Софийски университет „Св. Климент Охридски“

Факултет по математика и информатика

Катедра „Компютърна информатика“

ДИПЛОМНА РАБОТА

на тема

Разпознаване автор на текст

Ръководител:

проф. д-р Мария Нишева

Дипломант:

Цветина Людмилова Хантова

Факултетен номер: М-24064

Магистърска програма: „Изкуствен интелект“

София, 2015

Съдържание

1. Въведение

1.1 Дефиниране на задачата

Задачата за разпознаване автор на текст може да бъде формулирана изключително просто. Нека е дадено затворено множество от автори и за всеки от тях е дадено непразно множество от текстове. Даден е и текст, който принадлежи на един от авторите от това множество. Задачата е да се определи авторът, на когото даденият текст принадлежи.

Въпреки своята проста формулировка и разглеждането на задачата вече повече от век, все още няма ясно установен и единодушно приет за правилен подход, който да бъде използван за нейното решаване.

Едва ли е и реалистично да смятаме, че ще бъде открит универсален подход за решаването ѝ поради няколко причини.

Въпреки че по-горе дефинираната задача може да бъде разгледана като стандартна задача за категоризиране на текст [1], са изключително редки случаите, в които при решаване на задача за разпознаване автор на текст разполагаме с малко затворено множество от автори и неограничено количество от обучаващи текстове за всеки от тях. Много по-често има различни ограничения:

1. Липсва множество от автори. В този случай задачата вече има за цел да създаде профили на различните автори – демографска и психологическа информация, която може да бъде извлечена.
2. Множеството от автори се състои от много кандидати и за всеки от тях има малко количество от текстове.
3. Няма затворено множество от автори, но има един предполагаем автор и трябва да се определи дали той наистина е авторът на текста – потвърждаване на авторство.

Друга причина е, че задачата дава възможност на всяка една от стъпките при нейното решаване да бъдат използвани множество различни подходи и методи, както статистически, така и изчислителни. През годините някои от тях са показали добри резултати, а други са били отхвърлени, но не са много подходите, които са тествани върху разнообразие от корпуси, за да се проследи ясно до каква степен особеностите на корпусите влияят на получените резултати и до каква степен подходите са мащабируеми.

Не на последно място, до този момент не е установено стандартно множество от корпуси, върху които различните подходи към задачата да бъдат тествани, което прави тяхното обективно сравнение много по-трудно.

В настоящата разработка ще бъдат разгледани два алгоритъма за избор на характеристики, които ще дефинират авторовия стил. Единият алгоритъм се базира на синтактични n-грами, а другият използва честотата на думите в естествения език. Тези два подхода не са използвани често при решаване на задачата за разпознаване на авторство, но в разработките, в които са приложени, показват обещаващи резултати.

Целта е да бъдат приложени тези методи върху няколко корпуса от данни с различни характеристики и по този начин да могат да бъдат анализирани техните предимства и

недостатъци, да се установи тяхната мащабируемост и да се съпостави постигнатата точност с тази на други подходи.

1.2 Кратка история

Интересът към авторството се заражда още в епохата на Ренесанса, когато достъпът до повече текстове и литературни творби прави възможно тяхното сравняване. Скоро това довежда и до по-критичния поглед на хуманистите към езиковите/лингвистичните дисциплини. Един от най-ярките примери е ренесансовият хуманист Lorenzo Valla, който показва през 15-ти век, че „Donatio Constantini“, документ, чрез който се дарява западната част на Римската империя на папа Силвестър, е късен фалшификат и позовавайки се на използвани граматически форми и думи установява, че не е възможно този документ да е написан преди VIII век.

В съвремието авторството на литературните текстове най-често се определя на база външни белези – името на творбата е съпроводено с нейния автор, но има случаи, в които това може да е измамно, и такива, в които външни белези изобщо не съществуват, особено в разрастващото се интернет пространство. Така за текстове, за които няма категоричен белег за авторство, започват да се вземат предвид множество вътрешни такива – тема, изказ, използвани фрази, художествени похвати.

През XVIII век Alexander Pope, най-известен с преводите си на Омир, твърди, че определянето на авторство въз основа на стила е твърде наивен подход. На пръв поглед изглежда, че е сравнително лесно за даден човек да имитира нечий чужд стил. Samuel Johnson [2] малко по-късно заявява своето мнение, което е в пълна противоположност. Неговият приятел и биограф James Boswell [2] го запитва дали всеки човек има свой собствен стил на писане, както всеки човек е уникален в своя почерк или физиономия. Той отговаря категорично утвърдително, казвайки, че всеки има своя „странен“ стил, който може да бъде разпознат и идентифициран след задълбочени изследвания и сравнения. Но авторът на дадена творба трябва да е наистина добър, за да успее да изгради ясно и явно различим стил.

Търсенето на стилометрични маркери, които да са извън съзнателния контрол на автора, води до разграничаването на литературната интерпретация и стилометрията. Както Horton [2] го поставя, текстовите характеристики, които са видими и забележими за един литературовед, често са отражение на съзнателно взети решения от автора и в голяма степен са податливи на имитация - съзнателна или не. Но има част от съставянето, построяването на изреченията, която се осъществява от части от мозъка, които работят по толкова сложен начин, можем да кажем, че в голяма степен не могат да бъдат лингвистично контролирани.

Именно върху това стъпват повечето подходи и методи за разпознаване автор на текст в края на XIX век, когато започват реално да се появяват разработки, които се опитват да оценят авторовия стил количествено.

Едни от най-важните разработки по това време са тези на Mendenhall [3], Zipf [4], Yule [5, 6]. Но изследването, което се откроява най-много, е на Mosteller & Wallace [7]. Те използват разпределението на функционалните думи (думи, чиято основна функция е граматическа, такива думи в английския език са: of, about, the и много други) като характеристики, за да установят оспореното авторство на т.нар. Federalist Papers между тримата автори (Alexander Hamilton, James Madison, John Jay). От тогава много изследвания се фокусират в търсенето на

различни стилистични маркери за разпознаване на авторство, използвайки статистически анализи.

До края на XIX век разработките са доминирани от прилагането на Multivariate statistical analysis за различаване между различните автори. Появата на все по-мощни компютри провокира развитието на изследванията в съвсем нова посока, стъпвайки върху компютърните науки и компютърната лингвистика. Все по-често учените използват техники от Извличане на информация (Information Retrieval), Машинно самообучение (Machine Learning) и Обработка на естествен език (Natural Language Processing) за разпознаването на авторство.

Трябва да се отбележи, че всички използвани подходи почиват на няколко важни предположения. Първото от тях е, че авторовият стил е повлиян от различни характеристики на автора – пол, личен характер, ниво на образованост и много други. Друго предположение е, че тези характеристики могат да бъдат извлечени от неговия авторов стил. Но най-важното предположение е, че авторовият стил е количествено измерим по отношение на характерния начин, по който е използван езикът.

Въпреки че е всеобщо прието, че авторовият стил се повлиява от външни фактори като време, тема, се смята, че има авторови характеристики, които са сравнително константни и могат да бъдат извлечени независимо от тези фактори.

В следващата секция ще бъдат разгледани най-използваните до този момент характеристики за идентифициране на авторовия стил, както и техниките, използвани за различаването на отделните стилове.

1.1 Методи и техники, използвани досега

1.1.1 Типове характеристики

Rudman [8] пресмята, че в различни разработки по темата за разпознаване на авторство са разгледани над хиляда различни типове характеристики, които да определят авторовия стил. Но и до този момент няма съгласие кои от тях биха били най-добрият избор, повечето учени, изследващи проблема, обаче са единодушни, че няма една единствена характеристика, която сама по себе си да бъде достатъчно информативна, за да може да бъде направено разграничението между стиловете на различни автори.

Характеристики, които изчисляват сложността на текста и богатството на лексиката са сред най-ранно използваните. Примери за такива характеристики са средна дължина на думите в текста (по-общо - разпределението на дължината на думите в текста), средна дължина на изреченията, броят на думите с определена честота в текста (често използвано – hapax legomena – думи, които се появяват само веднъж в творбите на даден автор или в определена творба), богатство на речника – type-token ratio: V/N, където V е броят на уникалните думи в текста, а N е общият брой на думите в текста, характеристиката на Yule – K [5], характеристиката на Honore – R [9].

За всяко проучване, което твърди, че тези характеристики са добре разграничаващи авторовия стил, последват проучвания, показващи обратното. С времето се стига до съгласие, че тези маркери биха били полезни, ако се използват в комбинация с други типове характеристики, но в изолация не биха могли да бъдат надежден подход за решаване на проблема.

В съвременните изследвания най-често се използват характеристики, които представят определен езиков/лингвистичен слой – символни, лексикални, синтактични, семантични.

Основното предимство на използването на символните n-грами за характеристики, описващи авторовия стил е, че те са независими от езика, но въпреки това могат да послужат да се определят лексикалните предпочитания на автора, както и в някаква степен граматичните такива. Друго голямо предимство е, че тяхното извличане е лесно и не изисква допълнителна обработка на текста. Въпреки че още в началото на 90-те има разработки, които показват успешно идентифициране на език, използвайки n-грами [10, 11], за разпознаване на авторство те започват да се използват едва в началото на новото хилядолетие [12, 13, 14, 15, 16, 17]. Размерността на проблема, използвайки този подход, се увеличава значително с нарастване на n. Друг проблем е, че е възможно събраната информация да е обвързана повече със съдържанието на текста, отколкото с авторовия стил.

Лексикалните характеристики са едни от най-използваните до този момент. Дори на интуитивно ниво е ясно, че те носят стилистична информация. Те са лесни за извличане, тъй като изискват единствено токанизация, а също така са и лесни за интерпретация. Двата основни типа лексикални характеристики са функционалните думи и контекстните думи.

Mosteller и Wallace [7], които работят по определянето на оспореното авторство на т.нар. Federalist Papers, са първите, които използват честотите на предварително дефиниран брой функционални думи. От тогава насам функционалните думи са от най-често използваните характеристики.

Повечето съвременни експерименти, които използват функционалните думи за определяне на авторовия стил, подбират множество от няколко стотин функционални думи – местоимения, предлози, спомагателни и модални глаголи, съюзи. Числата и междуметията също често се добавят. Те не се включват в дефиницията на функционални думи, но са независими от темата и могат да носят ценна отличаваща информация.

Причината да се използват именно функционалните думи е, че се очаква тяхната честота да е в голяма степен независима от темата на творбата и по този начин да е възможно разпознаването на автор дори в случаите на разнообразно в тематично отношение творчество. Друго предимство на функционалните думи е, че честотата на употреба и подборът им е в по-малка степен съзнателно контролиран от автора и така може да се предотврати рискът от опити за измама.

Резултатите, които се постигат с този подход, много често са сравними с тези, постигнати от много по-сложно конструирани характеристики.

Контекстните думи (content words) също се използват за характеристики. В някои изследвания най-рядко използваните думи в корпуса, както и тези, които са равномерно разпределени в него, се премахват, за да има множество от думи, което с по-голяма вероятност да представи авторовата лексика, независимо от темата. В повечето случаи, използването само на контекстни думи за определяне на авторовия стил обаче води до повече проблеми и е по-разумно да бъдат използвани в комбинация с други характеристики.

Когато за характеристики се използват определено множество от думи (bag-of-words), важна информация, която не се взема предвид, е контекстът, в който се използват тези думи. Поради тази причина се появяват експерименти, които конструират характеристиките на база най-често срещаните n-грами от думи [14, 18, 19]. Резултатите показват, че не винаги постигната точност е по-висока от тази постигната, когато се използват отделни думи за характеристики [18, 19]. Един от проблемите на този подход е, че често векторите от честоти на n-грами съдържат много

нулеви стойности, защото много от комбинациите от думи не присъстват в разглеждания текст, особено ако той е по-кратък. Това прави ефективното обучение на класификатор много по-трудно. Друг проблем на този подход е отново вероятността да бъде уловена информация, която е специфична за съдържанието на текста, а не за авторовия стил [20].

Синтактичните характеристики се смятат за по-надеждни в сравнение с контекстните думи, защото не се контролират съзнателно от автора, а също така предоставят едно по-високо ниво на абстракция, вече не се използват конкретните думи. Baayen [21] е един от първите, които прилага синтактичните характеристики в задачата за разпознаване автор на текст. Той показва, че честотите на правилата на пораждащата ги граматика могат надеждно да разграничат различни автори и типове текстове. Повечето разработки, които последват, не изискват пълната обработка на текста, което се изисква при използването на правилата от пораждащата граматика. По-често се използват за характеристики n-грами от частите на речта или граматическите връзки [22, 23, 24, 25]. Въпреки това, синтактичните характеристики все още не са от най-често използваните, защото не са много езиците, за които има надеждни и стабилни платформи за обработка на естествен език.

По-нестандартен подход за текстове, които не са преминали редакция, могат да бъдат ортографичните и синтактични грешки и особености за определяне на авторство. Koppel и Schler [26] анализират текстове на имейли, като използват MS Word за проверка за синтактични и граматически грешки. Те идентифицират различни типове грешки – повтаряща се буква, разменени букви, замяна на букви, сливане на думи, и чрез тях се опитват да изолират особености на автора, които могат да бъдат използвани при последващо разпознаване на авторство на непознат текст.

Двата алгоритъма, които ще бъдат разгледани в дълбочина в рамките на тази разработка, надграждат два от най-използваните подходи за избор на характеристики – n-грами и Delta подход на Burrows [27].

Първият алгоритъм използва синтактични n-грами. Те се конструират на база на последователността, в която елементите се появяват по пътищата в синтактичното дърво, а не на база реда, в който се появяват в текста. Най-голямото предимство на синтактичните n-грами е, че те използват синтактичните връзки между думите, което липсва при използването на обикновени n-грами.

Другият подход, който е базиран на нормалната честота на думите в естествения език, има за цел да надгради един добре известен метод, който се основава на честотата на думите в текста – Delta подхода на Burrows [27]. Идеята, която описват Chen, Huang, Yang, Meng и Miao [28], разделя думите в текста на зони на база нормалната честота на думите в естествения език и анализира разпределението на групите от думи в текст. Групирането на думите постига едно по-високо ниво на абстракция, което позволява методът да бъде по-малко повлияван от темата.

1.1.2 Подходи за разграничаване на отделните автори

След като бъдат извлечени характеристиките, които ще представят всеки един от текстовете в корпуса, те трябва да бъдат анализирани, за да се определи кой текст от кого е написан. Приложените до този момент подходи са разнообразни.

Няколко важни фактора, които е добре да бъдат обмислени преди избор на конкретен подход, са количеството и типа на обучаващите данни, както и изискванията към крайния резултат, който искаме да получим. Особено когато говорим за съдебен доклад за авторство, е важно не само да бъде представен резултатът, но е важно да има аргументация на база на кои особености и аспекти на документите е постигнат.

Първите приложени подходи използват статистически анализи и вероятностни разпределения. Например, ако имаме множество от документи от двама различни автора, лесно могат да бъдат пресметнати дължините на думите в текстовете (брой на сричките в думата, брой на думи в изречението и т.н.) и прилагайки t-тест да се установи дали двамата автори имат различни средни. Student t-тест предполага независимост на данните, както и тяхното нормално разпределение. За да бъде избегнато това предположение, могат да бъдат използвани непараметрични статистики, например Wilcoxon тест.

Тези прости статистики не дават достатъчно добри резултати, но те могат да бъдат комбинирани. Един такъв подход, който оказва най-голямо влияние, е методът, използван от Burrows [27] – Delta. Методът е прилаган за решаването на много различни задачи за разпознаване на авторство, а също така има и много опити за неговото разширяване и подобряване по различни начини.

Burrows анализира първите 150 най-често срещани думи в колекция от текстове на различни поети. За всяка една от думите той пресмята подходящо нормално разпределение (оценка на средната честота на думата и също така оценка на отклонението от нея). За всеки един от текстовете се пресмята колко се отклонява над/под нормата честотата на всяка една от 150-те думи. Положителен z-score отразява дума, по-често срещана от средното, а отрицателен такъв - дума, по-рядко срещана от средното. Burrows дефинира Delta метриката, както следва:

Средното на абсолютните стойности на разликите между z-score на подбраните думи в текстове от дадена категория и z-score на същите думи в непознатия текст.

Категорията, към която се определя текстът, е тази, при която се постига най-малка стойност на Delta метриката.

Резултатите, които се постигат, използвайки този подход, са добри и в много случаи се използват като база за сравнение за новите методи, които се изследват.

Решението за авторството на текст може да бъде взето и, ако разпределението на характеристиките се разглежда като вероятностно разпределение, което да спомогне да се дефинират разликите между текстовете, използвайки някои от стандартните вероятностни разлики – отклонение на Kullback‑Liebler, разстояние на Kolmogorov‑Smirnoff и др. Това може да бъде основата за прилагане на алгоритъма за k най-близки съседи. Проблемът с този подход е, че той разчита на независимост на характеристиките в избраното множество.

Появата на техниките, базирани на машинно самообучение, е повратна точка в изследванията за разпознаване на авторство. Прилагането на тези методи е просто – обучаващите текстове се представят като числови вектори и всеки от тях е асоцииран с етикет, обозначаващ автора на текста. Обучаващи алгоритми се прилагат, за да се открият границите на класовете, които ще минимизират грешката при класификация.

Трите подхода, които най-често се използват, са невронни мрежи, дърво на решенията и наивен Бейсов класификатор.

Невронните мрежи [29, 30, 31] се организират в три или повече слоя и се обучават, използвайки метода на обратното разпространение на сигнал за грешка (backpropagation), който има за цел да минимизира грешката на база получения на изходния слой резултат и желания резултат. Основен недостатък при използването на невронните мрежи е, че дори да се осъществи точна класификация, не са ясни факторите, на които тя е базирана.

За разлика от невронните мрежи, подходът, използващ дърво на решенията [32], е замислен да осъществява описателна класификация. Дървото на решенията е рекурсивна структура от данни, която съдържа правила, които разделят пространството от характеристики на множества от подслучаи и така се дефинира съпоставяне на области от пространството към категории, в разглеждания случаи - различните автори.

Пример за такова правило е:

(Rule 1)
if feathers = no **and** warm-blooded = yes
then type is MAMMAL
else apply rule 2.

