Nyelvtech kidolgozás

Leichner Dávid

2016. december 12.

Tartalomjegyzék

1. Karakterek, kódolási szabványok, rendezések, n-gramok, karakter- és n-gram-alapú statiszikai megfigyelések

Ez egy elég alapvető informatikai tétel. Először is néhány definícióval érdemes kezdeni.

Karakterkódolás

- **Karakter:** absztrakt objektum, a számítógépen tárolt szövegek legkisebb, tovább nem bontható egysége. Egy karakter egy betű, szám, írásjel, speciális jel, illetve egyetlen egyszerűbb szövegformázó parancs ábrázolására alkalmas
- **Karakterkód:** a karakternek mint absztrakt objektumnak megfeleltett számkód, amely számítógépen közvetlenül ábrázolható
- **Karakterkészlet:** karaktereket és számkódjaikat összerendező táblázat. Meghatározza, hogy milyen karakterek használhatók és azt is, hogy az egyes karakterkódokat miképpen kell értelmezni. A karakterkészlet megjeleníthető karaktereket és speciális műveleteket előíró vezérlőkaraktereket tartalmaz
- **Kódlap:** több karakterkészletet is kezelő számítógéprendszerben egy saját számkódjával azonosított karakterkészlet
- **Betűkészlet:** olyan táblázat, amely adott karakterkészlet(ek) megjeleníthető karaktereihez karakterképeket rendel, amelyek a számítógép képernyőjén, vagy nyomtatásban jelennek meg. Az alkalmazott betűkészlet a szöveg számítógépbeli értelmezésére - például a szöveget nyelvi szempontból feldolgozó programokra - nincs befolyással, csak annak megjelenítésekor használatos

Kódolási szabványok címszó alatt igazából a szokásos mesékre gondolhatunk. Elkezdtük a jó kis BCD-t meg EBCDIC-t, de rájöttek, hogy ez kevés is (asszem 6 bit) meg gagyi, szóval jött pár új tábla, pl ASCII 7 bittel, de a drága amcsik elfelejtették a többi nyelvet, így jött be még sok-sok karaktertábla. Amikről az órai dikából szó volt az kettő:

1. UCS: Universal Character Set
2. Unicode szabvány

Na hát az 1. az európai a 2. az amerikai. Ezek 16 bitesek, ami baromi sok karakter. Gyakorlatilag minden beszélt és holt nyelv, illetve sok egyéb nyelv (pl. Braille). Minden egyes karakternek száma és neve van:

- U+0041 → Latin capital letter A
- U+0151 → Latin small letter o with double acute

Manapság viszont már ez is kevés. Ma már inkább 4 bájton (32 biten) tárolják a karaktertáblát.

- UTF-8 (Unicode Transformation Format): változó hosszúságú kódolással (1-6 bájt) képezi le a Unicode karaktertáblát, pl. 1 bájton tárolt kódjai az ACII-nak felelnek meg, így a latin betűs UTF-8 kódolású szövegek a régi ASCII környezetben is olvashatóak maradnak

- UTF-16: 1112064 karaktert kódol
- UCS-4: 4 bájt (32-bit) = UTF-32 (pontosan 32 bit)

Na ezek után 1980-ra igény volt egy egységes rendszerre. Nosza maradt az amcsi-európai vonal is, csak az európaiak nevet változtattak:

- **ISO 10646** - az ISO szervezet szabványa
- **Unicode** - amerikai cégek konzirciuma

1991-ben szépen egységesítették őket, de azért mind a kettő megmaradt. Teljesen kompatibilisek azóta is. Vannak különbségek, pl Unicode-ban vannak egyeztetési szabályok.

Rendezési elvek

Najó izgi. Szóval van sok nyelvünk, ezeknek van ABC-je és ABC rendje. Mondjuk a képírásnál már bonyolultabb a helyzet, de hagyjuk. Az átjárást az ábécék között transzliterálásnak hívjuk, itt vannak problémák. (Sok betűt összeegyeztetni két nyelvben például.)

Na most itt van sok féle rendezés, érdemes megemlíteni a könyvtári rendezést. Ehhez baromi nagy és sok sajátságos szabály kell. A magyar akadémiai rendezés ezen szabályokat gyűjti össze. Itt minden szépen le van írva. Van pár példa a diasorban is, ezeket nem írtam ide mert van életem.

Karakter- és n-gram-alapú statiszikai megfigyelések

Létezik a világban olyan, hogy betűstatisztika. Itt arra kell gondolni, hogy mondjuk a betűk milyen valószínűséggel fordulnak elő egy szövegben.

- Hangstatisztikák, betűstatisztikák
- Titkosírás-megfejtések
- Morse-ábécé, írógép-billentyűzet
- A nyelvek tipizálhatók n-gráfjaik alapján (pl. mássalhangzótorlódással kezdődő szavaink régen egyáltalán nem voltak, és ma is csak bizonyosak vannak)
- Információ (Shannon): kisebb valószínűségű üzenetnek az információtartalma nagyobb, és független események együttes információtartalma az események információtartalmának összege. (Tehát szétbonthatóak (lineárisak), akár anno infocode-ból)

Ide lehet kötni az infocode-t is.

- Az x jel információtartalma: $I(x) := p(x)log_2[p(x)^{-1}]$
- Itt még volt a lineráis tul. leírása ezt gondolom mindenki jól ismeri

Most pedig következhetnek a híres-neves n-gramok. Az egymás után következő n darab karakter egy szövegben az n-gram. Ezek lehetnek teljes szavak is, de nem feltétlenül azok. Az előnyük, hogy nyelvfüggetlenek és hibatűrő megoldások megvalósítására igen alkalmasak.

Ezekből lehet például magyar betűstatisztikát összedobni (1-gramok) ekkor látható, hogy melyik betű, milyen valószínűséggel fordul elő. (A: 9.35, X 0.01).

Mire jók ezek?

1. nórmlgenytnaeőettesy sdrúk üze neée issgnis k, őentetnsmrdttca.

2. Bégiz érj, ercszépot, ekalfönembenyett kmáb os alőrre.

3. Ezenlapoltike lehátszökmég hog, aztradon két. Korcsant ván.

4. Hárone - Az a léhez, meg, akarózsák hogy öregész egény látán, járnagyon kérdeket egy emberet. Tordárd sem tan.

5. Senki adsaadott, hogy egy ki tornak, a kapta a mielőtte szor ismerni most vele aótsap a eipővel. - Ó, biztosítás úr csak társallatság vele. Hallj ott.

6. Történik. Telt év óta meglátta bevallott kissé lehajtott és mindig, légéz erligyúneáenl eltűnt.

Megfigyelhető, hogy a 4 után nem javul igazán a minőség, Miért? Mert kicsi tanítókorpusz esetén (esetünkben 8 Mbájt) a négyesnél nagyobb csoportok eloszlásának megbízhatósága már nem elegendő.

Egy kis statisztika

A **Zipf-törvény** lényege, hogy összeszámoljuk egy szövegben az előforduló szavak gyakoriságát, majd sorba rendezzük őket gyakoriságuk alapján. Ábrázoljuk rang szerint (azaz annak a függvényében, hogy az illető szó hanyadik a sorban) egy hatványfüggvény szerint lecsengő görbét kapunk -1-hez közeli kitevővel. $R_i \sim \frac{1}{i^\alpha}$

A **Heaps törvény** szerint egy N szóból álló korpusz esetén a különböző szavak száma:

$$V \approx K \cdot N^\beta \tag{1}$$

ahol K : a szövegtől függő konstans ($10 \leq K \leq 100$), β : a szövegtől függő konstans (magyarnál: $0.6 \leq \beta \leq 0.7$)

2. Véges állapotú automaták (és kiterjesztéseik) nyelvtechnológiai alkalmazásai

No hát akkor először is el kell hogy merüljünk a szófák csodaszép világában. A szófa (Fredkin 1960) egy olyan, a szavak rákövetkező karaktereivel címkézett élsorozatokat tartalmazó fa, amelyben egy szót úgy találunk meg, hogy végigjárjuk karakterenként.

