Модел за автоматично сричкообразуване на български думи основан на ненадзиравано машинно самообучение

Красен Пенчев

An Unsupervised Machine Learning Model for Automatic Syllabification of Bulgarian Words

Krasen Penchev

Abstract

There are a lot of definitions of the syllable, and many discussions about it's role in the structure of the spoken languages. Some linguists put it in a central place in their theories. Having in mind that every person speaking a language, which is his/hers mother tongue, can divide the words into syllables, it could be concluded that the syllable is a structural entity of the spoken languages. The automatic syllabification, at least in theory, is applicable in a broad range of problems. Unfortunately it's not as popular as one would imagine. The small number and the low quality of the training resources are the main reasons for the low adoption rate of the automatic syllabification. A model for an unsupervised automatic syllabification is presented in this report. The aim is to design a general purpose model which would address the outlined existing problems of the automatic syllabification in the context of the Bulgarian language. The presented method is not constrained by the volume of the training data or the field of knowledge it's coming from.

Keywords: syllabification, machine learning, automatic, unsupervised, model

JEL Code:O300

Въведение

Съществуват много дефиниции и дискусии за същността и значението на сричката в речта. Някои лингвисти отдават голямо значение на сричката, поставяйки я на централно място в теориите си. Изповядващите другата крайност я смятат за несъществена единица на говоримия език. Една от причините за различните гледни точки е предизвикателството точно да бъде определена същността на сричката.

За текущия доклад се приема определението, че сричката е единица на говоримия език, която е по-голяма от фонема, и се състои от един гласен звук, който може да бъде предшестван и, или последван от един или повече съгласни звуци[1] [Huang,Acero & Hon 2001]. Дефиницията за сричка е подходяща, защото обхваща споменатите в предходния абзац, общоприети признаци на сричката, без да засяга такива, за които учените не са единодушни.

Въпреки различните мнения на учените, хората използващи даден език са способни да броят сричките в думите единствено използвайки интуицията си. Имайки предвид това, може да се каже, че сричката, макар и не всепризнато от лингвистите, е структурна единица в изграждането на думата. Възможността за надеждно определяне на границите на сричката би имала широко приложение. Правилното „разбиване“ на думите на срички би спомогнало за моделирането на думи в системите за автоматично преобразуване на реч в текст [Marchand,Adsett & Damper 2009]. Някои от съществуващите индекси за оценка на четимост на текст, например Flesch-Kincaid, включват като задължителна стъпка анализ на сричките в думите съставящи текста. Сричките могат да бъдат използвани като индексирани термини при изграждането на текстови търсачки.

Идентифицирането на граници на срички има широко приложение, на теория, но не е толкова широко използвано, колкото някой би предположил. Малък брой и ниско качество на ресурсите, подходящи за обучение, са едни от причините автоматичното

[1] В литературата, съчетанието между ядрото и крайната част на сричката често се нарича *рима*.

сричкообразуване да бъде слабо застъпено в сверите, където се предполага, че би имало широко приложение [Rogova,Demuynck & Van Compernolle 2013]. Поради множеството дефиниции и дискусии около същността на сричката, е трудно да бъдат създадени качествени ресурси за обучение на модели за сричкообразуване.

Текущият доклад представя модел за автоматично сричкообразуване, основан на машинно самообучение. Основната цел е създаване на модел с общо предназначение, който да адресира споменатите в предходния абзац проблеми на автоматичното сричкообразуване. Разглежданият подход не е ограничен нито от количество тренировъчни данни, нито от предметната област на данните. Напротив, колкото по-всеобхватна е тематиката на обучителните данни, толкова по-добри ще са възможностите на модела за идентифициране на границите на срички.

