



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Méréstechnika és Információs Rendszerek Tanszék

Watson és a DeepQA

Tartalomjegyzék

1. Bevezetés	2
1.1. IBM és az QA	2
1.2. Jeopardy!	2
1.3. Kiindulási pont és teljesítmény mérés	2
2. DeepQA	3
2.1. Content Acquisition	4
2.2. Question Analysis	5
2.3. Hypothesis Generation and Soft Filtering	6
2.4. Hypothesis and Evidence Scoring	6
2.5. Final merging and ranking	7
3. Összefoglalás	8
Irodalom	9

1. Bevezetés

1.1. IBM és az QA

Az IBM Research célja a számítástechnika fejlesztése, olyan módú alkalmazása ami pozitív hatással van a tudományok, üzleti élet és a társadalom számára. 2007-ben kezdődött egy olyan kutatási kihívás keresése, ami a 2002-ben készített Deep Blue sakkozó számítógép által elért népszerűséggel vetekedik és egyúttal jelentős üzleti vonatkozással is rendelkezik. Egy ilyen területnek számítottak a kérdés válaszoló rendszerek(QA), amik képesek egy tág tárgyterületre vonatkozó kérdések megválaszolására.

Az Automatic Open-Domain Question Answering a számítástechnika és mesterséges intelligencia egyik legnehezebb problémája, ami feltételezi a visszatérített információk egyesítését, természetes nyelv feldolgozást, gépi tanulást és ember-számítógép interfészeket. A feladatot megoldó rendszertől elvárjuk hogy helyes választ adjon, bizonyítékot szolgáltatson a válasz helyességére és megadja hogy milyen mértékben hiszi helyesnek az adott választ. A gyorsaság is egy fontos tényező, hiszen alkalmazási területtől függően akár percek, másodpercek alatt szükség lehet a válaszra. Ahogy a Deep Blue esetében a sakk volt az a terület ahol annak képességeit bemutatták, az új rendszernek a *Jeopardy!* nevű kvízzjáték volt megmérettetésének helyszíne.

1.2. Jeopardy!

THE DINOSAURS	NOTABLE WOMEN	OXFORD ENGLISH DICTIONARY	NAME THAT INSTRUMENT	BELGIUM	COMPOSERS BY COUNTRY
\$200	\$200	\$200	\$200	\$200	\$200
\$400	\$400	\$400	\$400	\$400	\$400
\$600	\$600	\$600	\$600	\$600	\$600
\$800	\$800	\$800	\$800	\$800	\$800
\$1000	\$1000	\$1000	\$1000	\$1000	\$1000