A szófák királyak, mert:

- Bináris keresőfák: $O(mlog(n))$ (n a fában tárolt elemek száma)
- Szófa: m hosszúságú kulcs megtalálása max. $O(m)$
- Nagy számú rövid füzér tárolása esetén a szófa kevesebb helyet igényel, mint a bináris keresőfa (ui. a kulcsokat nem tároljuk, a csomópontokat meg közösen használják az egyforma kezdőszeletű füzérek kulcsai)
- Hasítótáblák helyett is használható (bár néha a szófák lassabbak)
- Nem mindenre jó: vannak füzérként nehezen ábrázolható kulcsok (pl. a lebegőpontos számok)
- De: szótárak ábrázolására alkalmas
- ADFSA (körmentes determinisztikus véges automata) a szófánál is jobb, de csak ha nincs kiegészítő információ (csak puszta szólista)

Alapvetően sok szófa fajta van, mi hármat nézünk meg: általános szófa, módosíttt (kompakt) szófa, erősen módosított szófa. A különbség, hogy az erősen módosított szófák élein több betű is lehet.

A Kay-féle szóábrázolás lényege, hogy tömörítunk numerikus prefixekkel. Másik érdekesség a szuffixum-szófa, melyben egy füzér minden lehetséges végszelete tárolva van. Ezeknek n levelük van és magasságuk is n. A szuffixum-szófának is vannak érdekes tulajdonságai.

- $O(m)$ időben eldönthető, hogy egy m hosszú P füzér a részfüzére-e
- $O(m)$ időben megtalálható egy tetszőleges m hosszú P részfüzérének első előfordulása
- $O(m + z)$ időben megtalálható mind a z darab előfordulása egy m hosszú részfüzérének
- Az S_i és az S_j füzérek leghosszabb közös részfüzére megtalálható $O(n_i + n_j)$ idő alatt

Na mostmár rátérhetünk a tétel lényegére, vagy is az **automaták és véges állapotú mofológiák** témakörére.

Nyevel, nyelvtanok, automaták

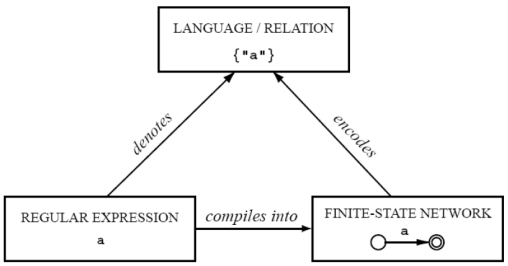
- **A formális nyelv** egy formális nyelvtan által leírt füzérek halmaza
- $G = (N, T, R, S)$ egy formális nyelvtan, ahol
  1. **N** $\rightarrow$ nemterminális szimbólumok véges halmaza
  2. **T** $\rightarrow$ terminális szimbólumok véges halmaza

3. $\mathbf{R} \to (T \cup N) * N(T \cup N)* \to (T \cup N)$ alakú szabályok véges halmaza
4. $\mathbf{S} \in \mathbf{N} \to$ mondatszimbólumok

- Formális nyelvek és formális nyelvtanok Chomsky-hierarchiája, ahol $A, B \in N; a \in T$; $\alpha, \beta, \gamma, \in (T \cup N)*$:
  - 0-típusú nyelvtan ($\alpha \to \beta$): rekurzív megszámlálható nyelvek
  - 1-típusú nyelvtan ($\alpha A \beta \to \alpha \gamma \beta$): környezetfüggő nyelvek
  - 2-típusú nyelvtan ($A \to \gamma$): környezetfüggetlen nyelvek
  - 3-típusú nyelvtan ($A \to a$ and $A \to aB$): reguláris nyelvek
- Az elfogadó automaták hierarchiája:
  1. Turing-gép
  2. Lineárisan kötött automata
  3. Veremautomata - $O(n^3)$
  4. Véges állapotú automata - $O(n)$

Véges állapotú automaták

- **Véges állapotú automata:** $A = (S, \Sigma, s, F, T)$ ötös, ahol
  - S $\to$ állapotok véges halmaza
  - $\Sigma \to$ egy ábécének nevezett véges halmaz
  - $s \in S \to$ kiinduló állapot
  - $F \subseteq S \to$ a végállapotok halmaza
  - $T : S \times (\Sigma \cup \{\epsilon\}) \to S \to$ átmenet függvény
- Az $X = x_0 x_1 \cdots x_n$ Σ ábécéből alkotott füzért **A elfogadja**, ha létezik S-ben az átmenetek $r_0, r_1, \cdots r_n$ sorrendje a következő feltételekkel:
  1. $r_0 = s$
  2. $r_{i+1} = T(r_i, x_i)(i = 0, \cdots, n-1)$
  3. $r_n \in F$



1. ábra. Reguláris kifejezés - reguláris nyelv - véges gép

Na szóval a reguláris kifejezéseket leírhatjuk ilyen kis csodaszép automatákkal.

Véges átalakítók (FST)

- **Véges átalakító:** $T(S, \Sigma, \Gamma, s, F, \delta)$ hatos, ahol
  - S $\rightarrow$ az állapotok véges halmaza
  - $\Sigma \rightarrow$ a bemenő ábécének nevezett véges halmaza
  - $\Gamma \rightarrow$ a kimenő ábécének nevezett véges halmaz
  - $s \in S \rightarrow$ a kezdő állapot
  - $F \subseteq S \rightarrow$ az elfogadó állapotok véges halmaza
  - $T : S \times (\Sigma \cup \{\epsilon\}) \times (\Gamma \cup \{\epsilon\}) \rightarrow S \rightarrow$ átmenetfüggvény
- A T átlakítja az $\alpha \in \Sigma*$ füzért $\beta \in \Gamma*$ füzérbe (röviden: $\alpha[T]\beta$) ha létezik út a kezdőállapotból egy végállapotba α bemenősorozat és β kimenő sorozat mellett
- Az FSA és az FST különbsége: az FSA kimenetén egy Boole-válasz jön létre, míg az FST egy füzért ad eredményül (a másik szalag tartalmát)

És végül az automaták köréből a velük elvégezhető és értelmezhető műveletek:

1. Konkatenáció
2. Unió
3. Iteráció
4. Metszett
5. Kompozíció

A nyelvtechnológiai felhasználás lényege, hogy megpróbáljuk a nyelvet kétszintes leírással megismerni, ezt automatákon keresztül tehetjük meg.

3. A számítógépes morfológia alapvető módszerei (kétszintes, unifikációs)

A kétszintes leírás felé...

Johnson felismerése a **reguláris reláció**. Ez röviden a következő:

1. $aa^nb^nb \rightarrow$ környezetfüggetlen nyelv
2. $a(ab)^nb \rightarrow$ reguláris nyelv

Kaplan & Kay (1981/1994)

Ez már egy nehezebb dióka. Egy hatékony elemzőt akartak írni a komák. Na a lényeg, hogy észrevették, hogy ha van két olyan szabály, amit átalakítóval modellálunk, akkor egyiket a másik bemenetére kötve egyetlen ekvivalens átalakítót kapunk. Ez azért király, mert nem generál köztes eredményt, illetve a szabályok számától függetlenül tetszőleges számú fonológiai újraírás következhet be.

De itt van egy kis bibi. A szabálymegfordítás problémája. Röviden: Ugyanazon generáló szabályok elemzésre való használatakor a kötelező szabályok opcionálisakká válhatnak: a kaNpat lexikális füzérből a szabályok egyetlen felszíni alakot generálnak, de a felszíni alakból az inverz leképezés „egy a sokhoz" típusú is lehet.

Ennek a problémának a megoldása a véges átalakítók soros vagy párhuzmaos kapcsolása.

