, чиито върхове свързва. Така изречения, които са с близост 0, също ще бъдат свързани, но ребрата между тези върхове ще са с тегло 0. Запазването на тези ребра опростява формулите, без да променя смисъла и резултата им.

3.3 Определяне на важност

Едно изречение е вероятно важно, ако е семантично близко до важни изречения. Това е философията за определяне на важност на страница или изречение съответно в PageRank и LexRank. В графите, които са образувани, това означава техните върхове да са свързани.

Определянето на важността на изреченията в текста се извършва чрез „гласуване“, като за едно изречение „гласуват“ неговите съседни, а всеки „вот“ е с коефициент на важност – важността на „гласуващото“ изречение. PageRank и LexRank имплементират това „гласуване“ като разпределяне на важността на всяко изречение на неговите съседни:

$$p(u) = \sum_{v \in \mathrm{adj}(u)} \frac{p(v)}{\deg(v)} \tag{3.4}$$

Където $p(u)$ е важността на изречение u, $\mathrm{adj}(u)$ е множеството от близки на u изречения (съседни върхове в графа), а $\deg(v)$ – броят на близките му изречения (съседни върхове в графа).

При пълен претеглен граф всеки връх е свързан с всеки и разпределянето на гласовете на един връх е според близостта на изречението с останалите, използвана като коефициент за гласа:

$$p_{\mathrm{cont}}(u) = \sum_{v \in D} \frac{p_{\mathrm{cont}}(v) \times \mathrm{sim}(u,v)}{\sum_{w \in D} \mathrm{sim}(w,v)} \tag{3.5}$$

Където $p_{\mathrm{cont}}(u)$ е важността на изречение u, а $\mathrm{sim}(u,v)$ е близостта на изречения u и v. След като всяко изречение е близко поне на себе си, сумата в знаменателя е винаги положителна.

Тъй като при работата с мрежата в уеб пространството е възможно съществуването на ранкова яма (rank sink) – подграф, към който има само входящи, но не и изходящи дъги – PageRank добавя коефициент на амортизация $d \in [0,1]$, определящ вероятността потребител да премине директно на случайна уеб страница. Това премахва проблема с ранковата яма, давайки изход от нея. Въпреки че при работата на LexRank, където графът е ненасочен, такива ранкови ями не

съществуват, Еркан и Радев запазват коефициента на амортизация с оглед на възможната несвързаност на графа:

$$p(u) = \frac{d}{N} + (1-d) \sum_{v \in \text{adj}(u)} \frac{p(v)}{\deg(v)} \tag{3.6}$$

$$p_{\text{cont}}(u) = \frac{d}{N} + (1-d) \sum_{v \in D} \frac{p_{\text{cont}}(v) \times \text{sim}(u,v)}{\sum_{w \in D} \text{sim}(w,v)} \tag{3.7}$$

За изчислението на тези стойности се използва итеративен алгоритъм, който ги приближава, докато промяната на някоя стойност на важност $\delta_p \geq \varepsilon$, където ε е предварително зададена позволена грешка.

Алгоритъм за изчисляване на относителната важност на изречения при непретеглен граф:

```
for s ∈ D do
        p_0(s) ≔ 1/N
done
δ ≔ 1
t ≔ 0
while δ ≥ ε do
        δ ≔ 1
        t ≔ t + 1
        for u ∈ D do
                p_t(u) ≔ d/N + (1 − d) Σ_{v∈adj(u)} p_{t−1}(v)/deg(v)
                if δ < |p_t(u) − p_{t−1}(u)| then δ ≔ |p_t(u) − p_{t−1}(u)| endif
        done
done
```

Алгоритъм за изчисляване на относителната важност на изречения при пълен претеглен граф:

```
for s ∈ D do
        p_cont 0 ≔ 1/N
done
δ ≔ 1
t ≔ 0
while δ ≥ ε do
        δ ≔ 1
        t ≔ t + 1
        for u ∈ D
                p_cont t(u) ≔ d/N + (1 − d) Σ_{v∈V} p_cont t−1(v)×sim(u,v) / Σ_{w∈D} sim(w,v)
                if δ < |p_cont t(u) − p_cont t−1(u)| then δ ≔ |p_cont t(u) − p_cont t−1(u)| endif
        done
done
```

В резултат на горните алгоритми, $p_t(u)$ и $p_{\text{cont}\,t}(u)$ ще съдържат мярката за важност на изречението u с грешка не по-голяма от ε.

4 Промени на алгоритъма LexRank с цел подобряване на неговия резултат

Тази глава разглежда някои възможности за промени на алгоритъма LexRank с цел подобряване на неговия резултат. Включени са два типа промени – промяна на мярката за близост между две изречения и предварителна лингвистична обработка на текста.

4.1 Промяна на мярката за близост на изречения

Предлагат се две мерки за семантична близост на изречения. Тези мерки са алтернативни на мярката, изчисляваща близост на изречения чрез косинусово разстояние на векторни представяния на изреченията.

4.1.1 Най-дълга обща подредица

Изчислението на семантична близост между две изречения, базирано на дължината на най-дългата обща подредица (longest common subsequence или LCS) на термове на двете изречения, се използва в мярката за качество на автоматично създадено обобщение ROUGE-L [7], описана по-долу (Раздел 6.2). За тази мярка за близост на изречения дефинираме подредица u на изречението s, записано $u \subseteq s$, както следва (t, t_i и t_j са термове):

$$t \in u \Rightarrow t \in s \tag{4.1}$$

$$\forall t_i, t_j : t_i \in u \wedge t_j \in u \Rightarrow \left(t_i <_u t_j \Leftrightarrow t_i <_s t_j\right) \tag{4.2}$$

Където $t_i <_q t_j$ означава, че термът t_i предхожда терма t_j в редицата q. Такива редици са изреченията и подредиците в тях. Най-дълга обща подредица $u_{\max}$ на изреченията s_1 и s_2 е дефинирана така:

$$u_{\max} \subseteq s_1 \wedge u_{\max} \subseteq s_2 \tag{4.3}$$

$$\forall u : u \subseteq s_1 \wedge u \subseteq s_2 \Rightarrow |u| \leq |u_{\max}| \tag{4.4}$$

За крайното числово измерване за тази мярка за близост е избрано средно хармонично на отношението на дължината на най-дългата подредица $\mathrm{LCS}(x, y)$ към дължините на двете изречения $x, y \in D$, където D е обобщаваният документ, а именно:

$$\mathrm{sim}_{\mathrm{LCS}}(x, y) = H\left(\frac{|\mathrm{LCS}(x,y)|}{|x|}, \frac{|\mathrm{LCS}(x,y)|}{|y|}\right) = \left(\frac{\left(\frac{|\mathrm{LCS}(x,y)|}{|x|}\right)^{-1} + \left(\frac{|\mathrm{LCS}(x,y)|}{|y|}\right)^{-1}}{2}\right)^{-1}$$

$$= \frac{2}{\frac{|x|}{|\mathrm{LCS}(x,y)|} + \frac{|y|}{|\mathrm{LCS}(x,y)|}} = \frac{2 \times |\mathrm{LCS}(x,y)|}{|x| + |y|} \in [0; 1] \tag{4.5}$$

Където 0 е пълна разлика (x и y нямат общи термове), а 1 е пълно съвпадение.

4.1.2 Най-значима обща подредица

Второто предложение за промяна на мярката за близост на изречения в алгоритъма LexRank разширява идеята за най-дълга обща подредица на изреченията с допълнителна информация за важността на термовете в подредицата. Тази допълнителна информация е isf коефициентът на термовете от изречението, числово определяща тяхната важност. Така мярката за близост разглежда общата подредица с най-голяма сума на isf коефициентите на термовете в нея. Дефинираме най-значима обща подредица isf-$\mathrm{LCS}(x, y)$ на изреченията $x, y \in D$ като:

$$\text{isf-LCS}(x, y) = \underset{u:u\subseteq x \wedge u\subseteq y}{\operatorname{argmax}} |u|_{\text{isf}} \quad (4.6)$$

$$|u|_{\text{isf}} = \sum_{t\in u} \text{isf}(t) \quad (4.7)$$

Изчисляваме близостта на две изречения $x, y \in D$ спрямо така получената тяхна най-значима обща подредица подобно на 4.1.1 – като средно хармонично на isf-претеглената дължина на подредицата (нейната важност) към isf-претеглените дължини на двете изречения:

$$\text{sim}_{\text{isf-LCS}}(x, y) = H\left(\frac{|\text{isf-LCS}(x, y)|_{\text{isf}}}{|x|_{\text{isf}}}, \frac{|\text{isf-LCS}(x, y)|_{\text{isf}}}{|y|_{\text{isf}}}\right) = \frac{2 \times |\text{isf-LCS}(x, y)|_{\text{isf}}}{|x|_{\text{isf}} + |y|_{\text{isf}}} \in [0; 1] \quad (4.8)$$

Където 0 е пълна разлика (x и y нямат общи термове), а 1 е пълно съвпадение.