1. ábra. Jeopardy tábla

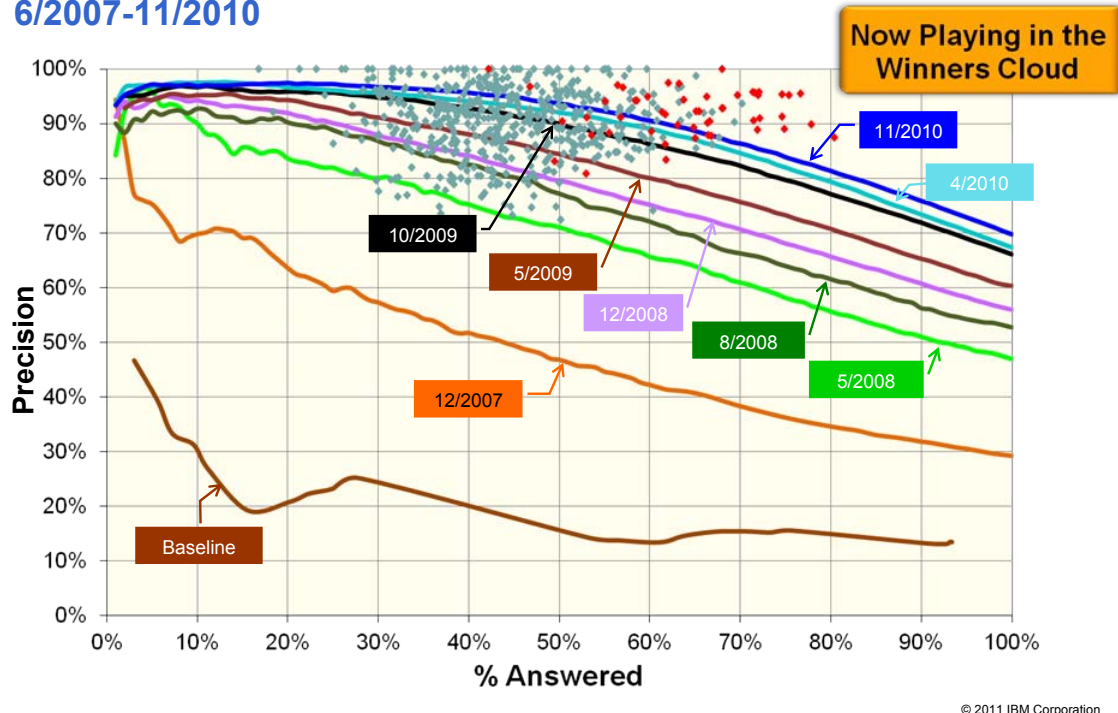
A Jeopardy játékban 3 játékos versenyez egymással, a győztes az aki a legtöbb kérdésre felelt pontosan és gyorsan. Egy 30 kérdéses Jeopardy játék esetén a kérdések 6 kategóriába vannak rendezve, növekvő nehézséggel és növekvő jutalommal a kategórián belül. Lehetséges kategóriák a történelem, tudomány, politika vagy valami kevésbé információdús, amik nem hogy segítenék a válaszadást hanem épp ellenkezőleg, félrevezetnek. Egy kérdés feltevése után a versenyzők, ha tudják a választ egy csengőt nyomnak meg. Ha aki először nyomta meg a csengőt helyes választ ad, megnyeri a kérdés nehézségének megfelelő pénzüsszeget. Ezzel ellentétben ha téved, eddigi nyereményéből levonnak és a többi versenyző is megpróbálhat válaszolni. Látható, hogy ahhoz hogy egy számítógép felvegye a versenyt ember versenyzőkkel, egy óriási adathalmazt kell elemezzen, kapcsolatokat felfedezzen, ezek alapján következtessen a helyes válaszra és a válasz pontosságára, mindezt nagyon rövid idő alatt.

1.3. Kiindulási pont és teljesítmény mérés

Az IBM Research kiindulási pontként két, már létező QA rendszer implementációt módosított (minimális erőfeszítést téve), hogy képesek legyenek Jeopardy kérdések megválaszolására. Az első rendszer a PIQUANT 2015. január 8.



Deep QA: Incremental Progress in Precision and Confidence 6/2007-11/2010



2. ábra. Emberek és Watson teljesítménye a Jeopardy! játékban

volt, ami a Watson projekt kezdete előtt már 6 éve fejlesztés alatt állt az IBM Research egy 4 tagú csapatja által és benne volt az akkori legjobb 5 TREC QA rendszerek csoportjában. A másik rendszer aminek a teljesítményét alapul vették a CMU által fejlesztett OpenEphyra, ami szintén TREC kérdésválaszolásra volt fejlesztve. A TREC a Jeopardy-val ellentétben kisebb corpusal dolgozott, egyszerűbbek voltak a kérdések és megengedte az webes keresést. Ezen a versenyen a PIQUANT 33% míg az OpenEphyra 44%-os pontosság körül mozgott.

Futtatva őket és mérve a válaszolt Jeopardy kérdések számát és azok pontosságát, megkapták az akkori technológia teljesítményét ami a 3. ábrán látható barnával rajzolva. Ez messze alulmarad az emberek teljesítményével szemben, a piros pöttyök jelentvént a Jeopardy legjobb játékosainak teljesítményét.

Három év fejlesztés után az IBM Watsonja és annak DeepQA-nak nevezett architektúrája már képes volt felvenni a versenyt az emberekkel a Jeopardy enyhén módosított változatában. A módosítás abban állt, hogy kizártak két típusú kérdést: audio vizuális kérdéseket és olyan kérdéseket amikhez magyarázatot igényeltek. Így a Watsonnak és a két másik versenyzőnek feltett természetes nyelvű kérdések az összes információt tartalmazták amik alapján meg lehetett őket válaszolni. A Jeopardy! szervezői is állítottak egy feltételt Watsonnal szemben, mégpedig azt hogy a verseny ideje alatt nem csatlakozhat az internetre, válaszait az offline rendelkezésre álló tudástárából kell kinyerje.