Véges átalakítók soros kapcsolása

- A fonológiai levezetések köztes állapotai mindig kiküszöbölhetők az egyes szabályokból kapott átalakítók kompozíciójával: az eredményül kapott átalakítónak csak két szintje van, a lexikális és a felszíni
- Az egyetlen generatív átalakító használata sokkal hatékonyabb felismerésre is, mintha az eredeti szabályoknak megfelelő átalakítók egyenként invertált sorrendben működnének
- Kaplan és Kay megoldotta ugyan az egyes szabályok sorozata átalakítóba való fordításának problémáját, azonban a nagy szabályrendszerek egyetlen átalakítóba való kompozíciója az akkori technikai korlátok miatt a gyakorlatban megvalósíthatatlan volt

Véges átalakítók párhuzamos kapcsolása

- Tudva, hogy a lexikális és a felszíni alakok közötti megfeleltetés leírható reguláris relációval, Koskenniemi egy lényegi változtatást, a szabályhalmazból származó átalakítók párhuzamos kapcsolását javasolta

A kétszintes szabályok alakja

- Három részből áll: megfeletetett párok, környezet, szabályoperátor
- Megfeletett párok: egy lexikális és egy felszíni karakterből álló pár
- Környezet: az adott jelenséget körülvevő fonológiai helyzetet specifikálja (az aláhúzás jelenti a megfeleltetett pár helyzetét az adott környezetben)

- Szabályoperátor: a megfeleltetett pár és a környezete között fennálló reláció; nagyjából a formális logika feltételeket és következményeket leíró operátorainak felelnek meg:
  - $\rightarrow$ a megfeleltetés csak akkor áll fenn, ha ez a környezet
  - $\leftarrow$ ha ez a környezet, akkor a megfeleltetés fennáll
  - $\leftrightarrow$ a megfeleltetés akkor és csak akkor áll fenn, ha ez a környezet
  - $/ \leftarrow$ a megfeleltetés soha nem fordul elő ebben a környezetben

Generatív vs. kétszintes szabályok

- $t \rightarrow c/\_\_\_i$ hát ez ilyen egyirányú
- $t \Rightarrow \_\_\_i$ ez meg kétirányú

Most jönnek ezek a hacacárék: "kizárólag, de nem mindig", "mindig, de nem kizárólag", "mindig és kizárólag", "soha". Nyilván itt is van jó sok konvenció és speciális szimbólum. Ezeket most nem írnám le, mert úgy se jegyzi meg senki.

Kétszintes lexikonok

- Tőlexikonok
- Alternációs minták lexikonjai
- Toldaléklexikon

Vannak lexikonjaink, ezeknek elemei (lexémák) meg van egy ilyen P folytatási osztály mely: azt állítja, hogy az eredeti lexikális elemet vagy a PS allexikon (birtokos toldalékok), vagy a K0 allexikon (kritikumok) valamelyik eleme követheti, vagy egy határszimbólum.

Párhuzamos szabályalkalmazás és a szótárak

Itt a másik szinten további megszorításokkal bevezethetőek új lexikonok. A lexikonok meg igazából szó-fa erdők, az egyik fa gyökere a másik levelénél van és így jó. Ezek a folytonos lexikális szűrők.

Nehézségek:

- A felszín és a lexikális alak kötelezően azonos hosszúsága
- A szuppletív (lexikalizálódott) alakok és a nem produktív toldalékolás kezelése
- Ha a szótár „mindent kibír", miért kell a „nehéz" alakokat is levezetésekkel kezelni?
- Mit ér a reguláris rendszer, ha vannak reguláris nyelvvel nem leírható morfológiai jelenségek?

Unifikáció

- A jegyszerekezetek irányított körmentes gráfokkal reprezentálhatók, ahol a csomópontok felelnek meg a változóknak és az ösvények a változónevek
- A jegyszerkezetek leggyakrabban jegy-érték mátrixokként írják le, melyeknek két oszlopuk van (az egyik a jegyek nevének, a másik az értékeinek)
- Az érték vagy atomi érték vagy egy másik jegyszerkezet
- Jegyszerkezet-unifikáció: két jegyszerkezeten működő művelet, mely csak akkor hiúsul meg, ha van olyan ösvény mindkét szerkezetben, melyeknek eltérő atomi értékei vannak; minden más esetben a létrejövő szerkezet az eredeti két szerkezet összes ösvényét tartalmazza (az azonosakat egyeejtve)
- Az unifikáció absztrakt művelet, és nem a morfológiához tartozik, csak a legegyszerűbb formáját itt tárgyaljuk először

Az unifikáció művelete

Az unifikáció olyan művelet, mely egyedül akkor nem hajtható végre, ha a tagokban egyazon jegy különböző értékekkel szerepel:

- $[NUMBERSG] \cup [NUMBERPL] = fail!$
- de:
- $[NUMBERSG] \cup [NUMBERSG] = [NUMBERSG]$
- $[NUMBERSG] \cup [NUMBER[]] = [NUMBERSG]$
- $[NUMBERSG] \cup [PERSON3] = [[NUMBERSG][PERSON3]]$

Az unifikáció implementációkban

Destruktív unifikáció:

- Két szerkezet unifikációjakor az egyik megszűnik, és helyén létrejön az eredményszerkezet
- Gyors
- Bizonyos esetekben nem kívánt hatása is lehet (pl. ha egy szabályt alkalmazunk, annak formáját nem szeretnénk a művelet következtében megváltoztatni)

Nem-destruktív unifikáció:

- A kiinduló szerkezetek nem változnak, az eredmény új, harmadik szerkezet
- Az implementációkban igen gyakori
- Könnyű kezelni, viszont másolási műveletet igényel
- Az unifikálhatóság-ellenőrzés egy reláció, igen / nem eredménnyel, más kimenet nélkül

Az unifikációról általában

- Az unifikáció, mint művelet nem a morfológiához tartozik, mindössze egy egyszerűbb változatát itt tárgyaljuk először
- Az unifikációt elsősorban irányított ciklusmentes gráfok esetében szokás használni
- A morfológiai leírás esetében egyszerűbb struktúrákon működtetjük az unifikációt
- A DAG több mint fa, és épp ezáltal használhatóbb nyelvi szerkezetek leírására, hogy egy csomópontnak a nyelvi szerkezetben sokszor kell, hogy több szülője lehessen
- A közös csomópontok elsősorban a szintaktikai szerekezet kezelésénél válnak fontossá

4. A számítógépes morfológia alkalmazásai (helyesírás, elválasztás, intelligens keresés, intelligens csere

A téma elején tekintsük a hibák és normák kezelésését:

- A helyesírási normák, a szabályok és a gépi helyesítás-ellenőrzés viszonya
- A helyesírási szabályzatok viszonya a gépi helyesírás-ellenőrzéséhez
- Formai és jelentéstani szempontok gépi kezelése
- A gép a "nyelvhelyességi tanácsadó" szerepében
- A nehezen értelmezhető elemzések okozta porblémák

No most, ha hibát szeretnék javítani, akkor vizsgálni kell a tipikus hibákat:

- Karakterhibák, elütések
- Valódi helyesírási hibák
- Nyelvhelyességi hibák
- Tipográfiai hibák
- A helyesírás-ellenőrzés lehetőségei és korlátai a szavak szintjén
- A szóellenőrzés és az ún. nyelvehelyesség-ellenőrzés viszonya
- A nyelvi programrendszer lehetséges hibái: a túlgenerálás

A javítási igényekenek két fajtájuk van:

1. Javítható (betűhibák, egybeírások, kötőjelezés, szóismétlés)
2. Nem javítható (Hibás különírások, központozási hibák, szórendhibák és még sok más)

A szóellenőrzés moduljai

1. Alapszótár illetve morfológia
2. Ajánlómodul és adatbázisai
3. Időleges sajtászótár
4. Kiegészítő szótár
5. Kizáró szótár
6. A ragozó nyelvek esetén hasznos a toldalékoló sajátszótárak lehetőségei és nehézségei

Levensthein táv

Hamming általánosítás. Van benne csere, beszúrás, törlés. Ide kötném a Damerau-Levensthein távot, ahol helycsere is van.

