Тервел Вълков, специалност *Изкуствен Интелект*, ФН 9MI3400058

**Проект по Семантичен Уеб**

*Тема 4: Entity linking на кандидати за избори, върху данни от elections.ontotext.com*

1. Увод

В България, както и в останалите държави от Европейския съюз, по Директива 2003/98/ЕO следва публично да се предоставят данни за проведените избори. Централната избирателна комисия публикува голямо количество такива данни, които всъщност представляват дигитализирани версии на протоколите от избирателните секции.

За съжаление, макар и данните да са детайлни, те са трудни за обработка – форматите на данните са нестандартни и неконсистентни, и липсва добър механизъм за идентификация на кандидатите и партиите. Имаме уникални идентификатори за самите кандидатури, но за партиите имаме просто номер на бюлетината (който се избира случайно, и съответно ще бъде различен в различните избори), а за хората имаме просто три имена, което в общия случай не е достатъчно за точна идентификация.

Поради този проблем с идентификацията е трудно да използваме данните за да анализираме участията в различни избори на конкретни политици – не може директно да видим за даден човек в кои избори е участвал, и така например да проследим как през годините си е сменял партията.

Целта на проекта е да опитаме да решим точно този проблем – в данните да намерим кои кандидатури всъщност са били на едни и същи хора.

1. Entity Resolution

Задачата на проекта приспада към областта на *Entity Resolution*[1]. В този тип задачи боравим с данни, представящи някакви реални обекти, описани чрез множество от атрибути – например информация за граждани от някаква правителствена агенция, или информация за клиентите на дадена компания.

За съжаление, при такива данни често има проблем с липсата на ясна идентификация на тези обекти. Когато боравим с данни от различни източници, дори и същият обект да присъства в няколко от тях, всеки ще има своя собствена система от идентификатори, която няма как директно да бъде съгласувана със същата на друг източник. Но дори и да имаме единствено множество данни, в което може да има няколко записа отнасящи се за същия обект, нямаме гаранция че ще може директно да идентифицираме това – възможно е идентификаторите в системата въобще да не се отнасят за обектите от които се интересуваме, и съответно за същия обект да има няколко записа с различни идентификатори. Например в данни за покупки на даден магазин, ние може да се интересуваме от клиентите, а идентификаторите в данните да се отнасят за покупките, и същият клиент да присъства многократно под множество идентификатори за всяка негова поръчка.

Щом директната идентификация по някаква уникална стойност за всеки обект не е възможна, може да опитаме да използваме останалата информацията за тях. Но тук срещаме други проблеми – когато боравим с различни източници, всеки източник може да пази напълно различни типове информация за дадените обекти, и дори когато същото поле присъства в няколко източника, може то да съдържа различна информация в различните източници. Има множество причини за това – може да има разлика във формата и нивото на детайлност (например един източник да пази три имена, а друг само инициал и фамилия), може да е резултат от неясност по правописа на дадени имена, може просто да е станала техническа грешка при въвеждането, а може и разликата всъщност да е напълно коректна, и да отразява реална промяна състоянието на обекта през времето.

Целта на *Entity Resolution* е разрешаването точно на такива проблеми - при дадени данни да открием кои записи всъщност се отнасят за същите обект. Има няколко различни задачи в тази област – при *record linkage* (или *entity matching*) работим с няколко източника и искаме да открием и свържем записи в тях които всъщност се отнасят за същия обект. При *deduplication* имаме единствен източник, и искаме да открием записите в него отнасящите за еднакви обекти, и да елиминираме повторенията на данни.

В конкретния случай, задачата е подобна на *deduplication* – искаме да открием кои записи се отнасят за едни и същи кандидати за избори, но не правим това с цел елиминиране на повторения от данните, а просто искаме да може за даден човек да намерим всичките негови участия в различни избори.

1. Данни

Базовите данни за изборите са предоставени от ЦИК[4], но са преминали допълнителна обработка и са въведени в онтология[3]. В този си вид те са достъпни на <https://elections.ontotext.com/>. Включват всички избори от 2013 година насам, като допълнително е извършено свързване на партиите между отделните избори и юрисдикции, и е добавена административната йерархия на България, свързана с Wikidata.