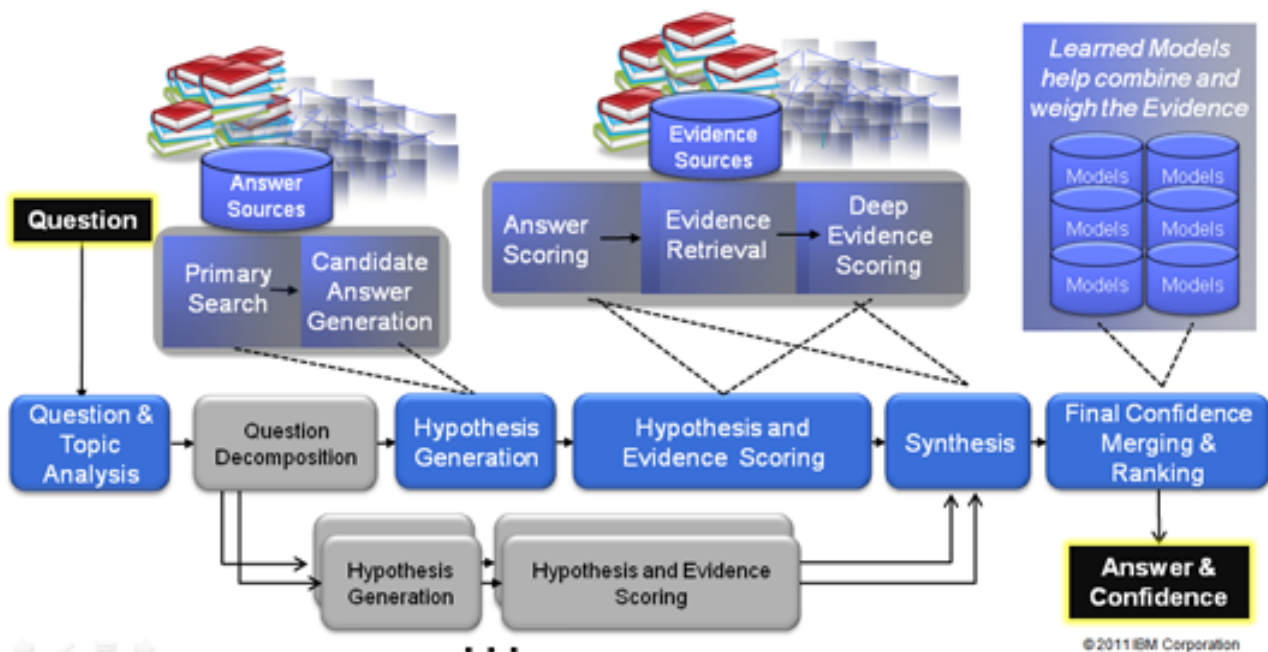
2. DeepQA

A projekt kezdetén próbálkoztak a szakirodalomban megtalálható algoritmusok implementálásával, azok adaptálásával. Mindez nehéznek bizonyult, lévén hogy minden ismert megoldás egy igen szűk problémakörre vonatkozott, nehéz volt azt a Jeopardy igényeihez alakítani, vagy egyáltalán működésre bírni. Ennek folytán az 2015. január 8.

IBM és CMU elindított egy Open Advancement Question Answering kezdeményezést, aminek a célja a QA kutatások egyesítése, azok megismételhetőségének növelése.

A OAQA összehozva a QA szakértőket, egy olyan egységes, kiterjeszhető platform megalkotását javasolta ami lehetővé tenné a komponensek által szolgáltatott eredmények egységes kiértékelését adott "kihívásokkal"(Challenge Problems) szemben. Ezek a kihívások a Jeopardy esetében többek között a nagy pontosság, helyes bizonyosság(confidence) meghatározás, komplex nyelv felismerés, tág ismeretkör és gyorsaság.

Egy ilyen platform a DeepQA, egy nagymértékben párhuzamosított, valószínűségi bizonyítékalapú keretrendszer, ami több mint 100 technikát használ természetes beszéd elemzésre, forrás keresésre, hipotézis generálásra, bizonyíték keresésre és pontozásra és azok egyesítésére. Magas szintű architektúrája a 3. ábrán látható. A továbbiakban a DeepQA által végzett lépések leírása következik.



