ALATI I METODE ZA OBRADU BITEKSTA I OBELEŽAVANJE KOMPOZITA

Kandidat:
Mirjana Ljuboja

Mentor:
Duško Vitas

Matematički fakultet
Univerzitet u Beogradu
2009

Sadržaj

Sa razvojem globalne svetske mreže, u poslednjih par decenija došlo je do povećanog protoka informacija širom sveta. Ovo je, neizbežno, stvorilo potrebu za bržim i lakšim sprazumevanjem između strana koje se koriste različitim jezicima, a time dalo podstrek razvoju načina da se jezičke barijere premoste. Uprkos početnim neuspesima da se stvore mašinski prevodioci za prirodne jezike, rad na tom polju je nastavljen.

Danas se metode obrade prirodnih jezika (eng. *natural language processing*, skr. ***NLP***) razvijaju pretežno u dva pravca: statističkom i lingvističkom.

Statistističke metode su podstaknute neprekidno rastućim brojem digitalnih dokumenata i potrebom da se oni obrade što brže i pouzdanije. Zasnivaju se na upotrebi nedeterminističkih metoda i metoda verovatnoće i statistike.

Lingvističke metode su, sa druge strane, više teorijske prirode. One pokušavaju da daju formalne modele koji prikazuju lingvističko znanje o konkretnim jezičkim sistemima. To ih čini manje efikasnim pri konkretnim, praktičnim primenama, ali zato mnogo pouzdanijim i detaljnijim. Dok je mogućnost ispravke greške u statističkim modelima minimalna zbog skrivenosti modela sa kojim se radi, modeli u lingvističkim metodama su eksplicitni i uvek podložni usavršavanju.

Za testiranje modela koriste se tekstualni korpusi. **Tekstualni korpusi** (eng. *corpus, text corpus* (pl. *corpora*)) su velike i struktuisane zbirke tekstova na kojima se vrši statistička analiza i testiranje hipoteza. Danas se najčešće nalaze u elektronskom formatu. Ti korpusi mogu da budu

sastavljeni od tekstova na jednom jeziku, jednojezični (eng. *monolingual corpus*), ili tekstova na više jezika, višejezični (eng. *multilingual corpus*).

Jedan od glavnih smerova istraživanja na području NLP-a je formiranje i upotreba višejezičnih paralelnih tekstualnih korpusa. Ovaj problem je jednako značajan u teorijskim područjima poput leksikografije i kontrastivne lingvistike, kao i u praktičnim primenama prevođenja i stvaranja mašinskih prevodilaca.

Osnovni pojmovi

Paralelni tekst (eng. *parallel text*) je tekst postavljen pored jednog ili više svojih prevoda. Ova ideja nije ništa novo. Poznata su višejezična izdanja "Biblije", sa po i do šest različitih verzija teksta. I ne zaboravimo najstariji primer, kamen iz Rozete.

Velike zbirke paralelnih tekstova nazivaju se **paralelni korpusi** (eng. *parallel corpus*).

Od 1988. god.-e uvodi se pojam biteksta. **Bitekst** (eng. *bitext*) je tekst sastavljen od dve verzije istog teksta. To su najčešće orginal i prevod, ali takođe mogu da budu i prevodi na različite jezike istog orginalnog teksta, ili, čak, različite verzije prevoda istog teksta na isti jezik.

Bitekst se formira iz dva koraka: prvo se na verzijama teksta odvojeno izvrši segmentacija na delove koji čine ekvivalentne jedinice, a zatim se tako dobijeni segmenti uparuju. **Paralelizovanje teksta** (eng. *parallel text alignment*) je proces uparivanja ekvivalentnih segmenata iz obe polovine paralelnog teksta. To se obično vrši softverom nazvanim *alignment tools*, a upareni segmenti su najčešće na nivou rečenica.

Skup bitekstova se naziva **baza bitekstova** (eng. *bitext database*) ili **dvojezički korpus** (eng. *bilingual corpus*) i može se koristiti pomoću pretraživačkih alata prilikom prevođenja. Po ovome, bitekst ima sličnosti sa **prevodilačkim memorijama** (eng. *translation memories*).

Prevodilačke memorije su baze podataka koje čuvaju delove teksta koji su ranije prevedeni. Delovi teksta mogu da budu reči, fraze, rečenice, paragrafi… Podaci se čuvaju u parovima orginal-prevod (eng. *source-target*) i pretražuju se po orginalu. Neke prevodilačke memorije traže unose koji se potpuno poklapaju, dok druge traže slične unose i nude više mogućih rezultata.

