Vektoralapú felügyelet nélküli jelentés-egyértelműsítés nagyméretű tanuló korpuszok esetében

Papp Gyula

Pázmány Péter Katolikus Egyetem, Információs Technológiai Kar
Interdiszciplináris Műszaki Tudományok Doktori Iskola
1083 Budapest, Práter utca 50/a
gyupa@digitus.itk.ppke.hu

Kivonat A cikk felügyelet nélküli jelentés-egyértelműsítési (JEÉ) algoritmusok egy lehetséges javítását mutatja be. A módosítás kulcsa a módszerek által alkalmazott tanítóhalmazok méretének megnövelése. Végeztünk egy kísérletet, amely során többféle vektoralapú felügyelet nélküli JEÉ algoritmust teszteltünk a SenseClusters ([1]) programcsomag segítségével. A módszerek kiértékeléséhez egy szabványos adathalmazt, a Senseval-3 JEÉ verseny ([2]) angol főneveit használtuk. A Senseval-3 tanulóadatok mellé a British National Corpus-ból gyűjtöttünk környezeteket annak érdekében, hogy növeljük az algoritmusok tanulóadathalmazainak méretét. A Senseval-2 verseny főnevein végzett paraméterhangolás után az eredmények javulást mutatnak a bővített méretű tanulóhalmazok alkalmazása esetén. Az így kapott rendszer versenyképes a legjobb felügyelet nélküli JEÉ rendszerekkel, például a HyperLex ([5]) algoritmussal.

Kulcsszavak: jelentés-egyértelműsítés, felügyelet nélküli jelentés-egyértelműsítés, környezet-reprezentáció

1. Bevezetés

A jelentés-egyértelműsítés (JEÉ) a nyelvtechnológia egyik legkutatottabb területe. A feladatnak két fő megoldási módszere van: a felügyelt és a felügyeleti nélküli gépi tanulás.

Felügyelt tanulás esetén kézzel címkézett szövegre van szükség ahhoz, hogy a vizsgált többjelentésű szó (a továbbiakban célszó) aktuális jelentését el tudjuk dönteni egy bizonyos környezetben. A felügyelet nélküli módszerek viszont nem igényelnek címkézett tanuló mintákat (környezeteket). Sőt, a célszó aktuális jelentését sem egy előre megadott jelentéslistából választják ki, mint a felügyelt tanulást alkalmazó módszerek. A felügyelet nélküli rendszerek célja a célszó különböző használati eseteinek elkülönítése. (Felügyelet nélküli esetben a szójelentés helyett a használati eset kifejezés használatos.) Klaszterezési algoritmusokat használnak a hasonló tanító minták csoportosításához. Szerencsés esetben az így kialakított klaszterek a célszó egyes használati eseteit reprezentálják.

Korábban még nem látott példa esetén az aktuális környezet reprezentációjához leginkább hasonló klasztert tekintjük az algoritmus által vélt használati esetnek.

A felügyelet nélküli JEÉ egyik előnye a felügyelt módszerekkel szemben az, hogy nem igényel a célszó jelentéseivel címkézett tanítókorpuszt. Mivel olyan címkézetlen környezetből, amely tartalmazza a célszót, tetszőlegesen sok gyűjthető, felvetődött, hogy érdemes lehet megnövelni a felügyelet nélküli algoritmus tanító mintáinak a számát. Feltételeztük, hogy több tanító példára jobb eredményt adnak a felügyelet nélküli JEÉ algoritmusok.

Sok felügyelet nélküli JEÉ módszer vektoralapú reprezentációt használ, amit [4] vezetett be. Emellett gráfalapú módszerekkel is jó eredmények születtek, például mind a HyperLex algoritmus ([5]), mind az optimalizált változata ([3]) nagyon jól teljesít.

Ennek a cikknek a célja annak bemutatása, hogy a környezetek számának növelése hogyan befolyásolja a felügyelet nélküli JEÉ algoritmusok teljesítményét. Elvégeztünk egy kísérletet több vektoralapú módszeren annak vizsgálatára, hogy több tanító környezet esetén javul-e a klaszterezési teljesítmény. A következő fejezet bemutatja a vektoralapú JEÉ módszerek lényegét. Ezt követően bemutatjuk a kísérlet során vizsgált programcsomagot. A 4. fejezet tárgyalja magát a kísérletet. Az elért eredményeket foglalja össze az 5. fejezet. Végül egy rövid összefoglalással zárul a cikk.