A szóellenőrzés menete

1. Morfológiai elemzés
2. Ajánlás
3. Ellenőrzés a generátumok morfológiai elemzésével

A mondantszintjén felismerhető hibajelenségek

Na jó meg kell hagyni itt csaltam egy mikonnyit, hisz ha nagyon figyelmes voltál rájöhettél, hogy eddig szó szintjén detektáltunk hibát. Most azonban mondant szintén fogunk, ez sokkal komplexebb és fincsibb, ezért csak megnézzük hol tartunk.

- Névelő egyeztetés
- Vesszőhiány
- Algoritmikus szóösszetétel
- Szemantikus szóösszetétel
- Hibás szóösszetétel
- Hiányos szerkezetek
- Téves szóhasználat
- Idegen szavak szűrése
- Terjengős kifejezések
- Trágár szavak szűrése
- Szóközhiány, -felesleg

Automatikus szövegelválasztás

Igen ez tényleg a szótagolásról fog szólni. Vannak alapszabályok, meg morfológiai felülbírálások. Beszélnünk kell arról is, ha valami kellemes, hosszabb szót választunk el, akkor az algoritmusunk mennyire legyen "engedékeny". Továbbá azért figyelni kéne a különleges elválasztásokra mondjuk az asz - szony - nyal (de kajak fontos).

A probléma például, hogy angolban alapból nem létezik egység, tök mindegy.

Számítógépes szinonimaszótárak és tezaszauruszok

- A szinonimákról
- Szinonimaszótár, vagy tezaszauruszok
- Tárolási és keresési problémák
- A rokonértelműség definíciója

- Az automatikus csere problémái
- Tő-visszaállítás
- Többértelműségek kezelése
- A lexikai és a szintaktikai szó különbségéből adódó nehzségek
- Az összetett szavak szinonímáinak problémája
- Morfológiai generálás minta alapján

5. Formális nyelvtanok, szintaktikai reprezentációk (közvetlen összetevős, függőségi, X-vonás), modatelemzési módszerek (TD, BU, kombinált)

Szóval na a formális nyelvtanokról esett már pár szó a második tételben. Tehát védjük a fákat és lapozzunk többet.

A szintaktikai reprezentációk gyakorlatilag fák. A közvetlen összetevős egy szép egy oldalra rendezett fa. A funcionális meg ki van egyensúlyozva. Az összetevős és függőségi reprezentáció pedig valahogy mindkettő. Kb az amit gyakon csináltunk. A X-vonás egy fokkal érdekesebb:

- $S \to NPVP$
- Az összetevős szerkezetben az NP és a VP "testvérek", azaz mindketten az S "gyermekei", de ezt nem fejezi ki a függőségi leírás
- Azt viszont a közvetlen összetevős leírás nem fejezi ki, hogy testvérek bár, de nem egyforma súllyal, ui. a VP a szerkezet feje
- X-vonás szabályként: $V'' \to N'V'$
- Azaz: a V'' a V maximális projekciója, tehát a mondat feje az ige!
- Csak endocentrikus

A tulajdonságok:

- Megőrzi mind az összetevős, mind a függőségi leírás előnyeit
- Szófajok feletti általánosítás: $X = V/N/P/Adj/Adv/Det/Q/\cdots$
- Az X-nél magasabb szerkezeteket is egységesen lehet vele leírni
- Megkülönböztethetők a különféle jellegű bővítmények (pl. előtte, mögötte)
- Szintjei: lexikális szerkezet, köztes szerkezet, maximális szerkezet

Az X-vonásnak vannak szabályai ezeket megint nem írom le, mert annyira azért nem viccesek.

Mondatelemzési módszerek

Alulról-felfelé (BU) elemzés - felülről-lefelé (TD) elemzés. **TD vs. BU**

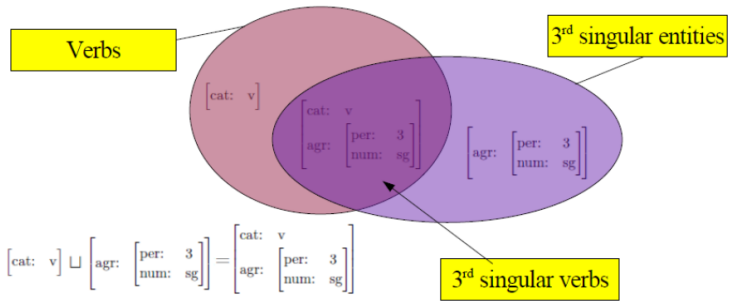
- TD: nem fordulhat elő, hogy ne mondat-szimólummal végződő elemzéssel tölsük az időt, és nincs olyan részszerkezet, melynek ne lenne helye a végső fában
- BU: nem fordulhat elő, hogy a bemenettel inkonzisztens szabályokkal kellene töltenie az időt, és nincs olyan részszerkezet, mely ne lenne kompatibilis a bemenet valamely részével
- Optimális megoldás: valamiféle kombináció, pl. BU-elemzés előfordulási valószínűségekkel; TD-elemzés "BU jóslással"

6. Az unifikáció és alkalmazásai a nyelvtevhnológiába (morfológiában, szintaxisban)

Először is beszélnünk kell a jegyszerkezetekről.

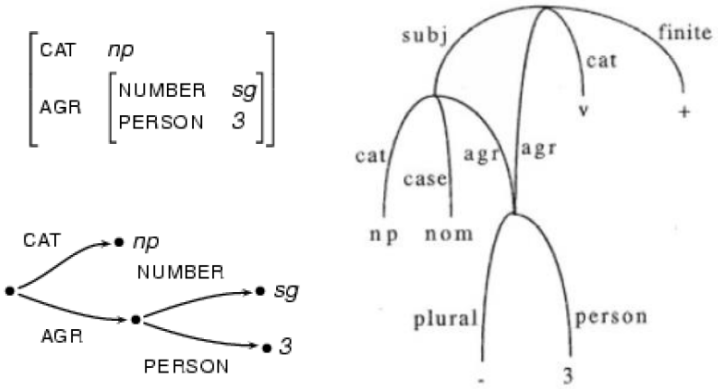
- A kategóriák (szófajok) nem atomiak, tovább alosztályozhatók: VPto, Sthat, Non3sgAux...
- Az efféle jegyek hierarchiába rendezhetők
- A szabályok így szabályosztályokká válnak
- Jegy-érték párok
- Jegyek: nyelvtani tulajdonságok, értékek: atomi szimbólumok és jegyszerkezetek
- Jegy-ösvény: nem egyetlen jegynek van értéke, hanem az ösvénynek
- Például: [egyeztetés[szám]] = Plur
- Fej-jegyek és láb-jegyek

De, hogy mi is volt ez? Nos a lényeg, hogy ezeket a diszjunk jegy-érték párokat unifikálhatjuk.



2. ábra. Az információk unifikációja a denotációk metszete

Ezeket lehet ábárzolni DAG-ként is.



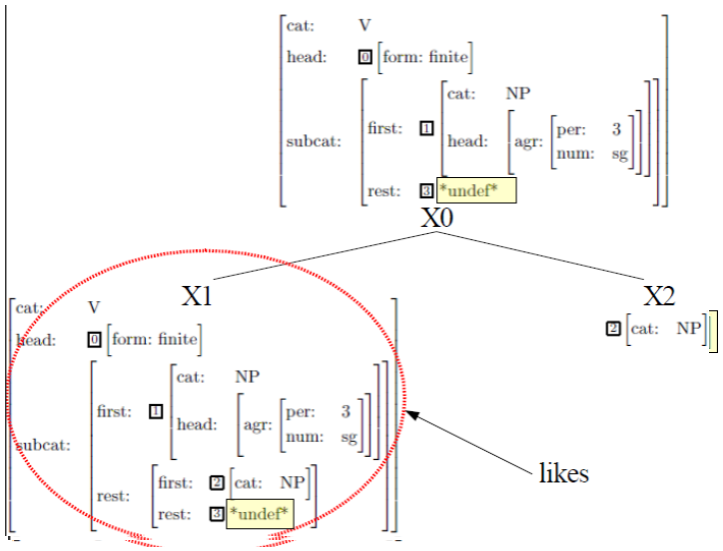
Ezeket még reprezentálni is lehet, hogy szépen tudjuk őket kategorizálni.