Bitekst se razlikuje od prevodilačkih memorija uglavnom po tome što zadržava raspored segmenata. Takođe, on je prvenstveno predviđen kao pomoć ljudskim prevodiocima, ne mašinama, pa manje greške koje se javljaju pri njegovom formiranju nisu od presudnog značaja.

Formiranje bitekst-a

Segmentacija

Prvi korak u formiranju biteksta je segmentacija tekstova na ekvivalentne delove. Segmentacija može da se vrši na više nivoa: paragrafi, rečenice, reči…

Iako bi se na prvi pogled moglo pretpostaviti da bi finija podela na reči dala bolje rezultate, najčešće se koristi segmentacija na rečenice. Dok na nivou rečenica u najvećem broju slučajeva imamo jedan-jedan korespondenciju između dva ekvivalentna teksta, isto ne može da se kaže na nivou reči. Takođe, na ovom nivou sintaksa i stil prevoda utiču na tekst u istoj meri koliko i sintaksa i stil orginala. Sve to uveliko otežava definisanje odgovarajućih pravila za uparivanje teksta segmentisanog na reči. Sa druge strane, pokazuje se da je segmentacija na rečenice i dalje dovoljno precizna da bi se dobio prihvatljiv paralelan tekst.

Sledeći problem je kako identifikovati rečenice i prepoznati njihov kraj. Kao vodeći princip, možemo da smatramo da je rečenica sintaksno autonoman skup reči. Prema ovome, osim celina koje intuitivno smatramo rečenicama, ovde bismo ubrojali i naslove, nabrajanja i slične celine.

Sada i našu intuitivnu predstavu rečenice treba prevesti u pravila koja će moći da koriste alati za segmentaciju teksta. Poznato je da većina rečenica završava nekim od znakova interpunkcije : **. ! ? ;**. 99% rečenica završava se tačkom. Ali, takođe, i samo deo tačaka u tekstu služi kao oznaka kraja rečenice. Evo nekoliko primera:

- internet i e-mail adrese: www.matf.bg.ac.rs, mr96035@alas.matf.bg.ac.rs;
- datumi, vreme, cene: 01.01.2010. god. , 13.15 ^{h}, 99.99£
- skraćenice: ex., npr., tj. …

Osnovni pravci rešavanja problema detekcije kraja rečenice su ručno generisanje pravila i mašinsko učenja. Kod ručnog generisanja pravila koriste se regularni izrazi, dok su neki od metoda za mašinsko učenje: neuronske mreže i drveta odlučivanja, skriveni Markovljevi modeli, maksimalna entropija...

Identifikovani segmenti se obeležavaju unapred predviđenim oznakama (npr. sa “<seg>” i “</seg>” tagovima). U nekim slučajevima, osim označavanja rečenica, potrebno je označiti i druge celine u tekstu: paragrafe, naslove, imena, datume i sl.

Jednom segmentisani ekvivalentni tekstovi se dalje poravnavaju.

Alignment

Poravnavanje teksta (eng. *text alignment*) može se vršiti na više nivoa, počev od poravnanja samog dokumenta, preko poravnanja paragrafa, rečenica, do poravnanja reči. Najčešće korišćene metode poravnanja su metode statističkog tipa, koje procenjuju verovatnoću nekog poravnanja. Ove metode su se pokazale kao veoma upotrebljive pri obradi velikih korpusa teksta.

Dva segmenta teksta možemo da smatramo ekvivalentnim ako izražavaju istu misao, makar u prihvatljivim granicama.

Prilikom prevođenja na razne jezike, moguće je da dođe do različitih problema. Ponekada nije moguće tekst prevesti jednoznačno. U toku rada mogu da se poremete logičke jedinice teksta (naslovi, podnaslovi, paragrafi…)

Najčešći problem je nemogućnost poravnanja teksta u odnosu jedan-jedan. Ovaj problem se javlja u više oblika:

- ubacivanje i izostavljanje (eng. *insertions* i *omissions*), kada segment jednog teksta nema odgovarajući segment u drugom tekstu. Ovo može da se reši uvođenjem praznog segmenta;
- jednom segmentu jednog teksta odgovara dva ili više segmenata drugog teksta;
- raspored ekvivalentnih segmenata je drugačiji u tekstovima (eng. *inversions*).

U ovakvim slučajevima, postojeće segmente grupišemo u veće celine, blokove, i poravnanje vršimo nad tim blokovima. Najčešći blokovi su oblika 1:1, 1:0 (0:1), 2:1 (1:2) i 2:2, mada su zabeleženi i blokovi oblika 8:1.