4.2 Предварителна лингвистична обработка на текста

Като част от възможните подобрения, предложени в тази дипломна работа, са няколко метода за предварителна лингвистична обработка на документите. Използваните инструменти са част от Българската многокомпонентна система за първична обработка и лингвистична анотация bglpc на Секцията по компютърна лингвистика към Института за български език към БАН[3] [37].

Различните методи за предварителна лингвистична обработка са използвани поотделно или заедно и с всяка алтернативна мярка за семантична близост на изречения.

4.2.1 Премахване на стоп думи

Стоп думите [38] представляват термове – частици, местоимения, съюзи, спомагателни глаголи, междуметия и други, които са често и сравнително равномерно срещащи се в документите (висок tf и нисък idf или isf) и се смята, че не носят смислова информация, необходима за да се разграничи съдържанието на един документ от друг. Тяхното премахване цели да намали шума, добавен от множество често срещащи се думи, грешно сигнализиращи за прилика между по-сложни изречения, при оценяване на семантичното съдържание на изреченията. Като част от системата за автоматично обобщаване на текстове на български език е използван списък със стоп думи, част от bglpc системата. Речникът съдържа 483 стоп думи, сред които са частици, съюзи, местоимения, числителни и други.

4.2.2 Филтриране по части на речта

Друга възможна предварителна лингвистична обработка на текста е определянето на частите на речта и филтрирането на термовете спрямо тях. За тази цел в системата ни за автоматично обобщаване е използван тагерът BgTagger от bglpc [37]. В имплементацията на системата под внимание за оценката на семантичната близост на изречения при използване на тагера се взимат съществителните имена и глаголите, следвайки предположението, че те са основните носители на съдържанието на текста.

4.2.3 Лематизация

Лематизацията представлява определянето на лемата – основната форма, на всеки терм. За лематизация е използван Българският лематизатор от bglpc системата. LexRank приема като различни всички термове, които имат разлика в записа си. Така термове, които има обща лема, но са в различна форма, са определени като семантично различни. След лематизация на текста, такива термове се смятат за един. Предположението, което ръководи добавянето на този метод

[3] http://dcl.bas.bg/webservices/

за предварителна обработка е, че лемата обединява формите на дадена дума и по този начин не се отчитат отделно различните форми, като това би трябвало да доведе до по-точно определяне на семантичната близост между изреченията, независимо коя мярка за близост се използва.

5 Имплементация на LexRank и предложените промени

В тази глава е представена имплементацията[4] на представеният в Глава 3 алгоритъм LexRank и предложените в Глава 4 промени.

5.1 Обща структура на системата за екстрактивно обобщаване на текстове

Работата със системата за екстрактивно обобщаване на текстове чрез алгоритъма LexRank, създадена в рамките на тази дипломна работа, е разделена на следните етапи на изпълнение:

1. Извличане на текстове и техните референтни обобщения от формата на документи, анотирани чрез ExtrSumAnnotator [39];
2. Разделяне на изречения и токъни, определяне на части на речта и лематизация на текстовете документи чрез bglpc [37];
3. Разделяне на изречения и токъни на референтните обобщения;
4. Създаване на конфигурационни файлове за екстрактно обобщаване на тестовите документи и оценяване на качеството на кандидатите за обобщения;
5. Изпълнение на алгоритъма LexRank в неговия основен вариант и с предложените промени;
6. Изпълнение на инструмента ROUGE [7], описан в Глава 6, за оценяване на качеството на кандидатите за обобщения.

5.2 Инструменти, езици и платформи, използвани за имплементация

Имплементацията на алгоритъма LexRank, както и на инструмента за извличане на текстове и референтни обобщения от формата на документи, анотирани с ExtrSumAnnotator, е осъществена на езика за програмиране C++ и неговия стандарт C++11 [40]. Допълнителните скриптове, създадени за подготовка на корпусите с обобщения БКО (Раздел 7.1) и Мултилинг [9], [10] и за работа със системата за екстрактно обобщаване на текстове, са написани на скриптовия език Bash [41]. Използван е редакторът за сорс код Visual Studio Code на Майкрософт [42]. За компилиране на имплементацията е използвана системата за компилация на софтуер CMake [43] и компилаторът gcc [44].

Имплементацията е създадена за изпълнение в операционната система Ubuntu на Canonical версия 16.04 [45]. Ubuntu е операционна система с отворен код, която използва ядрото Linux [46] и е базирана на операционната система Debian [47].

5.3 Имплементация на алгоритъма LexRank

Алгоритъмът LexRank е имплементиран на езика C++ и прилежащата му библиотека от стандартни шаблони (Standard Template Library) [40]. Имплементиран е клас LexRank със следната структура:

```
class LexRank
{
public:
    static const int CONTINUOUS = -1.0;
```

[4] https://github.com/deemhristov/Summarisation

```
    void init(const std::string & measure, const
            std::vector<std::vector<std::wstring>> & sents);
    std::vector<double> get_rank(double thres, double eps, double damp);

protected:
    std::vector<double> get_rank_cont(double eps, double damp);
    double cosine(int x, int y);
    double lcs(int x, int y);
    double isf_lcs(int x, int y);
    void calc_tf_isf();

private:
    std::vector<std::wstring> _keywords;
    std::vector<std::vector<std::wstring>> _sentences;
    std::vector<std::vector<int>> _sentkw;
    std::vector<double> _stfisf;
    std::vector<std::vector<int>> _kwtf;
    std::vector<double> _kwisf;
    std::vector<std::vector<double>> _similarity;
    std::vector<double> _sumsim;
};
```

Основните методи на класа LexRank са `init`, get_rank и get_rank_cont. Те изпълняват следните функции:

- `init` – инициализира обект от класа LexRank, като:
  - записва подадения списък от изречения в `_sentences`;
  - извлича списък с различните термове, срещащи се в изреченията, и ги записва в `_keywords`;
  - изчислява оценката за семантична близост на изречения спрямо зададения параметър `measure`;
  - изчислява векторните представяния на изреченията и ги записва в `_sentkw`;
- `get_rank` – изчислява относителната важност на изреченията в текста според алгоритъма за непретеглен граф, описан в Раздел 3.3;
- `get_rank_cont` – изчислява относителната важност на изреченията в текста според алгоритъма за пълен претеглен граф, описан в Раздел 3.3.

5.4 Имплементация на мерките за семантична близост на изречения

Методите `cosine`, `lcs` и `isf_lcs` изчисляват оценката за семантична близост между две дадени изречения x и y. Методът `cosine` имплементира мярката косинусово разстояние между векторните представяния на две изречения. Методът следва формула (3.3).

Методът `lcs` на класа LexRank е имплементиран чрез динамично програмиране – подход за решаване на оптимизационни и комбинаторни задачи, при който задачата се разделя на по-малки части, които се решават отделно и техният резултат се използва за изчислението на крайния. Дефинираме операцията $s - k$ за изречение s: $|s| \geq k$, както следва:

$$s = [t_1 \quad \dots \quad t_n]$$

$$s - k = [t_1 \quad \dots \quad t_{n-k}]$$

$$|s - k| = n - k = |s| - k$$

Тогава мярката $|\mathrm{LCS}(x,y)|$ за изречения $x = [t_1^x \quad \dots \quad t_{|x|}^x]$ и $y = [t_1^y \quad \dots \quad t_{|y|}^y]$ се изчислява по следния алгоритъм:

```
for i ≔ |x| … 0 do
        |LCS(x − i, y − |y|)| ≔ 0
done
for j ≔ |y| … 0 do
        |LCS(x − |x|, y − j)| ≔ 0
done
for i ≔ |x| − 1 … 0 do
        for j ≔ |y| − 1 … 0 do
                if t^x_{|x|−i} = t^y_{|y|−j} then
                        |LCS(x − i, y − j)| = max(|LCS(x − i − 1, y − j)|,
                                |LCS(x − i, y − j − 1)|,
                                |LCS(x − i − 1, y − j − 1)| + 1)
                else
                        |LCS(x − i, y − j)| = max(|LCS(x − i − 1, y − j)|,
                                |LCS(x − i, y − j − 1)|,
                                |LCS(x − i − 1, y − j − 1)|)
                endif
        done
done
```