3. ábra. DeepQA architektúra

2.1. Content Acquisition

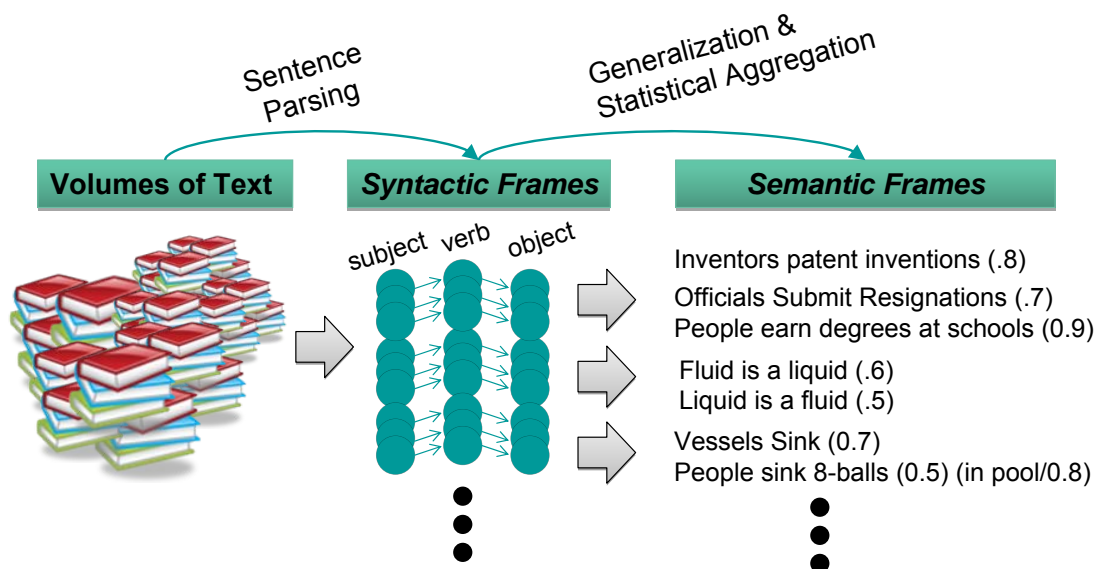
Az első lépés a DeepQA alkalmazásakor, függetlenül hogy Jeopardy-ról, vagy orvosi területről van szó, a tartalom beszerzése, amiből a hipotézisek és bizonyítékok fognak generálódni. Ez egy félig automatizált lépés: manuális elemzése a példa kérdéseknek, hogy megtudjuk milyen típusú kérdésekre kell tudjon a rendszer felelni. Ezt követi a tárgyterület elemzés ami szintén manuális, de támogatva van automatikus eljárásokkal mint a LAT (lexical answer type) vagy egyéb statisztikai elemzés. A Watson esetében a tudástár alapját képezik enciklopédiák, szótárak, cikkek, irodalmi művek stb. Az alap corpus kialakítása után, a DeepQA egy automatikus kiterjesztést végez:

- megkeresi a forrás műveket és letölti a hozzájuk kapcsolódó dokumentumokat
- az így kapott dokumentumokból önálló szövegrészeket von ki
- pontozza az így kapott részeket aszerint hogy mennyire kapcsolódnak a forrás művekhez
- hozzáadja a bővített corpushoz a legjobb találatokat

Ezen kívül a DeepQA használ egyéb strukturált és félig-strukturált adatforrásokat, mint például adatbázisok, taxonómiák, ontológiák(dbPedia, WordNet, Yago).