2. Vektoralapú felügyelet nélküli JEÉ

A vektoralapú JEÉ algoritmusoknak minden egyes célszóhoz szükségük van egy-egy tanítókorpuszra. Minden egyes korpusz olyan környezetekből áll, amelyek tartalmazzák az aktuális célszót. A környezet egy rövid szövegegység, általában egy mondat, egy bekezdés vagy egy k méretű szóablak, középen a célszóval. A korpuszok formátumára nincs semmilyen megkötés; a módszerek nem igényelnek semmilyen címkézést, így egyszerű szöveg is lehet a korpuszok tartalma.

Ez a fejezet a vektoralapú JEÉ algoritmusok általános működését mutatja be egy adott célszóhoz tartozó tanító korpusz esetén.

2.1. Jegykiválasztás

A vektoralapú módszerek első lépésben jegyeket választatnak ki a korpuszból. (A jegykiválasztás történhet más adatból is, azonban nem ez a szokás.) A jegyek általában szavak, bigramok vagy szó-együttelőfordulások. Egyszavas jegyek lehetnek például a tanítókorpuszban bizonyos számnál gyakrabban előforduló szavak. A bigramok egy kis (2-4 méretű) szóablakban gyakran együtt előforduló rendezett szópárok. Az együtt-előfordulások jellemzően nagyobb szóablakban (például azonos mondatban vagy bekezdésben) gyakran előforduló rendezetlen szópárok. A jegyek kiválasztására a minimális gyakoriságon kívül statisztikai tesztek is alkalmazhatók.

A jegykiválasztás a JEÉ algoritmusok egyik legfontosabb lépése, mert ezek adják majd a környezeteket reprezentáló vektorok dimenzióit.

2.2. Környezet-reprezentáció

A jegykiválasztás eredménye a célszó használati esetei szempontjából releváns-nak vélt jegyek halmaza. A környezet-reprezentációs lépés során az algoritmusok minden egyes környezethez egy-egy vektort rendelnek. Léteznek első-, ill. másodrendű környezetvektorok.

Az elsőrendű környezetreprezentációs vektorok úgy állnak elő, hogy a vektor i-edik eleme az előző lépés során gyűjtött i-edik jegy előfordulási száma az adott környezetben.

A másodrendű környezetvektorok ([4] vezette be őket) bonyolultabb módszerrel számíthatók ki. Bigram és együtt-előfordulási jegyek esetén értelmezzük őket. Első lépésben egy mátrixot készítünk, amelynek sorai a jegyek első, oszlopai pedig a második szavai. A mátrix celláit a sorhoz, ill. az oszlophoz tartozó szavaknak a korpuszbeli együtt-előfordulási száma alapján töltjük ki. Az aktuális környezetet úgy reprezentáljuk, hogy a környezet azon szavait, amelyek szerepelnek a mártix sorcímkéi között, helyettesítjük a sorukkal. Az így kapott vektorok átlaga lesz a környezetreprezentáció.

[6] összehasonlította az első, ill. másodrendű környezet-reprezentációkat. Megmutatta, hogy nagy mennyiségű szöveg esetén az elsőrendű, míg kis méretű korpusz esetén a másodrendű ábrázolás esetén jobb a JEÉ algoritmusok teljesítménye.

2.3. Dimenziószám-csökkentés

A környezet-reprezentációs vektorok általában elég ritkák, azaz viszonylag sok 0 elem található bennük. Néhány esetben a dimenziószámuk is túl nagy a klaszterezési algoritmusok számára. Emiatt javasolta [4] a szinguláris érték-felbontást (SVD), amellyel a másodrendű jegyek mátrixának méretét lehet csökkenteni, mindezt simítással egybekötve. Elsőrendű jegyek esetén a jegyvektorokból mint sorokból előállított mátrixra is alkalmazható az SVD transzformáció. Az SVD javíthatja a JEÉ rendszerek hatékonyságát. (Egyéb módszerek is alkalmazhatók a dimenziószám-csökkentésére.)

2.4. Klaszterezés

Miután rendelkezésre állnak a környezet-reprezentációs vektorok, már tetszőleges klaszterező algoritmus használható a célszó használati eseteinek elkülönítésére. Általában a klaszterező algoritmus bemenő paraméterként igényli a kialakítandó klaszterek számát. [7] javasolt néhány függvényt a megfelelő klaszterszám előre történő meghatározására.

2.5. Kiértékelés

Több módszer is létezik felügyelet nélküli JEÉ rendszerek kiértékelésére; [3] foglalja össze ezeket. Egy lehetőség az algoritmus eredményét „kézzel” elemezni. Másik alternatíva lehet JEÉ rendszer teljesítményét egy alkalmazásban mérni.