- Ösvényekre vonatkozó egyenletekkel: $< cat >= v$, vagy $< finite >= +$
- Jegy-érték mátrixokkal:

$$\begin{bmatrix} cat : v \\ finite : + \end{bmatrix} \tag{2}$$

  Itt persze lehetnek mátrixok a mátrixban.

Ezekkel tudunk alkotni szabályokat is:

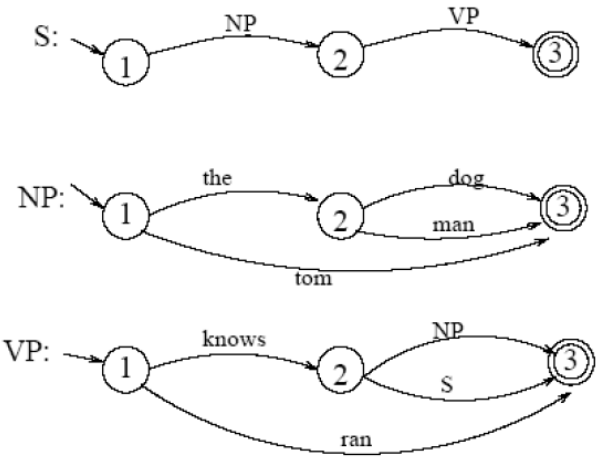


Az unifikációra már jó sok algoritmus létezik pl. Earley-algoritmus. További kiterjesztések:

- Negáció
- Diszjunkció
- Ösvény-egyenlőtlenség
- Halmaz-értékű jegyek
- Jegyszerkezet-leíró metanyelv

A véges állapotú eszközök kiterjesztései szintaktikai elemzéshez

Itt van mondjuk az RTN (Recursive Transition Network). Ezekkel tudunk alkotni szabályokat is:



Itt a szokásos FSA működtetésén túl figyelni kell:

- az aktuális bemeneti pozíciót,
- az aktuális állapotot és
- hogy hova kell visszatérni
- összegezve: veremkezelés kell

Az RTN összefoglalása:

- Az RTN egymást hívó FSA-k hálózata: az élek címkéi közt megjelenik más FSA-k "neve"
- A FSA (a reguláris nyelvek) $O(n)$ idő alatt elemzhetők
- Az RTN valójában egy veremautomata, azaz környezetefüggetlen nyelvek elemzésére is alkalmas
- Az RTN-nel tehát $O(n^3)$ elemzési idő garantálható

Van itt még ATN ami az RTN bővítése. Itt vannak élcímkék, értékek, tesztek és akciók. Az ATN formalizmusra is van 1-2 algoritmus hagyományosan létezik XML leírása is.

7. Jelentésreprezentációk, szemantikus hálók, fogalmi gráfok, ontológiák, WordNet

Jelentésábrázolás

Jelentés vagy világismeret? Fontos mind a kettő, hiszen a környezet nélkül esélytelen a jelentéstreprezentálni. A világismeret reprezentáció mindig a környezet megismerésére fókuszál, vagy is nem szó szinten akarja megérteni az élet nagy dolgait, hanem mondat, vagy még ennél is nagyobb egységek szintjén.

Ez eltolódott (mint kb minden) a számítógépes jelentésábrázolásban

- Matematikai logikai reprezentációk
- Konceptuális reprezentációk:
  - fogalmi gráfok
  - fogalmi függőség
- Lexikális szemantikai reprezentációk:
  - lexikális szemantikus hálók, ontológiák
- Mélytanulás, neurális hálók (suprise...)

Itt is van egy sorrendiség, hogy hogyan kellenek ezek a logikák, hogy fejlődtek:

- Elvi problémák
- Elsőrendű logika
- Magasabb rendű logikák
- Modális logikák
- Intenzionális logikák
- Montague elmélete (1970)

Montague-nyelvtanok

Az alapfeltételezés: a mondatok jelenzése igazságfeltételekkel megadható. Vegyük a következő mondatot: Péter olvas egy könyvet. $\Leftrightarrow$ Péter olvas egy könyvet. Érthető nem de? Tehát mindig írhatunk egy egyszerű szabályt, amivel lefedhető a mondat igazságtartalma, így az egész nyelvtan. Ezeket megadhatjuk logikai formulákkal (formalizálás meghitt folyamata):

$$\text{Péter olvas egy könyvet} \rightarrow \exists x(\text{könyv}(x) \wedge \text{olvas}(p*, x)) \tag{3}$$

Ez után jöhet az indirekt interpretáció: TNY $\rightarrow$ logika $\rightarrow$ modellek.

Még azért ide leírom a kompozicionalitás elvét: egy komplex kifejezés jelentése a részei jelentéseinek és az őket leíró szintaktikai szerkeztnek a függvénye.

Na, ez a gyakorlatban eléggé hasonlít arra, amit a Borival csináltunk gyakon. Itt kategorizáljuk a szavakat (Mondat: S, intranzitív igék: V, főnevek: N). Megtudunk adni ilyen levezetett kategóriákat ebből:

- Főnévi csoportok: S/V
- Tranzitív igék: V/(S/V)
- Determinánsok: (S/V)/N

A logikai levezetésről több szót nem ejtenék, tanultuk DM II-ből ezeket, aki esetlen nem emlékszik, az talál kis infót a 6. dia 23. slide-ján.

Szemantikus hálók

- Címkézett gráf
- Csúcsai: objektumok (fogalmak)
- Élek: a csúcsok közötti különféle lehetséges relációk
- Az attribútumokat egyes relációk közvetíthetik a relációk mentén

Lexikális szemantikus hálók

Na ez a fontos. A WordNet is pl ezt használja. Pszicholingvisztikai irányzatokat követ, hierarchiák vannak benne.

Az ontológiát is ide kell írni, ami a "felfogás leírása". Ez egy jó kis filozófiai tudományág, a létezést, létező dolgokat vizsgálja, megadja a vizsgált univerzumban létező fogalmak kategóriáit, metafizikai kereteket. A célja a világnak a formális leírása. Az ontológia értelmezésével érvényes logikai következtetések végezhetők.

Fogalmi gráfok

Volt egy ember Sowa volt a neve, és voltak neki fogalmi gráfjai. Ehhez tartozik gráf-alapú tudásreprezentáció meg következési modell, illetve grafikus interfész az elsőrendű logikához. Itt igazából rengeteg példa van, ezeket ide nem írom le, mert mindenki csak kinevetne, de leírom, hogy a hatodik előadás 28 diájától kezdődően találtok párat. Amúgy vettük gyakon, még háziba is benne volt. Amikor valahogy az egész mondatból levezetünk egy csini gráfot úgy, hogy a fogalmakat tekintjük és ezekből következtetünk.

WordNet

- A legelső WordNet (1990): a Princeton WordNet
- Eredetileg az emberi agy nyelvi tudásreprezentációjának modellje
- A legnagyobb ingyenes egységes gépileg feldolgozható lexikai adatbázis
- Ma: WordNet 3.0
- A WordNet legfőbb jellemzői:
  - szemantikus háló
  - a csúcsok címkéi: a "jelentések" mint szinonímahalmazok (synset)

– az élek címkéi: a leggyakoribb lexikai relációk
– az élcímkék szófajfüggőek: főnéviek, igeiek, melléknéviek

A WordNet-ben is van csomó reláció. Ezeket megint nem írom le, mert senki nem jegyzi meg, viszont van főnév, ige, melléknév és határozószó is.