Въпреки добрата интерпретация, която позволява този подход, както и възможността да се използват нечислови характеристики, в повечето изследвания до момента той не успява да постигне по-добри резултати от машините с поддържащи вектори (Support Vector Machines).

Наивният Бейсов класификатор осъществява подобна класификация, но не използва дървовидна структура, а теоремата на Бейс, за да открие най-вероятната категория на база обучаващите данни. Използвайки теоремата на Бейс, може да бъде изчислена вероятността за настъпване на дадено събитие, след като е известна част от информацията за него:

$$P(A|B) = P(B|A) * P(A)/P(B),$$

където $P(A|B)$ – условната вероятност за настъпване на събитието A при положение, че събитието B е настъпило (апостериорна вероятност), $P(A)$ – вероятност за настъпване на събитието A (априорна вероятност), $P(B|A)$ - условната вероятност за настъпване на събитието B при положение, че събитието A е настъпило, $P(B)$ - вероятност за настъпване на събитието B.

Недостатък и на този подход, както и на повечето вероятностни подходи, е предположението за независимост на характеристиките. Например, използвайки този подход, се приема, че честотите на срещане на думите „I“ и „me“ са независими, което няма как да бъде вярно.

Подходът, който дава най-стабилни и добри резултати в повечето изследвания, в които е включен, е базираният на машините с поддържащи вектори [25, 33, 34]. Те успяват в голяма степен да се справят с две от най-сериозните предизвикателства пред повечето алгоритми в машинното самообучение. А именно, машините с поддържащи вектори успяват да се справят с пространства с изключително голяма размерност и в голяма степен се справят с опасността от прекалено нагаждане към обучаващите данни (overfitting). Abbasi [35] показва, че в повечето случаи SVM дават по-добри резултати от другите методи за класификация като невронни мрежи, дърво на решенията или LDA (Linear Discriminant Analysis).

Това е подходът, които ще бъде използван и в настоящата разработка.

2. Мащабируемост в задачата за разпознаване на автор

2.1 Влияние на големината на множеството от автори

Правилното идентифициране на автор на текст, когато възможните автори са двама или трима, е задача, която в повечето случаи може да бъде решена с точност често над 90%. Но тестването на различни подходи само върху малки корпуси от данни няма как да даде информация за надеждността и мащабируемостта на подхода.

Ако експериментите се извършват само върху малко множество от автори, то оценката на подхода няма да бъде точна, резултатите ще бъдат нереалистично високи и характеристиките, извлечени от обучаващите данни, дори и да са отличаващи за това малко множество от автори, не биха се запазили при неговото увеличаване.

Едва в последните няколко години се появяват проучвания, които започват да използват по-големи множества от автори и анализират тяхното влияние.

Koppel [36] е един от първите, които се фокусират върху влиянието на големината на множеството от автори върху постигнатите резултати. Koppel използва множество от блогове – 10 000 автора (блога) като се вземат по 2 000 думи на автор за обучение и 500 за тестване. Той използва статистически подход и изпробва различни големини на множеството от автори, както и големини на фрагмента за обучение и за тестване. Съвсем очаквано, резултатите показват намаляване на точността при увеличаване на множеството от автори. Независимо от това, точността на оценяване (precision), с която се присвоява даден тестов текст към един от 1 000 автори е 93.2%, като точността на връщане (recall) е 39.3%.

Abbasi & Chen [35] използват богато множество от стилистични характеристики на различни лингвистични нива (лексикални, синтактични, структурни, както и специфични за съдържанието). Множеството от характеристики наброява няколко десетки хиляди. Точността, която постига Whiteprint системата върху Enron Email корпус (28 000 думи на автор) е 83%, когато големината на множеството от автори е 100. В експеримента върху текстове, взети от Java форум, с над 40 000 думи на автор, точността пада от 88% при 25 автора до 53%, когато авторите са 100. Подходът е обещаващ, но е важно да се отбележи, че разчита на голямо множество от обучаващи данни и използва тематично специфични характеристики. Това поставя въпроса какви биха били резултатите, ако в корпуса от данни има множество различни теми.

Zhao & Zobel [32] изследват поведението на различни алгоритми за машинно самообучение за разпознаване автор на текст, като множеството от автори е с големина 2, 3, 4 и 5. Резултатите показват, че точността спада с повече от 20%, когато авторите са петима, в сравнение с резултатите, получени при изследване само на двама автори.

Последващите експериментите ще покажат доколко алгоритмите, които се разглеждат в настоящата разработка, се повлияват от промяната в големината на множеството от автори.

2.2 Влияние на големината на множеството от данни за автор

Голяма част от проучванията за разпознаване автор на текст разчитат също на голям обем от текстове за всеки от авторите, повече от десетки хиляди думи за автор. Съвременните приложения на задачата изискват подходът да се справя с много по-малко данни, например, когато корпусът се състои от имейли, блог постове, туитове.

В изследванията на алгоритмите за машинно самообучение е прието, че повечето обучаващи данни водят до по-добри резултати или още известно като принципа – There is no data like more data [37].

Не са много изследванията, които се фокусират върху корпуси с ограничено количество от данни [17, 32, 38, 39]. Тяхната цел е да оцени спада в точността на различни подходи при намаляване на количеството на текстове за даден автор.

Според Biber [40, 41] 1 000 думи са достатъчни, за да може надеждно да бъдат пресметнати стилистичните особености в даден текст. След експериментите на Burrows [42] се смята, че 10 000 думи за автор са минимумът, нужен, за да се установи авторство на литературни текстове. Някои типове характеристики, като богатство на речника, например, не са надеждни, когато се прилагат върху текстове с дължина по-малка от 1 000 думи [43]. Eder [44], показва, че Delta подходът не е стабилен и надежден, когато данните за автор са съставени от по-малко от 2 500 думи. Но със сигурност няма ясен отговор за необходимото минималното количество данни за автор и от резултатите от направените експерименти се вижда, че отговорът е зависим от жанра и тематиката на творбите в корпуса от данни.

Hirst & Felguina [17], в изследване на авторството на кратки откъси от творби на Anne и Charlotte Bronte, представят систематично проучване за влиянието на различните големини на откъсите – броят на думите в текста (200, 500 думи) – и също така влиянието на увеличаващия се брой от откъси, използвани за обучение. Резултатите показват, че използването на голямо множество кратки текстове отчасти преодолява пречката от наличието само на кратки текстове, дори когато ‘кратки’ означава само 200 думи за автор.

Abbasi & Chen [35] разглеждат четири различни множества от данни с различна големина и характеристики. CyberWatch Chat съдържа средно по 1 400 думи за автор, Java Forum съдържа по 44 000 думи за автор, Enron Email корпусът - 28 000 думи за автор. Подходът им постига точност от 83% върху Enron Email, 53% върху Java Forum и 32% върху CyberWatch Chat. Резултатите показват, че големината на множеството от данни за автор е важен фактор, но жанрът, темата и шумът в съответните корпуси също имат роля в получената точност.

Резултатите от експериментите в настоящата разработка също ще потвърдят, че количеството данни за даден автор е от голямо значение, но разнообразието от теми и жанрове в корпуса, също има ефект върху точността при класификация.

3. Корпуси от данни

В тази секция ще бъдат описани корпусите, които ще бъдат използвани за тестване на двата избрани подхода за подбор на авторови характеристики. Целта при избора на корпуси е те да бъдат с различни характеристики, което ще даде по-точна оценка за мащабируемостта на методите. Единият от корпусите е доста популярен и ще бъде използван като базов, другият се състои от романи, третият – от блог постове. Последните два корпуса са доста различни както като жанр, големина на текстовете, от които са съставени, така и по броя на авторите, включени в тях.

3.1 Federalist Papers

Federalist Papers е колекция от 85 политически есета, публикувани през 1787 г. и 1788 г. от анонимен автор под псевдонима Publius. По-късно става ясно, че зад псевдонима стоят John Jay, Alexander Hamilton и James Madison. Авторството на есетата е изследвано първо от Monsteller и Wallace [7].

Корпусът ще бъде използван като базов. Повечето познати подходи към задачата са тествани върху този корпус, което позволява да бъде направено сравнение с получените до този момент резултати. Друга причина, това да бъде един от използваните корпуси, е, че той изпълнява повечето от изискванията за добре контролиран корпус – малко на брой автори, от които има достатъчно документи, за да се идентифицира техният авторов стил. Също така корпусът е контролиран относно тематиката и повечето текстове са с приблизително еднаква големина.

3.2 Английски романи (The Corpus of English Novels)

The Corpus of English Novels (CEN) е съставен от Hendrik De Smet [56]. Състои се от романи, написани от 25 новелиста, както британски, така и американски (Фигура 3-1). Всички романи са написани между 1881 и 1922. Авторите представят грубо едно поколение от новелисти. Корпусът не е контролиран относно жанр и тематика.

От текстовете на всеки един от романите е премахната информацията за автора и името на творбата.

За да се анализира по какъв начин алгоритмите се повлияват от броя на авторите в корпуса, както и количеството от данни за всеки един от тях, алгоритмите ще бъдат приложени върху няколко различни конфигурации на корпуса (Фигура 3-2, Фигура 3-3).

AUTHOR	NR. OF NOVELS	YEAR OF PUBLICATION	NR. OF WORDS
Andy Adams (1859-1935)	5	1903-1911	450,564
Arthur Conan Doyle (1859-1930)	18	1888-1913	1,566,987
Edith Nesbit (1858-1924)	8	1899-1907	537,969
Edith Wharton (1862-1937)	11	1900-1922	872,824
Emerson Hough (1857-1923)	9	1900-1922	751,315
Frances Burnett (1849-1924)	11	1881-1922	974,948
Francis Marion Crawford (1854-1909)	13	1882-1903	1,396,223
George Augustus Moore (1852-1933)	10	1885-1901	996,682
George Gissing (1857-1903)	20	1884-1905	2,408,767
Gertrude Atherton (1857-1935)	10	1888-1922	634,864
Gilbert Parker (1862-1932)	16	1893-1921	1,398,355
Grant Allen (1848-1899)	8	1884-1899	590,205
Hall Caine (1853-1931)	4	1885-1913	665,937
Henry Rider Haggard (1856-1925)	25	1885-1910	2,556,621
Henry Seton Merriman (1862-1903)	12	1892-1913	988,647
Humphrey Ward (1851-1920)	17	1881-1916	2,252,823
Irving Bacheller (1859-1950)	8	1892-1922	511,064
Jerome Kapla Jerome (1859-1827)	10	1886-1919	706,389
Kate Douglas Wiggin (1856-1923)	14	1893-1915	677,656
Lyman Frank Baum (1856-1919)	14	1900-1916	622,7
Marie Corelli (1855-1924)	11	1886-1921	1,719,829
Ralph Connor (1860-1937)	11	1898-1921	974,84
Robert Barr (1850-1912)	10	1893-1910	731,329
Robert Louis Stevenson (1850-1894)	9	1881-1893	676,472
Stanley John Weyman (1855-1928)	6	1890-1901	563,418
TOTAL	292	1881-1922	26,227,428

Фигура 3-1 Таблица с характеристиките на корпуса от английски романи [56]

Data Set Size	Limited	Balanced	Unbalanced
Novels Count	3	8	Original Count - (2, 21)

Фигура 3-2 Таблица с различните конфигурации на множеството от текстове за всеки автор в корпуса от английски романи

Authors Set Size	Small	Medium	Big
Authors Count	5	15	25

Фигура 3-3 Таблица с различните конфигурации на множеството от автори в корпуса от английски романи

3.3 Блог постове (The Blog Authorship Corpus)

Оригиналният корпус се състои от 19 320 блога, събрани от blogger.com през 2004 година. Всеки един от блоговете е представен в отделен xml файл. Името на файла носи информация за автора на блога – идентификационен номер, пол, години, сферата, в която работи, и неговата зодия.

Корпусът е събран и обработен от J. Schler и M. Koppel през 2006 година [55]. Всеки един от блоговете в корпуса съдържа поне по 200 срещания на основни думи в английския език. Форматирането на текстовете е премахнато, връзките в постовете са заменени с етикета – urllink. Корпусът е използван за първи път от J. Schler, M. Koppel, S. Argamon и J. Pennebaker [45]. Те се опитват да отговорят на няколко важни въпроса, свързани с особеностите на авторовия стил – по какъв начин съдържанието и стилът на писане се повлияват от това дали авторът е мъж или жена и каква информация можем да извлечем за някого от написан от него текст.

В настоящата разработка се използва подмножество от корпуса с блогове. Блоговете, които са подбрани, са на хора, работещи в сферата на образованието. Общият брой на тези блогове е 150. Общият брой на постовете – 9 264.

Както за корпуса от английски романи, за да се изследва по какъв начин алгоритмите се повлияват от броя на авторите в корпуса, както и от количеството от данни за всеки един от тях, алгоритмите ще бъдат приложени върху няколко различни конфигурации на корпуса (Фигура 3-4, Фигура 3-5).

Data Set Size	**Limited**	**Balanced**	**Unbalanced**
Blog Posts Count	5	20	Original Count - (1, 813)

Фигура 3-4 Таблица с различните конфигурации на множеството от документи за всеки автор в корпуса от блог постове

Authors Set Size	**Small**	**Medium**	**Big**
Authors Count	10	50	150

Фигура 3-5 Таблица с различните конфигурации на множеството от автори в корпуса от блог постове

За по-удобното прилагане на алгоритмите е написан скрипт, който обработва xml файловете и поставя всеки един от постовете в отделен файл. Имената на файловете с постове се конструират по следната схема – post_<blogId>_<postIndex>, която е по-различна от оригиналната схема на именуване в корпуса, включваща демографска информация за автора, която няма да е нужна в настоящата работа.

4. Софтуерна архитектура, разработена и използвана за експериментите

4.1 Основна структура

За демонстриране на използваните подходи за разпознаване автор на текст е разработено уеб приложение. Уеб приложението използва Full-stack JavaScript Framework – MEAN (MongoDB, Express, Angular.js, NodeJS). **MongoDB** е open-source нерелационна база от данни. В нея приложението пази получените резултати от експериментите върху различните корпуси. **NodeJS** и **Express** се използват за разработката на server-side функционалността, която включва разработката на алгоритмите за извличане на авторовите характеристики от документите, прилагането на SVM за класификация, както и инфраструктурата за осъществяване на експериментите. **Angular.js** е front-end JavaScript framework, който се използва за визуализиране на получените резултати.

За удобното осъществяване на експериментите е създаден клас **Classifier**. При създаването на обект от този клас се посочват:

- **corpusDictionary** – път към корпуса, който ще бъде използван
- **corpusSize** – използват се предефинирани стойности. Параметърът е низ от вида – authorsSetSize_dataSetSize. Стойностите за големината на множеството от автори са small, medium, big. Стойностите за големината на множеството от данни – limited, balanced, unbalanced. Конкретните стойности за всяко едно от множествата са описани в секцията **„Корпуси от данни“**.
- **featuresExtractionMethod** – методът, използван за извличане на авторовите характеристики от текста. За момента са разработени - SN_GRAMS и NFZ_WD.

Интерфейсът на класа е както следва:

- **constructAllFeatureSets(options, done)** – извлича векторите с характеристиките за авторите от всички текстове в корпуса в зависимост от избрания **featuresExtractionMethod**
- **train(options, done)** – обучава SVM класификатор на база на извлечените характеристики от текстовете в корпуса
- **save(done)** – запазва обучения модел, за да може да бъде използван за последваща класификация
- **trainAndSave(options, done)** – обучава SVM класификатор и го запазва

Опциите, които се подават, са различни в зависимост от алгоритъма, който се използва за извличане на множеството от характеристики.

Когато се прилага алгоритъмът, базиран на синтактични n-грами, може да бъде посочена дължината на синтактичните n-грами, както и големината на характеристичния вектор, т.е броят на най-често срещаните синтактични n-грами, които се вземат предвид при построяване на вектора за всеки един от текстовете.

Параметърът, който може да бъде конфигуриран, когато се използва алгоритъмът, който конструира зони на база нормалната честота на думите, е функцията за разделяне на множество от зони. Възможните стойности на параметъра са: LINEAR, RADIX, LOGARITHMIC.

След като се извърши извличането на характеристичните вектори и обучението на машините с поддържащи вектори за всяка една от конфигурациите на корпусите се прави запис в MongoDB базата от данни. Записът е със следната структура:

- corpusName – име на корпуса
- corpusSize – големина на корпуса (authorsSetSize_dataSetSize)
- featuresExtractionMethod – алгоритъма, който е използван за извличане на характеристичните вектори
- params – параметри на използвания алгоритъм
    - n
    - snGramsCount
    - partitionFunction
- modelFilePath – относителен път към файла с обучения SVM модел
- report – обобщени резултати след обучението на SVM модел, както и параметрите (C, gamma), при които са постигнати
    - C
    - gamma
    - fscore
    - accuracy

4.2 Файлова система на корпусите

Файловата структура, чрез която са представени различните корпуси, е една и съща. Папката, която съдържа данните за корпуса, е със следната структура:

- all – папка, съдържаща всички документи от корпуса. Всеки от текстовете на авторите от корпуса е в отделен файл.
- trees – папка, съдържаща всички документи от корпуса, след като са обработени от синтактичен парсер.
- authors – папка, съдържаща файл, в който се пази съответствието между документ (представен чрез името на файла) и автор. Авторът е представен от число, за да бъде подаден коректен вектор на SVM алгоритъма за обучение, който очаква числов вектор.
- sets – папка, съдържаща файлове, в които са описани всички документи, които се включват в съответната конфигурация на корпуса. Общият брой на конфигурациите, анализирани в настоящата разработка, са 9 в зависимост от големината на множеството от автори и множеството от документи, налични за тях.

4.3 Машини с поддържащи вектори (Support Vector Machines)

Машините с поддържащи вектори [50] са модел за машинно самообучение с учител, който все по-често се използва за класификация на данни.

Ако има обучаващо множество и всеки елемент от обучаващото множество принадлежи на една от две категории, SVM обучаващият алгоритъм създава модел, който да може да определи категорията на непознати примери, които не са част от обучаващото множество. Машините с поддържащи вектори могат да бъдат дефинирани като невероятностен двоичен линеен класификатор (non-probabilistic binary linear classifier).