Pravci rešavanja problema poravnanja rečenica uglavnom se kreću u tri pravca:

- statističke metode (metode zasnovane na dužini rečenica);
- geometrijske metode (bitext prostor, dot-plot…);
- leksičke metode.

Statističke metode se zasnivaju na pretpostavci da će dužine rečenica u različitim varijantama teksta biti međusobno ekvivalentne, tj. da će u većini slučajeva dužim rečenicama orginala odgovarati duže rečenice prevoda, dok će kraćim rečenicama orginala odgovarati kraće rečenica prevoda. Dužinu rečenica je moguće meriti brojem reči u rečenici, ali je češći slučaj da se meri brojem karaktera u njoj (Church-Gale metod).

Church-Gale-ova metoda dodeljuje mogućim parovima ekvivalentnih segmenata indeks verovatnoće (eng. *probabilistic score*) zasnovan na njihovoj međusobnoj meri udaljenosti δ (eng. *distance measure δ*), i na osnovu tog indeksa, algoritmom najveće verovatnoće (eng. *maximum likelihood algorithm*) bira najboljeg kandidata. Ako pretpostavimo da dužine rečenica u tekstovima slede normalnu distribucije, vrednost δ za par rečenica l_1 i l_2 dobijamo formulom

$$\delta = (l_1 - l_2 c)/\sqrt{(l_1 s^2)},$$

gde je **c** prosečni odnos dužina rečenica prvog reksta u odnosu na dužine rečenica drugog teksta, a **s** odstupanje od normalne distribucije.

Ovaj metod daje poravnate blokove tipa 1:0, 0:1, 1:1, 2:1, 1:2 i 2:2. U tekstovima u kojima je poravnanje pretežno 1:1, dobijaju se odlični rezultati. Ali, u slučajevima kada se jezici veoma razlikuju (npr. engleski i korejski), ili su delovi teksta oštećeni, ili loše obeleženi, potrebno je primeniti druge, geometrijske metode.

Melmed-ova metoda posmatra tekstove orginala (S) i prevoda (T) kao skupove njihovih rečenica. Formira se Dekartov proizvod ovih skupova i identifikuju parovi rečenica sa približno jednakim dužinama. Kod srodnih jezika traže se rečenice koje imaju približno jednak broj karaktera, dok se kod veoma različitih jezika traže rečenici čije su dužine u približno istoj razmeri kao i dužine kompletnih tekstova. Parovi rečenica koji su u zadatoj relaciji označavaju se kao tačke u grafiku. U ovom trenutku, korespondencija nije obavezno jednoznačna. Zatim se definiše pojas oko glavne dijagonale u kome se dalje traže mogući ekvivalentni segmenti.

Varijacije ove metode ja da se umesto dužina rečenica koriste kognate. **Kognate** (eng. *cognates*) su reči koje u različitim jezicima imaju isto značenje i sličan zapis (npr. eng. *international* i srp. *internacionalni*). Sada se prvo u tekstu identifikuju kognate a zatim u grafiku parovi tih kognata predstave kao tačke.

Neki od algoritama za prepoznavanje kognata su odsecanje (traži poklapanje prvih n karaktera), DICE koeficijent, Levenštajnovo odstojanje…

Ipak, i upotreba kognata ima svoje probleme. Dok su kod bliskih jezika kognate relativno česte, kod raznorodnih jezika one se svode uglavnom na tuđice i imena. Razne promene reči u nekim jezicima mogu da sakriju sličnosti između nekih oblika kognata (npr. eng. bank i srp. banka bi bile prepoznate kao kognate, ali oblik srp. banci ne bi). Različiti alfabeti mogu da onemoguće prepoznavanje kognata, čak i na nivou ličnih imena (Vlada i Влада). Sa druge strane, kao kognate mogu da budu prepoznate i reči sličnog zapisa, ali potpuno različitih značenja (npr. rus. товариш i srp. товариш).

Formati teksta

Vanilla aligner

Za potrebe ovog primera ilustrovaćemo rad sa “Vanilla” alignerom, open source alatom koji služi za poravnavanje teksta. “Vanilla” aligner koristi Church-Gale-ov metod poravnanja. Dakle, polazi se od pretpostavke da su ekvivalentni blokovi približno jednake dužine (imaju približno jednak broj karaktera). Ovaj metod uveliko zavisi od pravilne segmenatcije teksta kako na rečenice, tako i na paragrafe. Dok su rečenice jedinice koje se poravnavaju, paragrafi služe kao kontrolne jedinice za sinhronizaciju teksta. Takođe, za upotrebu ovog alignera potrebna je dodatna priprema teksta. Sa druge strane, ovo je jedan od retkih open source alignera koji ne zavisi od specifičnosti jezika koji se obrađuju.

