Adaptív dokumentumelemzés információkinyeréshez

DEZSÉNYI CSABA, MÉSZÁROS TAMÁS, DOBROWIECKI TADEUSZ

BME, Villamosmérnöki és Informatikai Kar, Méréstechnika és Információs Rendszerek Tanszék
dezsenyi@mit.bme.hu

Reviewed

Kulcsszavak: összetett dokumentumelemzés, elemzési terv készítése, információkinyerés

Manapság egyre nagyobb szerepet kapnak a különböző információ- és tudásmenedzsment alkalmazások, mind az ipari, mind a tudományos területeken. Komoly kihívást jelent viszont a folyamatosan növekvő méretű információtengerben megtalálni a számunkra érdekes információszeletet, kinyerni a számunkra értékes tudást. Ennek megfelelően az alkalmazásokban különösen fontos elemek lettek a különböző információkinyerési módszerek és technikák, melyek segítségével természetes nyelvű dokumentumokból tudunk releváns információt kiemelni.

1. Bevezetés

Ahhoz, hogy megfelelő teljesítményt érjünk el, nem elég egy-egy izolált algoritmust felhasználni, több különböző dokumentumelemzési módszert kell összehangoltan alkalmazni. Egy gazdasági tényeket kinyerő alkalmazásban például a következő feldolgozási lépésekre lehet szükség:

(1) szöveges tartalom kivágása a HTML oldalból,
(2) szavak és mondatok szegmentálása,
(3) személy- és intézménynevek felismerése,
(4) szófajtani elemzés és végül
(5) mondattani elemzés.

Az ilyen és ehhez hasonló, összetett információkinyerést alkalmazó rendszerek fejlesztése eddig elszigetelten történt, az elemzések megvalósítását pedig az egyedi alkalmazásoknak megfelelően alakították ki, különböző architektúrákra, adatmodellekre és implementációkra alapozva.

A bemutatásra kerülő dokumentumelemző keretrendszer célja az, hogy egységes elméleti és szoftver keretet nyújtson olyan alkalmazások fejlesztéséhez, melyben természetes nyelvű szövegek összetett elemzésére és feldolgozására van szükség. A keretrendszerben tetszőleges dokumentumelemzési feladat megvalósítható, így az ilyen alkalmazások alapvető platformjaként képes szolgálni.

A keretrendszer alapötlete az, hogy egy összetett dokumentumelemzési feladatot automatikusan dekomponálunk kisebb és egyszerűbb műveletekre, majd így elvégezve az elemzést, az eredményt konzisztensen egyesítjük. A tervezés fázisai között az egyik legnagyobb kihívás annak az adatmodellnek a kialakítása, mely segítségével a független elemzőmodulok eredményei konzisztensek és összefüggőek lesznek a végrehajtás után. Az elméleti alapokról és a megalkotott adatmodellről bővebben [1]-ben olvashatunk.

Jelen cikkben a keretrendszer rövid bemutatása után egy új adaptív dokumentumelemzési megoldást ismertetünk.

2. A dokumentumelemző keretrendszer bemutatása

Az információkinyerés teljes folyamata három fázisra osztható: dokumentumbeszerzés, dokumentumelemzés, és végül az információkinyerés. Első lépésként be kell szerezni azokat a dokumentumokat a forráskörnyezetből, melyek az alkalmazás által igényelt információt tartalmazhatják. Ez legegyszerűbb esetben egy lokális fájlrendszerből való iterált beolvasást jelent. Bonyolultabb megoldást kíván, ha a megfelelő dokumentumokat az interneten kell megkeresni és onnan letölteni. Ilyen intelligens dokumentum kereső és beszerző rendszer fejlesztése szintén a kutatás része, melyről bővebben [2]-ben olvashatunk. Az első fázis eredménye a beszerzett dokumentum, valamilyen kiindulási struktúrába öntve.