Esetleg a célszó jelentéseivel címkézett korpusz is használható a kiértékelésre. Végül azt is megtehetjük, hogy a tökéletesnek vélt klaszterekkel hasonlítjuk össze az algoritmus eredményét.

3. A vizsgált jelentés-egyértelműsítő rendszer

A kísérletünk során alkalmazott JEÉ rendszer az ingyenesen elérhető SenseClusters programcsomagból ([1]) és egy saját fejlesztésű kiértékelő modulból állt. Ez a fejezet röviden bemutatja a rendszer moduljait és a számukra szükséges bemenő paramétereket.

3.1. A SenseClusters moduljai

A SenseClusters bemenetként a célszót tartalmazó bekezdéseket vár. A jegykiválasztó modulja segítségével lehetőség van egyszavas, bigram, ill. együttelőfordulási jegyek gyűjtésére. Egyaránt lehetséges minimális előfordulási gyakoriságot, valamint valamilyen statisztikai mértéket megadni a jegyek elfogadásához.

A jegykiválasztás után a környezetreprezentációs modul hajtódik végre. A modul egyszavas jegyek esetén elsőrendű, bigram és együtt-előfordulási jegyek estén pedig mind első-, mind másodrendű környezet-reprezentációra képes.

A dimenziószám-csökkentő modul SVD transzformáció segítségével próbálja a reprezentációs vektorokat simítani. (Ennek a modulnak a végrehajtása opcionális.)

A SenseClusters a CLUTO programot ([8]) alkalmazza klaszterezésre. A CLUTO egyaránt támogat agglomeratív, particionális és hibrid klaszterezési algoritmusokat. Ezek a módszerek bemenő paraméterként igénylik az előállítandó klaszterek számát. Ennek meghatározásában a SenseClusters PK1, PK2, PK3 ([7]), ill. GS ([9]) mértékei nyújtanak segítséget.

3.2. A kiértékelő modul

Annak érdekében, hogy a kísérletünk eredményei más hasonló munkákéval összehasonlíthatóak legyenek, [3]-hoz hasonlóan a célszó jelentéseivel címkézett környezeteken végeztük a kiértékelést. Ezeket a környezeteket felosztottuk tanító és kiértékelő részre. A kiértékelési folyamat egy felügyelt tanulási feladat: a címkézett tanulókörnyezeteken tanuljuk meg a klaszter-jelentés hozzárendeléseket, a hatékonyságot pedig a címkézett tesztkörnyezeteken mérjük.

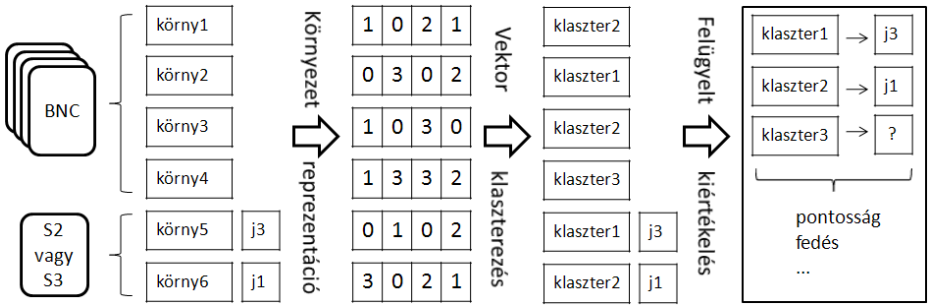
Habár a SenseClusters is nyújt kiértékelő szolgáltatásokat, ezeket nem lehetett a fent leírt módon alkalmazni, úgyhogy egy saját kiértékelő modul elkészítésére volt szükség.

Ugyan ez a kiértékelési módszer a rendszert felügyelet nélküli és felügyelt tanulás keverékévé teszi, fontos megjegyezni, hogy a klaszterek előállítása teljesen felügyelet nélkül zajlik, csupán a klaszter-jelentés párok kialakítása történik felügyelt módon.

3.3. Paraméterek

A kísérlet során az egész SenseClustert egy felügyelet nélküli JEÉ rendszernek tekintettük. A 3.1-es alfejezetben bemutatott különböző funkcióit (például hogy első- vagy másodrendű reprezentációt alkalmazunk) a rendszer szabad paramétereinek tekintettük. Ezeket próbáltuk hangolni.