Proces počinje od dve odvojene datoteke koje sadrže ekvivalentne verzije istog teksta. Svaka od ovih datoteka se prvo parsira tako da se dobije datoteka koja sadrži u svakom redu ili po jednu reč, ili granični marker. Granični markeri su oznake za kraj rečenice i kraj paragrafa. Ove oznake mogu biti proizvoljne, i mogu da se navedu prilikom poziva aligner-a. Smatrajmo da je .EOS oznaka kraja rečenice (eng. *end of sentence*), a .EOP oznaka kraja paragrafa (eng. *end of paragraph*). Aligner će strogo da poštuje oznake za kraj paragrafa koje služe kao kontrolne tačke. Oznake za kraj rečenica aligner po potrebi može i da ignoriše.

Fajl koji može da posluži kao ulaz u aligner izgledao bi otprilike kao u tabeli na sledećoj strani.

Parsirani fajlovi služe kao ulaz u aligner:

align –D ‘.EOF’ –d ‘.EOS’ file1 file2

Algoritam sada pokušava da nađe i upari rečenice približno iste dužine. Iako je ovo uspešno u najvećem broju slučajeva, nailazi se i na situacije da je jedna rečenica (α) mnogo veća od njoj odgovarajuće rečenice (β). Tada se pokušava da se rečenici β pridruži manja rečenica (β_1) koja dopunjava dužinu rečenice β do dužine rečenice α. Ako ovo nije moguće, rečenicu uparujemo sa praznom rečenicom. Na ovaj način se poravnava kompletan tekst. Rezultat se smešta u fajl pod nazivom *file1*.al (file1 je ime prvog ulaznog fajla). Fajla se sastoji od blokova teksta sledećeg oblika:

Ovo	istog
je	fajla
primer	.EOS
ulaznog	.EOP
fajla	A
u	ovo
aligner	je
.EOS	drugi
Ovo	paragraf.
je	.EOS
druga	.EOP
rečenica	

Ulaz u Vanilla aligner

*** Lnk: 1-1 ***

The European Council began its proceedings by exchanging ideas with Mr Klaus Hnsch, President of the European Parliament, on the main subjects for discussion at this meeting. .EOS

Der Europische Rat hat zunchst einen Gedankenaustausch mit dem Prsidenten des Europischen Parlaments, Herrn Klaus Hnsch, ber die wichtigsten auf dieser Tagung zur Errterung anstehenden Themen gefhrt. .EOS

Prvi red u bloku daje informaciju o odnosu poravnatih segmenata. 1-1 znači da je jedna rečenica (segment) prvog teksta poravnata sa jednom rečenicom (segmentom) drugog teksta. To je najčešći slučaj. Ostale mogućnosti su: 1-0 (rečenica iz prvog teksta nema ekvivalentnu rečenicu u drugom tekstu, tj. pridružena joj je prazna rečenica), 0-1 (rečenica iz drugog teksta nema ekvivalentnu rečenicu u prvom tekstu), 2-1 (dve rečenice iz prvog teksta su poravnate sa jednom rečenicom iz drugog teksta), 1-2 (jedna rečenica iz prvog teksta je poravnata sa dve rečenice iz drugog teksta) i 2-2 (dve rečenice iz prvog teksta su poravnate sa dve rečenice iz

drugog teksta, tj. u jednom tekstu se nalazi velika rečenica za kojom sledi mala, dok se u drugom tekstu nalazi mala rečenica za kojom sledi velika). Mogućnost poravnanja 3-1 je isključena jer aligner ne bi bio u stanju da razlikuje taj slučaj od npr. slučaja poravnanja 2-1, za kojim sledi 1-0.

Sledeća dva reda su poravnati segmenti, po jedan red za segment iz svakog fajla.

Po dobijanju izlaznog fajla, potrebno je izvršiti kontrolu. Pri radu sa dovoljno bliskim jezicima, ovaj aligner dostiže i 95% tačnosti. Greške se javljaju uglavnom u slučajevima kada je potrebno poravnanje tipa 3-1, 1-3 i sl.

Ipak pri radu sa značajno različitim jezicima, bez sličnosti u rečenicama, preporučljivo je koristiti drugačije metode.

Primer drugačijeg formata izlaza iz alignera bi bio:

(EN-d2p4seg1) "Not a bad guess," said I.
(FR-d2p3seg1) - Ta conjecture n'est pas fausse, dis-je.

(EN-d2p5seg1) "But you see how many we are?" he said.
(FR-d2p4seg1) - Et vois-tu combien nous sommes ? dit-il.

