Борислав Ризов

ПРЕМАХВАНЕ НА СЕМАНТИЧНА МНОГОЗНАЧНОСТ СЪС СКРИТИТЕ МАРКОВСКИ МОДЕЛИ

Увод

В компютърната лингвистика възниква следната кратко дефинирана задача: Даден ни е текст. Да се определят значенията на думите в текста. И както често се оказва, просто дефинираните задачи имат трудни отговори и решения. Задачата получава подобаващо внимание и причината за това е ясна. Тя е основна за много практически приложения, а нейното решение се нуждае от прилагането на разнообразни методи и източници на информация. В определени случаи определянето на значението е трудна задача и за хората – те се нуждаят от голям контекст и широко познание за света, за да го направят. Тъй като задачата е дефинирана доста свободно, една от първите трудности е да се назове какво е значение. Отговорът не е еднозначен и лесен (Агире и Едмъндс 2006). Изборът на формулировка на значение повлиява и избора на подход за категоризация на думите в текста (определяне на тяхното значение) с тези значения и обратно. В този текст ще предложим едно наивно приложение на един комплексен подход за решаване на горната задача. За значения на думите ще използваме синонимните множества от Българския WordNet (Фелбаум 1998, Коева и др. 2004). За определяне на тяхното значение в даден текст ще моделираме този текст с дискретен времеви модел, по-точно със *Скрития модел на Марков* (СММ). Този добре познат и широко използван статистически метод е подходящ при опитите за разрешаване на назования проблем. СММ може да се класифицира като подход от типа методи с управлявано обучение (supervised learning), но донякъде може да се разглежда и като хибриден, давайки възможност за самообучение.

Управлявано и неуправлявано обучение

Повечето алгоритми за премахване на многозначност се характеризират в една от категориите *управлявано обучение* (supervised learning) и *неуправлявано обучение* (unsupervised learning). При управляваното обучение знаем истинската стойност, която ни интересува, за данните, с които тренираме програмата. В нашия случай това са значенията на думите в тренировъчни текстове. От друга страна, при неуправляваното обучение нямаме класификация на входните данни, може дори да нямаме и предварително дефинирани стой-

ности, с които да ги класифицираме. Затова често неуправляваните алгоритми са определяни като *клъстеризиращи* – разделящи подобните феномени в множества (клъстери), които се явяват като резултат. Създаването на тренировъчни данни (тяхното анотиране) за премахването на семантична многозначност, а и за много други подобни задачи, е много трудна и продължителна дейност и затова е желателно да се използват и неанотирани тренировъчни данни. В много случаи системите в начален етап на анализ използват анотирани тренировъчни данни, а по-нататък им се подават неанотирани за самообучение. В конкретната задача, която сме си поставили, прилагането на СММ е много подходящо за такъв комбиниран подход. От една страна, моделът се обучава с данните от анотиран корпус, Семантично анотирания корпус на българския език (Коева и др. 2006), после може да се обучи с чисти текстове, като се прилага алгоритъм за максимизиране на очакването като Баум-Уелч. За да направим всичко това, първо трябва да дефинираме множеството от възможни значения на думите.

Избор на множество от значения на думите и WordNet.

За нуждите на експеримента трябва да се използва речник, който да дефинира значенията. Така анотирането на тренировъчния текст се свежда до определянето на най-близката дефиниция на думата (от предложените в речника) до употребата, която се класифицира. В нашия случай речникът е Българският WordNet, а, по-конкретно, представители на значенията на дадена дума са синонимните множества, в които думата участва.Днес WordNet е най-използваният лингвистичен ресурс за премахване на семантична многозначност. Това има своите предимства и недостатъци. Едно предимство е наличието на лингвистични мрежи WordNet за много езици. От тази гледна точка създаването на анотирани корпуси, семантично анотирани със значенията в WordNet, има големи предимства пред използването на други речници. Друго предимство е йерархичната структура на синонимните множества, зададена от релацията хиперонимия. Тази и другите семантични релации могат да се използват за изграждането на по-добри модели при представянето на езика. Недостатък спрямо така поставената задача представлява ситното гранулиране на значенията в WordNet – една дума е представена с много, понякога трудно различими, значения. По-нататък ще обосновем, че едно подходящо групиране на хипонимните значения спрямо съответното хиперонимно значение може да представлява приложение на предимствата на WordNet за преодоляването на недостатъците му.