Аналогично на $|\mathrm{LCS}(x,y)|$, мярката $|\text{isf-LCS}(x,y)|_{\text{isf}}$ за изречения $x = [t_1^x \quad \dots \quad t_{|x|}^x]$ и $y = [t_1^y \quad \dots \quad t_{|y|}^y]$ се изчислява по следния алгоритъм:

```
for i ≔ |x| … 0 do
        |isf-LCS(x − i, y − |y|)|isf ≔ 0
done
for j ≔ |y| … 0 do
        |isf-LCS(x − |x|, y − j)|isf ≔ 0
done
for i ≔ |x| − 1 … 0 do
        for j ≔ |y| − 1 … 0 do
                if t^x_{|x|−i} = t^y_{|y|−j} then
                        |isf-LCS(x − i, y − j)|isf = max(|isf-LCS(x − i − 1, y − j)|isf,
                                |isf-LCS(x − i, y − j − 1)|isf,
                                |isf-LCS(x − i − 1, y − j − 1)|isf + isf(t^x_{|x|−i}))
                else
                        |isf-LCS(x − i, y − j)|isf = max(|isf-LCS(x − i − 1, y − j)|isf,
                                |isf-LCS(x − i, y − j − 1)|isf,
                                |isf-LCS(x − i − 1, y − j − 1)|isf)
                endif
        done
done
```

6 Оценка на обобщения с ROUGE

Най-точна оценка за работата на система за автоматично обобщаване на текст може да бъде дадена от хора. Цената (като работа) на такова оценяване би била прекалено голяма, взимайки предвид броя на тестовите конфигурации, разглеждани в тази работа – 24 за всеки тест, общо

3240 (Глава 8). Това е причина за използване на инструмента за автоматизирано оценяване на обобщения ROUGE [7].

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) е инструмент за сравняване на два текста с цел оценка на семантичната им близост – идея, която е използвана и в мярката BLEU [48] за оценка на кандидати за превод. ROUGE предлага няколко варианта за сравнение на съдържанието на два текста, представени в следващите раздели.

6.1 ROUGE-N

ROUGE-N е мярка за семантична близост между автоматичното обобщение и референтното обобщение, базирана на броя на общите n-грами. N-грамът е редица от n на брой последователни термове, между които не присъстват други термове. Основата на мярката е покритие – каква част от n-грамите в референтното обобщение се съдържа в автоматичното. Изчислението на ROUGE-N е описано със следната формула:

$$\text{ROUGE-N} = \frac{c_{\text{match-}n}(S,R)}{c_n(R)} \tag{6.1}$$

Където S е кандидатът за обобщение, R е референтното обобщение, n е дължината на n-грамите, $c_n(R)$ е броят на n-грамите в обобщението R и $c_{\text{match-}n}(S,R)$ е броят на общите n-грами на обобщенията S и R. Тъй като ROUGE-N е базирана на покритие, за успешното ѝ използване се предполага, че размерът на обобщенията е ограничен.

Мярката ROUGE-N може да бъде сравнена с BLEU [48] – мярка, използвана за оценка на кандидати за превод на текст. BLEU изчислява верността на автоматичния превод, като базата на мярката е точност – каква част от кандидата присъства в съдържанието на референтния превод:

$$\text{BLEU-N} = \frac{c_{\text{match-}n}(S,R)}{c_n(S)} \tag{6.2}$$

Където S е кандидатът за превод и R е референтът. Тук се вижда и основната разлика между двете мерки и задачите – BLEU изчислява точността на превода, докато ROUGE-N изчислява покритието на обобщението.

6.2 ROUGE-L

Мярката за оценка ROUGE-L се основава на хипотезата, че два семантично близки текста ще имат по-дълга обща подредица от термове в сравнение с два семантично далечни текста [49], [50].

6.2.1 ROUGE-L за изречения

Дефинираме нерефлексивна антисиметрична транзитивна релация $<_s$ за последователност, като $x <_s y$ за $x, y \in s, x \neq y$ означава, че термът x се намира преди терма y в изречението s. Подредица u на изречението s, записано $u \subseteq s$, е дефинирана по следния начин (t, t_i и t_j са термове):

$$t \in u \Rightarrow t \in s \tag{6.3}$$

$$\forall t_i, t_j : t_i \in u \wedge t_j \in u \Rightarrow \left(t_i <_u t_j \Leftrightarrow t_i <_s t_j\right) \tag{6.4}$$

ROUGE-L разглежда най-дългата обща подредица между автоматичното обобщение и референтното обобщение. Най-дългата обща подредица $u_{\max}$ на изреченията s_1 и s_2 е дефинирана като:

$$u_{\max} \subseteq s_1 \wedge u_{\max} \subseteq s_2 \tag{6.5}$$
$$\forall u: u \subseteq s_1 \wedge u \subseteq s_2 \Rightarrow |u| \leq |u_{\max}| \tag{6.6}$$

Където $|u|$ е дължината на редицата u. Дефинира се и функция $\mathrm{LCS}(s_1, s_2) = u_{\max}$. Така се получава оценката ROUGE-L на кандидат изречение s спрямо референтно изречение r, представено като точност, покритие и мярка F:

$$P_{\text{L-sent}} = \frac{|\mathrm{LCS}(s,r)|}{|s|} \tag{6.7}$$

$$R_{\text{L-sent}} = \frac{|\mathrm{LCS}(s,r)|}{|r|} \tag{6.8}$$

$$F_{\text{L-sent}} = \frac{(1+\beta^2) \times P_{\text{L-sent}} \times R_{\text{L-sent}}}{\beta^2 \times P_{\text{L-sent}} + R_{\text{L-sent}}} \tag{6.9}$$

6.2.2 ROUGE-L за обобщения на текстове

Тъй като обобщенията съдържат едно или повече изречения и те не задължително съответстват между автоматичното и референтното обобщения, всяко от изреченията в автоматичното обобщение е оценено спрямо всички изречения в референтното обобщение. Оценката се базира на дължините на обединенията на най-дългите общи подредици за всяко референтно изречение [7, Раздел 3.2].

$$P_{\text{L}} = \frac{\sum_{r \in R} |\bigcup_{s \in S} \mathrm{LCS}(r,s)|}{|S|} \tag{6.10}$$

$$R_{\text{L}} = \frac{\sum_{r \in R} |\bigcup_{s \in S} \mathrm{LCS}(r,s)|}{|R|} \tag{6.11}$$

$$F_{\text{L}} = \frac{(1+\beta^2) \times P_{\text{L}} \times R_{\text{L}}}{\beta^2 \times P_{\text{L}} + R_{\text{L}}} \tag{6.12}$$

6.3 ROUGE-W

Оценките на кандидатите за обобщение при използване на мярката ROUGE-L зависят от дължините на най-дългите подредици, които са открити. Това не включва информация за разстоянието между термовете в тези подредици. Едно възможно допълнение към хипотезата, на която се базира мярката ROUGE-L, е това, че подредица от термове, които са последователни в едно изречение, е по-вероятно да бъде смислово носеща от такава, между термовете на която в оригиналното изречение има други термове. Допълнението към хипотезата е с цел ROUGE да се доближи възможно най-много до човешка оценка [7, Глава 4].

Разширената хипотеза е имплементирана като ROUGE-W – претеглен вариант на ROUGE-L, в който стойността на една подредица зависи от дължината ѝ и дължините на изцяло последователните (без дупки) подредици в нея.