Ezután többféle módon elemezni kell a dokumentumot, aminek eredményeképpen az eredeti forrás különböző strukturált reprezentációi állnak elő, melyeket *nézeteknek* nevezünk. Ezeknek a nézeteknek kell tartalmaznia az alkalmazás által igényelt információelemeket. Az elemzés során létrejött nézetek a bemenetei a harmadik fázisnak, ahol az alkalmazás számára érdekes információt le lehet kérdezni belőlük.

A rendszerbe illeszthető dokumentumelemző modulok az alapvető építőkövek, segítségükkel lehet összetett elemzési sémákat kialakítani és így bonyolult dokumentumelemzési feladatokat elvégezni. Számos eszköz vizsgálatára alapozva kialakítottunk egy absztrakt dokumentumelemző meta-modellt. A keretrendszer ennek segítségével egységesen tudja kezelni az egyes modulokat, implementációtól függetlenül. Az interfészek és adatmodellek kialakításánál különös figyelmet fordítottunk arra, hogy tetszőleges dokumentumelemzési művelet megvalósítható legyen.

Egy dokumentumelemző modul feladata az, hogy a bemenetként kapott dokumentumban felismerjen bizonyos elemeket, majd ezeket egy kimeneti dokumentumba transzformálja. Mind a bemeneti, mind a kimene-

ti dokumentumok speciális formátumúak az elemzők számára, ezek a már említett nézetek. Egy létrejövő nézetet tekinthetünk úgy, mint az eredeti forrás egy bizonyos típusú információs vetületét, mely valamilyen ismert struktúrában tartalmazza az elemző által azonosított információ-elemeket. Minden nézetnek van egy típusa, mely megmondja, hogy milyen fajta információt tartalmaz, illetve amely definiálja annak struktúráját (sémadefiníciók segítségével). A nézetek megvalósítása tipikusan XML-el lehetséges, azaz egy nézet, egy XML dokumentum.

Mivel egy elemzőmodul nézeteket állít elő, a bemenetei pedig szintén nézetek, az egyes modulok fel tudják használni mások eredményeit. A keretrendszer bemenete egy kezdeti nézet, amely az elemezni kívánt forrásdokumentumot tartalmazza valamilyen (alkalmazás függő) kezdeti struktúrába öntve. A teljes elemzési folyamat eredménye az egyes modulok által előállított nézetek szemantikailag összefüggő halmaza, melyet *nézethálózatnak* hívunk [1].

3. Dokumentumelemzési séma tervezése

A beszerző rendszer szolgáltatja az elemző keretrendszer számára a forrásdokumentumokat (mint kiindulási nézetek), míg az információkinyerés fázisa előírja, hogy milyen nézetek szükségesek ahhoz, hogy az alkalmazás számára igényelt információt ki tudjuk nyerni. A keretrendszer feladata tehát, hogy előállítsa a megfelelő nézeteket a különböző elemzőmodulok alkalmazásával. Ehhez ki kell választani a szükséges modulokat, meg kell tervezni a futási szekvenciát, végre kell hajtani az elemzési folyamatot és létre kell hozni az elemzés eredményét, azaz a teljes nézethálózatot.

Problémát okoz azonban az, hogy a megfelelő modulok kiválasztása, illetve a végrehajtási sorrend nem triviális, mert pl. modulok igényelhetik mások kimenetét, esetleg több fajta bemeneten képesek dolgozni, vagy egy tipikus eset lehet, hogy egy fajta nézetet többféleképpen tudunk előállítani. Az általunk kidolgozott módszer alapja, hogy klasszikus MI tervkészítő algoritmusokat alkalmazunk (például STRIPS alapú részben rendezett tervkészítő algoritmust [3]) az elemzési sémák részben vagy teljesen automatizált készítéséhez.