Automatic Learning for "Reading"



© 2011 IBM Corporation

4. ábra. Példa szövegből való információ kinyerésre

2.2. Question Analysis

A kérdés értelmezése már a játék közben történik, célja hogy a rendszer megértse hogy mit kérdeznek és eldöntse hogy a kérdést hogy is kellene tovább kezelni. Ebben a lépésben a DeepQA több szakértőt használ(mixture of experts), amik felszíni és mély elemzést, logikai alak, szemantikus szerep címkék, koreferencia és egyéb technológiák alapján próbálják a kérdést megérteni. Kiemelendők a következő lépések:

- *Kérdés osztályozás* Azt próbálja meghatározni hogy a kérdés milyen típusú, milyen tárgyterületre vonatkozik. Lehet például rejtvény kérdés, matek kérdés, definíciós kérdés stb.
- *Fókusz és LAT meghatározás* A fókusz a kérdésnek azt a szavát jelenti, amit ha kicserélünk a válasszal, egy igaz állítást kapunk. Ilyen például a "this form" a "When hit by electrons, a phosphor gives off electromagnetic energy in this form" kérdésben. A LAT mint már említettük, meghatározza hogy milyen típusú választ kéne adjunk, ami többek között segít a helyes válasz jelöltek kiválasztásában.
- *Dekompozíció* Vagyis a kérdések felbontása és annak eldöntése hogy egyáltalán van-e szükség felbontásra alkérdésekre.

2.3. Hypothesis Generation and Soft Filtering

A hipotézis generálás felhasználva a kérdés elemzés eredményét, válaszjelölteket generál, keresést végezve a rendszer tudástárában. A válaszjelölt behelyettesítődik a kérdésbe, amiről a rendszer el kell döntse hogy igaz-e. A jelöltek keresésekor a fő cél hogy minnél többet kapjon a rendszer, a következő lépésre bízván annak eldöntését hogy a kapott válasz tényleg helyes-e. A rendszer egész tudástárát felhasználva, néhány száz válaszjelölt generálódik, amik egy gyors szűrésen esnek át, lecsökkentve számukat százra. A szűrés például megvizsgálja a jelölt LAT-ját, hogy az megfelel-e a kérdésével. A szűrést és a szűrésnél használt pontozást gépi tanulást alkalmazva tanulta a rendszer, tehát alkalmazás függő. Azok a jelöltek amik nem mentek át a szűrésen továbbítódnak a végső, a válasz egyesítés és rangsorolás lépésbe.

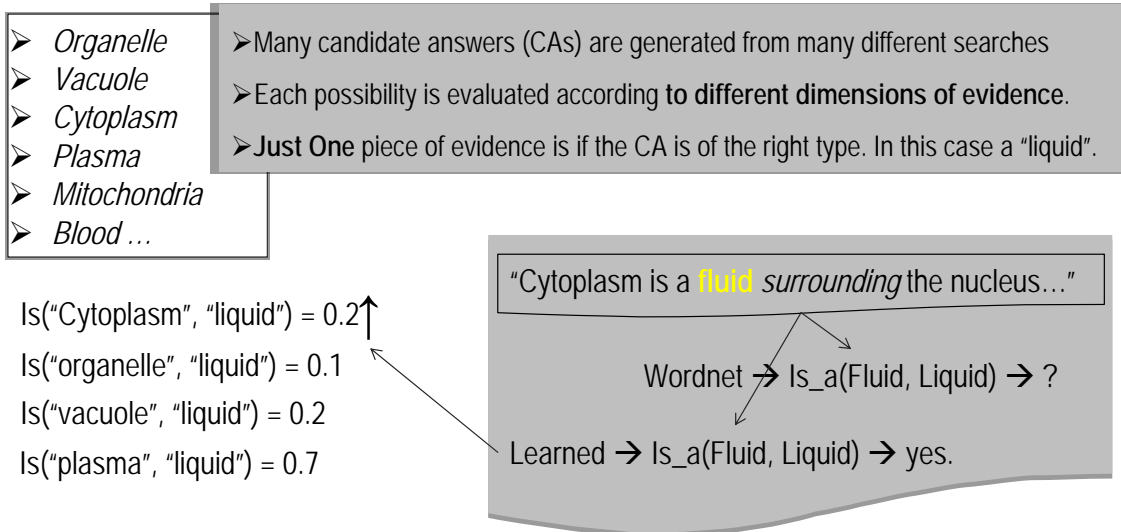
2.4. Hypothesis and Evidence Scoring

A szűrésen átesett jelöltekhez a rendszer megpróbál bizonyítékot keresni a tudástárában, hogy meg tudja indokolni őket. A kapott bizonyítékok pontozáson esnek át, hogy eldöntsük, mennyire is támasztják alá a válaszjelölteket. A jelöltek, bizonyítékaik és a bizonyítékok pontjai egy profilba csoportosítva lesznek továbbküldve a rendszer utolsó részéhez ahol az egyesítés és rangsorolás meg végbe.