Ovde je kodom u zagradama data kako informacija o jeziku iz koga je rečenica iz para, tako i o njenom polozaju u polaznom tekstu.

XAlign

Kao sledeći primer alata za poravnanje teksta, možemo da navedemo XAlign. XAlign kao ulaz uzima tekst u kome su već obeležene kako rečenice tako i druge logičke celine.

Reprezentacija poravnatog teksta u XAlign-u ima oblik:

```
<link targets="n5 n6" type="linking" id="l1" />
<link targets="n1 x1" />
<link targets="n2 x2" />
.....
<link targets="l1 x5" />
<link targets="n7 x6" />
```

Ako tekstove oznacimo sa *n* i *x*, ova reprezentacija označava da je prva rečenica teksta *n*, n1, poravnata sa prvom rečenicom teksta *x*, x1, druga rečenica teksta *n*, n2, sa x2, drugom rečenicom teksta *x*, itd.

Sa druge strane, peta i šesta rečenica teksta *n* čine blok koji se označava sa l1. Taj blok je poravnat sa petom rečenicom teksta *x*.

Ovakva reprezentacija se zatim lako može prebaciti u neki drugi željeni format, npr. TMX ili HTML.

TEI

Opišimo ovde jedan od formata teksta koji može da posluži kao ulaz u aligner-e.

Veoma popularan standard za zapis elektronskog teksta je TEI (eng. *Text Encoding Initiative*). Dok je na početku bio zasnovan na SGML-u, u kasnijim verzijama je usvojio XML kao način kodiranja. Kompletna TEI šema obeležavanja ime nekoliko stotina različitih elemenata. Zato je radi lakše upotrebe definisano nekoliko njegovih podskupova, među kojima su poznatiji TEI-Lite i TEI-Barebone.

Svaki Tei-dokument počinje i završava tagovima **<tei.2>** i **</tei.2>**.

Zatim sledi element **<teiHeader>**, koji može da ima do četiri dela **<fileDesc>**, **<encodingDesc>**, **<profileDesc>** i **<revisionDesc>**.

<fileDesc> je bibliografski opis datog teksta, uključujući i naslov. Ovaj element je obavezan.

<encodingDesc> opisuje način i alate kodiranja teksta.

<profileDesc> je nebibliografski opis teksta.

<revisionDesc> daje podatke o promenama u tekstu.

Posle <teiHeader> elementa obavezno sledi **<text>** element koji sadrži kompletan tekst koji se kodira. Tekst je zatvoren u elementu **<body>**. Ako se tekst sastoji od više odvojenih celina, kao npr. antologija, svaka od tih celina može da se smesti u poseban <body> element.

Tekst se dalje deli na podceline elementom ** **. sadrži atribut "**type**" koji određuje vrstu podele. element može neograničeno da se ugnježdava.

Paragrafi se označavaju sa **<p>**, a segmenti sa **<seg>**.