Конкретните данни използвани за проекта са на ниво кандидатура – за всяка имаме кандидатът (идентифициран само по три имена), партия (локална и главна) и информация за самите избори (тип, МИР – многомандатен избирателен район, община, имена , дата). Получени са чрез заявки към упоменатия endpoint, и изтеглени като *.csv* файлове. За работа с готовите данни се използва популярната библиотека *pandas*[8], като те се зареждат от .csv във dataframe, и се добавя нова *id* колона за по-лесна работа по-нататък. Взимаме само част от полетата – основно премахваме тези, отнасящите се за IRI-тата на различните обекти в онтологията (освен за самата кандидатура), и оставяме само съответните етикети, за по-четима демонстрация на данните. Извършва се съвсем лек preprocessing – конвертиране на всички низове до малки букви, тъй като в оригиналните данни не е консистентно изписването на имената, и в някои случаи те са изписани само с главни букви.

1. Анализ на данните

Възможните типове избори са общински (municipal), парламентарни (parliamentary), европейски (european) и президентски (presidential). Само общинските избори имат информация за община, и само общинските и парламентарните имат информация за МИР – за европейските и президентските имаме само партии и самите избори. Освен това, само в общинските и парламентарните избори имаме разделение на информацията за самите избори на две нива – по-общо за самия изборен цикъл (например „Парламентарни Избори 2017“) и по-специфично за неговата конкретна локална част (например „Избори за Парламент на РБ МИР 01. БЛАГОЕВГРАД“).

Общо данните се състоят от 110462 реда, като най-голямата част се отнасят за общински избори (около 72 хиляди реда), последвани от парламентарните избори (около 38 хиляди реда). Има 72186 уникални имена на хора, но след конвертиране на низовете до малки букви, получаваме 68037 уникални стойности – с около 4000 по-малко. Уникалните партии са съответно 1212 за местните и 111 за главните, като при конвертиране на низовете само местните намаляват, с около 30. Някои имена на хора се срещат много често – например Георги Иванов Георгиев се среща цели 58 пъти, Иван Георгиев Иванов – 46, а Петър Иванов Петров – 39.

При анализа на данните беше забелязана нещо интересно при местните избори от 2019 година. По неизвестна причина, ЦИК са третирали ситуациите с коалиции на партии по леко объркващ начин – същата кандидатура (не просто същото име на човек, а и същото IRI на кандидатура) има няколко записа, като локалната партия е съответната коалиция, а в главната партия е записана една от партиите-членове на коалицията. Така в данните изглежда че същият човек едновременно е член на няколко партии, и не става ясно всъщност точно към коя партия от коалицията той принадлежи. Един такъв пример е Борислав Константинов Банчев, който според данните едновременно е бил в 5 партии по време на Врачанските местни избори за общински съвет. Има и други кандидати, които са в подобна ситуация, със записи показващи едновременно участие в 4-5 партии, както и множество които са с по 2 участия (при по-малките коалиции).

Подобни аномалии не забелязах при изборите от предишните години, така че вероятно това е резултат от някаква специфика в представянето на данните през 2019 година.

1. Реализация

За целите на проекта беше избрана библиотеката *RecordLinkage*[6], тъй като тя поддържа работа върху единствено множество данни (докато *Magellan* например поддържа само свързване на две отделни таблици). След зареждане на данните, започваме със създаването на *Index* обект от *RecordLinkage*, чрез който ще получим двойки записи от данните, за които след това да проверим дали реферират към същия човек. Възможно е пълно индексиране, при което за всеки запис създаваме двойка със всеки друг запис, но това води до огромен брой двойки, поради което използваме *blocking* – така създаваме само двойки от записи които съвпадат по дадени атрибути, в случая име. Така получаваме 98056 кандидат-двойки.

За сравнение на записите в двойките създаваме *Compare* обект, към който добавяме различни правила за сравнение. В случая използваме най-простото правило – *exact*, което просто проверява дали съответните стойности на даден атрибут съвпадат или не. В библиотеката се поддържат и по-сложни проверки – в случая, тъй като работим с низове, бихме могли да ползваме правилото *string*, което може да се конфигурира с няколко метрики за сходство, като косинусова, разстояние на Левенщайн, и др. Освен това, възможно е да създадем и наш собствен алгоритъм, чрез който да извършваме по-специфични проверки според конкретната задача.

За полетата които ще проверяваме обаче, не очакваме да имаме ситуации с различни изписвания на същото име, и съответно предполагаме че точното съответствие е достатъчно като проверка. Въвеждаме проверки за местната и главна партия, общината и МИР, като няма нужда да проверяваме имената, тъй като сме подбрали кандидат-двойки които вече съвпадат по тях. След като имаме правилата за проверка, ние ги прилагаме и получаваме таблица, в която за всяка двойка имаме резултата от всяка проверка.