8. Számítógépes lexikográfia: szótárépítés, szótárátalakítás, szótárhasználat támogatása nyelvtechnológiával

Először is miért kell ez? Mi a szerepe a nyelvtechnológiának a lexikográfiában.

- A szótárak célja ma
  - emberek számára készülnek
  - gépek számára készülnek
- Miből hozunk létre ma szótárakat
  - semmiből, technikai támogatással
  - szövegekből, gépi támogatással
  - meglévő szótár(ak)ból, gépi támogatással
- A géppel támagotatt szótárkészítés típusai
  - embernek írt forrásokból, ember számára
  - embernek írt forrásokból, gép számára
  - gépi forrásokból, ember számára
  - gépi forrásokból, a gép számára

Itt jön az a mese, hogy csak akkor tudsz jól szótárazni, ha valamennyire ismered a nyelvet, hogy az ember megért mindent, de lusta ezért a legjobbat akarja és ilyesmik.

Szótárak csoportosítása

A nyelvek száma szerint. Vannak egynyelvűek és kétnyelvűek, ja meg többnyelvűek. Az egynyelvűek között is két különböző van: címszavak és szócikkek.

A szótárak általános szerkezete

- Önálló és utaló szócikkek
- Szócikkfej (baloldal): címszó, homonimák, alak- és írásváltozatok, kiejtés, elválasztás, szófaj, főbb toldalékos alakok, nyelvtani megjegyzés, stílusminősítés
- Jelentéscsoportok (jobboldal) alapjelentés, jeletésárnyalatok, közmondások, más szavakkal alkotott összetételek, származékszók

A szótárelemek nyelvtechnológiai felhasználása

- Címszó: kiindulás helyesírási programokhoz
- Variánsok és toldalékolt alakok: a morfológiai rendszerhez
- Szótagolás: elválasztó programokhoz
- Kiejtés: beszédkeltő rendszerekhez

- Szófaj: egyértelműsítőkhöz
- Témakód: szövegtípus-azonosításhoz
- Definíciók: jelentés-egyértelműsítéshez
- Példák: a címszó körüli többszavas kifejezések azonosításához
- "Lásd még" szavak: szinonimák, antonimák

A többi dolog

Hát ezek eléggé rizsa szagúak nekem. Mármint nyomtatott vs. elektronikus szótárak. Manapság full sok platformra kijönnek, mobimouse, telóra. Vannak terminológiakezelők.

A korszerű internetes szótárszolgáltatás kritériumai

- Folyamatosan bővíthető szótárkínálat
- Sajátszótár-készítési lehetőségek
- Tetszőleges webes tartalom integrált megjelenítése
- A kifejezések intelligens kezelése
- Közösségi jelenlét
- Egymás segítésének és a (jogos) kritikának a fóruma
- A rendszer szemantikus ismereteinek erősítése a felhasználó keresési szokásainak elemzésével
- Könnyű keresés-indítási lehetőség
- Saját menthető beállítások a környezet személyre szabásához

Nyelvfüggő szótárproblémák

- A forrás- és célnyelv karakterkészleteinek ismerete
- A forrás- és célnyelv ábécérendjeinek ismerete
- A fonetikai információ kezelése
- Egységes jelölés: nyelvi keresésnél a szótár grammatikai információval való kompatibilitás

Keresési technikák

- Betű szerint
- Csonkolt keresés
- Hasonlósági keresés (fuzzy, soundex, spell)
- Nyelvi alapú keresés a bemeneti oldalon
- Nyelvi alapú keresés a találati oldalon
- A kifejezések kezelésének problémái: alcímszók, kulcsszó-választás, indexek, egyazon kifejezés több címszó alatt

Szótármegjelenítés

Itt van az LMF ami a Lexical Markup Freamwork.

1. Létező szótárak struktúráinak konzisztens feltérképezése
2. Az összes feldolgozott szótárat lefedő leírás létrehozása
3. 61 szakértő bevonásával az összes szóba jövő szótárszerkezet megvizsgálása

Az "intelligens" szótárak készítésének problémái

- A legfőbb baj: a szótárforrások XML-változatainak "amatőr" vagy legalábbis nyomtatáscentrikus megoldásai
- A második ok: a szótár az embereknek, nem a gépeknek készül
- Egy sor technikai probléma, ami a szótárak "papírszótár" mivoltából ered, ám a gépi változatban át kell ezeket alakítani

Emellé még van jó sok próbléma ezeket felsorolom, akit érdekel bővebben nézze meg (hatodik diasor 48. slide-tól).

- Perjel probléma
- Többszörös előfordulások problémája
- Az ellentmondó előfordulások problémája
- A tilde-probléma
- A morfológia-probléma
- A nagybetű-probléma
- A vonzat-probléma
- A példa-probléma
- A "lásd"-probléma
- A pontos találatok problémája

9. Párhuzamos korpuszok, szövegszinkronizálás, fordítómemóriák

A párhuzamos korpuszok

- Egy könyvben átlagosan néhány százezer szó van
- Egy művelt ember átlagosan nap tízezer szót olvas
- Ez összesen évi 3.5 millió és egy életben hozzávetőlegesen néhány százmillió
- Sok lefordított szöveg akár többszázmillió szava elektronikus úton elérhető
- A gép tehát több szóval tud találkozni, mint egy ember egész életében...
- Egy lehetséges konklúzió: a fordítási probléma a minták alapján statisztikai alapon megoldható, de legalábbis a fordító munkája másokéval jól támogatható, ha megtaláljuk az épp fordítandóhoz leghasonlóbb mondatot
- Van magyar-angol párhuzamos korpusz is

Szövegszinkronizáció

- "Alignment"
- Kétnyelvű szövegpárban az egymás fordításának tekinthető szövegegységek meghatározása
- Nyelvi egységek szerinti szövegszinkronizáció-típusok:
  - Bekezdésszintű
  - Mondatszintű
  - Szószintű
- Pontosság/fedés szerinti szövegszinkronizáció-típusok
  - teljes
  - részleges

Felhasználása

- Párhuzamos korpuszok építése fordítástudományi felhasználásra
  - két- vagy többnyelvű korpuszok szinkronizálása fordítási megfelelések vizsgálatra
- Fordítások terminológiai konzisztenciájának vizsgálata
  - a forrásszövegben megtalált szakkifejezéseket a fordításban megkeressük
- Kétnyelvű teminológia gyűjtése
- Fordítómemória építése
  - a párok kész fordításként felhasználhatók a későbbiekben

- Statisztikai gép fordítás
- Eszközök: TRADOS, MemoQ

Módszerek

Tudjuk őket csoportosítani: hossz-alapú, lexikai alapú, horgonykereső módszerek, hibrid módszerek.

Hossz-alapú szinkronizáció

- Teljes szinkronizáció
- Valószínűségi modell
- Dinamikus programozás
- Pozitívumok: globálisan optimális megoldást keres, teljes szinkronizációra képes, robosztus
- Negatívumok: nem elég a szövegységek hosszait vizsgálni, a beszúrásokat és az elhagyásokat rosszul kezeli, költséges $O(n^2)$.

Horgony-alapú szinkronizáció

- A horgonyok a szövegegységek egymásnak nagy pontossággal megfeleltethető pontjai
- Kezdetben két horgony a szövegpár végpontjait köti össze
- Ezek után olyan szavakat keres, amelyek gyakran fordulnak elő együtt a szövegpár két oldalán nagyjából azonos pozícióban (heurisztika), majd ezekből horgonyokat határoznak meg (de pl. az egymást keresztező horgonyokat elvetik)
- A horgonyok által kisebb szinkronizálandó szegmensekre bontható a szöveg, ezeken belül újra horgonyokat keresnek
- Egyes horgonyokkal csak részleges szinkronizáció valósítható meg
- Az eredmények nem elég jók

Fordítómemóriák

No hát ez egy olyan valami, ami szegmenspárokat tárol. TM forrásnyelvi szegmensek és célnyelvi fordítások párjait tárolja, némi adalékinformációval (rögzítés dátuma, fordító neve, fordítás státusza stb.). Ha később egy hasonló forrásnyelvi szegmenst kell fordítani, a TM program megtalálja és felkínálja a szegmens korábbi fordítását.