В теорията на машините с поддържащи вектори всяка точка от множеството от данни се разглежда като вектор с размерност p и целта е всички точки от множеството да бъдат разделени с хиперравнина с размерност $(p-1)$. Това се нарича линеен класификатор. В повечето случаи има много хиперравнини, които ще удовлетворят това условие. Ограничението, което поставя SVM моделът, за да избере най-добрата хиперравнина, е тя да осигурява най-голямото разделение между двата класа. Т.е избира се хиперравнина, така че разстоянието до най-близките точки от двата класа е най-голямо.

Нека D е обучаващо множество:

$$D = \{(x_i, y_i) | x_i \in R^p, y_i \in \{-1,1\}\}_{i=1}^{n}$$

y_i обозначава категорията на x_i.

Целта е да се намери хиперравнина, която да разделя точките, за които $y_i = 1$, от точките, за които $y_i = -1$, и да осигурява максимално разстояние между двете категории. Всяка хиперравнина може да бъде описана с множеството от точки x, които удовлетворяват следното равенство:

$$w \cdot x - b = 0,$$

$\cdot$ е скаларно произведение, а w е нормалният вектор към хиперравнината.

Ако обучаващите данни са линейно разделими, то могат да бъдат избрани две хиперравнини, които да разделят данните от двете категории и да няма точки между тях, и след това да се максимизира разстоянието между тях. Тези хиперравнини могат да бъдат описани със следните равенства:

$$w \cdot x - b = 1$$

и

$$w \cdot x - b = -1$$

Разстоянието между тях е $2/\|w\|$, т.е задачата е да се минимизира $\|w\|$.

За да се осигури, че няма точки между двете хиперравнини е необходимо да бъде изпълнено:

$$y_i(w \cdot x_i - b) \geq 1, i = 1, \dots, n$$

Това е оптимизационна задача на квадратичното програмиране (Quadratic Programming Optimization Problem), по конкретно:

$$arg\min_{(w,b)} 1/2\|w\|^2$$, където трябва $y_i(w \cdot x_i - b) \geq 1, i = 1, \ldots, n$

Възможно е обаче да не съществува хиперравнина, която да разделя двете категории. Тогава трябва да бъде избрана хиперравнина, която разделя точките, колкото е възможно по-добре. За тази цел се въвеждат променливите ξ_i, които измерват степента на грешка при класификацията на елемента x_i, а $C > 0$ е параметър, който определя степента на наказание при $\xi_i \neq 0$.

Така оптимизационната задача вече може да бъде дефинирана по следния начин:

$$arg\min_{w,\xi,b}\{1/2\|w\|^2 + C\sum_{i=1}^{n}\xi_i\}$$ и $y_i(w \cdot x_i - b) \geq 1 - \xi_i, i = 1, \ldots, n$

Описаният по-горе алгоритъм за намиране на оптималната хиперравнина е линеен класификатор, но много често данните не са линейно разделими. Така през 1992 г., Bernhard E. Boser, Isabelle M. Guyon Vladimir N. Vapnik [46] предлагат създаването на нелинеен класификатор, като се използва функция на ядрото (kernel function). Полученият алгоритъм по своето същество е същият, но вече не се използва скаларно произведение, а нелинейна функция на ядрото. Това позволява на алгоритъма да намери хиперравнина, която постига най-добро разделение, в преобразувано пространство от характеристики. Преобразуването може да не бъде линейно и новото пространство да бъде от висока размерност, по този начин въпреки че класификаторът е хиперравнина в пространството с по-висока размерност, в оригиналното пространство класификаторът може да бъде нелинеен.

Едни от най-често използваните функции на ядрото са:

- Линейна – $K(x_i, x_j) = x_i \cdot x_j$
- Полиномиална – $K(x_i, x_j) = (\gamma x_i \cdot x_j + r)^d, \gamma > 0$
- RBF (Radial Basis Function) – $K(x_i, x_j) = \exp\left(-\gamma\left\|x_i - x_j\right\|^2\right), \gamma > 0$
- Sigmoid – $K(x_i, x_j) = \tanh(\gamma x_i \cdot x_j + r)$

В основата си, както споменахме още в началото, SVM моделът е двоичен, т.е. моделът разграничава две категории. За да бъде обобщен подходът и да бъде използван, когато категориите са повече от две, задачата трябва да бъде сведена до множество от двоични класификационни задачи.

Могат да бъдат построени двоични класификатори, които да различават между:

- Всяка двойка категории (one-versus-one) – в този случай се взема категорията, за която има най-много класификатори, които са я избрали.
- Една категория и всички останали (one-versus-all) – в този случай се взема категорията, за която е класификаторът с най-висок резултат.

За осъществяване на експериментите ще бъде използван node-svm – NodeJS модул [57], който стъпва върху C++ библиотека, която имплементира машини с поддържащи вектори – LIBSVM [52].

Модулът дава възможност за избор между различни имплементации на SVM. За решаване на задачата за разпознаването автор на текст ще бъде използван multiclass класификатор – CSVC (Фигура 4-1). Има няколко параметъра, които могат да бъдат настроени:

- kernelType – в експериментите по-долу се използва за функция на ядрото – RBF (Radial Basis Function). Тя може да работи върху данни, които не са линейно разделими като

проектира данните в пространство с по-висока размерност. Линейната функция на ядрото е частен случай на RBF, а също така и sigmoid функция на ядрото при определени параметри се държи, както RBF, което прави RBF често най-разумния избор. Но има ситуации, в които тя не е подходяща, особено когато броят на характеристиките е изключително голям, тогава е по-добре да бъде използвано линейното разделяне.

- gamma и C – параметри на функцията на ядрото. Няма как предварително да бъде определено кои стойности са подходящи за определена задача. За тази цел се извършва „grid-search“, т.е обхождат се различни комбинации за (C, gamma), за да се установи коя дава най-добри резултати.
- nFold – стойността на n при n-fold cross-validation, т.е на колко подмножества ще бъде разделено обучаващото множество. По-подробно ще бъде разгледана cross-validation техниката в следващата секция. Експериментите са направени с разделяне на 4 подмножества. По-точни резултати могат да бъдат постигнати с по-висока стойност на параметъра, но това води и до по-високи изисквания откъм време и ресурси.
- normalize – параметърът определя дали да бъде извършена нормализация на данните преди обучение на модела.
- reduce – параметърът определя дали да бъде извършен PCA (Principal Component Analysis), за да бъде намалена размерността на данните. В експериментите по-долу се работи с оригиналното множество от характеристики, определени от съответните алгоритми.

```
var svm = new nodesvm.CSVC({
    kernelType: nodesvm.KernelTypes.RBF,
    gamma: [0.125, 0.5, 2, 5, 10],
    C: [0.125, 0.5, 2, 5, 10],
    nFold: 4,
    normalize: true,
    reduce: false,
});
```

Фигура 4-1 Извикване на функцията за конструиране на SVM със съответните параметри

4.4 Cross-validation

Cross-validation [51] е техника за валидация на модели, използва се при прилагане на алгоритми за машинно самообучение. Използвайки cross-validation се дава възможност по-точно да се оцени доколко полученият модел дава добри и точни резултати, когато бъдат подадени нови данни.

В задачите, в които се прилагат алгоритми за машинно самообучение с учител, се използва обучаващо множество, което служи за обучение и получаване на модел, и тестово множество, което служи за оценяване на получения модел. Целта на cross-validation е да дефинира множество, което да се използва за тестово още в обучаващата фаза – валидационно множество (validation dataset), за да се предотврати прекаленото нагаждане към обучаващите данни (overfitting) и да се даде по-точна оценка за представянето на класификатора върху непознати данни.

Има различни типове cross-validation. Те могат да бъдат разделени на две основни групи – изчерпателна и неизчерпателна. Когато се използва изчерпателният подход, оригиналното

множество се разделя на обучаващо и валидационно по всички възможни начини, докато неизчерпателните не изчисляват всички разделения.

Подходът, който ще бъде използван в настоящата работа, е неизчерпателен и един от най-често използваните – k-fold cross-validation. При него обучаващото множество се разделя на k по-малки и равни по големина множества. След това за всяко едно от тях се извършва следната процедура:

- Моделът се обучава върху останалите k-1 множества
- Полученият модел се валидира и се оценява неговата точност, използвайки избраното множество

Оценката на класификатора след k-fold cross validation се изчислява като средното аритметично на стойностите, получени на всяка една от итерациите.

5. Синтактични n-грами

5.1 Теория

За решаването на много задачи при обработката на естествен език се използват традиционните n-грами. Те са последователност от елементи в реда на появата им в разглеждания текст. Като елементите могат да бъдат както конкретните думи от текста, така и отделни символи или части на речта, както и много други. Тук n обозначава броя на елементите в последователността.

Основната идея на синтактичните n-грами е те да бъдат конструирани на база на последователността, в която елементите се появяват по пътищата в синтактичното дърво, а не на база на реда, в който се появяват в текста. Най-голямото предимство на синтактичните n-грами е, че те използват синтактичните връзки между думите.

Първите, които прилагат синтактичните n-грами за решаването на задачата за разпознаване автор на текст, са Grigori Sidorov и Efstathios Stamatatos [47].

Анализите, които те правят, показват, че подходът дава по-добри резултати от обикновените n-грами, независимо дали те са символни, съставени от думи или от съответстващите части на речта. Корпусът обаче, който е използван за експериментите е малък – 39 документа от трима автори. Точността на класификацията при повечето експерименти е над 90%, което е очаквано предвид малкия брой автори и достатъчното текстове за всеки един от тях. Резултатите показват, че точността е най-голяма, когато се използват 2-грами и 3-грами. Броят на n-грамите също оказва влияние. За конкретния корпус точността е по-добра, когато броят на n-грами е по-голям, но ако бъде избран прекалено голям брой n-грами е възможно да се стигне до прекалено нагаждане към обучаващите данни.

Синтактичните n-грами дават възможност да бъде премахнато влиянието на някои от специфичните за езика конструкции, които пречат да бъде ефективно оценено сходството между текстове. Например, правилото, прилагателното да се поставя преди съществителното, ще попречи, ако се използват обикновени n-грами, да бъде открито възможно съществуващо сходство на изреченията. Това важи и за подчинените изречения, както и за още други конструкции в езика.

Нека разгледаме следните две фрази на английски език: „eat with wooden spoon“ и „eat with metallic spoon“. Двете фрази нямат общи обикновени 3-грами, но ако използваме синтактични 3-грами, ще получим „eat with spoon“ като обща и за двете.

Важно е, че синтактичните n-грами имат реално лингвистично (езиково) тълкуване, защото те отразяват реални синтактични връзки, което е изключително ценно при разпознаване автор на текст.

Предимството на синтактичните n-грами, n-грами, които са конструирани, използвайки пътищата в синтактичните дървета, е, че те са по-малко произволни от обикновените n-грами. Техният брой често е по-малък от този на обикновените n-грами. Синтактичните n-грами могат да бъдат лингвистично тълкувани, докато традиционните n-грами са по-скоро статистически характеристики. Именно опитът да се включи езикова информация в статистически базиран метод прави синтактичните n-грами полезни. Така се премахва един от основните недостатъци на традиционните n-грами – много елементи, които не носят ценна информация, което води до голям шум в извлечените характеристики.

Основният проблем, когато се използват синтактични n-грами, е необходимостта от извършване на синтактичен анализ на текстовете, което в повече случаи е задача, изискваща доста време и ресурси. Също така въпреки огромния напредък в Обработката на естествен език, все още точността при такъв тип анализи не винаги е удовлетворителна, особено когато става дума за езици, различни от английски.

Има опити недостатъците, които имат традиционните n-грами, да бъдат преодолени, като се използват изцяло статистически подходи. Такива примери са skip-грамите и Maximal Frequent Sequences (MFS) [48].

Skip-грамите са сходни на традиционните n-грами. Разликата е в това, че някои елементи, докато се конструира последователността, се пропускат. Стъпката, с която се пропускат елементи, може да бъде различна и да се установи експериментално. Това е опит да се премахне част от шума, като се добавят произволни отклонения в текста. Проблемът със skip-грамите е, че техният брой също нараства много бързо.

Maximal Frequent Sequences са skip-грами с голяма честота, т.е. вземат се под внимание само skip-грами, чиято честота е над предварително зададена граница. Проблемът с MFS е, че за да бъдат конструирани, е нужно използването на сложни алгоритми, чието прилагане изисква значително време. Друг недостатък е, че отново, за разлика от синтактичните n-грами, са зависими от колекцията от текстове и не подлежат на езиково тълкуване.

Могат да бъдат изградени различни типове синтактични n-грами в зависимост от елементите, които се използват за тяхното получаване.

Елементи на синтактичните n-грами могат да бъдат:

- Конкретните думи в документа
- Частите на речта, които съответстват на думите
- Синтактичните връзки
- Комбинация от изброените по-горе

Както и в статията, в която синтактичните n-грами за първи път са използвани за разпознаване автор на текст [47], и в настоящата разработка, ще бъдат използвани синтактичните връзки за

конструиране на n-грамите. За да бъдат извлечени синтактичните връзки, всички текстове от изследваните корпуси са обработени от Stanford Parser [53].

Stanford Parser е анализатор на естествен език. Това е софтуерна система, която има за цел да извлече граматичната структура на изреченията. Например, кои групи от думи са свързани и образуват фрази, определяне частите на речта, частите на изречението.

Вероятностните анализатори използват знания за езика, които са придобити от ръчно анализирани от хора изречения, за да се опитат да произведат най-вероятния анализ за новите изречения. Тези вероятностни анализатори все още допускат грешки, но средната им производителност се е повишила значително с годините. Тяхното развитие е едно от големите постижения в обработката на естествен език през 90-те години.

Статистическият анализатор, разработен от Университета в Станфорд, е Java имплементация на вероятностни анализатори на естествен език – PCFG (Probabilistic Context Free Grammars) и Lexicalized Dependency Parsers, както и Lexicalized PCFG Parser.

При разработката на алгоритъма за разпознаване автор на текст, който се основава на синтактични n-грами, е необходимо всички текстове, които са част от корпуса, да бъдат обработени и да се генерират дървета на зависимостите (dependency trees), които ще бъдат използвани при конструирането на синтактични n-грами.

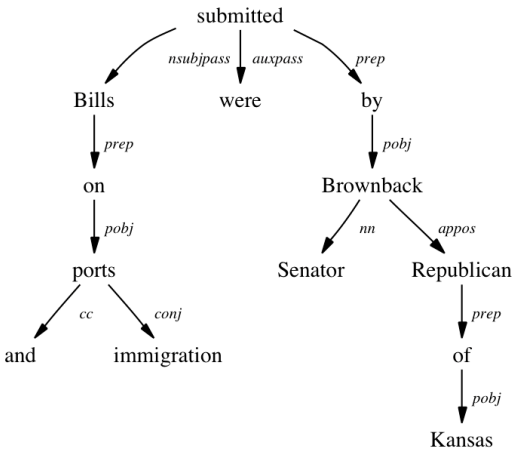
За тази цел в настоящата разработка ще бъде използван модулът Stanford Dependencies [53]. Stanford Dependencies осигурява представяне на граматическите връзки между думите в изречението. Всяка една зависимост се представя по следния начин – име на връзката, управляващ (governor), подчинен (dependent).

Например, стандартните зависимости за изречението

Bills on ports and immigration were submitted by Senator Brownback, Republican of Kansas.

са:

nsubjpass(submitted, Bills)
auxpass(submitted, were)
agent(submitted, Brownback)
nn(Brownback, Senator)
appos(Brownback, Republican)
prep_of(Republican, Kansas)
prep_on(Bills, ports)
conj_and(ports, immigration)
prep_on(Bills, immigration)



Фигура 5-1 Dependency tree [47]

Настоящата имплементация на Stanford Dependencies [53] различава приблизително 50 различни граматически връзки. Всички зависимости са двоични – граматическите връзки са между управляващ и подчинен. Дефинираните граматически връзки са в йерархия. Най-общата граматическа връзка – dep (dependent) – се използва единствено, когато не съществува в йерархията по-точна или системата не успява да извлече по-точна.

5.2 Реализация

Първата стъпка при разработката на алгоритъма, базиран на синтактични n-грами, е да се направи синтактичен анализ на корпусите от данни, от който ще бъде получено дърво на зависимостите (dependency tree) за всяко едно от изреченията в корпуса. Както споменахме, това е задача, която отнема доста време и ресурси, особено когато корпусите от данни са големи. Поради тази причина и трите корпуса, които ще бъдат използвани, са обработени, използвайки Stanford Parser, предварително. За всеки текст от съответния корпус се създава отделен файл, който съдържа дърво на зависимостите за всяко едно от изреченията в него. По-долу на Фигура 5-2 е съдържанието на bash script файла, който се грижи това да се случи, като стартира java програмата, която извършва обработката на всеки един от файловете.

```
for filepath in $(find .)
do
    filename=$(basename $filepath)
    newfilename="../trees/$filename"

    java -cp "../../stanford-parser.jar"
         -mx4g edu.stanford.nlp.parser.lexparser.LexicalizedParser
         -retainTmpSubcategories -outputFormat "typedDependencies"
         -outputFormatOptions "basicDependencies" "../../englishPCFG.ser.gz"
         $filepath > $newfilename
done
```

Фигура 5-2 Bash script за генериране на файловете, съдържащи обработените с Stanford Parser документи

След като за всеки един от документите в корпуса са построени синтактичните дървета е необходимо да бъдат извлечени синтактичните n-грами от тях. За да бъдат получени всички синтактични n-грами с определена дължина в дадено изречение, което е представено от синтактично дърво, трябва да бъдат намерени всички пътища с тази дължина с начало всеки един от възлите в дървото.

За тази цел е написана помощна функция (Фигура 5-3), на която се подава възелът, от който да започват пътищата, дълбочината и дървото, в което ще се извърши търсенето. Функцията извършва търсене в дълбочина и връща всички пътища от начално зададения възел с определената дължина. В пътищата се пазят не конкретно използваните думи в текста, а граматическите връзки между думите, които сме получили след прилагането на Stanford Parser.