1. Подходи за автоматично сричкообразуване

Съществуват два основни подхода за автоматично сричкообразуване, единият от които базиран на предварително дефинирани правила, а другият, на машинно самообучение [Marchand,Adsett & Damper 2007]. Първият подход използва универсални правила за сричкообразуване, каквито бяха споменати във въведението. Основаният на машинно самообучение подход използва корпус, чийто думи са правилно „разбити“ на срички, за да обработва непознати думи. Доказано е, че подходът основан на предварително съставени правила показва по-слаби резултати от базирания на машинно самообучение [Marchand et al. 2007; Marchand et al. 2009]. По-слабите резултати се срещат в случаите на прилагане на моделите, основани на предварително дефинирани правила, върху данни с различна тематика. Обикновено подходите основани на предварително определени правила не са с универсално предназначение. Например модел, който е създаден за обработка на правни документи би се справил слабо в обработката на поетични текстове в сравнение с модел с общо предназначение.

1.1. Сричкообразуване основано на универсални правила

Съществуват три популярни метода за идентифициране на граници на срички, наречени съответно принцип за максимизиране на начална част (на сричка) [Kahn 1980], принцип на звуковата последователност [Selkirk 1984], и принцип на легалност [Goslin & Frauenfelder 2001].

При принципа на звуковата последователност, на всеки звук в сричката бива дадена числова стойност според скалата на звучност. Гласните звуци са с най-висок ранг, следвани от носовите съгласни, фрикативите, и плозивите. Стойностите на последователните звуци нарастват в началната част на сричката и намаляват в крайната част, кодата. Основният проблем на този подход е невъзможността за правилно определяне на границите на сричките, когато повече от една възможност за поставяне на граница е възможно. Повече от една възможност за поставяне на граница има в случай на струпване на съгласни звуци. Предимство на подхода е, независимостта му от тренировъчните данни.

Принципът на легалност позволява струпване на съгласни звуци да бъде валидна начална или крайна част на сричка, само ако е било срещано като начално или крайно струпване на съгласни в дума. Това значи, че начална част на сричка може да бъде валидна единствено ако се е срещала като начална последователност от звуци на дума от тренировъчните данни. Този принцип има същия недостатък като принципа на звуковата последователност - когато има няколко начина за разбиване на струпване на съгласни звуци, границите между сричките са неясни.

При третия метод, принципа за максимизиране на началната част на сричката, когато има няколко варианта за разбиване на струпване на съгласни звуци се предпочита този вариант при който началната част е с по-голяма дължина. Подходът е силно зависим от качеството и количеството на обучителните данни.

1.2. Сричкообразуване основано на машинно самообучение

Определянето на границите на сричката изцяло чрез предварително съставени правила често не постига желаните резултати. В някои ситуации, правилата са недостатъчни за ясно поставяне на границите на сричките, съставящи обработваната дума. Недостатъците на предварително съставени правила водят до нуждата от по-добри методи за автоматично сричкообразуване. Съществуват множество подходи, основани на машинно самообучение.

Müller [2006] предлага подход, който включва разработване на граматики за характеризиране на фонологичната структура на думите. Използвайки граматики, думата бива представена като последователност от срички. Всяка сричка, от своя страна е представена от съставните ѝ начална част и рима. Римата се състои от ядрото и кодата на сричката. Всички граматики в предложения подход разграничават едносрични и многосрични думи, и различни по брой струпвания на съгласни звуци.

При подхода на Bartlett,Kondrak and Cherry [2009] са постигнати едни от най-добрите резултати в автоматичното сричкообразуване. В подхода се използва комбинация от метода на опорния вектор и скрити модели на Марков. Всяка фонема бива класифицирана спрямо позицията си в думата, взимайки под внимание съвкупност от характеристики. Скритите модели на Марков преодоляват недостатъка на третирането на фонемите самостоятелно. При тренирането на модела, към всяка дума бива асоцииран клас, избран от съвкупността на всички възможни класове. Класовете представляват последователности от срички, представени като последователност от начална част, ядро, и крайна част.

Представеният в текущия доклад модел стъпва на предложения от Mayer [2010] подход за независимо от езика, автоматично, сричкообразуване. За всяка една дума в тренировъчните данни, авторът намира всички възможни комбинации от срички, използвайки универсалните правила за сричкообразуване. В същото време, честотите на срещане на възможните срички биват акумулирани. При обработка на непознати думи, моделът избира тази комбинация от срички, която има най-голям сбор на честотите на срещане на отделните срички.