Скрити марковски модели

В математиката Марковска верига, по името на Андрей Марков, е дискретен (времеви) стохастичен процес, изпълняващ условията на Марков (Рабинер 1989). Във всеки времеви момент системата се намира в някакво

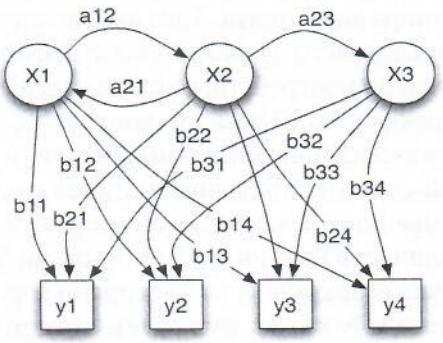
състояние. Преминаването в друго или оставането в дадено състояние при преминаването в следващ времеви момент се нарича *преход*. Преход от състояние в състояние става с някаква вероятност. Приемат се и начални вероятности за това в кое състояние се намира системата в началния момент.Ето условията на Марков:

1. Само текущото състояние дава информация за бъдещото поведение на процеса. Историята на процеса не внася допълнителна информация.
$P(X(t_{n+1})=s_{n+1}|X(t_n)=s_n, X(t_{n-1})=s_{n-1},..., X(t_0)=s_0) = P(X(t_{n+1})=s_{n+1}|X(t_n)=s_n)$

2. Процесът е хомогенен по отношение на времето, т.е.:
$P(X(t_{n+1})=si|X(t_n)=s_j) = P(X(t_{k+1})=si|X(t_k)=s_j)$

Скрит модел на Марков е Марковски процес с неизвестни параметри. Целта на разглежданията е да се установят скритите параметри, като се използват видимите. За разлика от обикновения Модел на Марков, при СММ състоянията са скрити за наблюдаващия, но всяко състояние извежда видими букви. Всяко състояние разполага с вероятностно разпределение на извежданите символи. По този начин изходната поредица предоставя някаква информация за преминатите състояния.

x – състояния
y – изходи
a – вероятности на преходите
b – вероятности на изходите
$\Sigma_j a_{ij} = 1, \Sigma_j b_{ij} = 1$



n-грами (n-grams)

СММ е пример за n-грамен езиков модел и по-точно, при дефинираните горе условия на Марков, биграмен (разбира се може да се използват n-грами от по-висок ред). n-грам е поредица от n символа от входната азбука, с която работим (напр. думи, синтактични категории и т.н.), а n-грамните модели се използват, за да се предсказва всеки символ в поредица, ако са известни предходните n-1 символа и вероятностните наблюдения в тренировъчен корпус. Въпреки че този основен принцип е много прост, възниква сериозният проблем, че има много повече възможни n-грами в езика, отколкото можем да извлечем от тренировъчен корпус. Друг проблем е, че тренирайки модел с фиксирано тренировъчно множество, правим модела неподходящ за специфични текстове. Например, ако моделът се тренира с юридически текстове, може да сме сигурни, че моделът няма да е подходящ за медицински текстове. Тази статичност се проявява още и в това, че се използват определени думи и моделът не би се справил с всякакви новопоявили се думи (Манинг и Шутце 1999). Можем да преодолеем до голяма степен посочените проблеми като дефинираме детерминистична функция E,

която прави някои от думите еквивалентни. Ако в определените класове на еквивалентност има повече от един елемент, по този начин се редуцира разглежданото множество от думи и контексти. Това намалява нуждата от памет за съхраняване на модела. Друго предимство е, че се намаляват необходимите примери в тренировъчния корпус, които биха представили един добър модел. Моделът става приложим и в случаите, когато се класифицират думи, рядко срещани или несрещани в корпуса. Използването на класовете води до загуба на диференциацията между отделни контексти, равностойна на отстраняване на многозначност, но това може да се компенсира с използване на n-грами от по-висок ред.