По този начин влиянието на темата на текста става по-малко и е възможно да се определи авторовият стил, като се залага на граматичните конструкции, които авторът използва.

Така се обработват всички изречения във всяка една от творбите. След като бъдат намерени синтактичните n-грами е необходимо да бъда изчислена тяхната честота във всеки един от документите.

Едновременно с това се пресмята честота на синтактичните n-грами и на база целия корпус. Така, след като бъде обработен целият корпус, се избират за характеристики на текстовете само най-често използваните в целия корпус. Големината на това множество се конфигурира, за да може да се анализира по какъв начин влияе на получените резултати. В експериментите, направени в тази разработка, използваните стойности за големина на множеството от характеристики са 200, 400, 700, 1000, 1500.

След като бъдат открити най-често използваните в корпуса синтактични n-грами, всеки един от документите се представя с вектор от съответните честоти на избраните синтактични n-грами в конкретния документ.

Възможно е някои от синтактични n-грами да присъстват само в част от текстовете. В този случай те влизат в множеството от характеристики на останалите текстове с честота 0.

```
var DFS = function(node, depth, path, tree){
    var paths = [];
    if(depth == 0){
        paths.push(path);
        return paths;
    }

    var children = tree[node];
    if(children){

        for(var i=0; i<children.length;i++){
            if(children[i])
            {
                var newPath = path.concat((children[i])[1] + " ");
                var shorterPaths = DFS((children[i])[0], depth-1, newPath, tree);

                if(shorterPaths) {
                    paths.push.apply(paths, shorterPaths);
                }
            }
        }
    }
    return paths;
};
```

Фигура 5-3 Имплементация на Depth First Search за генериране на синтактични n-грами

6. Разпознаване автор на текст, анализирайки разпределението на думите в текста на база честотата им в естествения език

6.1 Теория

Както повечето подходи при разпознаване автор на текст, и този разчита, че всеки автор притежава присъщ единствено за него стил на писане. Всеки един от алгоритмите търси характеристиките на текста, които най-добре биха го описали.

Разглежданият по-долу подход има за цел да надгради един добре известен такъв, който се основава на честотата на думите в текста – Delta подходът на Burrows.

Идеята, която описват Chen, Huang, Yang, Meng и Miao [28], има за цел да раздели думите в текста на групи и да се анализира разпределението на групите от думи в текста.

Този подход анализира всички думи, които авторът е използвал в даден текст. Думите се разпределят в зони на база на честота им в естествения език. Методът разчита, че всяка дума в разглеждания текст допринася за авторовия стил, независимо дали думата е функционална или контекстна такава.

Всяка една зона се описва с две характеристики, които ще бъдат описани по-долу. Целият текст се представя с множеството от характеристиките на всяка една от зоните. Те имат за цел да опишат авторовия стил.

Разпределението на думите се описва на три нива:

1. Богатство на лексика
2. Честота на срещане на всяка от зоните в текста
3. Места на поява на всяка от зоните в текста

Първите две нива са широко разпространени, но до този момент в познатите методи не се взема под внимание къде в текста се появява дадена дума.

В обобщение подходът се отличава от останалите по няколко важни точки:

- Всички думи от текста биват използвани при анализа
- При групирането на думите в зони се използва нормалната честота на срещане на думите в езика, което прави алгоритъма по-малко чувствителен към съдържанието и се фокусира към стиловите характеристики
- Освен честотата на думите в текста се използва и информация за това къде те се появяват в него

Анализът на текста се осъществява в четири стъпки (Фигура 6-1):

1. Представяне на позицията на всяка дума в текста

   Текстът се представя като поредица от думи, като не се взема под внимание пунктуацията.

$$T = \{w_0, w_1, w_2, \dots, w_{n-1}\}$$

w_i $(0 \leq i \leq n-1)$ е $(i+1)$-вата поред дума в текста. Дължината на текста е n. За да се представи позицията, на която всяка една дума се появява в текста, се използва индексът на думата. Индексът се нормализира с дължината на текста, за да бъде премахнато влиянието на текстове с различна дължина.

Позицията на w_i в текста T се дефинира, както следва:

$$l_i = \frac{i}{n}, 0 \leq i \leq n-1$$

2. Групиране на думите в зони

На тази стъпка думите се групират в зони на база на нормалната честота на срещането им.

Нормална честота на срещане на дума - средната честота на срещане на думата в текстове на естествен език. Тази честота може да бъде оценена, като се пресметне честотата на срещане на думата в достатъчно голям и представителен корпус от текстове.

Зона от думи със сходна нормална честота – група от думи, които имат сходна нормална честота. Определянето на думите като сходни по нормална честота може да стане, използвайки различни функции. Някои от тях ще бъдат описани по-долу.

Нека означим с F – множеството от нормалните честоти на срещане на думите от текста T.

$$F = \{f(w_0), f(w_1), f(w_2), \dots, f(w_{n-1})\},$$

където функцията $y = f(x)$ съпоставя на дума x нейната нормална честота – y.

$$Z_k = \xi(T, f(x), f_{max}), k = 1,2, \dots, K(f_{max})$$

където ξ е функцията, избрана за да се осъществи разделението на зони, k е броят на зоните, f_{max} е максималната стойност за нормална честота, която се среща в текста.

Възможни функции за разделяне на думите на зони по сходството на тяхната нормална честота са линейно разделение, базово разделение (radix), логаритмично разделение.

Линейно разделение – при това разделение всички зони са с еднакъв размер – L – параметър на функцията.

$$Z_k = \left\{ w \middle| k = \left\lfloor \frac{f(w)}{L} \right\rfloor, w \in T \right\}, k = 0,1,2, \dots, K-1;\ K = \left\lfloor \frac{f_{max}}{L} \right\rfloor + 1$$

Базово разделение (Radix) – при това разделение големината на зоните е различна в зависимост от нормалната честата. Първите R зони са с базовата големина L, следващите R зони са с големина RL, следващите R зони с големина R^2L и т.н.

$$Z_k = \{ w \in T | k = \begin{cases} B, & B < R \\ (R-1) * E + \left\lfloor \frac{B}{R^E} \right\rfloor, & B \geq R \end{cases} \}$$

където L е базовата големина на зона, R е основата ($R > 1$ и $R \in \mathbb{N}$), $E = \lfloor \log_R B \rfloor$ е максималната експонента, а $B = \lfloor f(w)/L \rfloor$ е базовата честота.

Логаритмично разделение –

$$Z_k = \{ w \in T | k = \begin{cases} 0, & f(w) = 0 \\ \lfloor \log_r f(w) \rfloor, & f(w) > 0 \end{cases} \}$$

където r е основата на логаритъма, $r > 1$ и $r \in \mathbb{R}$.

3. Представяне на зона от думи със сходна честота

Текстът може да бъде разглеждан като поредица от думи, които принадлежат на различни зони. Ако предположим, че текстът T съдържа думи от зона Z_k n_k пъти, то $\sum_{k=o}^{K-1} n_k = n$.

Текстът T може да бъде представен по следния начин:

$$T = \langle Z, L \rangle,$$

където
$Z = \{Z_k | 0 \leq k < K\}$ – множество от зоните на нормална честота
$L = \{L(Z_k) | 0 \leq 0 < K\}$, където $L(Z_k) = \{l_{k0}, l_{k1}, l_{k2}, \dots, l_{kn_k-1}\}$ – множество от позициите на думите в T, които попадат в зона Z_k.

4. Пресмятане на стила на текста

Разстояние между w_i и w_j дефинираме така:

$$d_{ij} = d_{ji} = |l_i - l_j|$$

Occurrence Distance Expectation – очакването на разстоянието между появите на думите има за цел да представи честотата на появяване на съответната зона в текста.

$$\alpha = \frac{1}{n_k + 1} \sum_{i=0}^{n_k} d_{i,i-1}^{(k)} = \frac{1}{n_k + 1} (1 - 0) = \frac{1}{n_k + 1}$$

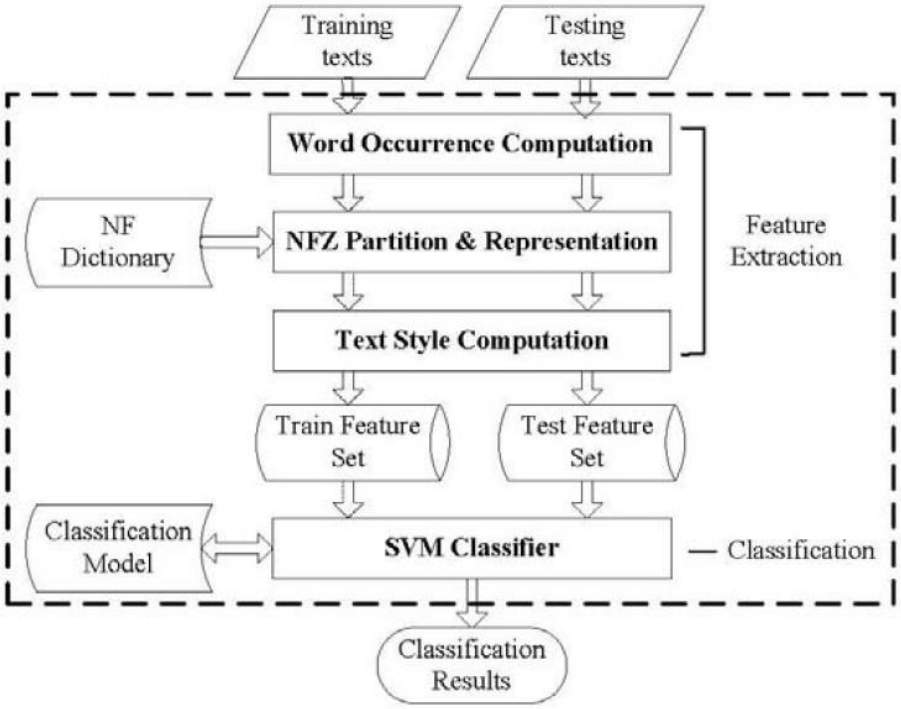
Occurrence Distance Variance – дисперсията на разстоянието между появите на думите изобразява разпределението на появите на думите от зоната в текста.

$$\gamma = 1/\alpha_k \sqrt{\frac{1}{n_k + 1} \sum_{i=0}^{n_k} (d_{i,i-1}^{(k)} - \alpha_k)^2}$$

След като комбинираме стойностите за очакването и дисперсията за всяка една от зоните получава вектор с характеристики, описващ стила на автора:

$$\Gamma = \{(\alpha_k, \gamma_k) | 0 \leq k < K\}$$

Върху получените характеристики вече може да бъдат приложени SVM, които ще обучат модел, който да класифицира нови данни.



Фигура 6-1 Схема на задачата за разпознаване автор на текст, използвайки NFZ WD подхода [28]

6.2 Реализация

Върху всеки един от документите в корпуса се изпълнява функция (Фигура 6-2), която да конструира множеството от характеристики.

Подзадачите, които функцията изпълнява:

- Извиква функция, която да представи текста като масив от думи (tokenize). За тази цел се използва готов NodeJS пакет [58].
- Извиква функция, която за всяка една от думите в текста намира нейната честота в естествения език. Речникът, който се използва за извличане на естествената честота на думите, е BNC (British National Corpus) [54].
- Извиква функция, която да осъществи разделянето на зони на база извлечените честоти и функцията на разделение (Фигура 6-3)
- Пресмята за всяка една от получените зони очакването и дисперсията, които ще съставляват множеството от характеристики на текста.

Някои зони ще бъдат представени само в част от текстовете в корпуса и е важно тези зони да бъдат добавени във вектора с характеристики и на останалите текстове със стойности за очакване и дисперсия, равни на 0, за да може обучението, което ще бъде извършено на база характеристичните вектори, да бъде коректно.

```
var constructFeaturesSet = function(filename, options, done){
    var partitionFunction = getPartitionFunction(options['partitionFunction']);

    fs.readFile(filename, "utf8", function(err, data){

        if(!err) {

            // tokenize the text
            var tokenizedText = nlpAlg.tokenizeText(data);

            // retrieve NF for each word in the text
            constructNaturalFrequencyArray(tokenizedText,
            function(wordFrequencies){

               // NFZ partition
               var partition = NFZPartition(tokenizedText, wordFrequencies,
                                            partitionFunction);
               var naturalFrequencyZones = partition[0];
               var wordNormalizedOccurrences = partition[1];

                // Text style computation
                var textFeatures = {};

                for(var key in naturalFrequencyZones){
                    if(naturalFrequencyZones.hasOwnProperty(key))
                    {
                        var expectation =
                        calculateOccurrenceDistanceExpectation(
                        naturalFrequencyZones[key]);

                        var variance =
                        calculateOccurrenceDistanceVariance(
                        naturalFrequencyZones[key],
                        wordNormalizedOccurrences[key]);

                        textFeatures[key] = [expectation, variance];
                    }
                }

                done(null, textFeatures);
            });
        }
        else{
            done(err, null);
        }
    });
};
```

Фигура 6-2 Функция за извличане на множеството от характеристики на база честотата на думите в естествения език

```
// Represents the tokenized text with the arrays of natural frequency zones and the
words occurrences
var NFZPartition = function(tokenizedText, wordFrequencies, partitionFunction){

    var naturalFrequencyZones = [];
    var wordNormalizedOccurrences = [];

    // NFZ Partition
    for(var i=0; i<tokenizedText.length;i++){
        var currentWord = tokenizedText[i];

        var k = partitionFunction(wordFrequencies[i]);

        if(!naturalFrequencyZones[k]){
            naturalFrequencyZones[k] = [];
        }
        naturalFrequencyZones[k].push(currentWord);

        if(!wordNormalizedOccurrences[k]){
            wordNormalizedOccurrences[k] = [];
        }
        wordNormalizedOccurrences[k].push(i/tokenizedText.length);
    }

    return [naturalFrequencyZones, wordNormalizedOccurrences];
};
```

Фигура 6-3 Функция за разпределяне на думите в зони

7. Резултати от експериментите

В настоящата глава ще бъдат представени и анализирани резултатите от направените експерименти с описаните по-горе алгоритми и корпуси от данни.

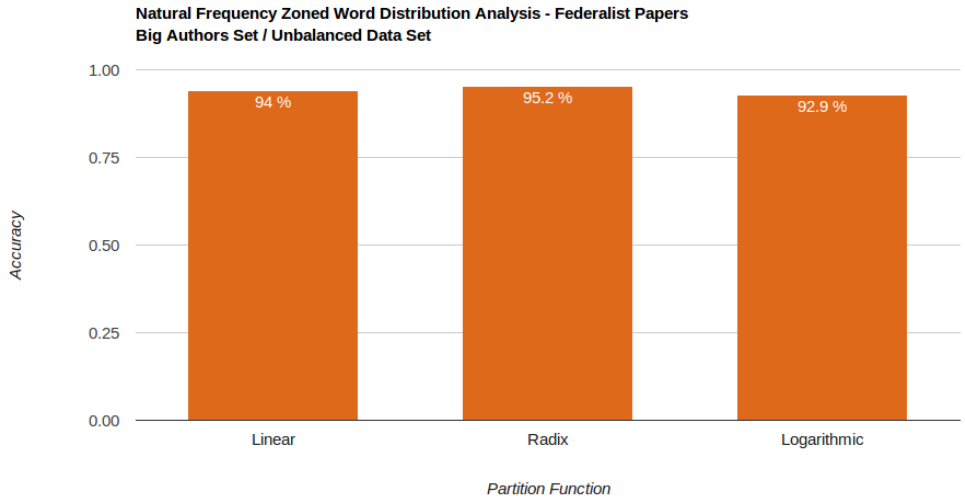
7.1 Federalist Papers

Корпусът, който се използва за базов и такъв, който да покаже, че по-задълбоченото разглеждане на използваните алгоритмите е обосновано, е колекцията от т.нар. Federalist Papers. Най-добрият досега резултат върху този корпус е от 2013 година [52] – 97.1% като за постигането му се използват множество от най-често срещани функционални думи и машини с поддържащи вектори.

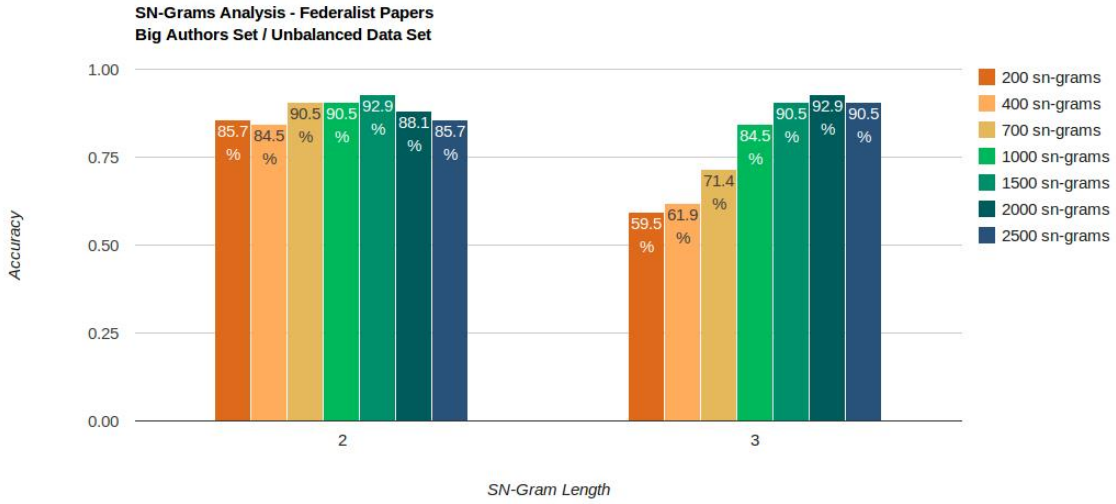
Алгоритмите, описани в настоящата работа, показват точност над 90%. Точността при направените експерименти е най-висока, когато се използва RADIX функцията за разделяне на зони в алгоритъма, базиран на нормалната честота на думите в естествения език - 95.2%. Резултатите получени, когато се приложат другите две функции за разделение са съпоставими с най-високия получен резултат, разликата е от порядъка на няколко процента (Фигура 7-1).