6.3.1 Последователни редици

Нека вземем изречението s и една негова подредица u.Терминът последователна подредица q се дефинира така:

$$q \subseteq u \tag{6.13}$$
$$\forall t_i, t_j: t_i \in q \wedge t_j \in q \wedge t_i <_s t_j \wedge \left(\nexists t_k: t_k \in q \wedge t_i <_s t_k \wedge t_k <_s t_j\right)$$
$$\Rightarrow \nexists t_l: t_l \in s \wedge t_l \notin q \wedge t_j <_s t_l \wedge t_l <_s t_j \tag{6.14}$$

Тогава максимална последователна подредица $q_{\max}$ е последователна подредица, която не е подредица на по-дълга последователна подредица. Т. е. за дадена последователна подредица $q \subseteq u$ следва:

$$q_{\max} \subseteq q \Rightarrow q_{\max} = q \tag{6.15}$$

Нека $\mathrm{Q}(u)$ е множеството от всички максимални последователни подредици на u. Тогава:

$$\forall q_i, q_j : q_i \in \mathrm{Q}(u) \wedge q_j \in \mathrm{Q}(u) \wedge q_i \neq q_j \Rightarrow q_i \cap q_j = \emptyset \tag{6.16}$$

$$\bigcup_{q \in \mathrm{Q}(u)} q = u \tag{6.17}$$

6.3.2 Коефициент за тегло и ROUGE-W

Идеята на ROUGE-W е да даде по-висока оценка на по-дългите последователни подредици [7, Глава 4]. Затова е необходима претегляща функция $f(|q|)$, която да дава коефициент на последователната подредица q спрямо нейната дължина и за която е вярно:

$$f(|q_1|) + f(|q_2|) < f(|q_1| + |q_2|) \tag{6.18}$$

Така най-високата възможна оценка на дадена подредица ще бъде определена от максималните последователни подредици в нея. Тогава дефиницията на оценката ROUGE-W е:

$$P_{\mathrm{W}} = f^{-1}\left(\frac{\sum_{r \in R} \sum_{q \in \mathrm{Q}(\mathrm{U}_{s \in S} \mathrm{LCS}(r,s))} f(|q|)}{f(|S|)}\right) \tag{6.19}$$

$$R_{\mathrm{W}} = f^{-1}\left(\frac{\sum_{r \in R} \sum_{q \in \mathrm{Q}(\mathrm{U}_{s \in S} \mathrm{LCS}(r,s))} f(|q|)}{f(|R|)}\right) \tag{6.20}$$

$$F_{\mathrm{W}} = \frac{(1 + \beta^2) \times P_{\mathrm{W}} \times R_{\mathrm{W}}}{\beta^2 \times P_{\mathrm{W}} + R_{\mathrm{W}}} \tag{6.21}$$

Съществуват различни възможности за имплементацията на претеглящата функция, като предложената от Лин [7] е:

$$f(x) = x^a, a > 1 \tag{6.22}$$

Този вариант е използван в имплементацията на Perl инструмента ROUGE с коефициент $a = 1.2$.

6.4 ROUGE-S

ROUGE-S е мярка, подобна на ROUGE-2 (ROUGE-N с дължина на n-грамите 2, т. нар. биграми), но с една съществена разлика – ROUGE-S разглежда пропускащи биграми: наредени двойки от термове, които се срещат в изречението в същата последователност, но не задължително непосредствено един след друг. Отново подобно на ROUGE-N, ROUGE-S оценява отношението на броя на общите пропускащи биграми между кандидата за обобщение и референтното обобщение към броя на всички пропускащи биграми в кандидата и референта съответно за точност и покритие.

$$P_{\mathrm{S}} = \frac{c_{\text{match-S}}(S, R)}{c_{\mathrm{S}}(S)} \tag{6.23}$$

$$R_{\mathrm{S}} = \frac{c_{\text{match-S}}(S, R)}{c_{\mathrm{S}}(R)} \tag{6.24}$$

$$F_{\mathrm{S}} = \frac{(1+\beta^2) \times P_{\mathrm{S}} \times R_{\mathrm{S}}}{\beta^2 \times P_{\mathrm{S}} + R_{\mathrm{S}}} \tag{6.25}$$

Където $c_{\text{match-S}}(S,R)$ е броят на общите пропускащи биграми за кандидата за обобщение и референтното обобщение, а $c_{\mathrm{S}}(S)$ и $c_{\mathrm{S}}(R)$ са броят на всички пропускащи биграми в кандидата и референта съответно.

ROUGE-S предлага и ограничаване на разстоянието на термовете в изречението в един пропускащ биграм. Това е свързано с предположението, че термове на по-малко разстояние в едно изречение е по-вероятно да са семантично обвързани. При ограничаване на разстоянието до 0 ROUGE-S се превръща в F мярка на ROUGE-2 (ROUGE-N при $n = 2$), тъй като всички разглеждани пропускащи биграми ще бъдат от два непосредствено последователни терма.

6.4.1 ROUGE-SU

ROUGE-SU е разширение на ROUGE-S, което добавя информация за общите термове (или униграми), следвайки предположението, че общи термове определят обща тематика. ROUGE-S оценява присъствието на обща семантика спрямо наредени двойки от термове и затова е възможно в случай на разлики в словореда (възможно за български език), да даде ниска оценка, тъй като термове, които се срещат в кандидата и референта, са в различен ред. Изчислението на разширената мярка за оценка ROUGE-SU е както следва:

$$P_{\mathrm{SU}} = \frac{c_{\text{match-S}}(S,R) + c_{\text{match-1}}(S,R)}{c_{\mathrm{S}}(S) + |S|} \tag{6.26}$$

$$R_{\mathrm{SU}} = \frac{c_{\text{match-S}}(S,R) + c_{\text{match-1}}(S,R)}{c_{\mathrm{S}}(R) + |R|} \tag{6.27}$$

$$F_{\mathrm{SU}} = \frac{(1+\beta^2) \times P_{\mathrm{SU}} \times R_{\mathrm{SU}}}{\beta^2 \times P_{\mathrm{SU}} + R_{\mathrm{SU}}} \tag{6.28}$$

Лин [7] предлага имплементацията на ROUGE-SU чрез ROUGE-S като се добави маркер за начало на изречение. Така униграмите ще бъдат представени от пропускащи биграми, в които първият елемент ще е маркерът за начало на изречение. Тази имплементация е възможна при неограничено разстояние между термовете в един пропускащ биграм, но би дала грешен резултат при ограничаване на това разстояние – при ограничение n ще бъдат разгледани единствено първите $n+1$ терма, тъй като останалите са на по-голямо разстояние в изречението от маркера за начало на изречение.

6.5 Промени на Perl инструмента ROUGE за използване с български език

Последната версия на оригиналния Perl инструмент е ROUGE 1.5.5[5] от 2005 г. Инструментът, освен директно оценяване на представените му текстове, дава възможност и за предварителна обработка – стеминг (евристично определяне на основата или корена на даден терм) и премахване на стоп думи.

До тази версия включително инструментът е имплементиран основно за оценяване на резултати от автоматично обобщаване на текстове на английски език и допълнителната предварителна обработка не е подходяща за български език и съответно тази функционалност не е използвана. Също това ROUGE 1.5.5 поддържа единствено стандартната английска латиница. За да може да бъде използван за оценяване на кандидати за обобщение на български език, функционалността

[5] https://github.com/andersjo/pyrouge/tree/master/tools/ROUGE-1.5.5

на инструмента беше разширена с поддръжката на българска кирилица, създавайки ROUGE 1.5.6.

7 Корпус

За оценяването на разработката, направена за целите на дипломната работа, е необходим корпус от текстове на български език и техните референтни обобщения. Използвани са два такива корпуса – извадка от Българския национален корпус [51] с ръчно съставени обобщения (Раздел 7.1) и документите на български език и техните обобщения от корпусите за трениране Мултилинг за 2015 и 2017 г. [9], [10]

7.1 Български корпус с обобщения (БКО)

Българският национален корпус [51] съдържа над 240 000 текста с общо 1,2 млрд. думи, разпределени йерархично в 8 стила и общо 88 тематични области.

7.1.1 Корпус с обобщения

Дипломната работа е насочена основно към публицистични текстове и затова използваме текстовете от стила публицистика (C-MassMedia). За построяване на корпус с обобщения и оценяване на работата на разработената система са използвани текстове съдържащи поне 1000 и по-малко от 3000 думи. Извлечени са 93 публицистични текста, съдържащи с 1000-1999 думи (клас 1xxx), и 54 текста, съдържащи 2000-2999 думи (клас 2xxx), които допълнително са филтрирани[6], за да се отстранят документи, съдържащи повече от един текст, повтарящи се текстове и други неподходящи текстове, като в двата класа остават съответно 57 и 6 документа (Таблица 7.1).

Клас	**Брой думи**	**Общо документи**	**Филтрирани документи**	**Размер на обобщенията**
1xxx	1000-1999	93	57	L – 40% (400-800 думи) M – 20% (200-400 думи) S – 10% (100-200 думи)
2xxx	2000-2999	54	6	L – 20% (400-600 думи) M – 10% (200-300 думи) S – 5% (100-150 думи)

Таблица 7.1. Видове документи от БНК, използвани за създаване на корпус с обобщения

На базата на останалите 63 текста с новини, съдържащи 1000-2999 думи, е създаден ръчен корпус с обобщения (БКО). За всеки изходен текст са извлечени обобщения в три размера – 10%, 20% и 40% за документите от клас 1xxx и 5%, 10% и 20% за документите от клас 2xxx (Таблица 7.1), използвайки инструмента ExtrSumAnnotator от пакета с инструменти SummaryAnnotationTools, оригинално създадени за анотирането на Полския корпус на обобщения [39].