СММ за премахване на семантична многозначност

При решаване на задачата за отстраняване на семантична (или друга, например по част на речта) многозначност трябва да се уточнят неизвестните в постановката на СММ: Какво представлява процесът? Кои са състоянията и изходите? Подхождаме, като представяме текста като дискретен времеви процес – генериране на текст дума по дума във всеки такт от процеса. Тоест във всеки времеви отрязък системата се намира в дадено състояние и извежда като изход по една дума. Състоянията на системата, които са скрити за нас, са значенията на думите. Значенията на думите определяме, като намираме най-вероятния път през състоянията, който системата преминава, докато извежда текста. За да се приложи този метод, е необходимо да попълним три вероятностни матрици:

tp – вероятностите на преходите между състоянията

ep – вероятностите на изходите

sp – началните вероятности системата да се намира в дадено състояние

Тройката <sp, tp, ep> представлява *Скрития модел на Марков*.При определянето на вероятностите са възможни два основни подхода (Манинг и Шутце 1999):

1. Имаме предварителна представа (или хипотеза) за разпределението – параметричен подход. Необходими са по-малко тренировъчни данни, тъй като трябва да се изчислят сравнително малко параметри. Може да се предвиди колко такива данни са необходими, за да се твърди, че е направено добро статистическо приближение.

2. Разчитаме единствено на тренировъчни данни. При този подход е необходимо по-голямо количество тренировъчни данни, за да се приближи реалното разпределение.В някои случаи може да се разчита, че разпределението е гладка функция, да се използва непрекъснато разпределение и да се направи интерполация. Тогава честотите трябва да се изгладят и модифицират, за да се използват ограничените количества данни.

Експериментът

В тази наивна реализация използваме само тренировъчни текстове, за да определим вероятностния модел. За целта преброяваме честотите на раз-

познатите преходи и входни точки и ги използваме за изчисляването на вероятностите. Например $tp_{ij} = C(t, i, j) / \Sigma_k C(t, i, k)$, където C(t, a, b) е честотата на преходите от a в b.

Необходимо е голямо количество данни, за да има някакво разпределение на вероятностните функции, дори без да се изисква приближаване към реалното разпределение. Понастоящем в Българския wordnet (Коева 2008) има повече от 31 хиляди значения. Задачата, да се попълни матрица с преходи tp, като се използва ръчно анотиран корпус, е трудно изпълнима. За да се доближим до реалното разпределение, са необходими повече примери. Например, ако хвърлим монета 2 пъти и се окаже, че се е паднало ези, можем ли да твърдим, че вероятността да се падне ези, е 100% – очевидно не! Но ако сме я хвърлили 1000 пъти и 800 са се паднали ези, то с голяма увереност можем да кажем, че е крива и в 4/5 от случаите се пада ези.

В действителното приложение е направено леко изглаждане на модела при използването на WordNet. Честотите на всички значения (синонимни множества) са малко завишени, така всяко значение има ненулева начална вероятност. По подобен начин са увеличени честотите на извежданите думи от синонимно множество (самите думи в синонимното множество). Разбира се, при това моделиране, примерите от корпуса имат много по-голяма тежест.

Тренировъчният корпус, създаден в Секцията по компютърна лингвистика (Коева и др. 2006), се разделя в отношение 3:1. Големият дял се използва за тренирането на модела. Малкият – за тестване. Не е необходимо да се търси най-вероятният път от състояния, който генерира целия текст. Достатъчно е да се намерят състояния между поредните немногозначни думи. В тях търсенето се рестартира. Ще дадем един условен пример. Да си представим, че пътуваме от град Незнаен до град Неизвестен и всички пътища между двата града минават през село Междинно. Става ясно, че сме принудени да преминем през Междинно, затова трябва да минимизираме пътя от Незнаен до Междинно и пътя от Междинно до Неизвестен. Очевидно тези оптимизации не са зависими помежду си. По тази причина всяка една от частите на тренировъчния корпус е преобразувана в поток от многозначни поредици, като за краища на поредиците служат еднозначни или неразпознати думи. За откриването на най-вероятния път се използва алгоритъмът на Витерби (Рабинер 1989). Този алгоритъм има квадратична сложност и предложеното разделяне на текста значително ускорява приложението му.

Резултати:

При така описаното наивно приложение получихме следните резултати:
покритие: 0.40279
прецизност: 0.667891

Покритието е отношението на изходните думи към входните, а прецизността – отношението на правилните към изведените. Първото нещо, което

веднага се забелязва, е ниското покритие. То е предизвикано от това, че в някои поредици алгоритъмът не намира път, който да изведе цялата поредица. Това е така, защото двете думи, при които се къса редицата, не са били срещнати при тренирането. За да се преодолее този проблем, при всеки случай на прекъсване алгоритъмът се рестартира. Новата оценка е следната:

покритие: 0.939349

прецизност: 0.621822

В посочените данни са взети под внимание само многозначните поредици.