Между подхода на Mayer и разглеждания в този доклад метод има две съществени разлики. Първата разлика е, че текущия модел няма за цел да бъде независим от езика. Представената разработка се отнася само за български език. Въпреки, че се отнася само за българския език, моделът има минимално „познание“ за него, разликата между гласни и съгласни звуци, и основните правила за сричкообразуване, посочени в определението за сричка, предложено в началото на тази точка. Втората разлика е, че текущият модел, в процеса на обучение, акумулира данни за характеристики на сричките, които методът на Mayer не взима под внимание.

2. Модел за автоматично сричкообразуване на български думи

Както името им подсказва, моделите основани на машинно самообучение трябва да бъдат „обучени“ преди да бъдат използвани върху непознати данни. Освен процеса на обучение, който е втори по ред на изпълнение, има още два процеса, на които също са важни и заслужават да им бъде обърнато внимание.

Първият процес, предшестващ обучението, включва събиране на тренировъчни данни и тяхната подготовката. Подготовката най-често включва „почистване“ на данните, което се изразява в унифициране на формата им, и всички действия, които биха намалили „шума“ в данните. Процесът по подготовка на тренировъчни данни е разгледан по-подробно в точка 2.1.

Обучението, което следва подготовката на данните, зависи пряко от качеството на тренировъчните данни. Въпреки, че зависи от качеството на обучителните данни, може да се каже, че процесът на обучение е също толкова важен, колкото и подготовката на данните. Тренировъчният процес на представяният в текущия доклад модел е разгледан подробно в точка 2.2.

След успешното обучение на модела следва процеса на оценка на резултатите. Този

процес дава реална информация за качеството на предходните процеси по събиране на тренировъчни данни, и обучение. В литературата се срещат различни подходи за осъществяване на оценката на резултатите, някои от които са разгледани в точка 2.3.

2.1. Подбор и подготовка на тренировъчни данни

Същността на разглеждания модел предполага, че той ще оперира върху отделни думи. Това в голяма степен опростява процеса по подготовка на тренировъчни данни. Извличането на отделни думи от текст е много по-проста операция от например анализ на изречения или параграфи, защото се избягва от сложни алгоритми за идентифициране на препинателни знаци и техния контекст. В подготовката на тренировъчните данни могат да бъдат идентифицирани две стъпки, които са зависими помежду си.

Първата стъпка включва събирането на „груби“ тренировъчни данни. В конкретния случай, тези данни представляват корпус от текстове. Важно е подбраните текстовете да бъдат на разнообразна тематика, за да може броят на уникалните думи да бъде възможно най-голям, представяйки същевременно богатството на българския език. Обучението на представяния подход за автоматично сричкообразуване би било повлияно благоприятно ако тренировъчните данни съдържат думи от възможно най-много сфери на познание.

Следващата стъпка в подготовката на тренировъчните данни е трансформирането на събрания в предходната стъпка корпус до отделни думи. Този процес се нарича токенизация. Токенизацията, в текущия случай, може да бъде дефинирана като сегментация на символен низ до думите, които го изграждат. Важно е да се има в предвид, че токенизацията има различен смисъл ако се използва в контекста на преобразуване на програмен код до абстрактно синтактично дърво. Токенизацията има съвсем друг смисъл в областта на електронните финанси, където представлява метод за сигурно съхранение на номера на дебитни и кредитни карти [Daz-Santiago,Maria Rodriguez-Henriquez & Chakraborty 2014] .

Съществуват множество подходи за токенизация, основаващи се на предварително съставени правила, на машинно самообучение, но те не са предмет на анализ в текущия доклад. В трудовете на Habert et al. [1998]; He and Kayaalp [2006] се разглеждат множество подходи за токенизация, тяхната успеваемост и скорост. Въпреки изобилието на подходи за сегментация е важно да бъде избран най-подходящият за нуждите на представения в текущия доклад модел.