Természetesen a paraméter-optimalizálást más adathalmazon kell végezni, mint amin a rendszer teljesítményét mérjük. A kiértékelést az elkülönített adatokon az optimalizált paraméterekkel végeztük.



1. ábra. A kísérlet folyamata

4. A kísérlet

A kísérletet egy szabványos adathalmazon végeztük el azért, hogy más rendszerekével összehasonlítható legyen az eredmény. A Senseval-3 jelentés-egyértelműsítő verseny 20 angol főnevét, ill. az ezekhez tartozó korpuszokat választottuk ki erre a célra. A kísérlet során a Senseval-3 tanító adathalmaz - tesztadathalmaz elkülönítését alkalmaztuk.

A paraméterek hangolására a Senseval-2 verseny 20 angol főnevét választottuk ki. A hozzájuk tartozó korpuszokat használtuk az optimalizálás során végzett kiértékeléshez.

4.1. A kísérlet menete

A kísérlet folyamata az 1. ábrán látható. Első lépésként az egyes célszavakhoz állítottuk elő a korpuszokat. Minden egyes célszóhoz a British National Corpusból (BNC) gyűjtöttünk olyan bekezdéseket, amelyek az adott többjelentésű szót tartalmazzák. A Senseval adathalmaz környezeteit, amelyek a célszavak jelentéseivel címkézettek, hozzáadtuk a megfelelő szóhoz tartozó, BNC-ből gyűjtött korpuszhoz. (A Senseval adathalmaz Senseval-2-t jelent paraméter optimalizálás, és Senseval-3-at kiértékelés esetén.) Ezzel a módszerrel minden egyes célszóhoz 2000-3000 környezetet sikerült gyűjteni.

Ezután a SenseClusters moduljait hajtottuk végre a kigyűjtött környezeteken. Végül a saját kiértékelő modul segítségével mértük az egyes módszerek hatékonyságát.

Az egész kísérletet megismételtük csupán a Senseval adatokon. Az így kapott eredményeket a bővített adathalmazon kapottakkal összevetve tudtunk következtetést levonni a környezetek számának szerepéről.

4.2. Az optimális paraméterek

A 4. fejezet bevezetésében már szerepelt, hogy a paraméterek hangolása a Senseval-2 adatokon történt. Abban az esetben, amikor a SenseClusters-t a bővített adathalmazon futtattuk, a legjobb eredményt elsőrendű együtt-előfordulási jegyek segítségével értük el. Ez összhangban van [6] következtetéseivel, melyeket a 2.2. alfejezetben említettünk. Egy particionális klaszterezési módszer, az ún. ,,Repeated Bisection" algoritmus bizonyult a legjobbnak. Az SVD transzformáció alkalmazása nem javított az eredményeken.

Az optimális paraméterhalmaz nagyon hasonló volt abban az esetben, amikor csak a Senseval-2 adatokon végeztük a kísérletet, mindössze a jegyek típusa bizonyult más esetben optimálisnak: az egyszavas jegyek nyújtották a legjobb teljesítményt.

5. Az eredmények

Az 1. táblázaton szerepelnek a kísérlet eredményei. A leggyakoribb jelentés heurisztika jelentette a baseline módszert. Az optimalizált paraméterekkel elindított algoritmus a BNC környezetekkel kiegészített adatokon futtatva kis mértékkel jobbnak bizonyult az alapadatokon futtatott esetnél. Mindkét verzió lényegesen felülmúlja a baseline algoritmust.

Az elért eredmények versenyképesek a többi Senseval-3 adatokon kiértékelt felügyelet nélküli JEÉ rendszerek eredményeivel. A 2. táblázaton látható, hogy egyedül az optimalizált HyperLex algoritmus ([3]) teljesített jobban. Az SCBNC, ill. az SCS3 nevek jelölik az általunk alkalmazott rendszereket. (A többi rendszer hatékonyságát [3] mérte meg.)

Habár ezek a rendszerek ugyanazon az adathalmazon lettek kiértékelve, mégis nehéz őket összehasonlítani a különböző alkalmazott tanítási módszerek miatt. Némelyik algoritmusnak szüksége van a leggyakoribb jelentés ismeretére (ezeket MFS-Sc jelöli, ha a leggyakoribb jelentést a SemCor, MFS-S3, ha a Senseval-3 adatok alapján számítja a módszer), némelyek a Senseval-3 tanítópéldák 10%-át használják a klaszter-jelentés hozzárendelés tanulására (10%-S3TR), mások erre a teljes tanítóhalmazt igénybe veszik (S3TR) [3].