Според изложените по-горе съображения е подходящо групирането на значенията – всяко значение се заменя със своя първи хипероним, ако има такъв. Това довежда до малко по-лоши резултати. Направеният анализ показва, че хиперонимията „повдига" значенията неравномерно. В Wordnet има както много дълбоки йерархии от значения, така и много плитки.

Оценка на резултатите

За някои текстове 60% вярност е отличен резултат, за други 70% е недостатъчен. На резултатите не може да се гледа като на абсолютни стойности и на тази основа да се преценява доколко те са добри или не. Резултатите са достоверни само при тестове с много големи обеми текстове, представителни за езика. Извършването на подобни тестове в този момент изглежда нереално. Затова практиката е да се определят сравнителни стойности за едни и същи тестови данни. Подходящо е да се дадат долна и горна граница на верността, които да показват доколко е релевантен прилаганият алгоритъм. За долна граница се използва някой прост и лесен за имплементация алгоритъм. Подходящ е например алгоритъмът, който приписва на всяка дума най-често срещаното ѝ значение. За горна граница се използва човешката преценка или по-точно степента на съгласие на двама или повече анотиращи. Използването на долна граница до голяма степен е достатъчно – тя отразява ентропията на текста и показва доколко използваният подход е подобрение в сравнение с най-елементарните методи.

Въпреки че не предлагаме сравнителни данни, абсолютната стойност на резултатите е достатъчно висока, за да се приеме, че използването на CMM е подходящо средство за отстраняване на многозначност.

Бъдещи планове

Резюмирайки слабите места на предложената реализация, предлагаме следните насоки за тяхното решаване:

1. Използването на синтактичен анализ преди прилагането на CMM може значително да подобри резултатите, както при покритието, така и при прецизността (в първата им форма, т.е. преди рестартирането на веригите). Ето един типичен пример, който създава проблеми при тренирането на модела:

Засадих ябълково и крушово дърво. Без синтактичен анализ корпус, в който се среща примерът *Засадих ябълково дърво*, не би дал тренировъчния материал, който може да подготви модела за примера.

2. Групиране на значенията (организирането им в класове на еквивалентност) ще отстрани до голяма степен негативите от малкия корпус за трениране. Ако се върнем към горния пример, това означава в корпуса да се среща примерът *Посадих овощно растение*.

3. СММ се съчетава подходящо с други методи за премахване на многозначности, които може да се разположат преди или след неговото прилагане. При различно представяне на множеството от значения може да се използват и различни версии на алгоритмичната схема.

ЛИТЕРАТУРА

Агире и Едмъндс 2006: *Eneko Agirre and Philip Edmonds* (Ed.). Word Sense Disambiguation, Springer.

Коева и др. 2004: *Koeva, Sv., S. Mihov, T. Tinchev*. Bulgarian Wordnet – Structure and Validation. – In: Romanian Journal of Information Science and Technology, volume 7, numbers 1-2: 61-78.

Коева и др. 2006: *Koeva, Sv., Sv. Leseva, I. Stoyanova, E. Tarpomanova, M. Todorova*. Bulgarian Tagged Corpora. – In: Proceedings of the Fifth International Conference Formal Approaches to South Slavic and Balkan Languages, 18-20 October 2006, Sofia, pp. 78-86.

Коева 2008. *Koeva, Sv.* Derivational and Morpho-Semantic Relations in Wordnet. – In: Intelligent Information Systems, pp. 359-368.

Манинг и Шутце 1999: *Manning, C. and H. Schutze*. Foundations of Statistical NLP. MIT Press.

Рабинер 1989: *Rabiner L.* A tutorial on Hidden Markov Models and selected applications in speech recognition. Proceedings of the IEEE 77 (2), pp. 257-286.

Фелбаум 1998: *Fellbaum, Ch.* (Ed.). WordNet: An electronic lexical database. Cambridge, MA: MIT Press.

Янг и др 1995: *Young S., G. Evermann, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, P. Woodland*. The HTK Book (rev. 2002).