Evaluating Possibilities and Their Evidence

In cell division, mitosis splits the nucleus & cytokinesis splits this liquid cushioning the nucleus.



© 2011 IBM Corporation

5. ábra. Példa bizonyítékra:

A 5. ábrán látható egy példa a válaszjelöltet alátámasztó bizonyítékra:

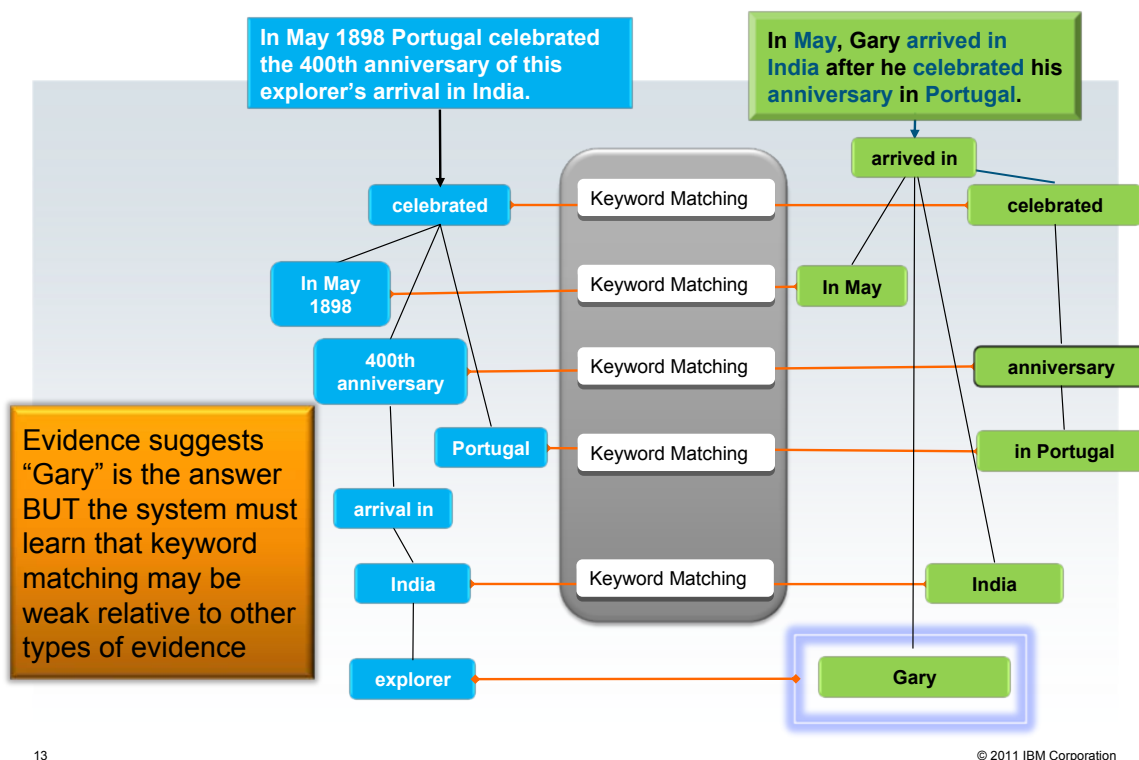
- A kérdésben megjelenik, hogy a keresett dolog egy folyadék(liquid)
- A generált válasz jelöltek között ott található a *Cytoplasm*, amiről talált a rendszer egy olyan állítást, amiben megjelenik hogy ez egy folyadék(fluid)

- A rendszer adatbázisából többek között azt is megtudtuk, hogy a fenti két folyadékot jelentő szók szinonimák, így a rendszer megnöveli a válaszjelölt(Cytoplasm) pontját és az azt alátámasztó bizonyíték pedig a talált idézet lesz.