С този резултат вече трябва да решим за кои двойки ще считаме че съвпадат. Един прост вариант е да филтрираме двойките които са минали през поне една от проверките – това постигаме като просто сумираме резултатите от всички проверки, и филтрираме тези със сума по-голяма или равна на 1. Така получаваме записите с еднакви имена (тъй като кандидат-двойките са подбрани като такива) и съвпадение в полетата за главна партия, локална партия, община, МИР, или едновременно няколко от тях, което общо ни дава 70007 двойки. Ако искаме да сме по-стриктни, може да изискваме поне две съвпадения, при което ще получим 45719 двойки, или поне три за още по-стриктно филтриране, при което ще имаме само 23669 съответствия. Алтернативен вариант е да претеглим проверките – те имат стойности само 0 или 1, но е логично да предположим че съвпадения в някои полета са по-важни от други. Например, ако счетем че е по-вероятно даден политик да смени партията си отколкото да смени областта или общината в която участва в избори, то област и община ще са по-стабилни индикатори, които е по-вероятно да останат същите през годините, и съответно може да ги оценим с по-голямо число – например да оценим съвпадение на партия с 0.5, а съвпадение на община с 1.5, и да променим прага за филтриране съобразно с това. Тук би се изисквало експериментиране с много различни комбинации от стойности, и за съжаление поради липсата на тестови данни е трудно резултатът да се оцени и така да преценим кои стойности са оптимални. Една примерна комбинация е 0.75 за локалната партия, 0.5 за главната (по-ниско число поради описания случай с коалиции в секция 4.), 1.5 за МИР и 1.25 за община, с праг за филтриране – сума на стойностите на проверките поне 1.25.

Финално, искаме да запишем резултатите във файл. Получената таблица е dataframe с мулти-индекс (съдържащ двойки от id-та на съвпадащи записи), така че за по-лесна работа с него конвертираме индексът до обикновени колони, и групираме по първата колона с id-та. Така за всяка кандидатура (за който има съвпадения) получаваме група с всичките ѝ съвпадения, след което обхождаме тези групи и във файла *output.txt* записваме информация за всеки кандидат – име, и полетата от записите които считаме че му принадлежат. При простия вариант с филтриране на записи с поне едно съвпадение измежду търсените полета, имаме 38558 групи, докато при примерния претеглен вариант получаваме 35802.

1. Резултати

Според конкретните параметри може да получим различен брой кандидати със съответствия – например при простия вариант с търсене на едно съвпадение получаваме информация за 38558 кандидата, като за по-голямата част от тях (около 22000) намираме само по 2 записа, но има и около 1000 случая с по 6 записа, и даже няколко с по над 10 записа.

Резултатите не са перфектни, но все пак имаме полезна информация – например, един произволно избран кандидат е Георги Стойчев Йорданов, за който може да намерим следните записи, които алгоритъмът е избрал за съвпадащи (съкратени за целите на демонстрацията):

*local\_party\_label, main\_party\_label, mun\_label, mir\_label, main\_el\_label, elLabel*

* *Местна коалиция АТАКА (ПП АБВ, ПП АТАКА), Атака, Община Каварна, Добрич, Местни Избори 2019, Местни Избори 2019 за общински съвет 0817. Каварна*
* *ПП АТАКА, Атака, Община Каварна, Добрич, Местни Избори 2015, Местни Избори 2015 за общински съвет 0817. Каварна*
* *ОБЕДИНЕНИ ПАТРИОТИ – НФСБ, АТАКА и ВМРО, ОБЕДИНЕНИ ПАТРИОТИ – НФСБ, АТАКА и ВМРО, nan, Добрич, Парламентарни Избори 2017, Избори за Парламент на РБ МИР 08. ДОБРИЧ*
* *АТАКА, Атака, nan, Добрич, Парламентарни Избори 2014, Избори за Парламент на РБ МИР 8*
* *АТАКА, Атака, nan, Добрич, Парламентарни Избори 2021, Избори 2021 за Парламент на РБ МИР 08. ДОБРИЧ*
* *АТАКА, Атака, nan, Добрич, Парламентарни Избори 2021, Избори 11 Юли 2021 за Парламент на РБ МИР 08. ДОБРИЧ*
* *АТАКА, Атака, nan, Добрич, Парламентарни Избори 2021, Избори 14 Ноември 2021 за Парламент на РБ МИР 08. ДОБРИЧ*