Prikažimo primer teksta u TEI formatu:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<tei.2>
        <teiheader>
                <filedesc>
                        <titlestmt>
                                <title>Europarl-es</title>
                        </titlestmt>
                </filedesc>
        </teiheader>
        <text>
                <body>
                        <div>
                        <!-- ep-99-10-04 -->
                        <p><seg>Reprise de la session </seg></p>
                        <p><seg>Je déclare reprise la session du Parlement européen, qui avait
                        été interrompue le vendredi 17 septembre . </seg></p>
                        <p><seg>Déclaration de la présidente </seg></p>
                        </div>
                        …
                </body>
        </text>
</tei.2>
```

Kao što je već napomenuto, TEI specifikacija opisuje više od četri stotine definisanih tagova, sa brojnim atributima. Samo manji broj njih je obavezan. Po potrebi korisnika, takođe je predviđena mogućnost uvođenja novih tagovi. Neki od specijalizovanih osnovnih skupova tagova su za rečnike, poeziju, dramu…

TMX

Za razliku od predhodnog primera, TMX je jedan od popularnih formata za izlaz iz aligner-a.

TMX (eng. *Translation Memory eXchange*) je standard za razmenu prevodilačke memorije u okviru XML standarda. Razvijen je oko 1998 god. od strane OSCAR-a (eng. *Open Standards for Container/Content Allowing Re-use*), podgrupe LISA-e (eng. *Localization Industry Standard Associacion*). Omogućava lakšu razmenu prevodilačke memorije između prevodilaca i drugih alata sa malo ili nimalo gubitaka.

Sam TMX dokument je XML fajl predviđen za automatsko generisanje i obradu od strane alata za obradu teksta.

Definicija TMX–a ima dva dela: format kontejnera i format sadržaja. Postoje dva nivoa implementacije:

- prvi nivo (eng. *Plain Text Only*): podržan je samo format kontejnera. Unutar oblasti označenih <seg> tagovima ne sme da bude nikakvih drugih oznaka.
- drugi nivo (eng. *Content Markup*): dozvoljeno je formatiranje kako kontejnera, tako i sadržaja teksta.

Svaki TMX dokument počinje tagom **<tmx>** i završava tagom **</tmx>**. Unutar ovih tagova uvek se nalaze dva elementa: <header> i <body>.

Header počinje i završava tagovima **<header>** i **</header>** (moguća i skraćenica **<header…/>**). On kao svoje obavezne atribute sadrži meta-podatke o dokumentu (naziv i verzija alata koji je kreirao fajl, izvorni jezik i sl.). Uz obavezne i opcione atribute, header može da sadrži i elemenate **<note>**, **<prop>** i **<ude>**. Ovde se čuvaju informacije koje važe unutar celog dokumenta.

Telo dokumenta se nalazi između tagova **<body>** i **</body>**. Sastoji se od niza **<tu>** elemenata (eng. *translation units*).

Ako želimo da više <tu> elemenata grupišemo u jednu logičku celinu, koristimo **<prop>** element za svaki takav <tu> element. <prop> opisuje razna svojstva roditeljskog elementa (ili celog dokumenta ako se nalazi u header-u). Njegov obavezan atribut je "**type**". <prop> i <note> takođe mogu da sadrže i bilo koje podatke specifične za taj <tu> element.

Svaki <tu> element sadrži makar jedan **<tuv>** element (eng. *translation unit variant*). Obavezan atribut za <tuv> element je "**xml:lang**", jezik sadržaja tog elementa. Podaci specifični za svaki <tuv> element smeštaju se unutar njega u <prop> i <note> elemente.

Unutar svakog <tuv> elementa dalje se nalazi **<seg>** element. <seg> element sadrži sam tekst dokumenta.

Primer TMX dokumenta bi izgledao ovako:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!--TMX comment to change...-->
<tmx version="1.3">
        <header creationtool="ACIDE" creationtoolversion="1.0" segtype="sentence"
        datatype="plaintext" adminlang="EN" srclang="ES" />
        <body>
                <tu>
                        <prop type="Domain">ef-08</prop>
                        <tuv xml:lang="ES" creationid="n1 " creationdate="20090428T101223Z">
                                <seg>Reanudación del período de sesiones  </seg>
                        </tuv>
                        <tuv xml:lang="FR" creationid="n1 " creationdate="20090428T101223Z">
```

```
                    <seg>Reprise de la session  </seg>
                </tuv>
            </tu>
            <tu>
                <prop type="Domain">ef-08</prop>
                <tuv xml:lang="ES" creationid="n2 " creationdate="20090428T101223Z">
                    <seg>Declaro reanudado el período de sesiones del Parlamento
                    Europeo , que fue interrumpido el viernes , 17 de septiembre .  </seg>
                </tuv>
                <tuv xml:lang="FR" creationid="n2 " creationdate="20090428T101223Z">
                    <seg>Je déclare reprise la session du Parlement européen, qui avait été
                    interrompue le vendredi 17 septembre .  </seg>
                </tuv>
            </tu>
            …
        </body>
</tmx>
```

U slučaju kada je izvorni tekst već označen na neki način, npr. HTML tagovima, potrebno je te tagove izuzeti iz obrade. Zato ih na Content mark-up nivou posebno označavamo.

<bpt> (eng. *begin paired tag*) i **<ept>** (eng. *end paired tag*) označavaju redom početak i kraj uparenih kodnih elemenata u tekstu. Oni imaju obavezan atribut "**i**" koji povezuje odgovarajuće <bpt> i <ept> elemente.

<hi> (eng. *highlight*) označava deo koda koji ne treba menjati, kao, na primer, datum ili lično ime.

<it> (eng. *isolated tag*) označava početak ili kraj koda koji počinje i završava u različitim segmentima. Ima obavezan atribut "**pos**" čije vrednosti mogu da budu "**begin**" i "**end**".

<ut> (eng. *unknown tag*) označava kod čije značenje nije poznato.

U sledećem primeru, prvi red je čist HTML tekst, dok je drugi red taj isti tekst sa TMX oznakama:

Deo teksta je u **** italiku ****, a deo u **** boldu ****.

Deo teksta je u **<bpt i="1">**<i>**</bpt>** italiku **<ept i="1">**</i>**</ept>**, a deo u **<bpt i="2">****</bpt>** boldu **<ept i="2">****</ept>.**

Od teksta do rečnika

Primena predhodne teorije može da se ilustruje primerom formiranja francusko-srpskog jezičkog korpusa.