7.1.2 Анотиране с ExtrSumAnnotator

Анотаторът за обобщения ExtrSumAnnotator от пакета SummaryAnnotationTools [7] [39] представлява инструмент с графичен интерфейс, позволяващ създаването на екстрактно обобщение на документ чрез извличане на анотирани части от оригиналния текст. Интерфейсът

[6] Допълнителното филтриране и създаването на ръчните обобщения от БКО са извършени от Виктория Петрова, специализант към Секцията по компютърна лингвистика в периода 1.5.2017 г. – 31.8.2017 г.

[7] http://zil.ipipan.waw.pl/SummaryAnnotationTools

се състои от основен прозорец, който показва текста на документа и в който се анотира обобщението, текстово поле за обобщението, което представя извлечения текст, и информационен панел за обобщението. Информационният панел показва дължината на целия текст като брой думи, желаната дължина на обобщението като брой думи и моментната дължина на обобщението като брой думи и като част (в проценти) от желаната дължина на обобщението.

Извличането на обобщение се извършва чрез маркиране на текст в основния прозорец, като маркираният текст се смята за анотиран като част от обобщението. Инструментът предоставя възможност за извличане на обобщения в три размера, като за всяко има таб със същия интерфейс. Всяко следващо и по-малко обобщение се извлича от предишното (и по-голямо) обобщение – най-дългото от целия текст на документа, средното от най-дългото и най-късото от средното. Резултатът се записва в оригиналния файл на документа, като след текста се допълват обобщенията, представени като списъци от индекси на избраните символи. Това представяне налага допълнителна обработка за извличане на текста на обобщенията, като инструментът за това е допълнително имплементиран.

Направени са няколко модификации на ExtrSumAnnotator 1.0 (версията, която е предоставена в пакета SummaryAnnotationTools), записани като ExtrSumAnnotator 1.1, за да бъде улеснено неговото използване за целите на дипломната работа. Първата промяна е, че при стартиране, версия 1.1 позволява въвеждането на желаните размери на трите обобщения (като проценти), за разлика от оригиналната версия, в която размерите са фиксирани като 20%, 10% и 5%. Втората промяна е добавянето за запитване при опит за запазване на мястото на оригиналния файл, както и на запитване за запазване при изход ако има направени промени, за да се подсигури запазването на резултата на извършената работа.

7.2 Данни от Мултилинг

Заедно със създадения специално за дипломната работа корпус на обобщения, базиран на Българския национален корпус, за оценяването на работата на имплементацията на алгоритъма LexRank и неговите промени са използвани и предварително създадени тестови и тренировъчни данни за семинарите за автоматично обобщение Мултилинг, издания 2015 и 2017 [9], [10]. Корпусите съдържат статии от Уикипедия на 38 езика, за всеки от които за извлечени по 30 документа с ръчно създадени обобщения. Сред тези езици е и корпусът на български, който използваме за проверка на работата на системата за обобщаване на текстове на български език. Документите са с дължина между 1343 и 6909 думи. Размерът на обобщенията е зададен отделно за всеки документ, като стойностите варират между 129 и 308 думи.

8 Резултати

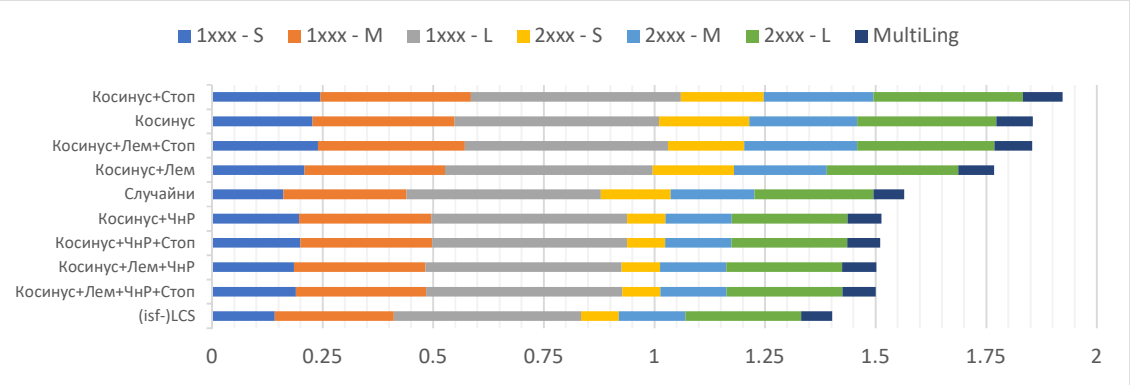
За оценяване на резултатите от работата на системата за екстрактно обобщаване, разработена за тази дипломна работа, използваме описания в Глава 6 инструмент ROUGE и представените в Глава 7 корпуси с обобщения. В следващите два раздела са представени оценките, които ROUGE дава на различните подходи и комбинации, и анализ на резултата от тестовете. Подробни резултати от оценката с инструмента ROUGE са представени в Приложение А.

8.1 ROUGE оценка

Обобщенията, чието качество е оценено с ROUGE, са извлечени със следните параметри: претеглен граф, $\varepsilon = 0.0001$ и $d = 0.15$. 8 различни оценки са изчислени – ROUGE-N за 1, 2, 3 и 4-грами (Разд. 6.1), ROUGE-L (Разд. 6.2), ROUGE-W с $f(x) = x^{1.2}$ (Разд. 6.3), ROUGE-S и ROUGE-SU

без ограничение на разстоянието в пропускащите биграми (Разд. 6.4). Крайната оценка е изчислена като средно геометрично на различните ROUGE оценки. Кандидатите за обобщение и референтните обобщения са подадени на ROUGE токънизирани, по едно изречение на ред. Резултатите от оценяването с ROUGE са представени таблично в Приложение А, като са представени 7 тестови групи – групи 1 - 6 върху данните от БКО (2 класа – 1xxx и 2xxx, и три размера за всеки клас – S, M и L, представени в Таблица 7.1) и група 7 върху данните от семинарите Мултилинг.

Заедно с различните конфигурации на алгоритъма LexRank и предложените промени са представени и оценките от ROUGE на примерни обобщения, създадени чрез случаен избор на изречения. Тези оценки сформират базовото ниво за качество на обобщение, което трябва да бъде надминато при изпълнение на системата за автоматично обобщаване, за да бъде признат подходът за успешен. Различните конфигурации са подредени спрямо крайната оценка, от най-висока до най-ниска. Фигура 8.1 представя общата оценка на отделните конфигурации като сбор от крайните оценки в различните тестови групи. Тези оценки са средно аритметичното на оценките на всички обобщения в дадена тестова група за определена конфигурация с доверителен интервал 5%-95%, т. е. изключени са най-ниските и най-високите 5% от оценките.



Фигура 8.1 Обща оценка на различните конфигурации на системата за автоматично обобщаване на текстове на български език

8.2 Анализ на резултатите

От получените оценки от ROUGE ясно се забелязва тенденция за влошаване на резултата на системата при използване на методи за предварителна лингвистична обработка за мярката косинусово разстояние (Фигура 8.1). Забележимо подобрение на резултата се наблюдава единствено при премахването на стоп думи от текстовете в случаите, в които термовете в изреченията не са филтрирани спрямо части на речта. Лематизирането не представя забележима промяна в резултата, докато работата само със съществителни имена и глаголи довеждат до по-лоши резултати, като в тестови групи 4, 5 и 6 (БКО, клас 2xxx, Таблица А.4, Таблица А.5, Таблица А.6) оценките са под базовото ниво. Това предполага, че за определяне на близост между две изречения е необходима повече информация за структурата на изречението и контекста, описани чрез останалите части на речта.

Резултатите също така показват и лоша работа на алгоритъма с алтернативните мерки LCS и isf-LCS. И при двата корпуса тези мерки генерират по-лоши обобщения от всички комбинации на косинусово разстояние между вектори с методи за предварителна обработка на текст. Изводът, който може да бъде направен, е за важното влияние на по-свободния словоред на българския език, който контрастира с по-строгия словоред на английския език [52]. Мерките, търсещи най-

дълги или значими подредици оценяват изреченията „Кучето излезе навън.“ и „Навън излезе кучето.“ като много различни, когато всъщност те са до голяма степен еднозначни. Тази еднозначност обаче е открита при използване на косинусово разстояние, тъй като векторното представяне на двете изречения е напълно еднакво.