Алгоритъмът, базиран на синтактичните n-грами, дава малко по-ниска точност. Най-високата точност, която се постига от синтактичните n-грами, е когато се използват 2-грами и големината на множеството от синтактични n-грами е 1 500 – 92.9%. Ясно се вижда от графиката (Фигура 7-2), че когато се използват по-дълги синтактични n-грами е необходимо да се използват и по-голям брой от тях, за да се постигне по-добра точност. Причината за това е отчасти фактът, че когато се използват по-дълги n-грами, характеристичните вектори стават по-разредени, т.е. има много нулеви честоти в тях, което прави обучението на класификатор по-трудно.

Експериментите върху Federalist Papers потвърждават, че по-нататъшно разглеждане и на двата подхода върху други корпуси е разумна стъпка.



Фигура 7-1: Резултати върху корпуса от Federalist Papers, използвайки NFZ WD.



Фигура 7-2 Резултати върху корпуса от Federalist Papers, използвайки синтактични n-грами

7.2 Английски романи (The Corpus of English Novels)

Експериментите върху корпуса от английски романи имат за цел да проследят по какъв начин повлиява различният брой на изследвани автори, както и промяната на броя налични текстове за всеки от тях върху получените резултати.

По-долу са графиките, които показват резултатите от експериментите върху корпуса от английски романи в неговите различни конфигурации за големина на множеството от автори и текстове (Фигура 7-1 – Фигура 7-20).

Ако има малко количество от данни за авторите и същевременно се увеличава техният брой, точността спада, но е важно да се отбележи, че спадането на точността с увеличаване броя на авторите е по-бавно, от това при намаляване на данните за автор. Въпреки че при увеличаване броя на авторите в корпуса е по-трудно да се обучи точен класификатор, тъй като трябва да се намерят характеристиките, които ще разграничат добре всеки двама от тях, по-ключов фактор се оказва количеството от текстове за всеки от авторите.

Интересно наблюдение е, че точността на алгоритмите не се повлиява значително от това дали броят на текстове за всеки един от разглежданите автори е еднакъв или не, дори в повечето случаи, когато бъдат използвани всички текстове от даден автор, получените резултати са по-добри (Фигура 7-6, Фигура 7-8). Възможна причина за това е, че корпусът от данни не е контролиран относно тема и жанр, което прави правилната класификация при намаляване на количеството от текстове още по-трудна.

За разлика от резултатите върху Federalist Papers, експериментите върху корпуса от английски романи показват, че върху повечето конфигурации на корпуса, по-добри резултати, макар и не с голяма разлика, дава алгоритъмът, използващ синтактични n-грами.

В експериментите върху английските романи се наблюдава, че точността при използване на синтактични n-грами с дължина 3 спада драстично (Фигура 7-10, Фигура 7-12, Фигура 7-16, Фигура 7-20). По-късно същият ефект ще се наблюдава и при експериментите, направени с блог постове. Възможна причина за това може да бъде фактът, че характеристичните вектори, които се получат на база 3-грами и то върху корпус, който не е контролиран относно тема, са много по-разредени.

Влиянието на броя синтактични n-грами, които се използват в експериментите, отново е по-голямо, когато се използват n-грами с дължина 3. За тези с дължина 2, той оказва по-малко влияние, както се наблюдава и в експериментите върху Federalist Papers.

Най-добри резултати при използването на алгоритъма, базиран на честотата на думите в естествения език, дава логаритмичната функция за разделяне на зони, независимо от конфигурациите на корпуса (Фигура 7-3, Фигура 7-7, Фигура 7-9, Фигура 7-11, Фигура 7-17, Фигура 7-19).

Глобално най-добри са резултатите върху този корпус, използвайки синтактичните n-грами, когато множеството от автори е малко и се използват всички текстове за авторите – 100% (Фигура 7-7, Фигура 7-8). Най-ниски са резултатите, когато множеството от автори е голямо и множеството от текстове е ограничено – 52% (Фигура 7-16, Фигура 7-17).

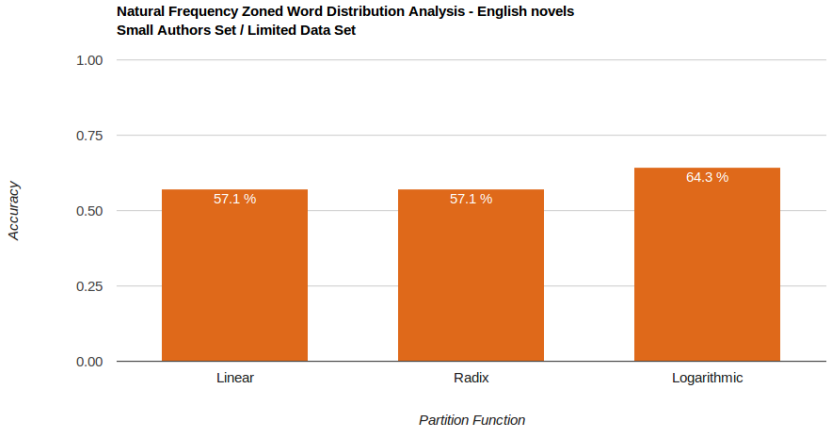
Експериментите на Abbasi & Chen [35] върху множеството от текстове, взети от Java форум с над 40 000 думи на автор при 25 автора, което е съпоставимо като корпус с настоящия разглеждан,

показват по-добри резултати - 88%. Възможни причини за това е по-голямото количество от използвани характеристики – няколко десетки хиляди. От графиките (Фигура 7-6, Фигура 7-18) се вижда, че точността, когато се използват по-голям брой синтактични n-грами, се увеличава, но нещо, с което трябва да се внимава при тяхното увеличаване, е да не се стигне до прекалено нагаждане към тренировъчните данни (overfitting).

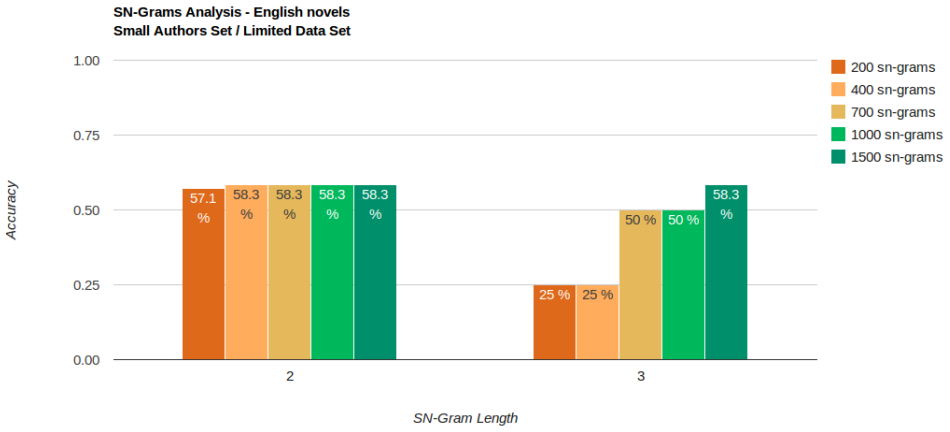
Следваща стъпка при изследването на алгоритъма, използващ синтактични n-грами, при наличието на повече ресурси, е тестването на по-разнообразно множество от конфигурации за дължината на синтактичните n-грами, както и големината на изследваното множество от синтактични n-грами.

Друга причина за по-ниските резултати е и фактът, че корпусът от английски романи не е контролиран относно тематика, за разлика от корпуса, използван от Abbasi & Chen. Както и изследванията, направени от самите Abbasi & Chen, показват – голямо влияние върху получените резултати оказва не само големината на корпуса, а и доколко разнообразен е относно тематика и жанрове.

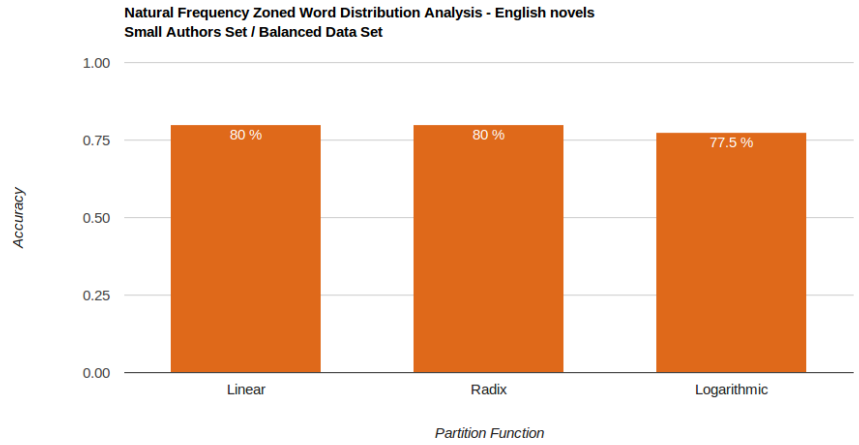
В обобщение, получените резултати потвърждават интуитивното и показаното в изследванията, направени до този момент [17, 32, 35, 36]. Точността и на двата подхода намалява с увеличаване броя на авторите и с намаляване на текстовете за всеки от тях. Но за разлика от изследванията, направени от Zhao & Zobel [32], които показват спад от над 20%, когато множеството от автори се увеличи от двама на петима, направените в настоящата разработка експерименти показват, че избраните два подхода осигуряват относителна стабилност и по-плавен спад в точността – до спад от 20% се стига при увеличаването от 5 разглеждани автора на 15 автора.



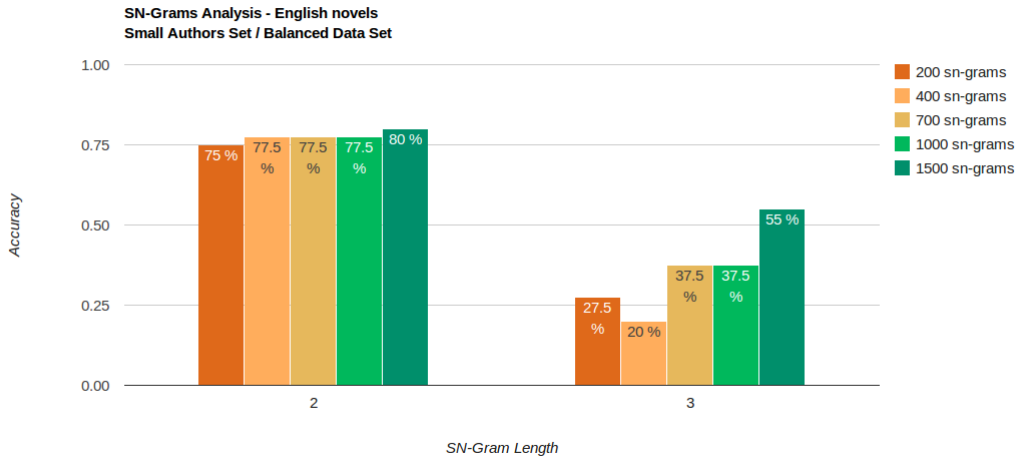
Фигура 7-3 Резултати върху корпуса от английски романи, използвайки NFZ WD – Small Authors Set и Limited Data



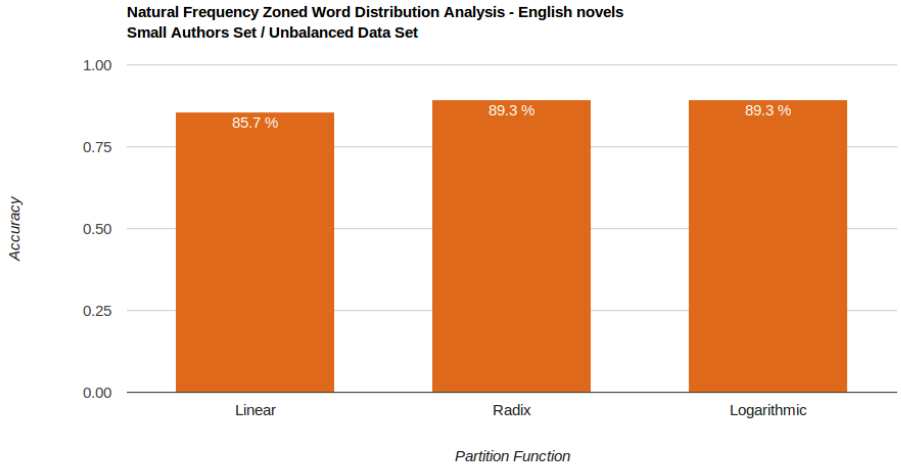
Фигура 7-4 Резултати върху корпуса от английски романи, използвайки синтактични n-грами – Small Authors Set и Limited Data Set



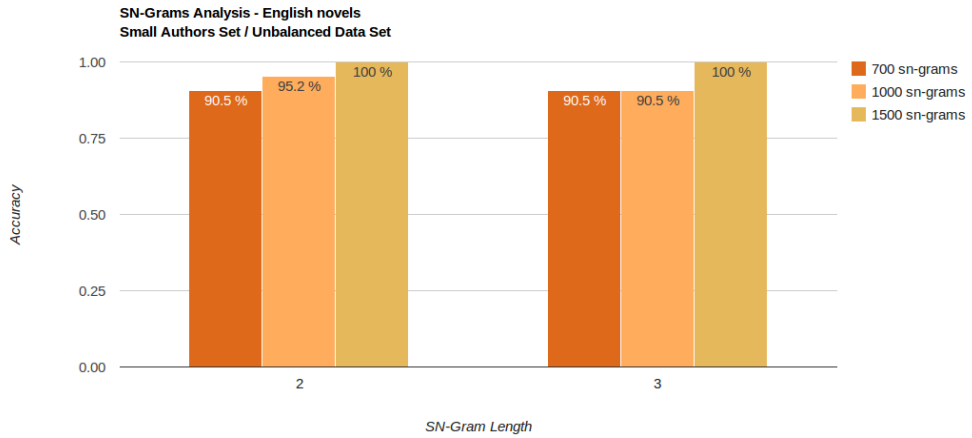
Фигура 7-5 Резултати върху корпуса от английски романи, използвайки NFZ WD – Small Authors Set и Balanced Data Set



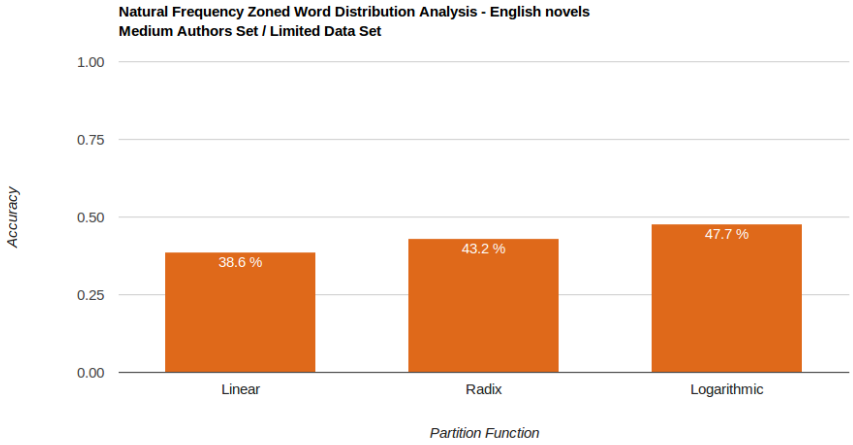
Фигура 7-6 Резултати върху корпуса от английски романи, използвайки синтактични n-грами – Small Authors Set и Balanced Data Set



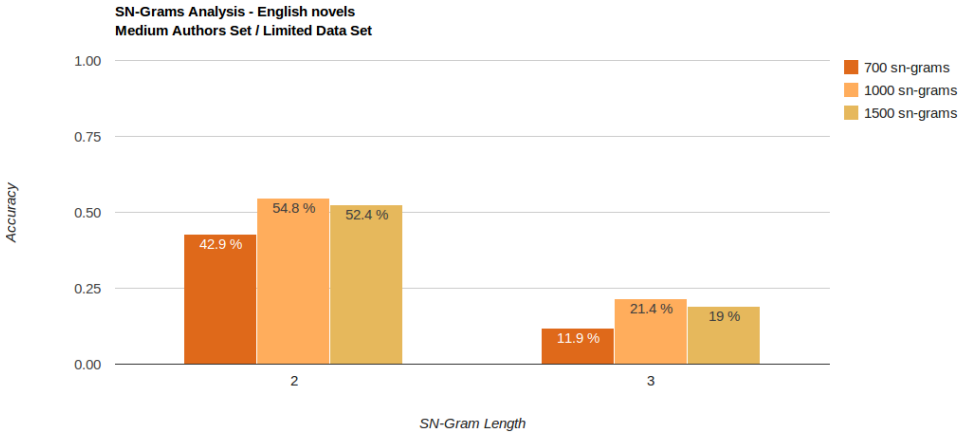
Фигура 7-7 Резултати върху корпуса от английски романи, използвайки NFZ-WD – Small Authors Set и Unbalanced Data Set



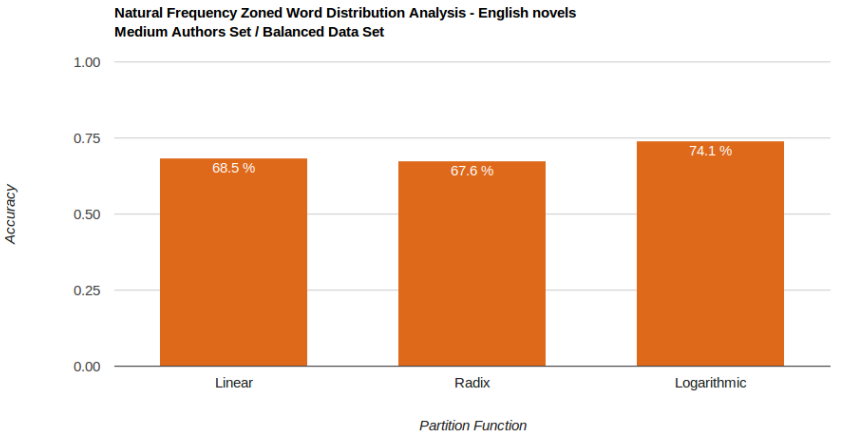
Фигура 7-8 Резултати върху корпуса от английски романи, използвайки синтактични n-грами – Small Authors Set и Unbalanced Data Set