След токенизацията на корпуса следва да бъдат премахнати дублираните думи. Също като за сегментацията, така и за премахване на повтарящите се думи съществуват множество подходи. Изборът на подход за премахване на дубликатите не е предмет на текущия доклад. Тъй като моделът не взима предвид семантичното значение на думата, размерът на буквите може да бъде унифициран. Тази стъпка е важна, защото при евентуалното ѝ пропускане може да се очакват некоректни резултати.

Разполагайки със списък от неповтарящи се думи, с унифициран размер на буквите, може да се пристъпи към обучението на модела.

2.2 Обучение на модела за автоматично сричкообразуване

Също като процесът по подготовка на тренировъчни данни, разгледан в предходните редове, процесът на обучение на модела за идентифициране на границите на срички също може да бъде разделен на няколко стъпки.

Първата стъпка включва извличането на възможните комбинации между сричките на думата. Правилата за извличане се основават на предложеното определение за сричка във въведението. Този процес повтаря за всяка една дума в тренировъчните данни.

Например думата „червен“. Тя може да бъде разделена на срички по няколко начина. В един случай се получава комбинацията (че, рвен), в друг (чер, вен), а в трети, (черв, ен). Всеки човек, говорещ български език, може да посочи, че втората комбинация съдържа сричките на думата „червен“.

След като бъдат извлечени възможните комбинации между сричките за всички думи,

резултатите трябва да бъдат акумулирани. Конкретната структура от данни в която да бъдат акумулирани обработените тренировъчни данни не е предмет на дискусия в текущия доклад. Въпреки това, е добре да се спомене, че структурата от данни е тясно свързана с възможностите за съхранение на тренирания модел при бъдеща практическа имплементация.

Както беше споменато в точка 1, при подходът на Mayer [2010] се изчислява честота на срещане на всяка една сричка от извлечените комбинации между срички за всяка една от думите в тренировъчните данни. Представения в текущия доклад метод също използва тази метрика, но е добавено още едно ниво на информация. Не само се изчисляват честотите на срещане за всяка сричка, а се пази и информация за местоположението ѝ в думата.

Mayer [2010] достига до извода, че съществуват три основни местоположения, които сричката може да заема в една дума, начално (initial), междинно (medial), и крайно (final). Ако бъде взета за пример думата „параван“, резултатът от тренирането на модела върху нея би бил $(\text{па}_i, \text{ра}_m, \text{ван}_f)$, $(\text{пар}_i, \text{ав}_m, \text{ан}_f)$, $(\text{па}_i, \text{рав}_m, \text{ан}_f)$, където (i, m, f) обозначават местоположението на сричката в думата. В случай, че думата е с повече от три срички ще има повече от една сричка в междинна позиция.

След края на обучението на модела, след обработката на всяка дума поотделно и акумулирането на резултатите, следва фазата на оценка на модела. Във фазата на оценка може да бъде получена реална мярка за качеството на предходните два процеса, събирането на обучителни данни и самото обучение на модела.

2.3 Проблеми при оценката на модела за автоматично сричкообразуване

Преди да бъдат разгледани проблемите при оценката на успеваемостта на модела, трябва накратко да бъде обяснен процеса по идентифициране на границите на сричките. Преди модела да „вземе решение“ за правилната комбинация от срички в думата която обработва, всички възможни комбинации от срички трябва да бъдат извлечени. Тази стъпка се повтаря и в процеса на обучение на модела, разгледан в точка 2.2. След извличането на всички възможни комбинации от срички, данните за тях, честотите им на срещане, и позициите им, се използват за намиране на правилната комбинация.

Ако се върнем на примера с думата „червен“ от точка 2.2, след обучението на модела е възможно да разполагаме с примерните данни за думата, показани на следното уравнение:

$$\text{че} = \text{че}_{[i,290]}, \text{че}_{[f,185]}$$
$$\text{рвен} = \text{рвен}_{[i,146]}, \text{рвен}_{[f,125]}$$
$$\text{чер} = \text{чер}_{[i,273]}, \text{чер}_{[f,254]}$$
$$\text{вен} = \text{вен}_{[i,233]}, \text{вен}_{[f,192]}$$
$$\text{черв} = \text{черв}_{[i,122]}, \text{черв}_{[f,108]}$$
$$\text{ен} = \text{ен}_{[i,157]}, \text{ен}_{[f,231]}$$

За целите на примера умишлено е избрана кратка дума, тъй като колкото по-дълга е една дума, толкова повече са възможните ѝ комбинации от срички. За най-дългата дума в българския език „непротивоконституателствувайте“, възможните комбинации от срички са 1613, а възможните срички са много повече.