Близките оценки от ROUGE за LCS и isf-LCS във всички комбинации с методи за предварителна обработка на текст показват независимостта на двете мерки от подобни характеристики на текста. Премахването на стоп думи и филтрирането на части на речта оставят относително същата дължина на най-дългата обща подредица на две изречения в сравнение с дължина на изреченията. Оценките също така показват, че в случаите с кратки документи, съдържащи неголям брой изречения, лематизацията също не променя измерената близост между изреченията. Възможно е тази обработка да има по-голям принос при обобщението на множество от документи, където може да има повече подобни, перифразирани изречения, или на документи с по-голям размер. Двете мерки, също така, получават ROUGE оценки по-ниски от базовото ниво при БКО, но при данните от семинарите Мултилиг, оценките им са по-високи. Това показва ниската ефективност на LCS и isf-LCS при кратки текстове (БКО, до 3000 думи), за разлика от по-дълги (Мултилинг, до 7000, Таблица А.7). Също така БКО съдържа текстове, които са относително дълги за новини и макар че са класифицирани като такива, често представляват анализи на събития и имат специфична структура в сравнение с новините. Всичко това потвърждава, че автоматичното обобщаване зависи от конкретния език, дължината и структурата на документите.

9 Заключение

Автоматичното обобщаване е сериозна задача, резултат на нарастването на количеството информация в електронна форма и в Интернет. Автоматичното обобщаване на текстове на български език е задача, за която съществуват малко разработки и малко ресурси. В тази дипломна работа са разгледани няколко алгоритъма за автоматично обобщаване на текстове, следващи различни подходи – екстрактен (извличане на непроменени части от текста и сглобяване на обобщение от тях), абстрактен (извличане и представяне на семантичната информация в текста чрез междинна структура и генерирането на изцяло нов текст на обобщение) и абстрактно ориентиран (генериране на екстрактно обобщение, съдържащо основната информация от текста, и автоматичното му преструктуриране, базирано на синтактично-лексикални методи). Избран е алгоритъмът LexRank, който е езиково независим и може да бъде използван за автоматично обобщаване на текстове на български език.

С цел допълнително подобряване на работата на алгоритъма LexRank са проучени възможни промени на мярката за оценяване на близост на изречения с мерки, използващи непретеглени и претеглени най-дълги общи подредици, различни методи за предварителна лингвистична обработка на текста – премахване на стоп думи, филтриране по части на речта и лематизация, както и комбинация на всяка от мерките за семантична близост на изречения с различните видове предварителна лингвистична обработка.

Алгоритъмът е имплементиран без използването на готови модули, с изключение на стандартните библиотеки. Допълнително са проучени и адаптирани за работа с наличните ресурси инструментът ROUGE за оценяване на автоматични обобщения и пакетът от инструменти за анотиране на обобщения SummaryAnnotationTools. Създаден за целите на дипломната работа и бъдещи разработки е Българският корпус с обобщения (БКО).

Резултатите от проведения експеримент, изведени като автоматични оценки на кандидатите за обобщения, създадени с имплементираната система и оценени с ROUGE, показват, че алгоритъмът LexRank е подходящ за създаване на обобщения на единични документи на български език. Предложената предварителна лингвистична обработка не подпомага неговата работа при кратки единични текстове. Също така, мерките за близост на изречения, базирани на общи подредици от термове, са неподходящи за български език, поради променливия му словоред – свойство, от което оригиналната мярка, косинусово разстояние между вектори, не зависи.

От интерес е продължаващото проучване на мерки за семантична близост на изречения, които биха били по-подходящи за особеностите на синтаксиса на българския език. Възможни такива мерки са ROUGE-S и ROUGE-SU, които използват последователността на термовете, подсигурявайки строг словоред на фразите, като допускат свобода на словореда на по-високо ниво в изречението. Също така, бъдещи експерименти включват и различни методи за отстраняване на семантична многозначност и разрешаване на анафорите. Възможно разширение са и методите на абстрактно ориентирано обобщаване, даващи възможност за създаване на свързано и последователно обобщение без необходимост от специализирани инструменти за семантичен анализ и семантично преобразуване. При реализация на бъдещи разработки за автоматично обобщаване на единични български текстове може да се използва имплементацията на алгоритъма LexRank в рамките на тази дипломна работа, референтния корпус с обобщения на български новини БНК и резултатите от оценяването на работата на алгоритъма LexRank и неговите модификации.

Библиография

[1] A. Broder, „A Taxonomy of Web Search“, *SIGIR Forum*, том 36, бр 2, с-ци 3–10, Сеп 2002.

[2] Google, „Create good titles and snippets in Search Results“. [Онлайн]. Available at: https://support.google.com/webmasters/answer/35624?hl=en. [Отворен на: 11-Ное-2017].

[3] Microsoft, „How Does Bing Choose The Title For My Web Page?“ [Онлайн]. Available at: https://blogs.bing.com/webmaster/2014/06/23/how-does-bing-choose-the-title-for-my-web-page/. [Отворен на: 11-Ное-2017].

[4] IDC, „EMC Digital Universe Infobrief“, 2014.

[5] Laurie Sullivan, „Google Estimates More Than 130 Trillion Web Pages“, *MediaPost*, 14-Ное-2016. .

[6] Maurice de Kunder, „WorldWideWebSize.com“. [Онлайн]. Available at: http://www.worldwidewebsize.com/. [Отворен на: 10-Ное-2017].

[7] C.-Y. Lin, „ROUGE: A Package for Automatic Evaluation of Summaries“, в *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, Barcelona, Spain, 2004, с-ци 74–81.

[8] G. Erkan и D. R. Radev, „LexRank: Graph-based Lexical Centrality As Salience in Text Summarization“, *J. Artif. Int. Res.*, том 22, бр 1, с-ци 457–479, Дек 2004.

[9] G. Giannakopoulos *и съавт.*, „MultiLing 2015: Multilingual Summarization of Single and Multi-Documents, On-line Fora, and Call-center Conversations“, в *Proceedings of the SIGDIAL 2015 Conference*, Prague, Czech Republic, 2015, с-ци 270–274.

[10] G. Giannakopoulos *и съавт.*, „MultiLing 2017 Overview“, в *MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres (MultiLing 2017)*, Valencia, Spain, 2017, с-ци 1–6.

[11] R. Mihalcea и P. Tarau, „TextRank: Bringing Order into Texts“, в *Proceedings of EMNLP 2004*, Barcelona, Spain, 2004, с-ци 404–411.

[12] L. Page, S. Brin, R. Motwani, и T. Winograd, „The PageRank citation ranking: Bringing order to the Web“, в *Proceedings of the 7th International World Wide Web Conference*, Brisbane, Australia, 1998, с-ци 161–172.
[13] R. Mihalcea, „Language Independent Extractive Summarization“, в *Proceedings of the ACL 2005 on Interactive Poster and Demonstration Sessions*, Stroudsburg, PA, USA, 2005, с-ци 49–52.
[14] J. M. Kleinberg, „Authoritative Sources in a Hyperlinked Environment“, *J. ACM*, том 46, бр 5, с-ци 604–632, Сеп 1999.
[15] F. Barrios, F. López, L. Argerich, и R. Wachenchauzer, „Variations of the Similarity Function of TextRank for Automated Summarization“, *arXiv:1602.03606 [cs]*, Фев 2016.
[16] H. P. Luhn, „A Statistical Approach to Mechanized Encoding and Searching of Literary Information“, *IBM J. Res. Dev.*, том 1, бр 4, с-ци 309–317, Окт 1957.
[17] K. S. Jones, „A Statistical Interpretation of Term Specificity and Its Application in Retrieval“, *Journal of Documentation*, том 28, бр 1, с-ци 11–21, 1972.
[18] D. Parveen и M. Strube, „Integrating Importance, Non-redundancy and Coherence in Graph-based Extractive Summarization“, в *Proceedings of the 24th International Conference on Artificial Intelligence*, Buenos Aires, Argentina, 2015, с-ци 1298–1304.
[19] R. Barzilay и M. Lapata, „Modeling Local Coherence: An Entity-based Approach“, в *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, Stroudsburg, PA, USA, 2005, с-ци 141–148.
[20] A. Khan, N. Salim, и Y. Jaya Kumar, „A Framework for Multi-document Abstractive Summarization Based on Semantic Role Labelling“, *Appl. Soft Comput.*, том 30, бр C, с-ци 737–747, Май 2015.
[21] L. Suanmali, N. Salim, и M. S. Binwahlan, „SRL-GSM: A Hybrid Approach based on Semantic Role Labeling and General Statistic Method for Text Summarization“, *Journal of Applied Sciences*, том 10, бр 3, с-ци 166–173, Мар 2010.
[22] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, и P. Kuksa, „Natural Language Processing (almost) from Scratch“, *arXiv:1103.0398 [cs]*, Мар 2011.
[23] G. A. Miller, „WordNet: A Lexical Database for English“, *Commun. ACM*, том 38, бр 11, с-ци 39–41, Ное 1995.
[24] C. Fellbaum, *WordNet: An Electronic Lexical Database*. MIT Press, 1998.
[25] F. Murtagh и P. Contreras, „Methods of Hierarchical Clustering“, *arXiv:1105.0121 [cs, math, stat]*, Апр 2011.
[26] A. Gatt и E. Reiter, „SimpleNLG: A Realisation Engine for Practical Applications“, в *Proceedings of the 12th European Workshop on Natural Language Generation*, Stroudsburg, PA, USA, 2009, с-ци 90–93.
[27] F. Liu, J. Flanigan, S. Thomson, N. M. Sadeh, и N. A. Smith, „Toward Abstractive Summarization Using Semantic Representations.“, в *HLT-NAACL*, 2015, с-ци 1077–1086.
[28] L. Banarescu *и съавт.*, „Abstract Meaning Representation for Sembanking“, *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, с-ци 178–186, 2013.
[29] P. Kingsbury и M. S. Palmer, „From TreeBank to PropBank“, 2002.
[30] M. Palmer, D. Gildea, и P. Kingsbury, „The Proposition Bank: An Annotated Corpus of Semantic Roles“, *Comput. Linguist.*, том 31, бр 1, с-ци 71–106, Мар 2005.
[31] J. Flanigan, S. Thomson, J. G. Carbonell, C. Dyer, и N. A. Smith, „A discriminative graph-based parser for the abstract meaning representation“, 2014.
[32] S. Dohare, H. Karnick, и V. Gupta, „Text Summarization using Abstract Meaning Representation“, *arXiv:1706.01678 [cs]*, Юни 2017.
[33] J. Flanigan, C. Dyer, N. A. Smith, и C. de J. Carbonell, „Generation from Abstract Meaning Representation using Tree Transducers“, в *HLT-NAACL*, 2016.
[34] I. Konstas, S. Iyer, M. Yatskar, Y. Choi, и L. Zettlemoyer, „Neural AMR: Sequence-to-Sequence Models for Parsing and Generation“, *arXiv:1704.08381 [cs]*, Апр 2017.