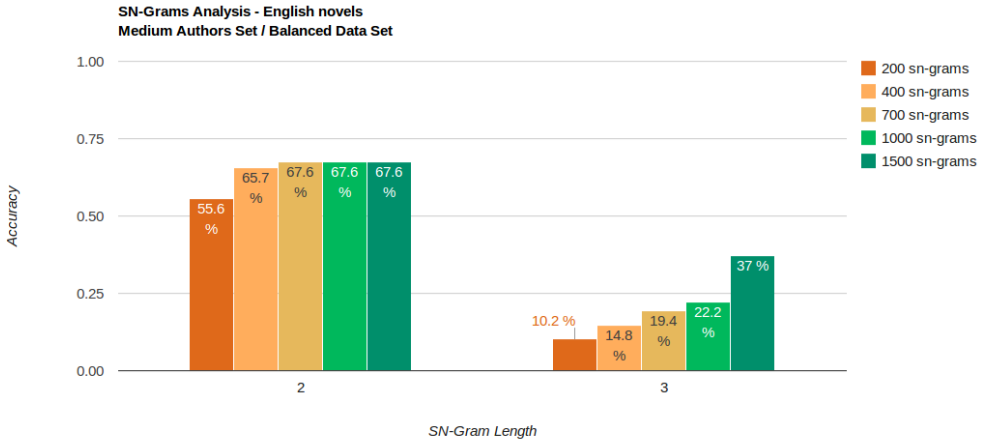
Фигура 7-9 Резултати върху корпуса от английски романи, използвайки NFZ-WD – Medium Authors Set и Limited Data Set



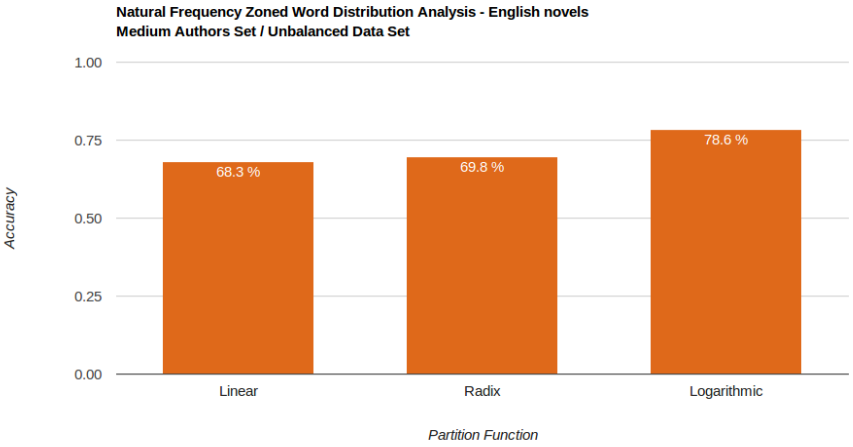
Фигура 7-10 Резултати върху корпуса от английски романи, използвайки синтактични n-грами – Medium Authors Set и Limited Data Set



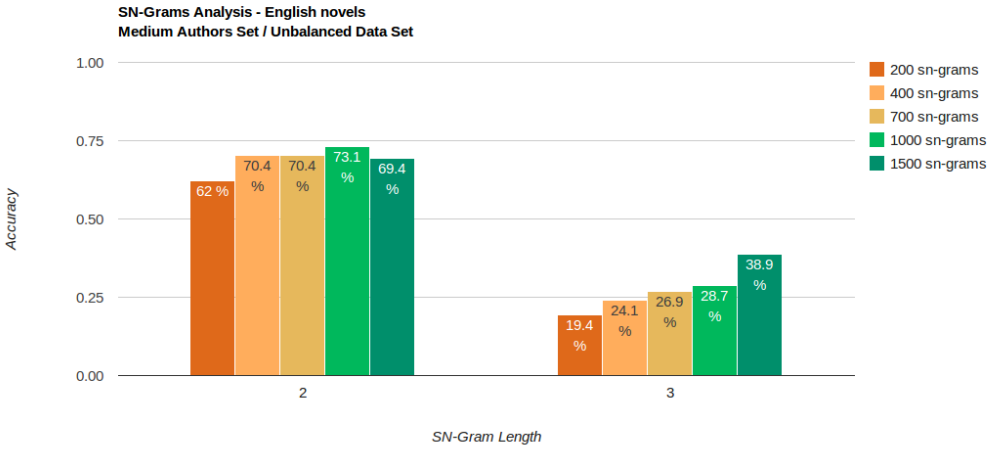
Фигура 7-11 Резултати върху корпуса от английски романи, използвайки NFZ-WD – Medium Authors Set и Balanced Data Set



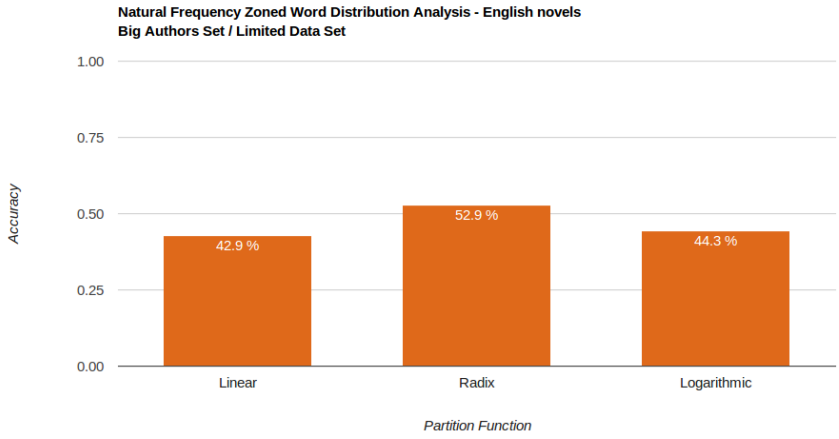
Фигура 7-12 Резултати върху корпуса от английски романи, използвайки синтактични n-грами – Medium Authors Set и Balanced Data Set



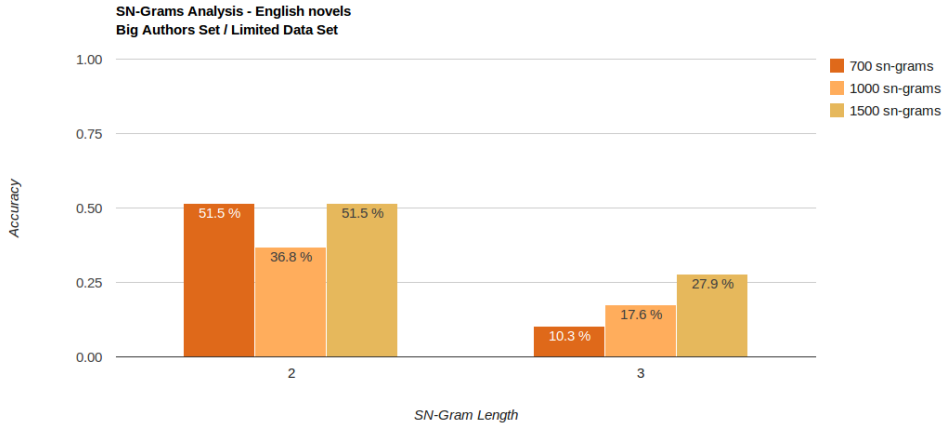
Фигура 7-13 Резултати върху корпуса от английски романи, използвайки NFZ-WD – Medium Authors Set и Unbalanced Data Set



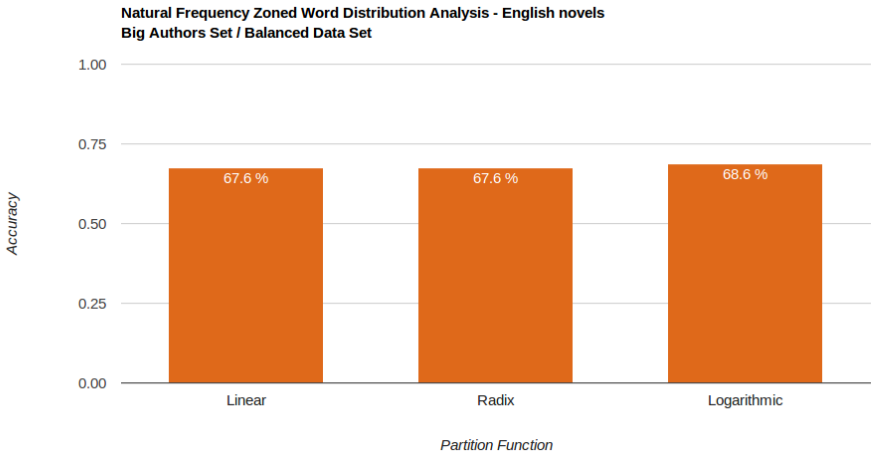
Фигура 7-14 Резултати върху корпуса от английски романи, използвайки синтактични n-грами – Medium Authors Set и Unbalanced Data Set



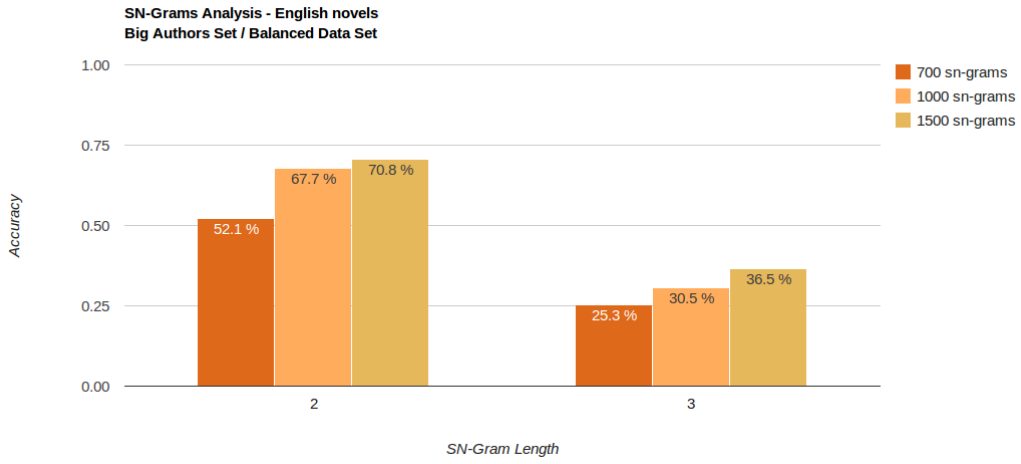
Фигура 7-15 Резултати върху корпуса от английски романи, използвайки NFZ-WD – Big Authors Set и Limited Data Set



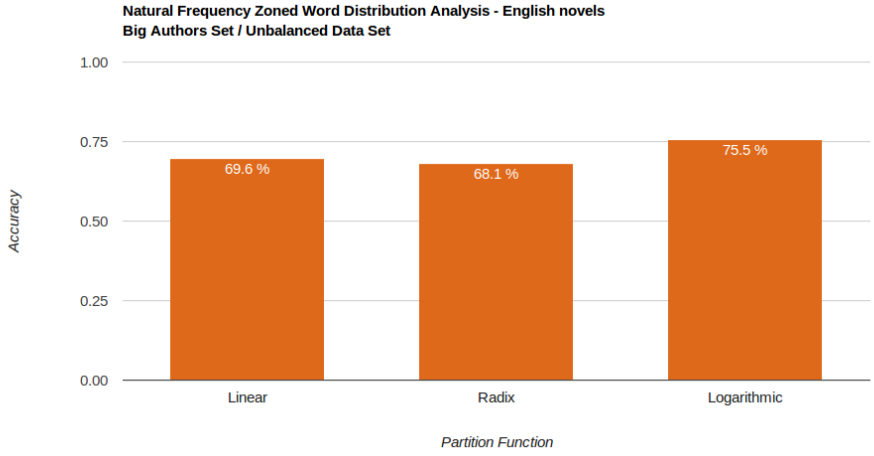
Фигура 7-16 Резултати върху корпуса от английски романи, използвайки синтактични n-грами – Big Authors Set и Limited Data Set



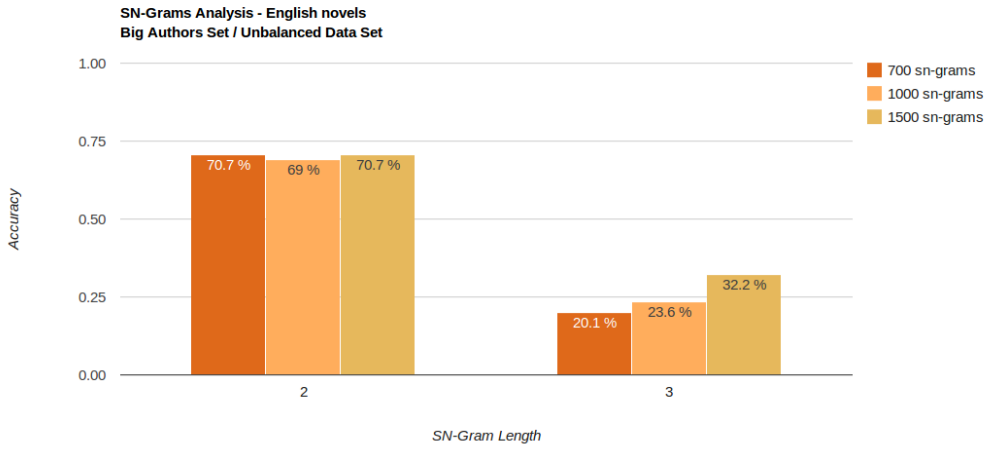
Фигура 7-17 Резултати върху корпуса от английски романи, използвайки NFZ-WD – Big Authors Set и Balanced Data Set



Фигура 7-18 Резултати върху корпуса от английски романи, използвайки синтактични n-грами – Big Authors Set и Balanced Data Set



Фигура 7-19 Резултати върху корпуса от английски романи, използвайки NFZ-WD - Big Authors Set и Unbalanced Data Set



Фигура 7-20 Резултати върху корпуса от английски романи, използвайки синтактични n-грами - Big Authors Set и Unbalanced Data Set

7.3 Блог постове (The Blog Authorship Corpus)

Резултатите, получени от експериментите върху корпуса от блог постове, потвърждават голямото влияние, което имат големината на множеството от автори и текстовете за всеки един от тях. При това наблюдаваните тенденции са сходни с получените върху корпуса от романи.

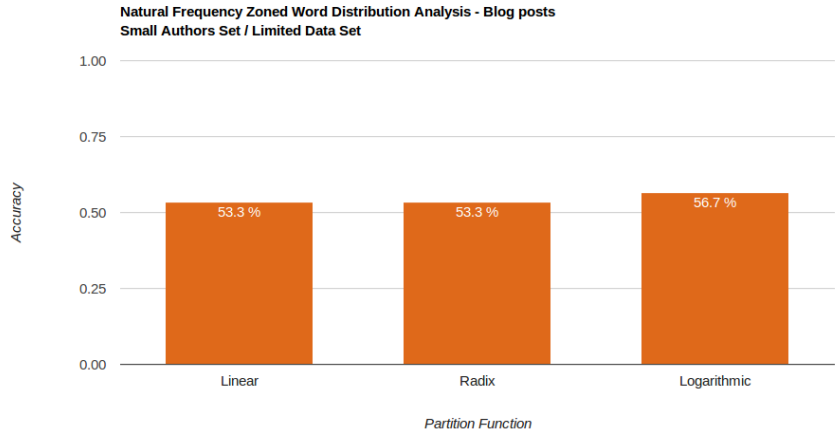
Важно наблюдение е, че в повечето експерименти върху корпуса от блогове, точността на алгоритъма, базиран на честотата на думите в естествения език, е по-висока от тази на алгоритъма, използващ синтактични n-грами. Логаритмичната функция на разделение отново се показва като най-надеждна.

Вероятна причина за по-добрите резултати на алгоритъма, използващ нормалната честота на думите в естествения език, може да бъде фактът, че синтактичните n-грами проследяват граматически конструкции, използвани от автора, които в по-свободните текстове, каквито са блог постове, могат да варират и да не бъдат така добре обособени, както в една литературна творба. В същото време подходът, базиран на честотата на думите в естествения език, не разчита толкова на контекста, в който са думите, колкото на правилното им групиране, което води до по-добри резултати за конкретния корпус. В подкрепа на това твърдение е и резкият спад в точността, когато се използват по-дълги синтактични n-грами (Фигура 7-28, Фигура 7-30, Фигура 7-32, Фигура 7-34, Фигура 7-36, Фигура 7-38).

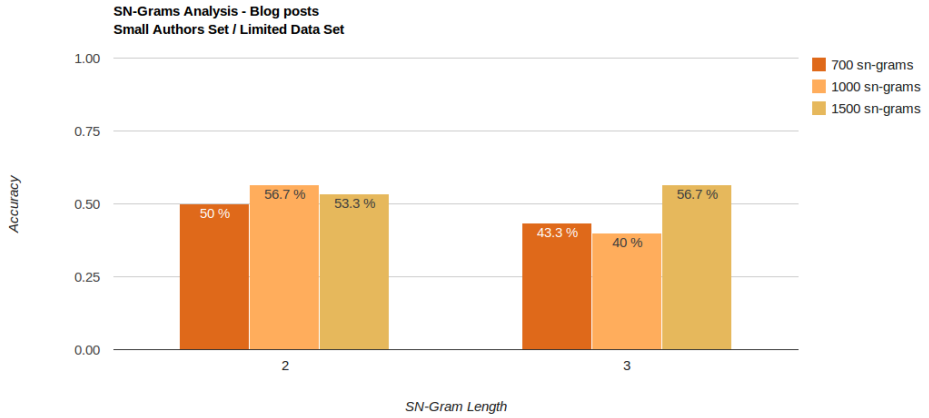
Представянето на двата подхода върху корпуса от блог постове за различните му конфигурации варира между 50% - 86%. Най-високият резултат от 86% се получава, когато множеството от автори е малко (в конкретния случай 50 автора) и се използват всички налични блог постове на авторите (Фигура 7-25, Фигура 7-26). Най-неточна – 50.5% - е класификацията, когато се използва корпусът, съдържащ 150 автора и минималното количество от блог постове в използваните конфигурации – по 5 блог поста на автор (Фигура 7-33).