6. Összefoglalás

Ez a cikk bemutatott egy lehetséges módszert felügyelet nélküli JEÉ algoritmusok teljesítményének növelésére. Ehhez mindössze olyan környezetekre volt

1. táblázat. A kiértékelés eredménye a Senseval-3 főneveken. Az első oszlopban szerepelnek a vizsgált célszavak. Emellett állnak a leggyakoribb jelentések arányai. Az utolsó két oszlop mutatja a kísérlet eredményét az alapadatok, valamint a BNC környezetekkel kiterjesztett korpuszok esetén. A táblázatban feltüntetett számok a pontossági értékek. (A fedés megegyezik a pontossággal.)

Szó	MFS	SCS3	SCBNC
argument	51.4	48.6	51.4
arm	82.0	85.0	85.7
atmosphere	66.7	72.8	71.6
audience	67.0	70.0	76.0
bank	67.4	72.7	72.0
degree	60.9	67.2	68.8
difference	40.4	48.2	43.0
difficulty	17.4	47.8	26.1
disc	38.0	71.0	66.0
image	36.5	60.8	60.8
interest	41.9	59.1	66.7
judgment	28.1	40.6	40.6
organization	73.2	73.2	69.6
paper	25.6	44.4	52.1
party	62.1	64.7	65.5
performance	32.2	42.5	46.0
plan	82.1	78.6	77.4
shelter	44.9	42.9	48.0
sort	65.6	65.6	65.6
source	65.6	50.0	50.0
Átlag:	54.5	61.9	62.9
(Senseval-2 adatokon)	51.9	59.0	59.8

2. táblázat. A Senseval-3 angol főnevein kiértékelt felügyelet nélküli JEÉ rendszerek összehasonlítása.

Rendszer	Típus	Pontosság	Coverage
HyperLex	S3TR	64.6	1.0
SCBNC	S3TR	62.9	1.0
SCS3	S3TR	62.0	1.0
Cymfony	10%-S3TR	57.9	1.0
Prob0	MFS-S3	55.0	0.98
MFS	-	54.5	1.0
Ciaosenso	MFS-Sc	53.95	0.90
clr04	MFS-Sc	48.86	1.0
duluth-senserelate	-	47.48	1.0

szükség, amelyek tartalmazták az éppen vizsgált célszót. Ezekkel a környezetekkel kiegészítve a tanuló korpuszt némileg javult az algoritmusok hatékonysága. A módszer hátránya, hogy a tanulási folyamat idejét megnöveli.

A paraméterek optimalizálása után kapott vektoralapú JEÉ algoritmus versenyképes a jelenlegi legjobb hasonló rendszerekkel, azonban az összehasonlítás elég nehéz feladat a különböző módon végzett klaszter-jelentés hozzárendelések miatt.

Hivatkozások

1. Purandare, A., Pedersen, T.: SenseClusters - finding clusters that represent word senses. In: Proc. of the Nineteenth National Conference on Artificial Intelligence (AAAI-04). San Jose, USA (2004) 1030–3031
2. Mihalcea, R., Chklovski, T., Kilgarriff, A.: The Senseval-3 English lexical sample task. In: Senseval-3 proceedings (2004) 25–28
3. Agirre, E., Martínez, D., de Lacalle, O.L., Soroa, A.: Evaluating and optimizing the parameters of an unsupervised graph-based WSD algorithm. In: Proc. of the TextGraphs Workshop: Graph-based algorithms for Natural Language Processing. New York, USA (2006) 89–96
4. Schütze, H.: Automatic word sense discrimination. Computational Linguistics, **24**(1) (1998) 97–123
5. Véronis, J.: HyperLex: lexical cartography for information retrieval. Computer Speech & Language, **18**(3) (2004) 223–252
6. Purandare, A., Pedersen, T.: Word sense discrimination by clustering contexts in vector and similarity spaces. In: Proc. of the Eighth Conference on Computational Natural Language Learning (CoNLL). Boston (2004) 41–48
7. Pedersen, T., Kulkarni, A.: Selecting the „right" number of senses based on clustering criterion functions. In: Proc. of the Posters and Demo Program of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics. Trento (2006) 111–114
8. Zhao, Y., Karypis, G.: Evaluation of hierarchical clustering algorithms for document datasets. In: Proc. of the 11th Conference of Information and Knowledge Management (CIKM). McLean, USA (2002) 515–524
9. Tibshirani, R., Walther, G., Hastie, T.: Estimating the number of clusters in a dataset via the Gap statistic. Journal of the Royal Statistics Society (Series B) **63**(2) (2001) 411–423