[35] E. Lloret, M. T. Romá-Ferri, и M. Palomar, „COMPENDIUM: A text summarization system for generating abstracts of research papers“, *Data & Knowledge Engineering*, том 88, бр Supplement C, с-ци 164–175, Ное 2013.
[36] Ó. Ferrández, D. Micol, R. Muñoz, и M. Palomar, „A Perspective-based Approach for Solving Textual Entailment Recognition“, в *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, Stroudsburg, PA, USA, 2007, с-ци 66–71.
[37] S. Koeva и A. Genov, „Bulgarian Language Processing Chain“, в *Proceedings of Integration of Multilingual Resources and Tools in Web Applications. Workshop in conjunction with GSCL*, 2011, том 26.
[38] H. P. Luhn, „Keyword-in-Context Index for Technical Literature (KWIC Index)“, *American Documentation*, том 11, бр 4, с-ци 288–295, 1960.
[39] M. Ogrodniczuk и M. Kopeć, „The Polish Summaries Corpus“, в *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, 2014.
[40] ISO/IEC, „14882:2011: Information technology – Programming languages – C++“.
[41] Free Software Foundation, *GNU Bash*. 2013.
[42] Microsoft, „Visual Studio Code - Code Editing. Redefined“. [Онлайн]. Available at: http://code.visualstudio.com/. [Отворен на: 23-Ное-2017].
[43] Kitware Inc., *CMake*. 2012.
[44] Free Software Foundation, *GNU GCC*. 2015.
[45] Canonical, *Ubuntu*. 2016.
[46] L. Torvalds, *Linux*. 2014.
[47] Software in the Public Interest, *Debian*. 2017.
[48] K. Papineni, S. Roukos, T. Ward, и W.-J. Zhu, „BLEU: A Method for Automatic Evaluation of Machine Translation“, в *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, Stroudsburg, PA, USA, 2002, с-ци 311–318.
[49] I. D. Melamed, „Automatic Evaluation and Uniform Filter Cascades for Inducing N-Best Translation Lexicons“, *CoRR*, том cmp-lg/9505044, 1995.
[50] H. Saggion, S. Teufel, D. Radev, и W. Lam, „Meta-evaluation of Summaries in a Cross-lingual Environment Using Content-based Metrics“, в *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, Stroudsburg, PA, USA, 2002, с-ци 1–7.
[51] S. Koeva, I. Stoyanova, S. Leseva, T. Dimitrova, R. Dekova, и E. Tarpomanova, „The Bulgarian National Corpus: Theory and Practice in Corpus Design“, *Journal of Language Modelling*, том 0, бр 1, с-ци 65–110, 2012.
[52] Е. Георгиева, *Словоред на простото изречение в българския книжовен език*. Българска Академия на Науките, 1974.

Приложение А

Легенда на съкращенията: Лем – лематизация; ЧнР – филтриране по части на речта; Стоп – премахване на стоп думи; R – ROUGE; Ср. г. – Средно геометрично на оценките от ROUGE

Мярка	**Лем**	**ЧнР**	**Стоп**	**R-1**	**R-2**	**R-3**	**R-4**	**R-L**	**R-W**	**R-S**	**R-SU**	**Ср. г.**
Косинус			X	0.40316	0.20375	0.17088	0.15688	0.3759	0.17745	0.15575	0.15912	*0.245809*
Косинус	X		X	0.39078	0.19267	0.16583	0.15521	0.36709	0.17732	0.14842	0.15173	*0.239128*
Косинус				0.39481	0.17647	0.14709	0.13582	0.3561	0.16849	0.14439	0.1475	*0.227005*
Косинус	X			0.37543	0.15755	0.12683	0.11537	0.34052	0.15764	0.13289	0.13604	*0.208848*
Косинус		X	X	0.35849	0.1429	0.11887	0.10885	0.33828	0.15567	0.12492	0.12823	*0.199616*
Косинус		X		0.36019	0.14109	0.1146	0.10372	0.33464	0.15314	0.12418	0.12743	*0.196703*
Косинус	X	X	X	0.34693	0.13529	0.11055	0.10139	0.325	0.14887	0.11683	0.12	*0.1899*
Косинус	X	X		0.34728	0.12892	0.10247	0.09252	0.32344	0.14607	0.11657	0.11979	*0.184853*
Случайни	**?**	**?**	**?**	**0.32484**	**0.10532**	**0.0796**	**0.07101**	**0.30338**	**0.135**	**0.09997**	**0.1031**	***0.161193***
(isf-)LCS	?	?	?	0.30704	0.08317	0.06221	0.05511	0.28796	0.12555	0.09017	0.0933	*0.142327*

Таблица А.1 Тестова група 1 - БКО, клас 1xxx, размер на обобщението S (10%)

Мярка	**Лем**	**ЧнР**	**Стоп**	**R-1**	**R-2**	**R-3**	**R-4**	**R-L**	**R-W**	**R-S**	**R-SU**	**Ср. г.**
Косинус			X	0.49255	0.29717	0.26309	0.2468	0.49925	0.21022	0.25884	0.26064	*0.339904*
Косинус	X		X	0.48405	0.29124	0.25862	0.24468	0.48957	0.20449	0.24518	0.24702	*0.331754*
Косинус				0.47848	0.2714	0.23865	0.22615	0.47953	0.20088	0.243	0.24479	*0.320957*
Косинус	X			0.47744	0.26871	0.2343	0.22068	0.4786	0.20055	0.23713	0.23894	*0.317156*
Косинус		X		0.45838	0.25335	0.2211	0.20762	0.46138	0.18646	0.21424	0.2161	*0.299016*
Косинус		X	X	0.4558	0.25197	0.22122	0.20846	0.45931	0.1866	0.2126	0.21446	*0.298167*
Косинус	X	X		0.45627	0.25235	0.22048	0.2073	0.4605	0.18677	0.2126	0.21447	*0.298073*
Косинус	X	X	X	0.45468	0.24724	0.21597	0.20335	0.45598	0.18534	0.20957	0.21142	*0.294449*
Случайни	**?**	**?**	**?**	**0.43878**	**0.22803**	**0.19245**	**0.17755**	**0.44338**	**0.17597**	**0.20447**	**0.20628**	***0.278642***
(isf-)LCS	?	?	?	0.42827	0.21999	0.19057	0.17949	0.43314	0.1715	0.18292	0.1848	*0.268625*

Таблица А.2 Тестова група 2 - БКО, клас 1xxx, размер на обобщението M (20%)