Моделът проверява сумите на честотите на сричките, взимайки предвид и позициите за които са записани. По този начин, автоматичното сричкообразуване се свежда до решаване на следното уравнение уравнение:

$$\text{че}_{[i,290]} + \text{рвен}_{[f,125]} = 415$$
$$\text{чер}_{[i,273]} + \text{вен}_{[f,192]} = 465$$
$$\text{черв}_{[i,122]} + \text{ен}_{[f,231]} = 353$$

Както се вижда от резултатите на трите уравнения, сричките „чер“ и „вен“ са най-вероятните градивни елементи на думата „червен“. От примера става ясно каква е причината

за добавяне на още едно ниво на информация, използване на контекста на вече срещаните срички за по-прецизно предсказване на сричките в непознати думи.

В реална обстановка е възможно да се случи така, че да има уравнения с еднакви резултати. В такъв случай се използва сумата на честотите на сричките, без позициите да бъдат взимани под внимание. Ако отново се стигне до ситуация на равни суми, се пристъпва до използването на принципа за максимизиране на началната част на сричката, разгледан в точка 1.1.

Въпреки, че начина на работа на модела за идентифициране на границите на срички е ясен, съществуват трудности пред процеса по оценка на успеваемостта. Една от причините за трудностите е съставянето на златен стандарт[2]. Според Duanmu [2009], дори в английски език, език с ясно дефинирани лингвистични правила, съществуват разногласия между учените за правилното сричкообразуване на дума като „happy“. Имайки предвид неединодушието относно значението на сричката в речта, съществуването на множество верни комбинации между срички за една и съща дума не е учудващо.

При тестването на методите им за автоматично сричкообразуване, Goldwater and Johnson [2005] правят разграничение между думи изградени от поне две срички, и думи, изградени от какъвто и да е брой срички. При някои от тестваните методи се забелязва разлика в резултатите от няколко процента, до почти 30% успеваемост. Авторите доказват, че е по-лесно да се постигне по-висока успеваемост на идентифициране на границите на сричката при тестване върху данни, съдържащи голям брой еднослични думи, и редки струпвания на съгласни звуци.

Съдържанието на тестовите данни е важен аспект при оценката на подходи за автоматично сричкообразуване. Съществуват различни виждания по въпроса дали данните трябва да съдържат произволна извадка думи, или е нужно да има някакви критерии при подбора на тестовите данни. Ако тестовите данни съдържат голям брой еднослични думи, успеваемостта на модела е много по-висока. Това е така, защото при еднослични думи няма нужда от разбиване на струпани съгласни звуци [Goldwater and Johnson 2005].

Заключение

Представеният в текущия доклад модел за автоматично сричкообразуване е все още в процес на активна разработка. Това стана ясно в предходната точка където, само в общи линии, бяха очертани характеристиките на модела. Макар детайлите за начина на работа на модела да са достатъчни за базова имплементация, остават много въпроси за изясняване преди модела да придобие завършен вид.

Един от въпросите, които трябва да бъдат решени преди модела да бъде определен като завършен, е свързан с подхода за третиране на „дифтонги“. Дифтонгът е комбинация от два съседни гласни звука в една сричка. Това определение противоречи на дефиницията за сричка, дадено в точка 1. Въпреки противоречието, справянето с дифтонг, който е заобиколен от съгласни звуци не е проблем, тъй като обикновено само единият от гласните звуци в дифтонга е сричкообразуващ. Предизвикателство пред идентифицирането на границите на сричката са случаите в които съществува дифтонг, който не е заобиколен от съгласни звуци.