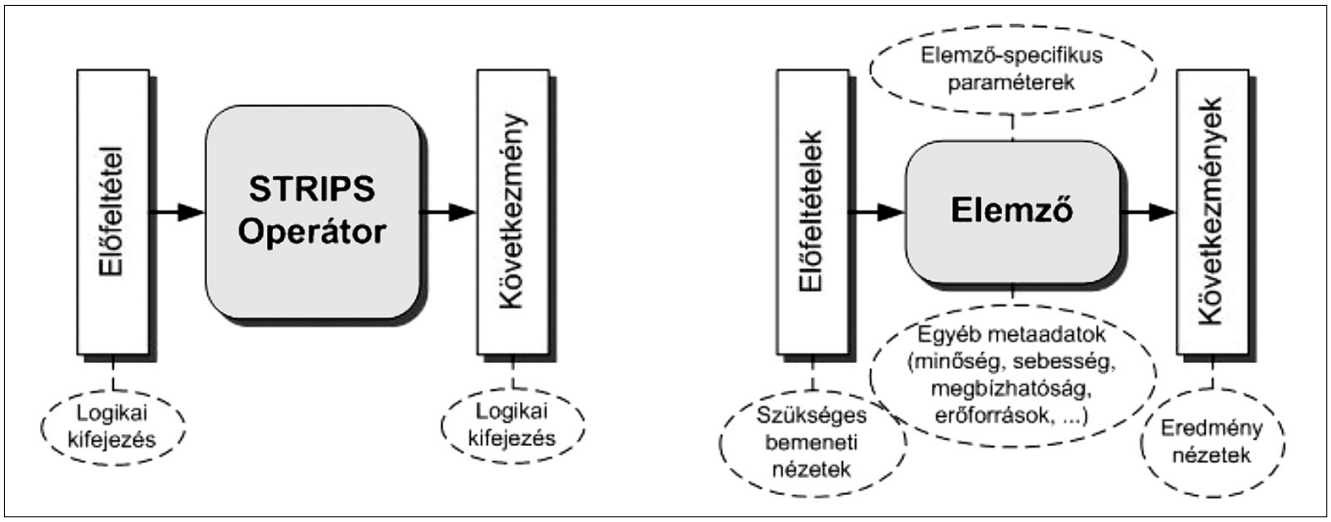
Mivel a klasszikus MI tervkészítési problémák és a keretrendszerben lévő elemzési séma problémaköre nagymértékben analóg, kézenfekvő megoldásnak tűnik a már kidolgozott módszerek és algoritmusok adoptálása. A keretrendszer egyes elemei, elnevezései könnyen megfeleltethetők az MI tervkészítés terminológiájában használatos fogalmaknak.

A tervben lévő operátorok (melyek tk. cselekvések, állapotváltozást okoznak) itt az elemzőmodulok lesznek, melyek meglévő nézetekből új nézeteket állítanak elő *(1. ábra)*. Egy modul előfeltétele a futáshoz igényelt bemeneti nézeteire vonatkozó logikai feltételekből áll, míg a hatás az eredményképpen létrejövő nézetek logikai leírása.

A terv kiindulási állapota az, amikor egy új dokumentum kerül a rendszerbe, mint kiindulási nézet, a célállapot pedig akkor valósul meg, mikor az összes olyan nézet előállt, ami szükséges az információkinyeréshez. A tervkészítő algoritmus által létrehozott részben rendezett terv itt a végleges tervként szolgál, mivel az egyes modulok párhuzamosan is tudnak futni, nincs szükség linearizálásra. A fogalmak megfeleltetése után már könnyű alkalmazni a megfelelő MI tervkészítő algoritmust, a működési mechanizmus ugyan az lesz, mint általános esetben *(2. ábra)*.

A terv kiindulási állapota két absztrakt elemzőt tartalmaz: „start” és „cél”. A start lépés a kiindulási nézeteket állítja elő és nincs előfeltétele, a cél pedig az alkalmazás által igényelt nézeteket definiálja előfeltételek formájában, nyílván következmény része nincsen. Az algoritmus feladata az, hogy a start és a cél közötti utat megtalálja megfelelő elemzőmodulok illesztésével. Regresszív tervkészítő esetében az illesztés a céltól visszafelé történik, azaz első lépésként a célhoz keres

1. ábra STRIPS operátor és Elemző



olyan alkalmas elemzőmodulokat, melyeknek a következmény része kielégíti a cél előfeltételeit. Amennyiben a tervnek létezik megoldása, akkor ezt folytatva előbb-utóbb eljutunk a startig.