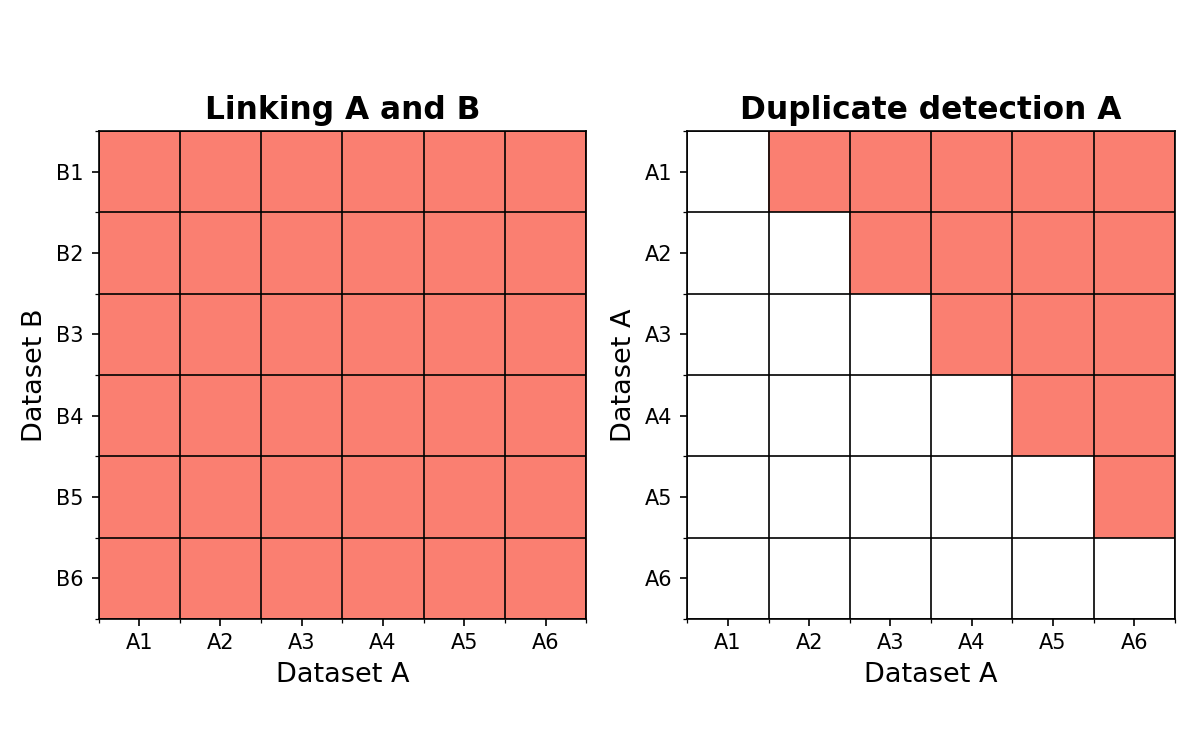
Партийната принадлежност е консистентна, като има два случая с коалиции, но те включват партията която присъства в другите записи. Районът също е консистентен, като в двата местни избора имаме и съвпадение на общината.

Има и случаи които показват че може би не сме достатъчно стриктни при проверките – например Николай Колев Диков, за който имаме:

* *ПП Глас Народен, nan, nan, Добрич, Парламентарни Избори 2021, Избори 2021 за Парламент на РБ МИР 08. ДОБРИЧ*
* *ПП ГЛАС НАРОДЕН, Глас народен, Община Добрич, Добрич, Местни Избори 2015, Местни Избори 2015 за общински съвет 0828. Добрич-град*
* *ПП Глас Народен, nan, nan, Варна, Парламентарни Избори 2021, Избори 2021 за Парламент на РБ МИР 03. ВАРНА*
* *ПП ГЛАС НАРОДЕН, Глас народен, nan, Варна, Парламентарни Избори 2021, Избори 11 Юли 2021 за Парламент на РБ МИР 03. ВАРНА*
* *ПП ГЛАС НАРОДЕН, Глас народен, nan, Варна, Парламентарни Избори 2021, Избори 14 Ноември 2021 за Парламент на РБ МИР 03. ВАРНА*

Тук важни са първи и трети запис, които се отнасят за същите избори, но имат различен район. Може би това всъщност е възможно, не съм достатъчно запознат със спецификите на организацията на изборите, но изглежда съмнително че същият човек едновременно ще има тези две участия.

Нещо което забелязваме в резултатите е че имаме голямо количество съвпадения на хора за които вече знаем че са същите (тъй като съвпадат не само по име, а и по идентификатора на кандидатура, присъстващ в поле “s” в данните), което се получава поради проблема с коалициите описан в секция 4. Ако счетем, че тази информация е безполезна за нас, може да искаме да премахнем тези случаи – един прост начин за това е да добавим сравняване и по полето “s”, но след това да умножим тези стойности по голямо отрицателно число, така че при филтрирането такива двойки записи автоматично да паднат под прага и да не бъдат включени в резултата.