A pontozásnál használt algoritmusok is sokfélék lehetnek, például kulcsszó alapú(6. ábra) vagy okosabb, következtető(7. ábra)

- Időbeli következtetés: 400th évforduló = Május 1498
- Ekvivalens kifejezések, szófordulatok: arrival in = landed in
- Térbeli következtetés: India = Kappad Beach

Different Types of Evidence: Keyword Evidence



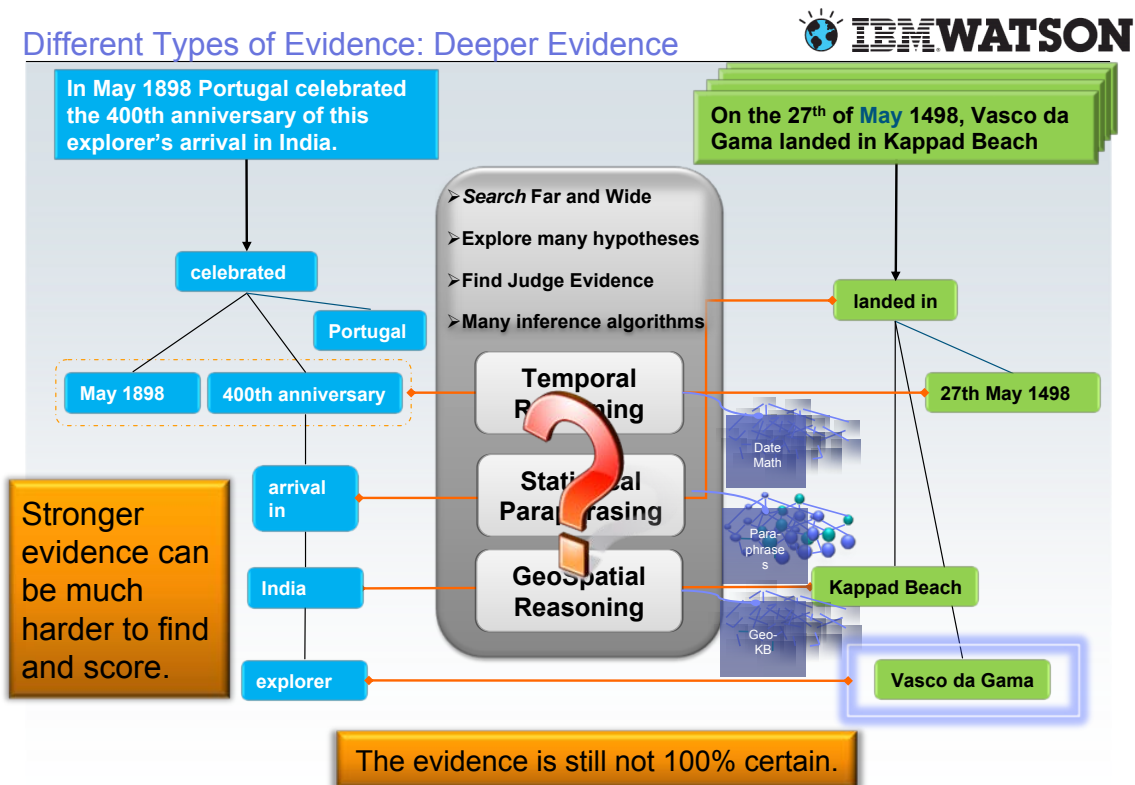
13

© 2011 IBM Corporation

6. ábra. Pontozás:Kulcsszó alapú

2.5. Final merging and ranking

Az utolsó lépés az ekvivalens válasz jelöltek fúziója illetve azok rangsorolása az őket alátámasztó bizonyíték profilok pontszámai alapján. A rangsorolás elvégzését a rendszer gépi tanulást alkalmazva tanulta. Minden kérdés típushoz tanítottak egy olyan rangsorolót, ami az adott típusnál fontos pontszámok alapján rangsorolja a válasz jelölteket.



14

© 2011 IBM Corporation

7. ábra. Pontozás: Következtetés alapú

3. Összefoglalás

Mint látható az IBM elérte célját, létrehozott egy rugalmas, kiterjeszhető architektúrát és látványosan bemutatta működését a Jeopardy! kvízzjáték során. A fent bemutatott lépésekből kiderül, hogy habár maga a Watson egy csak is Jeopardy-ra specializálódott kérdésválaszoló rendszer, a megvalósításához használt lépések, elvek bevethetők komolyabb területeken is, mint például a gyógyászati, jogi és üzleti problémák megoldására.

Hivatkozások

[1] <http://www.cs.hku.hk/watson/>

[2] <http://www.aaai.org/Magazine/Watson/watson.php>