Alapfunkciók

- Szövegelemzés
  - központozás: egyfajta előfeldolgozás
  - markup: lehetséges segítség
  - "kódolt" szövegrészek azonosítása

- Szegmentálás
  - a fordítási egységek megállapítása
  - egynyelvű felszíni elemzéssel
  - a szinkronizálás alapvető feltétele
- Szövegszinkronizálás
  - forrásnyelvi-célnyelvi párok létrehozása
- Terminológiakivonatolás
  - induló szótárral, szövegstatisztikák segítségével
  - a fordítási munkával kapcsolatos becslések alapjául is szolgál
- Egyéb funkciók
  - Visszakeresés
  - Módosítás, javítás
  - Automatikus fordítás

Előnyök és hátrányok

Előnyök:

- Konzisztens dokumentumok előállítása: nagy projekteknél elengedhetetlen
- Gyorsítja a folyamatot: egyszer kell fordítani
- Pénzt takarít meg: a fordítóiroda tudja, mit tartalmaz az induló memória, illetve meg tudja számolni, hogy mit kellett "igazán" fordítani
- Egyetlen projekten belül is gazdaságos

Hátrányok:

- Azon alapul, hogy a mondatok "újrafelhasználhatók"
- A ma szoksás fordítói munkafolyamat átszervezésével jár
- Vannak dokumentumformátumok, melyeket még mindig nem támogatnak közvetlenül
- Idő, míg valaki megtanulja és rászokik
- Szabadúszó fordítóknak drága és sokszor bonyolult
- Minden bemenetnek elektronikusnak kell lennie
- A korábbi fordítások is átalakítandók a fordítómemóriának megfelelő alakúra
- A hibák is jobban szaporodhatnak a segítségével

Keresés a fordítómemóriákban

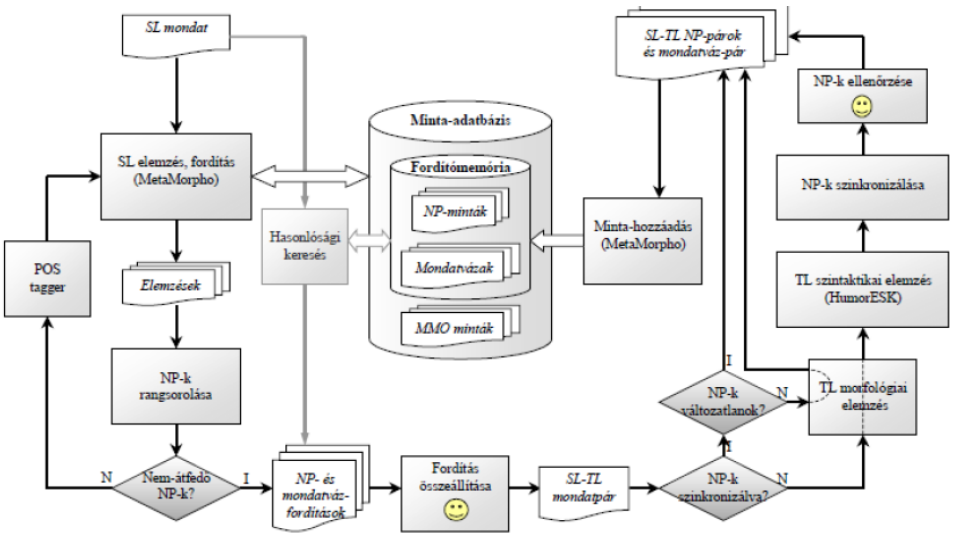
A TMZ-hez numerikus értékeket generáluk (fuzzy kulcs) és egy speciális indexfájlban tároljuk egyértelmű azonosítóval. Kereséskor ezt fejtik vissza (fuzzy index). A fuzzy kulcs általában valamilyen n-gram. Normalizálás: hosszú mondatok kis helyen való tárolását meg kell oldani.
A TMX két implementációs szintje:

- 1. szint: az adatok a $< seg >$ elemeken belül sima szövegek Content Markup nélkül, ami szoftverüzenetek fordításakor elég, formázott dokumentumoknál nem
- 2. szint (Content Markup): formázott dokumentumokra (ld. SRX)

Szabványok

1. TMX
2. TBX
3. SRX
4. GMX
5. OLIF
6. XLIFF
7. TransWS
8. xml:tm

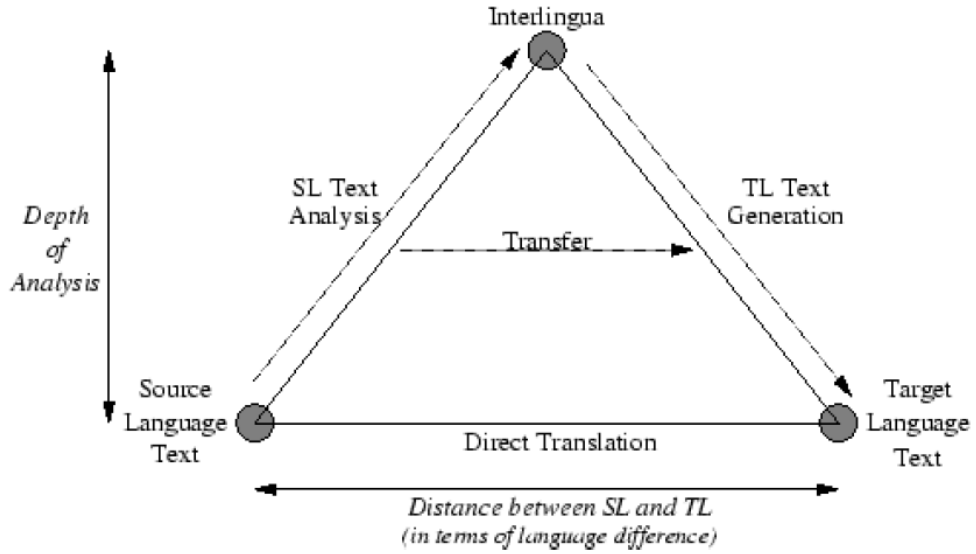
Fordítómemóriák a gyakorlatban



3. ábra. Az intelligens fordítómemória felé

10. Szabályalapú és statisztikai gépi fordítási módszerek

Szabályalapú fordítási stratégiák



4. ábra. Szabályalapú fordítási stratégiák

No hát ennek több formája van, ezeket fogjuk szépen sorban megvizsgálni.

Közvetlen fordítás

SL text → Morphological Analysis → Bilingual Dictionary Lookup → Local Reordering → TL Text

Ezeknek a lépései a következők:

1. morphological analysis
2. lexical transfer of content words
3. various work relating to prepositions
4. SVO rearrangements
5. miscellany
6. morphological generation

A közvetlen fordítás jellemzői

- Soha nincs kész semmilyen állapot sem, csak a végén
- Rengeteg lépésből áll
- Minden lépés egy adott nyelvi jelenséget dolgoz fel

- A legfontosabb eszköz a kétnyelvű szótár
- Komoly probléma a helyes kimeneti szórend és a megfelelő toldalékolás előállítás
- Ezt a feladatot átrendezésnek hívják

Példa: Systran és Meteo.