Po izboru tekstova koji će biti uključeni u korpus, tekstovi su označeni minimalnim skupom tagova <head>, <p> i <seg>. <head> tagovi su ubačeni ručno, pri čemu je dodat bibliografski opis teksta u skladu sa TEI standardom. Poglavlja su označena automatski ili polu-automatski.

Prepoznavanje kraja rečenice vršeno je programom Sentence.grf koji je deo Intex sistema. Kako se težilo što strožijem poravnanju 1-1, po potrebi su neki segmenti (rečenice) ručno podeljeni na manje segmente (delove rečenice).

Poravnanje je obavljeno alignerom XAlign, dok su potrebne prepravke napravljene programom Concordancier.

Dobijeni tekstovi su zatim pomoću WS4LR-a prebačeni u TMX i HTML format.

WS4LR

WS4LR (eng. *Workstation for Lexical Resources*) nastao je na Matematičkom fakultetu Univerziteta u Beogradu iz potrebe da se u malom vremenu obradi velika količina raznovrsnih leksičkih resursa. Sistem može da se koristi za različite jezike, sve dok se poštuju zadati formati i metodologije.

WS4LR radi na .NET platformi pod Windows-om 2000/XP/2003. Sastavljen je od modula koje korisnik po potrebi sam bira. Omogućeno je i pozivanje potrebnih funkcija iz komandne linije i korišćenje spoljnih skriptova na AWK-u, Perl-u i sl.

Kao prvo, ovaj sistem omogućava prebacivanje skupa tekstova iz jednog seta karaktera u drugi. Moguće je prebaciti i samo izbrani deo teksta unutar fajla, dok ostatak teksta ostaje u starom kodnom setu. Ovo je naročito korisno u našem jeziku gde postoje dva pisma, ćirilica i latinica, i gde treba prebaciti označen tekst (npr. HTML) u jedan od ta dva pisma.

Takođe je omogućen rad sa rečnicima u formatima DELAS i DELAC. DELAF fajlovi mogu da se generišu iz DELAS-a upotrebom transduktora. DELAS je format rečnika koji sadrži osnovne oblike (pojedinačne) reči (leme) u zapisu: lema.*oznaka[+opis]*, gde oznaka sadrži

informaciju o načinu promene leme, a opis daje dodatne osobine te reči. DELAF je rečnik koji sadrži različite oblike (forme) leme, i sastoji se od niza unosa oblika: forma, lema[:kategorija], gde je kategorija gramatička kategorija datog oblika reči. DELAC je rečnik izraza sastavljenih od više lema, a DELACF od više formi. Njihovi formati su slični predhodnim slučajevima, osim što mogu da sadrže i neke dodatne znakove (blanko, apostrof…). Takođe, način promene lema ili formi u njima je značajno komplikovaniji. Osim načina promene svake leme koja se nalazi u izrazu, mora da se obrati pažnja i kako te leme i forme međusobno utiču jedna na drugu. O obeležavanju ovakvih izraza biće reči u kasnijem tekstu. WS4RL omogućava pretraživanje, unos novih podataka, brisanje i editovanje starih za formate DELAS i DELAC. Sami rečnici se uključuju u rad po potrebi, što omogućava lakšu manipulaciju.

Pored morfoloških rečnika, moguć je rad i sa rečnicima sinonima (*Wordnet*). Osim pretraživanja i editovanja pojedinačnih wordnet-a, moguće je njihovo međusobno sinhronizovanje, kao i razmena informacija sa morfološkim rečnicima.

Puna snaga WS4LR-a dolazi do izražaja tek pri radu sa paralelnim tekstom. Izlazni format teksta iz XAlign-a se postojećim modulima prebacuje u TMX format. Tekst u tom formatu dalje može da se prebaci u neki od formata koji su čitljiviji ljudima, kao što je HTML. U saradnji sa drugim modulima unutar WS4LR, morfološkim i rečnicima sinonima, dalje je moguće naći i obeležiti potrebne reči u tekstu, kao i njihove prevode.

Ovo nam dalje omogućava da formiramo dvojezičke liste koje sadrže u svakom redu reč i njen prevod.

BiTMark

Glavna funkcija BiTMark-a je označavanje kompozita u paralelnom tekstu. **Kompozite** su fraze sastavljene od više reči koje zajedno imaju posebno značenje (npr. “radni dan”).

BiTMark omogućava označavanje kompozita u bitekstu.