Събирането на „груби“ тренировъчни данни е друг проблем, който предстои да бъде решен. Както беше споменато в точка 2 има характеристики, които корпусът от тренировъчни данни трябва да притежава. Това усложнява проблема. В литературата са описани множество случаи в които се стига до създаване на модели, основани на машинно обучение, които служат за набавяне на тренировъчни данни. Както става ясно в точка 2, качеството на тренировъчните данни има много голямо значение за обучението на модела.

[2] Прието е данните, които служат за оценка на успеваемост на алгоритми и модели да се наричат златен стандарт.

Имплементацията на модела е процес, който изисква голямо внимание при избора на структури от данни и възможности за сериализация на самия модел. Сериализацията и десериализацията[3] са процеси, които позволяват тренираният модел да бъде съхраняван. Бързодействието е задължително качество, което прави избора на подходящи структури от данни още по-важен [Sulov 2014].

Литературни източници

1. Bartlett, S.; Kondrak, G. and Cherry, C. (2009). *On the Syllabification of Phonemes*, Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics : 308-316.
2. Duanmu, S., 2009. *Syllable Structure: The Limits of Variation*. OUP Oxford, .
3. Daz-Santiago, S.; Maria Rodriguez-Henriquez, L. and Chakraborty, D. (2014). *A Cryptographic Study of Tokenization Systems*, 4 : 6.
4. Goldwater, S. and Johnson, M. (2005). *Representational Bias in Unsupervised Learning of Syllable Structure*, Proceedings of the Ninth Conference on Computational Natural Language Learning : 112-119.
5. Goslin, J. and Frauenfelder, U. (2001). *A Comparison of Theoretical and Human Syllabification*, Language and Speech 44 : 409-436.
6. Habert, B.; Adda, G.; Adda-Decker, M.; de Marëuil, P. B.; Ferrari, S.; Ferret, O.; Illouz, G. and Paroubek, P. (1998). *Towards tokenization evaluation*, 98 : 427-431.
7. He, Y. and Kayaalp, M. (2006). *A Comparison of 13 Tokenizers on MEDLINE*, .
8. Huang, X.; Acero, A. and Hon, H.-W., 2001. *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Prentice Hall PTR, Upper Saddle River, NJ, USA.
9. Kahn, D., 1980. *Syllable-based Generalization in English Phonology*. Garland, .
10. Marchand, Y.; Adsett, C. and Damper, R. (2007). *Evaluating automatic syllabification algorithms for English*, Proceedings of SSW6 .
11. Marchand, Y.; Adsett, C. and Damper, R. (2009). *Automatic Syllabification in English: A Comparison of Different Algorithms*, Language and speech 52 : 1-27.
12. Mayer, T. (2010). *Toward a Totally Unsupervised, Language-Independent Method for the Syllabification of Written Texts*, Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology, SIGMORPHON 2010, Uppsala, Sweden, July 15, 2010 : 63-71.
13. Müller, K. (2006). *Improving Syllabification Models with Phonotactic Knowledge*, Proceedings of the Eighth Meeting of the ACL Special Interest Group on Computational Phonology and Morphology : 11-20.
14. Reitermanov, Z. (2010). *Data splitting*, WDS'10 Proceedings of Contributed Papers : 31-36.
15. Rogova, K.; Demuynck, K. and Van Compernolle, D. (2013). *Automatic syllabification using segmental conditional random fields*, COMPUTATIONAL LINGUISTICS IN THE NETHERLANDS JOURNAL 3 : 34-48.
16. Selkirk, E. (1984). *On the major class features and syllable theory*, Language Sound Structure .
17. Sulov, V. (2014). *On the Essence of Hardware Performance*, Research Journal of Economics, Business and ICT 9 : 13-18.

За контакти

Красен Пенчев, докторант
Икономически университет - Варна
krasen_penchev@ue-varna.bg

[3] Сериализацията е процес на преобразуване на структури от данни до поток от байтове, запазвайки формата и свойствата на данните. Десериализацията пресъздава структурите от данни, прочитайки потока от данни.