Като цяло резултатите са малко по-ниски от тези получени върху корпусът от английски романи. Това е напълно очаквано, имайки предвид, че авторовият стил в творчеството на един писател дори и да се променя спрямо тема, време, когато е написан, е много по-консистентен. В един блог авторовият стил е в много случаи по-неформален и променящ се динамично. Също така количеството на текстовете за всеки автор, с които разполагаме в корпуса от романи, е много по-голямо от съответното такова в корпуса с блог постове.

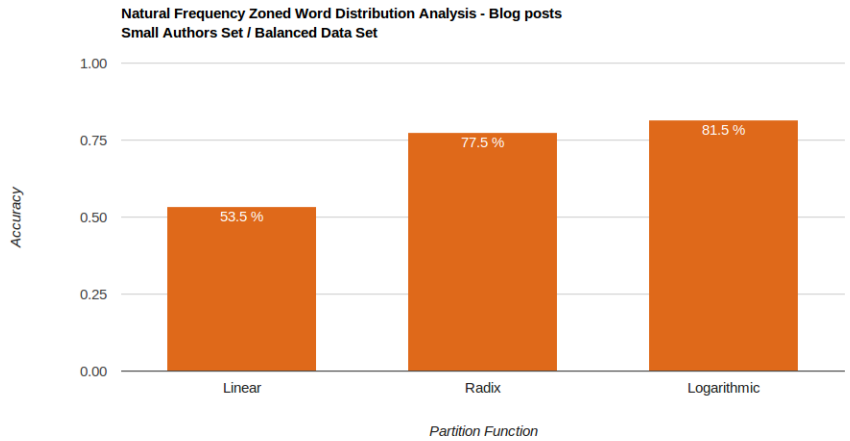
Подходящо сравнение, за да бъде оценено доколко постигнатите резултати върху корпуса от блогове са добри, е изследването на Abbasi & Chen върху корпуса CyberWatch Chat, в който за всеки автор се съдържат средно по 1 500 думи. Получените резултати са: 25 автора – 50.4%, 50 автора – 42.6%, 100 автора – 31.7%. Ясно се вижда, че точността постигната с двата подхода, разгледани в настоящата работа е по-висока. Това показва стабилност на разглежданите методи, когато са приложени върху корпуси с различни характеристики, в случая романи и блог постове.



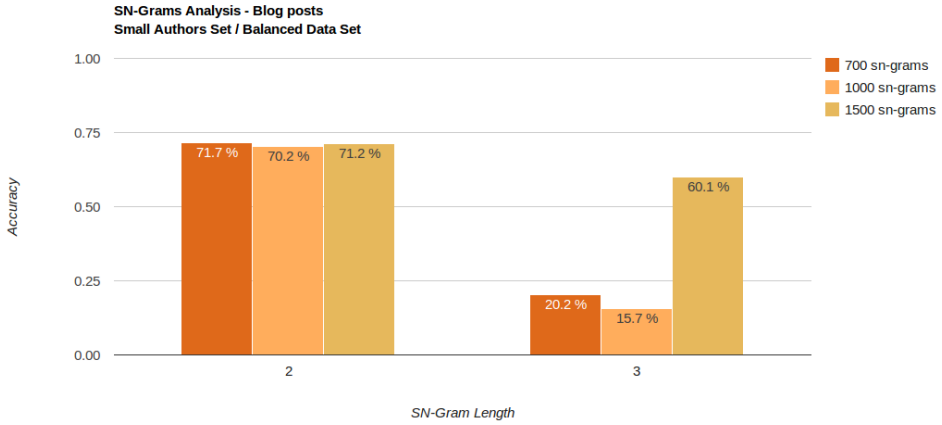
Фигура 7-21 Резултати върху корпуса от блог постове, използвайки NFZ-WD - Small Authors Set и Limited Data Set



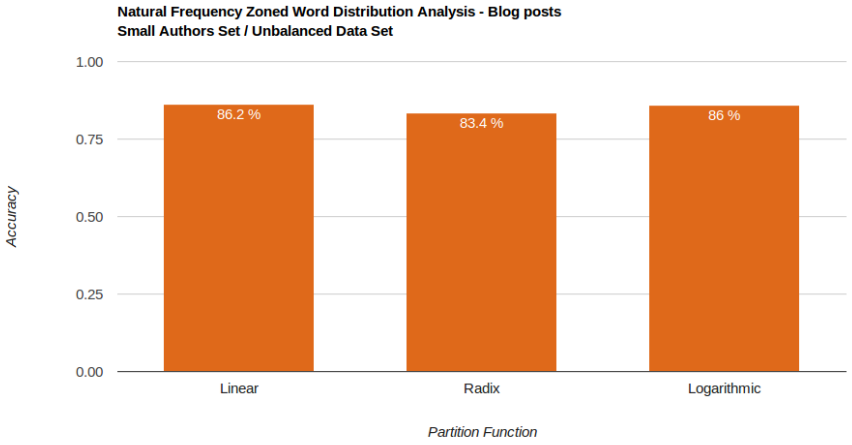
Фигура 7-22 Резултати върху корпуса от блог постове, използвайки синтактични n-грами – Small Authors Set и Limited Data Set



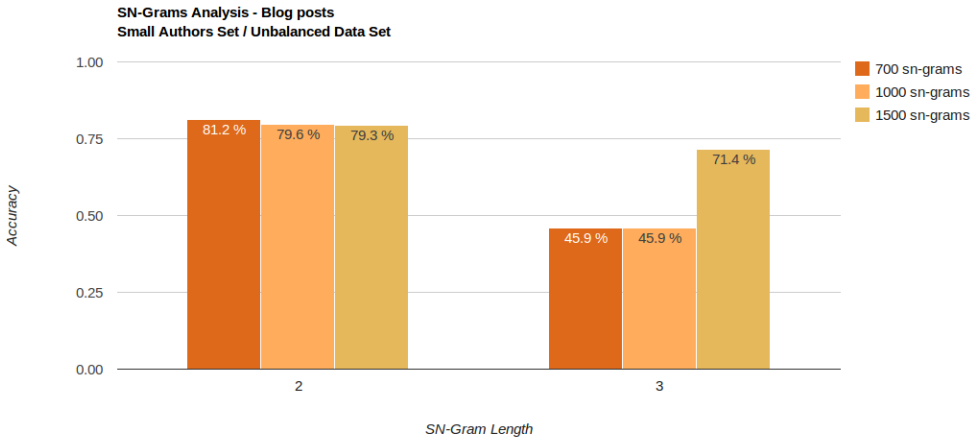
Фигура 7-23 Резултати върху корпуса от блог постове, използвайки NFZ-WD - Small Authors Set и Balanced Data Set



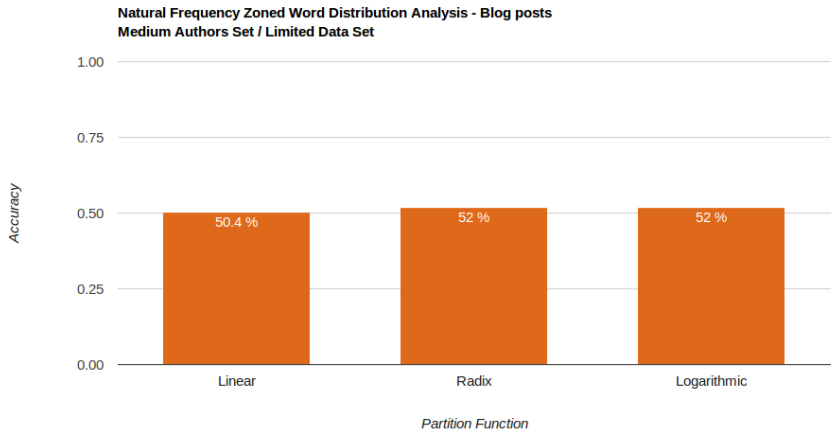
Фигура 7-24 Резултати върху корпуса от блог постове, използвайки синтактични n-грами - Small Authors Set и Balanced Data Set



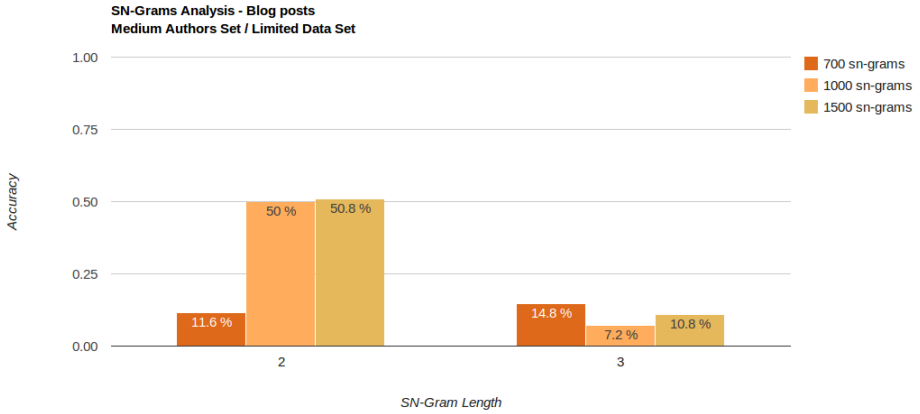
Фигура 7-25 Резултати върху корпуса от блог постове, използвайки NFZ-WD - Small Authors Set и Unbalanced Data Set



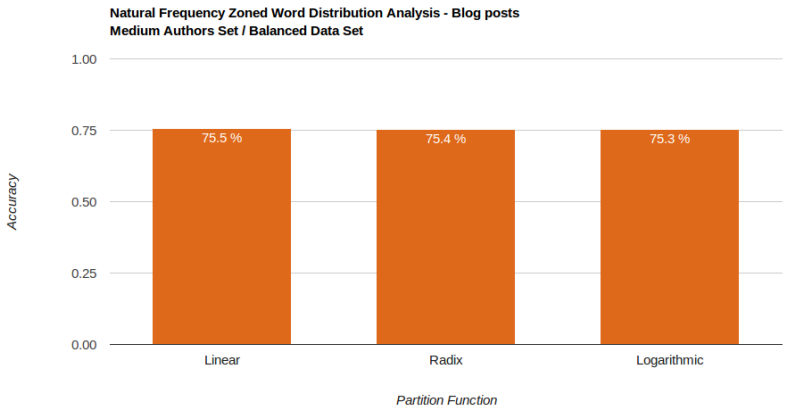
Фигура 7-26 Резултати върху корпуса от блог постове, използвайки синтактични n-грами - Small Authors Set и Unbalanced Data Set



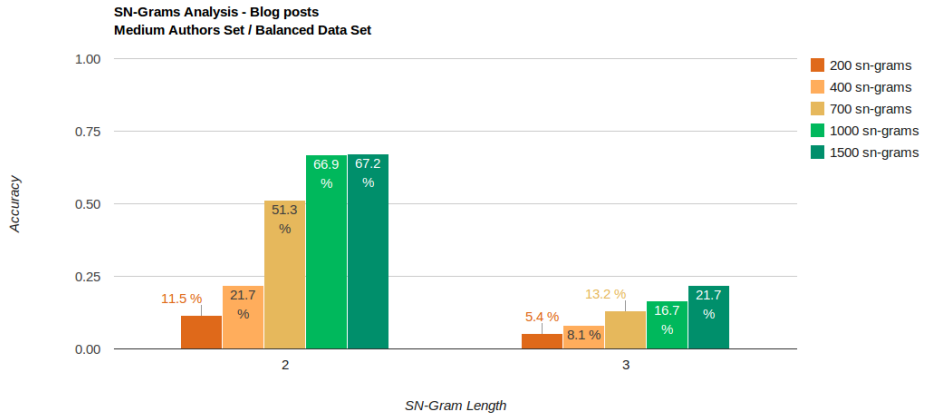
Фигура 7-27 Резултати върху корпуса от блог постове, използвайки NFZ-WD - Medium Authors Set и Limited Data Set



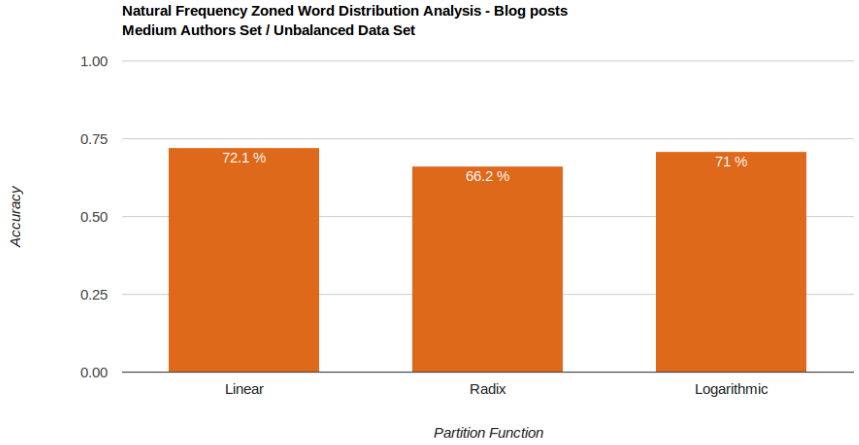
Фигура 7-28 Резултати върху корпуса от блог постове, използвайки синтактични n-грами - Medium Authors Set и Limited Data Set



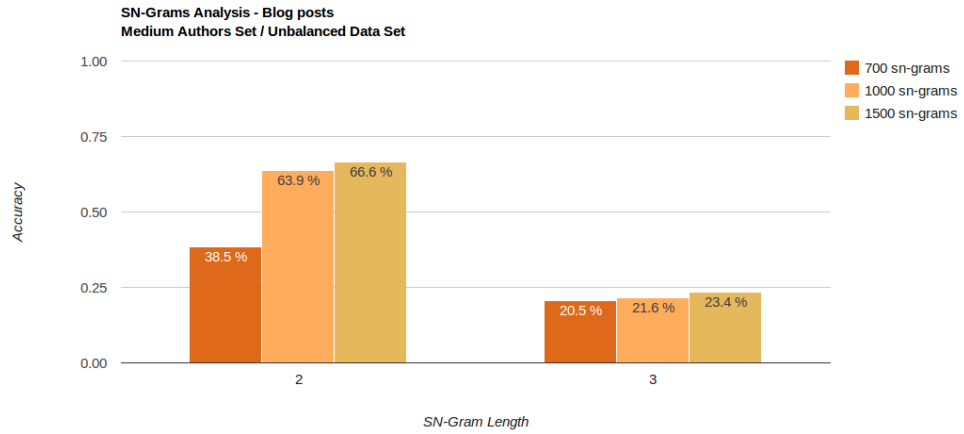
Фигура 7-29 Резултати върху корпуса от блог постове, използвайки NFZ-WD – Medium Authors Set и Balanced Data Set



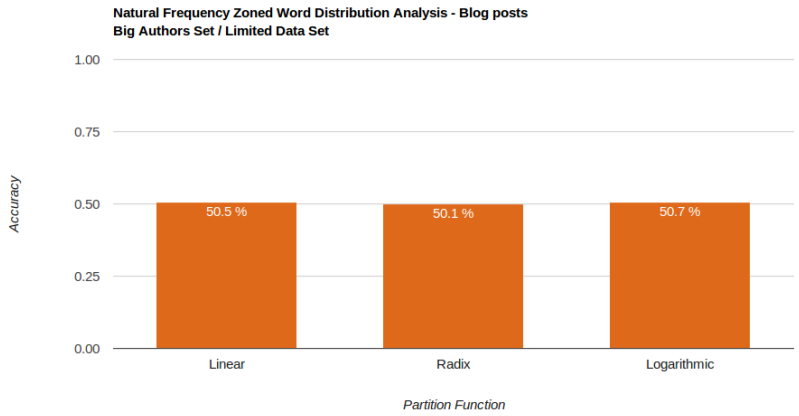
Фигура 7-30 Резултати върху корпуса от блог постове, използвайки синтактични n-грами - Medium Authors Set и Balanced Data Set



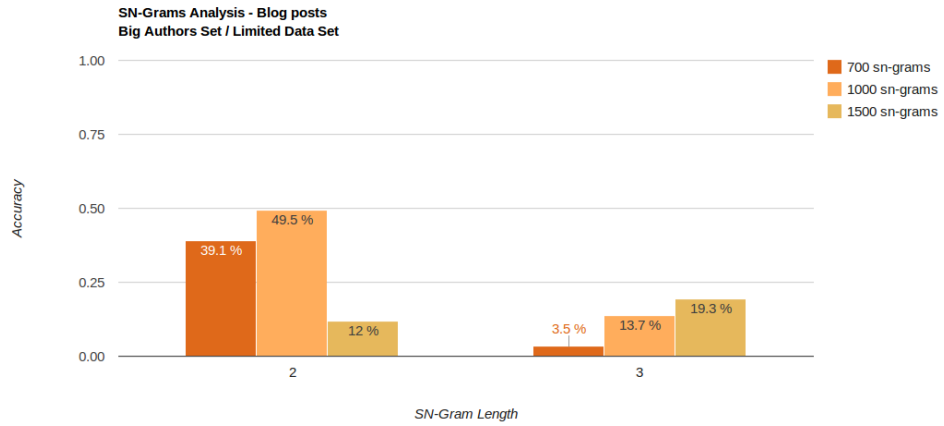
Фигура 7-31 Резултати върху корпуса от блог постове, използвайки NFZ-WD - Medium Authors Set и Unbalanced Data Set



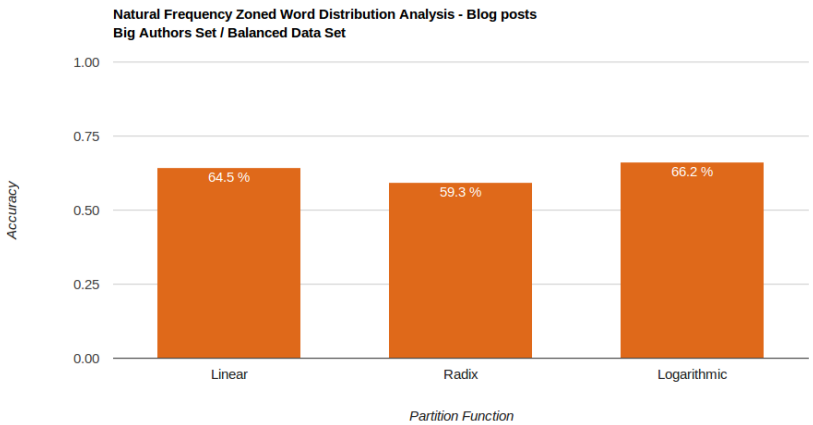
Фигура 7-32 Резултати върху корпуса от блог постове, използвайки синтактични n-грами - Medium Authors Set и Unbalanced Data Set