Друго неприятен аспект на резултата е че имаме до някаква степен повторение на информация – за същото име може да имаме няколко групи съвпадения, които споделят записи, като например първата съдържа записи A, B, C и D, втората съдържа записи B, C и D, а третата само записите C и D. Така информацията че записите B, C и D се отнасят за същия човек присъства на две места, а че C и D се отнасят за същия човек – на три места. Това най-вероятно се получава поради метода по който се генерират кандидат-двойките – за записа A ще получим двойките (A, B), (A, C) и (A, D), след което за B ще получим двойките (B, C) и (B, D), без (B, A) тъй като вече я имаме – взимаме само едната посока на съвпадението). Евентуално, при групирането ще създадем група за A, която ще включва B, C и D, и отделно групи за B и C.

Потенциално може това да се елиминира с допълнителна обработка на групите, така че някак да ги обединим в тези случаи, и във финалния резултат да запишем само една група, съдържаща всичките записи заедно. Не всички случаи на няколко групи записа за същото име обаче спадат към тази категория – например за Веселин Найденов Марешки имаме общо 7 отделни групи, като 2 от тях са с по 5 записа, и съвпадат по 4 от тях, тоест по-голямата част от информацията се повтаря, но не напълно.

1. Заключение

В този проект, чрез библиотеката *RecordLinkage* върху данните от elections.ontotext.com беше приложен сравнително прост алгоритъм за намиране на записи принадлежащи на същия човек. Резултатите са далеч от перфектни, но все пак има някакъв прогрес в идентифицирането на кандидати, и потенциално резултатните данни биха могли да бъде подложени на допълнителна обработка, за да се извлече от тях по-полезна информация.

Възможно е след промени на алгоритъма да получим по-добри резултати - има различни параметри с които може да се експериментира, като например прагът на оценка за съвпадение, според който решаваме дали наистина два записа се отнасят за същия човек или не. В проверките използваме само едно подмножество от полетата – потенциално бихме могли да включим и информацията за датата и самите избори в един по-усъвършенстван алгоритъм (например ако забележим, че две кандидатури със същото име са участвали на различни избори в същия изборен цикъл, да може да заключим че те няма как да съвпадат).

Една интересна, но най-вероятно трудна посока на разработка би била да използваме по някакъв начин информация за връзки между партии. Например, ако имахме данни че партия А е създадена от бивши членове на партия Б (която вече не съществува), и намерим два записа на същия човек, който по-рано е бил от партия Б, но вече е от партия А, то бихме могли с по-голямо ниво на сигурност да заключим че той е същият човек – смяната на партията е резултат от разпадането на партия Б и преминаването на част от членовете към новата партия А.

За съжаление, тук липсват тестови данни, както често се случва в този тип задачи, поради което е трудно да оценим резултатите по математически начин. Макар и данните тук да не са чак такова огромно количество (относително, имайки предвид масивните данни върху които често се налага да се прилагат такива алгоритми), все пак ръчното им преглеждане не е лесно осъществимо, и съответно не е лесно да заключим до колко успешно всъщност сме решили задачата.

1. Използвана литература
2. Basics of Entity Resolution, Kyle Rossetti & Rebecca Bilbro, District Data Labs

[<https://www.districtdatalabs.com/basics-of-entity-resolution>]

1. An introduction to Entity Resolution — needs and challenges, Sonal Goyal, Towards Data Science

[<https://towardsdatascience.com/an-introduction-to-entity-resolution-needs-and-challenges-97fba052dde5>]

1. 5-Star Linked Open Elections Data, Nikola Tulechki, Ontotext

[<https://www.ontotext.com/blog/5-star-linked-open-elections-data/>]

1. Presentation on 5-Star Linked Open Elections Data, Nikola Tulechki, Ontotext

[<https://docs.google.com/presentation/d/1bGfJPm0SBxWebtiVnIcZ5F0lbTvkez-fFO3QhUKdo8Y>]

1. Semantic Elections, Nikola Tulechki, GitHub

[<https://github.com/nikolatulechki/semanticElections>]

1. RecordLinkage

[<https://github.com/J535D165/recordlinkage>]

1. Python Record Linkage Toolkit Documentation

[<https://recordlinkage.readthedocs.io/>]

1. pandas

[<https://pandas.pydata.org/>]

1. pandas Documentation - API reference

[<https://pandas.pydata.org/docs/reference/index.html>]