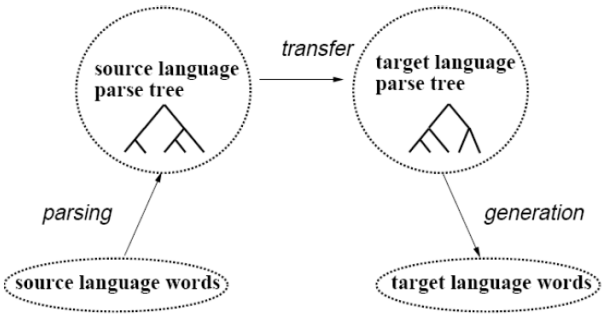
Közvetítőnyelves fordítás

Hát itt mint a nevében is benne van, használunk egy interlingua központot és akkor azzal dolgozunk.

Jellemzői

- Nem létezik mindent kielégítő közvetítőnyelv
- Kísérletek: a logikai formalizmusoktól a formalizált eszperantóig
- A soknyelvű rendszerek esetén nagyon "vonzó"
- $n > 3$ esetén $n \cdot (n-1) > 2 \cdot n$
- Végül is a két fordítási lépés nem más, mint egy-egy közvetlen fordítás

Transzfer fordítás



5. ábra. A transzfer fordítás menete

A transzfer fordítás menete és típusai

- Morfológiai elemzés
- Egyértelműség (POS, WSD)
- Lexikális transzfer: kétnyelvű szótárakkal
- Szerkezeti transzfer: frázisok, csonkok
- Morfológiai generálás
- Szintaktikai/felszíni transzfer: hasonló struktúrájú, különösen rokon nyelvek között
- Szemantikus/mély transfer: távolabbi rokonságban lévő nyelvek között

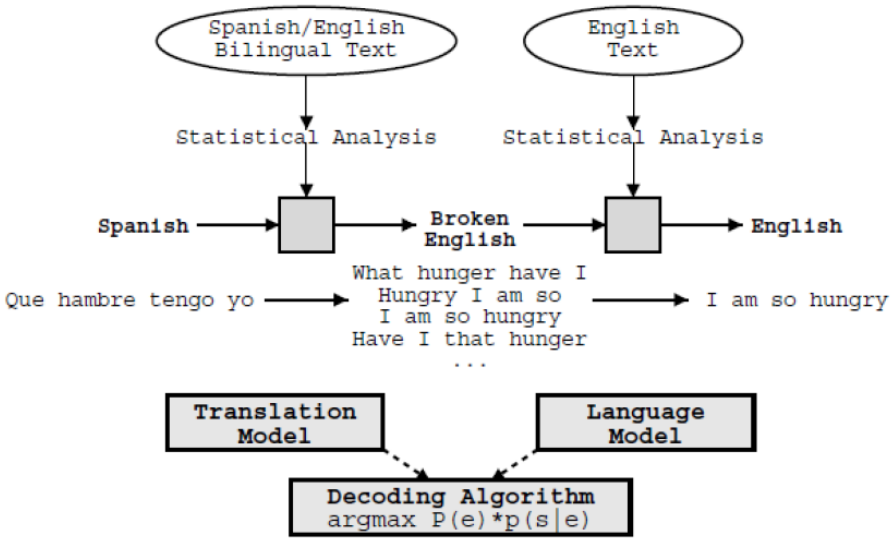
Egyebek

- Példa-alapú fordítás
- MetaMorpho: egy hibrid szabály/példa-alapú fordító

Nehézségek

- Lexikai többértelműségek kezelése
- PP-kapcsolások kezelése
- Struktúrális többértelműségek kezelése
- Többszavas kifejezések kezelése
- Idiómák kezelése

Statisztikai fordítás



6. ábra. A statisztikai fordítás alapjai

A három fő modul

- Nyelvmodell: adott egy e angol string, hozzá kell rendelni egy $P(e)$-t
  - jó angol string: magas $P(e)$
  - rossz angol string: alacsony $P(e)$
- Fordítási modell: adott egy $< f, e >$ stringpár, a hozzárendelt $P(f|e)$ valószínűséggel
  - ha $< f, e >$ egymás fordítása: magas $P(f|e)$

  - ha $< f, e >$ nem egymás fordítása: alacsony $P(f|e)$

- Dekódoló algoritmus: nyelvmodell + fordítási modell + bemenő f mondat, amikhez meg kell találni az e fordítást, a $P(e) \cdot P(f|e)$ maximalizálásával

A statisztikai GF lényege

- A naív Bayes-tétel:

$$\tilde{e} = argmax_{e \in e*} p(e|f) = argmax_{e \in e*} p(f|e)p(e) \tag{4}$$

- Az ötlet a beszédfelismerésből jön
- A legtöbb "minőségi" probléma a korpusz méretéből adódik
- Csak egy fordítási modell (feltételes valószínűségi rész) és egy nyelvmodell, (teljes valószínűségi rész) kell, meg a dekóder, ami számol

A fordítási modell

- A fordítási modell feladata megtalálni a legjobban illeszkedő bemenetet az e (angol) kimenethez
- $P(f|e)$ tanulását párhuzamos korpuszból lehet csak megoldani
- Tipikus helyzet: sokszor nincs elég adat a $P(f|e)$ közvetlen becslésre

Hogyan hozunk létre fordítási modellt

- EM-algoritmus (expectation maximization)
- Inicializálás: minden kapcsolat azonos súlyú
- Valószínűségi hozzárendelése a hiányzó adatokhoz
- Paraméterbecslés a teljes adatokból
- Iterálás

Megjelennek a frázis-alapú fordítási modellek

- Szószinten nehéz a törlés és a beszúrás, ezért megéri a bemenetet frázisokra szegmentálni
- Itt a frázis nem nyelvtani fogalom, hanem csak a gyakran előforduló összefüggő szósorozatok
- Először minden frázist lefordítunk a célnyelvre
- Ezután frázisokat átrendezzük
- Így nem kompozícionális frázisok is fordíthatók
- Minél több az adat, annál több hosszú frázis tanulható meg

A nyelvmodell kialakítása

- Ez azoknak a mondatoknak a halmazából indul ki, amit az adott nyelven mondani szoktak
- Milyen a "jó" nyelvmodell? Milyen a "jó" nyelv?
- Trigram-valószínűségekkel szokás közelíteni
- Pl. p(witch—the green) > p(green—the witch)
- A nyelvmodell kiindulhat a web n-gramjaiból is!

Ha nem elég nagy a korpusz

- Sokszor szükséges ún. simítás a nyelvmodellhez
- Ha a z sose követte xy-t a szövegeinkben, akkor még mindig megkérdezhetjük, hogy z legalább y-t követte-e?
- Ha igen, akkor xyz talán nem olyan rossz
- Ha nem követte, még mindig megkérdezhetjük, hogy z elfogadható hétköznapi szó-e? Ha nem, akkor xyz tényleg nagyon kis valószínűségű kell legyen!
- a = előfordulás("xyz") / előfordulás("xy") simítva:

$$\begin{aligned} b(z|xy) = \\ & 0.95 \cdot \text{előfordulás}(\text{"}xyz\text{"})/\text{előfordulás}(\text{"}xy\text{"})+ \\ & 0.04 \cdot \text{előfordulás}(\text{"}yz\text{"})/\text{előfordulás}(\text{"}z\text{"})+ \\ & 0.008 \cdot \text{előfordulás}(\text{"}z\text{"})/\text{összes-látott-szó}+ \\ & 0.002 \end{aligned} \tag{5}$$

A statisztikai GF problémái és megoldásuk(?)

- Tulajdonnevek: másolás, transzliterálás, szótárból újraalkotás
- Számmal alkotott kifejezések, dátumok, mennyiségek: saját fordítási táblák kellenek
- Főnévi csoportok általában: ezeket érdemes nyelvileg "elő-elemezni"
- A fejlődés a hibrid rendszerek felé mutat
- Alternatív elképzelések: létező rendszerek/szolgáltatások kombinációja
- webfordítás.hu → iTraslate4.eu
- Soknyelvű fordítás: újra "közvetítőnyelves" megoldások?
- Elindult a világ a neurális hálós fordítás felé