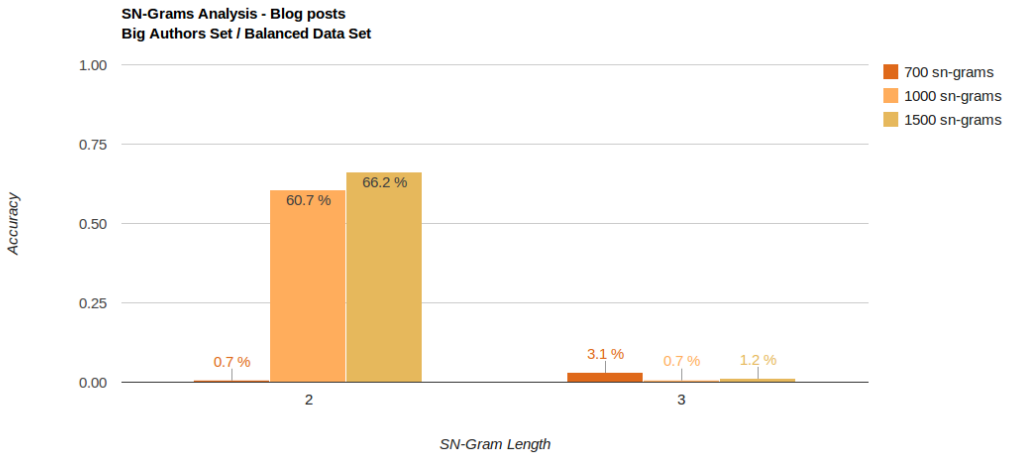
Фигура 7-33 Резултати върху корпуса от блог постове, използвайки NFZ-WD - Big Authors Set и Limited Data Set



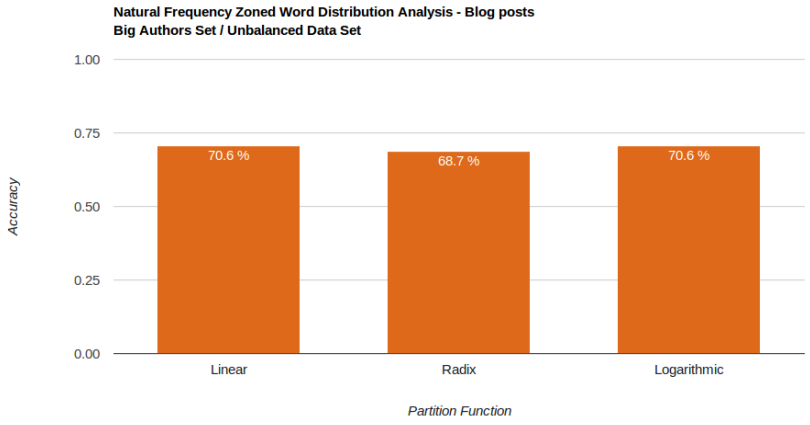
Фигура 7-34 Резултати върху корпуса от блог постове, използвайки синтактични n-грами – Big Authors Set и Limited Data Set



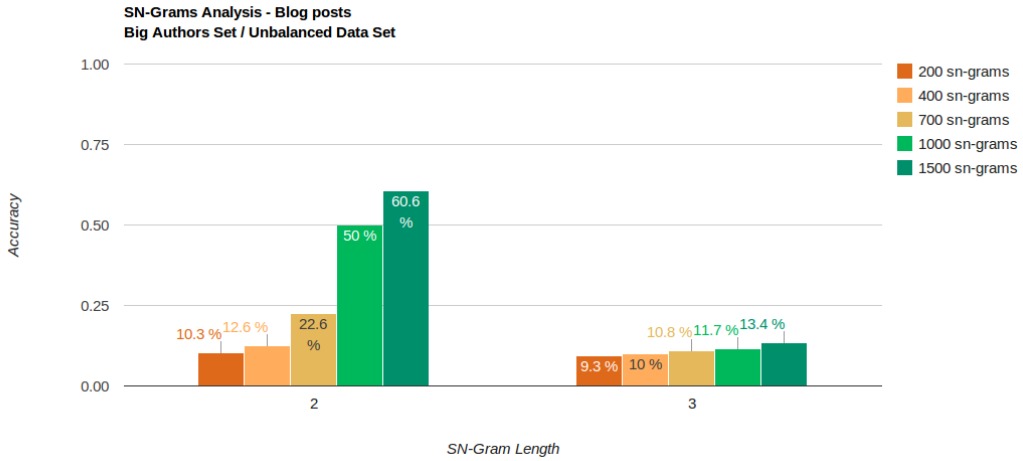
Фигура 7-35 Резултати върху корпуса от блог постове, използвайки NFZ-WD - Big Authors Set и Balanced Data Set



Фигура 7-36 Резултати върху корпуса от блог постове, използвайки синтактични n-грами - Big Authors Set и Balanced Data Set



Фигура 7-37 Резултати върху корпуса от блог постове, използвайки NFZ-WD - Big Authors Set и Unbalanced Data Set



Фигура 7-38 Резултати върху корпуса от блог постове, използвайки синтактични n-грами- Big Authors Set и Unbalanced Data Set

8. Заключение

Настоящата дипломна работа се фокусира върху разглеждането на два алгоритъма за извличане на характеристики от текст, определящи авторов стил, които да бъдат използвани за решаването на задачата за разпознаване автор на текст. Единият алгоритъм е базиран на синтактични n-грами, а другият на разделяне на зони от думи, вземайки честота на думите в естествения език.

Получените резултати от експериментите потвърждава влиянието, което имат характеристиките на корпуса, който се използва при тестването на различни подходи за решаването на задачата за разпознаване автор на текст. Очаквано най-лоши са резултатите, когато множеството от автори е голямо, а множеството от текстове за всеки от тях малко. Най-добри са резултатите, когато корпусът разполага с по-голямо количество от текстове, написани от авторите, и броят на авторите е по-малък. Въпреки очаквания спад на точността при увеличаване броя на авторите и намаляване на множеството от текстове, точността при подходящо избрани параметри и за двата подхода е винаги над 50%.

Това не е достатъчно, за да може да се разчита на правилна класификация от разглежданите методи, но със сигурност е основа, върху която може да се надгражда. Получените резултати след експериментите са сравними, а в някои случаи и по-добри от резултатите, постигнати от, до този момент малкото, изследвания, в които се използват различни конфигурации на корпусите от данни, в които има по-голямо множество от автори и по-малко множество от текстове.

Един от големите недостатъци и на двата подхода, особено на подхода, разчитащ на синтактичните n-грами, както и на избрания метод за класификация – машини с поддържащи вектори, е тяхната ресурсоемкост, което попречи и настоящото изследване да бъде по-изчерпателно в използването на различни параметри. Друг недостатък и на двата подхода е, че не са независими от езика на текстовете в корпуса – при използването на синтактичните n-грами е необходим синтактичен анализ на текста, а при използването на нормалната честота на думите в езика е необходим достатъчно изчерпателен корпус от честотите на думите в съответния език, за да се разчита на добри резултати от алгоритъма.

Направените експерименти потвърждават, че едва ли ще бъде намерен универсален подход, който да дава надеждни резултати, независимо от характеристиките на корпуса от данни, както като големина, така и като жанрово и тематично разнообразие. Целта е да се установи, на база изследвания, кои подходи са най-подходящи в определен контекст. За постигането на тази цел е необходимо разнообразието от методи да бъде изследвано върху множество от базови корпуси с различни характеристики.

Литература

1. Fabrizio Sebastiani. (2002). Machine Learning in Automated Text Categorization. ACM Computing Surveys, Vol. 34, No. 1, March 2002.
2. Susan Schreibman, Ray Siemens and John Unsworth. (2004). A Companion to Digital Humanities. Published by Blackwell Publishing Ltd.
3. Mendenhall T. C. (1887). The characteristic curves of composition. Science, IX, 237–49.
4. Zipf G. K. (1932). Selected studies of the principle of relative frequency in language. Harvard University Press, Cambridge, MA.
5. Yule G.U. (1938). On sentence-length as a statistical characteristic of style in prose, with application to two cases of disputed authorship. Biometrika, 30, 363-390.
6. Yule G.U. (1944). The statistical study of literary vocabulary. Cambridge University Press.
7. Mosteller F., Wallace, D.L. (1964). Inference and disputed authorship: The Federalist. Addison-Wesley.
8. Rudman J. (1997). The state of authorship attribution studies: Some problems and solutions, Computers and the Humanities, vol. 31, pp. 351–365, 1998.
9. Honore A. (1979). Some simple measures of richness of vocabulary. Association for Literary and Linguistic Computing Bulletin, 7(2), 172–177.
10. Cavnar, W., & Trenkle, J. (1994). N-gram-based text categorization. In Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval, (pp. 161–175). Las Negas, NV.
11. Dunning, T. (1994). Statistical identification of language. Technical Report MCCS 94-273, Computing Research Lab (CRL), New Mexico State University, Las Cruces, New Mexico.
12. Clement, R., & Sharp, D. (2003). Ngram and Bayesian classification of documents for topic and authorship. Literary and Linguistic Computing, 18(4), 423–447.
13. Keselj, V., Peng, F., Cercone, N., & Thomas, C. (2003). N-gram-based author profiles for authorship attribution. In Proceedings of the 6th Conference of the Pacific Association for Computational Linguistics, (pp. 255–264). Halifax, Canada: Pacific Association for Computational Linguistics.
14. Peng, F., Shuurmans, D., Keselj, J., & Wang, S. (2003). Language independent authorship attribution using character level language models. In Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics, (pp. 267–274). Morristown, NJ.
15. Stamatatos, E. (2006). Ensemble-based author identication using character n-grams. In B. Stein, & O. Kao (Eds.) Proceedings of the 3rd International Workshop on Text-Based Information Retrieval, (pp. 41–46). Trento, Italy.
16. Grieve, J. (2007). Quantitative authorship attribution: An evaluation of techniques. Literary and Linguistic Computing, 22(3), 251–270.
17. Hirst, G., & Feiguina, O. (2007). Bigrams of syntactic labels for authorship discrimination of short texts. Literary and Linguistic Computing, 22(4), 405–417.
18. Sanderson, C., & Guenter, S. (2006). Short text authorship attribution via sequence kernels, Markov chains and author unmasking: An investigation. In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP), (pp. 482–491). Syndney, Australia.
19. Coyotl-Morales, R., Villase ˜nor Pineda, L., Montes-y G´omez, M., & Rosso, P. (2006). Authorship attribution using word sequences. In Proceedings of the 11th Iberoamerican Congress on Pattern Recognition, Lecture Notes in Computer Science 4225, (pp. 844–853). Cancun, Mexico: Heidelberg: Springer Verlag.

20. Gamon, M. (2004). Linguistic correlates of style: Authorship classification with deep linguistic analysis features. In Proceedings of the 20th International Conference on Computational Linguistics, (pp. 611–617). Geneva, Switserland: Association for Computational Linguistics.
21. Baayen, H., Van Halteren, H., & Tweedie, F. (1996). Outside the Cave of Shadows: Using syntactic annotation to enhance authorship attribution. Literary and Linguistic Computing, 11(3), 121–131.
22. Stamatatos, E., Fakotakis, N., & Kokkinakis, G. (2000). Automatic text categorization in terms of genre and author. Computational Linguistics, 26(4), 461–485.
23. Khmelev, D., & Tweedie, F. (2001). Using Markov chains for identification of writers. Literary and Linguistic Computing, 16(4), 299–307.
24. Kukushkina, O., Polikarpov, A., & Khmelev, D. (2001). Using literal and grammatical statistics for authorship attribution. Problemy Peredachi Informatsii, 37(2), 96–108.
25. Diederich, J., Kindermann, J., Leopold, E., & Paass, G. (2003). Authorship attribution with Support Vector Machines. Applied Intelligence, 19(1-2), 109–123.
26. Koppel, M. and Schler, J. (2003), Exploiting Stylistic Idiosyncrasies for Authorship Attribution, in Proceedings of IJCAI'03 Workshop on Computational Approaches to Style Analysis and Synthesis, pp. 69-72.
27. Burrows, J. (2002). 'delta': A measure of stylistic difference and a guide to likely authorship. Literary and Linguistic Computing, 17(3), 267–287.
28. Zhili Chen, Liusheng Huang, Wei Yang, Peng Meng, and Haibo Miao. (2012). More than Word Frequencies: Authorship Attribution via Natural Frequency Zoned Word Distribution Analysis. Cornell University Library.
29. Matthews, R., & Merriam, T. (1994). Neural computation in stylometry I: An application to the works of Shakespeare and Fletcher. Literary and Linguistic Computing, 8(4), 203–209.
30. Zheng, R., Li, J., Chen, H., & Huang, Z. (2006). A framework for authorship identification of online messages. Journal of the American Society for Information Science and Technology, 57(3), 378–393.
31. Tearle, M., Taylor, K., & Demuth, H. (2008). An algorithm for automated authorship attribution using neural networks. Literary and Linguistic Computing, 23(4), 425–442.
32. Zhao, Y., & Zobel, J. (2005). Effective and scalable authorship attribution using function words. In G. Lee, A. Yamada, H. Meng, & S. Myaeng (Eds.) Proceedings of the 2nd Asian Information Retrieval Symposium, Lecture Notes in Computer Science 3689, (pp. 174–189). Jeju Island, Korea: Heidelberg: Springer Verlag.
33. Koppel, M., Argamon, S., & Shimoni, A. (2003b). Automatically categorizing written texts by author gender. Literary and Linguistic Computing, 17(4), 401–412.
34. Argamon, S., Whitelaw, C., Chase, P., Dawhle, S., Hota, S., Garg, N., & Levitan, S. (2007). Stylistic text classification using functional lexical features. Journal of the American Society of Information Science and Technology, 58(6), 802–822.
35. Abbasi, A., & Chen, H. (2008). Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. ACM Transactions on Information Systems, 26(2), 7:1–7:29.
36. Koppel, M., Schler, J., Argamon, S., & Messeri, E. (2006). Authorship attribution with thousands of candidate authors. In E. Efthimiadis, S. Dumais, D. Hawking, & K. J¨ arvelin (Eds.) Proceedings of the 29th International Conference of the Special Interest Group on Information Retrieval, (pp. 659–660). Seattle, WA, USA: Association for Computing Machinery.
37. Moore, R. (2001). There's no data like more data (but when will enough be enough?). In Proceedings of IEEE International Workshop on Intelligent Signal Processing. Budapest,Hungary.
38. Luyckx, K., & Daelemans, W. (2008b). Personae: a corpus for author and personality prediction from text. In Proceedings of the 6th International Conference on Language Resources and Evaluation, (p. no pages). Marrakech, Morocco: ELDA.

39. Koppel, M., Schler, J., & Argamon, S. (2011). Authorship attribution in the wild. Language Resources and Evaluation, Special Issue on Plagiarism and Authorship Analysis.
40. Biber, D. (1990). Methodological issues regarding corpus-based analyses of linguistic variations. Literary and Linguistic Computing, 5, 257–269.
41. Biber, D. (1993). Representativeness in corpus design. Literary and Linguistic Computing, 8, 1–15.
42. Burrows, J. (2007). All the way through: Testing for authorship in different frequency strata. Literary and Linguistic Computing, 22(1), 27–47.
43. Tweedie, F., & Baayen, H. (1998). How variable may a constant be? Measures of lexical richness in perspective. Computers and the Humanities, 32, 323–352.
44. Eder, M. (2010). Does size matter? Authorship attribution, small samples, big problem. In E. e. a. Pierrazo (Ed.) Proceedings of Digital Humanities 2010, (pp. 132–135). London, UK: Centre for Computing in the Humanities, Kings College London.
45. J. Schler, M. Koppel, S. Argamon and J. Pennebaker (2006). Effects of Age and Gender on Blogging in Proceedings of 2006 AAAI Spring Symposium on Computational Approaches for Analyzing Weblogs.
46. Boser, B. E.; Guyon, I. M.; Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. Proceedings of the fifth annual workshop on Computational learning theory - COLT '92. p. 144.
47. Grigori Sidorov, Francisco Velasquez, Efstathios Stamatatos, Alexander Gelbukh, and Liliana Chanona-Hernández. Syntactic Dependency-based N-grams as Classification Features. LNAI 7630, 2012, pp. 1–11
48. Garcна-Hernбndez, R.A., Martнnez Trinidad, J.F., Carrasco-Ochoa J.A.(2010). Finding Maximal Sequential Patterns in Text Document Collections and Single Documents. Informatica
49. Ebrahimpour M., Putnins T. J., Berryman M. J., Allison A., Ng B. W.-H., Abbott D. (2013). Automated authorship attribution using advanced signal classification techniques, PLoS ONE, Vol. 8, No. 2, Art. No. e54998, 2013
50. Support Vector Machine, Wikipedia: http://en.wikipedia.org/wiki/Support_vector_machine
51. Cross-Validation, Wikipedia: http://en.wikipedia.org/wiki/Cross-validation_%28statistics%29
52. LIBSVM: http://www.csie.ntu.edu.tw/~cjlin/libsvm/
53. Stanford Parser: http://nlp.stanford.edu/software/lex-parser.shtml
54. British National Corpus: http://www.kilgarriff.co.uk/bnc-readme.html.
55. The Blog Authorship Corpus: http://u.cs.biu.ac.il/~koppel/BlogCorpus.htm
56. Corpus of English Novels: https://perswww.kuleuven.be/~u0044428/
57. Node.js module for SVM – node-svm - https://www.npmjs.com/package/node-svm
58. Node.js module for NLP – natural - https://www.npmjs.com/package/natural