Мярка	**Лем**	**ЧнР**	**Стоп**	**R-1**	**R-2**	**R-3**	**R-4**	**R-L**	**R-W**	**R-S**	**R-SU**	**Ср. г.**
Косинус			X	0.6175	0.46041	0.42801	0.40989	0.64045	0.24219	0.41489	0.41574	*0.473701*
Косинус	X			0.61561	0.45541	0.42405	0.40782	0.63693	0.24129	0.40976	0.41061	*0.470427*
Косинус				0.60856	0.44458	0.41228	0.39597	0.62917	0.2403	0.39985	0.40072	*0.462217*
Косинус	X		X	0.60584	0.44523	0.41424	0.39773	0.62804	0.23379	0.39516	0.39604	*0.459827*
Косинус	X	X		0.5928	0.42873	0.39874	0.38359	0.61491	0.22369	0.37115	0.37207	*0.443555*
Косинус		X		0.59118	0.42946	0.39982	0.38522	0.6138	0.22029	0.37075	0.37167	*0.442896*
Косинус	X	X	X	0.59064	0.42884	0.3999	0.38516	0.61353	0.22205	0.36974	0.37066	*0.442884*
Косинус		X	X	0.58956	0.42618	0.39668	0.38199	0.61078	0.22014	0.3672	0.36812	*0.440367*
Случайни	**?**	**?**	**?**	**0.59021**	**0.41791**	**0.3816**	**0.36194**	**0.61239**	**0.2235**	**0.3781**	**0.37899**	***0.438656***
(isf-)LCS	?	?	?	0.57374	0.40856	0.38208	0.36985	0.59481	0.21036	0.33881	0.33977	*0.422547*

Таблица А.3 Тестова група 3 - БКО, клас 1xxx, размер на обобщението L (40%)

Мярка	**Лем**	**ЧнР**	**Стоп**	**R-1**	**R-2**	**R-3**	**R-4**	**R-L**	**R-W**	**R-S**	**R-SU**	**Ср. г.**
Косинус				0.35289	0.15477	0.12322	0.11199	0.34823	0.15743	0.12846	0.13144	*0.204586*
Косинус			X	0.33175	0.13829	0.10582	0.09109	0.33137	0.14776	0.11955	0.12237	*0.187402*
Косинус	X			0.33701	0.13123	0.09623	0.08524	0.33503	0.14755	0.11912	0.122	*0.183383*
Косинус	X		X	0.31098	0.11451	0.08871	0.07978	0.32356	0.13913	0.11305	0.11594	*0.1723*
Случайни	**?**	**?**	**?**	**0.29358**	**0.10524**	**0.07737**	**0.06793**	**0.30925**	**0.13073**	**0.09913**	**0.10199**	***0.157528***
(isf-)LCS	?	?	?	0.23792	0.03382	0.02158	0.01888	0.2377	0.09765	0.05946	0.06195	*0.085811*
Косинус		X		0.23792	0.03382	0.02158	0.01888	0.2377	0.09765	0.05946	0.06195	*0.085811*
Косинус		X	X	0.23792	0.03382	0.02158	0.01888	0.2377	0.09765	0.05946	0.06195	*0.085811*
Косинус	X	X		0.23792	0.03382	0.02158	0.01888	0.2377	0.09765	0.05946	0.06195	*0.085811*
Косинус	X	X	X	0.23792	0.03382	0.02158	0.01888	0.2377	0.09765	0.05946	0.06195	*0.085811*

Таблица А.4 Тестова група 4 - БКО, клас 2xxx, размер на обобщението S (5%)

Мярка	**Лем**	**ЧнР**	**Стоп**	**R-1**	**R-2**	**R-3**	**R-4**	**R-L**	**R-W**	**R-S**	**R-SU**	**Ср. г.**
Косинус	X		X	0.40623	0.19857	0.16114	0.14597	0.43935	0.16313	0.19297	0.19464	*0.255235*
Косинус			X	0.38751	0.18943	0.1541	0.1405	0.43106	0.16306	0.18908	0.19072	*0.248592*
Косинус				0.39467	0.1784	0.14374	0.1295	0.4293	0.15793	0.19145	0.19308	*0.243017*
Косинус	X			0.36949	0.14371	0.10623	0.09471	0.40035	0.14449	0.16508	0.16669	*0.209136*
Случайни	**?**	**?**	**?**	**0.35635**	**0.12129**	**0.08698**	**0.07465**	**0.38838**	**0.14097**	**0.15211**	**0.15373**	***0.190044***
(isf-)LCS	?	?	?	0.31379	0.08818	0.05659	0.04979	0.3364	0.11819	0.11673	0.11826	*0.15003*
Косинус		X		0.31379	0.08818	0.05659	0.04979	0.3364	0.11819	0.11673	0.11826	*0.15003*
Косинус		X	X	0.31379	0.08818	0.05659	0.04979	0.3364	0.11819	0.11673	0.11826	*0.15003*
Косинус	X	X		0.31379	0.08818	0.05659	0.04979	0.3364	0.11819	0.11673	0.11826	*0.15003*
Косинус	X	X	X	0.31379	0.08818	0.05659	0.04979	0.3364	0.11819	0.11673	0.11826	*0.15003*

Таблица А.5 Тестова група 5 - БКО, клас 2xxx, размер на обобщението M (10%)

Мярка	**Лем**	**ЧнР**	**Стоп**	**R-1**	**R-2**	**R-3**	**R-4**	**R-L**	**R-W**	**R-S**	**R-SU**	**Ср. г.**
Косинус			X	0.47382	0.28559	0.24416	0.22588	0.53899	0.18277	0.29292	0.29375	*0.337578*
Косинус				0.46427	0.25376	0.21635	0.20139	0.51595	0.17234	0.27337	0.27419	*0.315293*
Косинус	X		X	0.45432	0.25488	0.21166	0.19329	0.51568	0.17014	0.26619	0.26704	*0.310197*
Косинус	X			0.44874	0.23342	0.19443	0.18125	0.50215	0.16625	0.25844	0.25927	*0.29809*
Случайни	**?**	**?**	**?**	**0.44083**	**0.20116**	**0.15713**	**0.14086**	**0.49195**	**0.1501**	**0.23962**	**0.24048**	***0.269567***
(isf-)LCS	?	?	?	0.41451	0.20097	0.16639	0.1577	0.46703	0.14726	0.2071	0.20798	*0.262053*
Косинус		X		0.41451	0.20097	0.16639	0.1577	0.46703	0.14726	0.2071	0.20798	*0.262053*
Косинус		X	X	0.41451	0.20097	0.16639	0.1577	0.46703	0.14726	0.2071	0.20798	*0.262053*
Косинус	X	X		0.41451	0.20097	0.16639	0.1577	0.46703	0.14726	0.2071	0.20798	*0.262053*
Косинус	X	X	X	0.41451	0.20097	0.16639	0.1577	0.46703	0.14726	0.2071	0.20798	*0.262053*

Таблица А.6 Тестова група 6 - БКО, клас 2xxx, размер на обобщението L (20%)

Мярка	**Лем**	**ЧнР**	**Стоп**	**R-1**	**R-2**	**R-3**	**R-4**	**R-L**	**R-W**	**R-S**	**R-SU**	**Ср. г.**
Косинус			X	0.30889	0.05632	0.0173	0.00577	0.26994	0.10578	0.08788	0.09008	*0.089082*
Косинус	X		X	0.30545	0.05279	0.01532	0.00501	0.26991	0.10582	0.08558	0.08779	*0.085298*
Косинус				0.32786	0.04751	0.01218	0.00347	0.27925	0.10857	0.09363	0.09583	*0.081644*
Косинус	X			0.33173	0.04737	0.01178	0.00313	0.28274	0.10947	0.09388	0.09613	*0.08073*
Косинус	X	X		0.29558	0.04173	0.01188	0.00411	0.2652	0.10337	0.07973	0.08193	*0.07717*
Косинус		X		0.29794	0.0415	0.01127	0.0038	0.2652	0.10366	0.08097	0.08316	*0.076355*
Косинус	X	X	X	0.28984	0.04023	0.01097	0.00398	0.25838	0.10108	0.07641	0.07857	*0.074633*
Косинус		X	X	0.29227	0.04057	0.01082	0.00375	0.2594	0.10149	0.07746	0.07963	*0.074455*
(isf-)LCS	?	?	?	0.28115	0.03579	0.00994	0.00354	0.25438	0.099	0.07238	0.0745	*0.070546*
Случайни	**?**	**?**	**?**	**0.29322**	**0.03571**	**0.00938**	**0.00235**	**0.26451**	**0.10275**	**0.07722**	**0.07939**	***0.068816***

Таблица А.7 Тестова група 7 – данни от Мултилинг