Hagyományos tervkészítő algoritmusok esetében fontos kérdés az, hogy az egyes operátorok ne rontsák el mások előfeltételeit (például egyik előállít egy szükséges feltételt, viszont később egy másik mellékhatásként törli azt).

Ezt az úgynevezett védett szakaszok elméletével építik bele az algoritmusokba. Ennek lényege, hogy az operátorok sorrendezését kényszerítik úgy, hogy ne fordulhasson elő ilyen szituáció. Mivel a keretrendszerben lévő elemzők előfeltételei és következményei kizárólag ponált elemeket tartalmazhatnak (inkrementális jelleg: mindig nézetet állít elő egy elemző, sose töröl), ezért a rendszer kedvező tulajdonsága, hogy ilyen jellegű probléma nem fordulhat elő.

4. Nyitott kérdések és az implementáció jellemzői

Általános esetekben az adaptáció tökéletesen működik, de összetettebb konfigurációk esetében adódhatnak olyan konfliktus szituációk, melyek automatikus feloldása nehézségeket okozhat. Például egy tipikus eset, amikor egy fajta nézetet két féle elemző is elő tud állítani. A tervkészítő algoritmus szempontjából ekvivalens a két elemző használata, ám a valós kimenetel nagyon különbözhet, egyes esetekben az egyik, máskor a másik lehet a jobb hatásfokú.

Vajon ilyenkor az volna célszerű, ha a tervkészítéskor eldöntenénk egy prioritás jellegű választással? Vagy nyitva hagyva a választást, a végrehajtás egy bizonyos fázisában lenne érdemes dönteni, hogy melyik fusson? Esetleg mindkettő futtatása után próbálja meg a rendszer a jobb eredményt kiválasztani, vagy egy optimális uniót képezni belőlük? A kutatás jelenlegi fázisában kézi konfliktusfeloldást használunk, így ezek a fontos és érdekes kérdések jelenleg még nyitottak, a teljesen automatikus és optimális tervkészítés megoldása még fejlesztés alatt áll.

A dokumentumelemző keretrendszer prototípus implementációja elkészült és néhány egyszerű elemzőmodul is készült tesztelés céljából. A rendszerben használatos adatmodellek és metaadat kezelés teljes egészében XML alapú.

Mivel az alapötlet, az elméleti keret és maga az implementáció sikeressége is egyaránt a gyakorlati használhatóságon múlik, ezért a rendszer tesztelése és értékelése folyamatosan történik. Emellett a kezdeti tapasztalatokra alapozva mindenképpen ígéretesnek mutatkozik a kutatás és számos valós célalkalmazás fejlesztését is tervbe vettük.

Irodalom

[1] Cs. Dezsényi, T. Mészáros, T. P. Dobrowiecki: "Parser Framework for Information Extraction", Proc. of EUROFUSE Workshop on Data and Knowledge Engineering, September 22-25, Warszawa, Poland, 2004.
[2] Cs. Dezsényi, P. Varga, T. Mészáros, Gy. Strausz, T. P. Dobrowiecki: „Ontológia Alapú Tudástárház Rendszerek", Proceedings of Networkshop 2003 Conference, Pécs, April 14-17, 2003.
[3] S. Russell, P. Norvig: Artificial Intelligence. A Modern Approach. Prentice Hall Inc., 1997.

2. ábra Elemzési terv példa: egyszerű ténykinyerés
(E1: tokenizáló, E2: nyelvfelismerő, E3: indexelő, E4: morfológiai elemző, E5: névfelismerő, E6: mondatelemző, N0: kiindulási nézet, N1: szavak és mondatok, N2: dokumentum nyelv, N3: index, N4: szavak morfológiája, N5: felismert nevek, N6: elemzett mondatok)