Ulaz u BiTMark je HTML fajl koji se dobija konverzijom biteksta pomoću WS4LR-a. Tekst je smešten u tabelu koja sadrži dve kolone, po jednu za svaki jezik. Svaki poravnati segment teksta je smešten u poseban red tabele. Ćelije tabele sadrže redni broj segmenta i tekst tog segmenta označen <seg> tagovima.

Prilikom učitavanja teksta u BiTMark editor, vrši se prepoznavanje i označavanje svih ranije obrađenih kompozita.

Dalje označavanje kompozita je moguće samostalno, ili putem prenošenja oznaka iz već označenog teksta na drugom jeziku. U slučaju da se radi paralelno na oba jezika i kompozite i njihove oznake se smeštaju u rečnik koji je kasnije moguće snimiti. Prilikom prenošenja oznaka,

same oznake mogu da se kopiraju ili prilagode delovima kompozite na ciljnom jeziku. Skupovi oznaka mogu da se po potrebi dopunjuju, ili da se definišu sasvim novi skupovi koji više odgovaraju jezičkom paru.

Pri izlasku iz programa, nudi se snimanje označenog teksta, snimanje rečnika generisanog tokom rada, i razdvajanje novodobijenog teksta na odvojene jezičke fajlove.

Tekst na kome se radilo snima se u istom formatu u kome je bio i učitan. Generisani rečnik se sastoji od redova koji sadrže redom sledeće informacije:

- oznaka kompozite na prvom jeziku
- kompozita na prvom jeziku
- oznaka kompozite na drugom jeziku
- kompozita na drugom jeziku

Opciono, mogu biti dodata i poređenja uparenih kompozita izračunata pomoću algoritama:

- *Levenshtein distance*,
- *Longest common subsequences*,
- *Cost depending on the length of the gaps* i
- *Local similarity*.

Po želji, sadržaj biteksta može biti razdvojen po jezicima u dva fajla, čiji su redovi segmenti zatvoreni <seg> tagovima. Ti segmenti sadrže u sebi oznake kompozita.

Označavanje kompozita

Oznake kompozita imaju oblik

{*vrsta_kompozite atributi_kompozite opciona_dopuna***}** … **{/***vrsta_kompozite***}**

Kompozite delimo na tri vrste:

- opšti nazivi (npr. apsolutna većina, radni dan…). Ovakve grupe reči označavamo sa **N**;
- imena (npr. Srpska Akademija Nauka i Umetnosti, Zemunski kej). Označavaju se sa **NP**;
- imenovani entiteti (npr. 20°C). Označavaju se sa **NE**.

Kompozite se opisuju sa nekoliko atributa.

- atribut **cat** opisuje vrste i raspored reči od kojih je kompozita sastavljena. Tako vrednost AN označava imenicu ispred koje stoji pridev, dok su NDN dve imenice između kojih je određeni član. Po default-u, ponuđene vrednosti su: AN, NA, NN, VV, XV, VN, NV1N, PN, XN, NDN, ND1N, ND1NN, ND2N, NDAN, ND1AN, NAD1N, NDNCDN, NPN, NP1N, NPNCN, AAN, ACAN, NAA, NACA. Ove atribute je moguće prilagoditi potrebama jezika sa kojima se radi. Cat se pridružuje N i NP vrstama kompozita.
- istim vrstama kompozita pridružuje se i **fs**. Ovaj atribut označava rod i broj imenice u kompoziti. Ponuđene opcije na početku rada su: fs (ženski rod jednine), ms (muški rod jednine), fp (ženski rod množine) i mp (muški rod množine).
- poslednji atribut je **type**. On prenosi semantičke informacije o kompoziti. Može da se pridruži vrstama NP i NE. Za imena, njegova default vrednost je iz skupa: Toponym, Hydronym, Oronym, Event, Legislation, Mesure i Organisation. Za imenovane entitete, te vrednosti su TIMEX i NUMEX.

U nekim slučajevima, imenica koja je deo kompozite data je samo implicitno. Takav je, na primer, slučaj sa kompozitom "*drumski saobraćaj*" u izrazu "*drumski i železnički saobraćaj*". U slučajevima ovakvih skraćenica, kompozitu možemo da dopunimo tom imenicom unutar samog taga. Takve dopune su vrednosti atributa **N**, koji se može naći uz imena i opšte nazive.

Navedimo nekoliko primera iz jednog od tekstova na španskom i francuskom:

{N cat="NA" fs="ms"} procès-verbal {/N}

{NP cat="NA" fs="ms" type="Organisation"} Parlement européen {/NP};

{NE type="TIMEX"} 17 septembre {/NE}

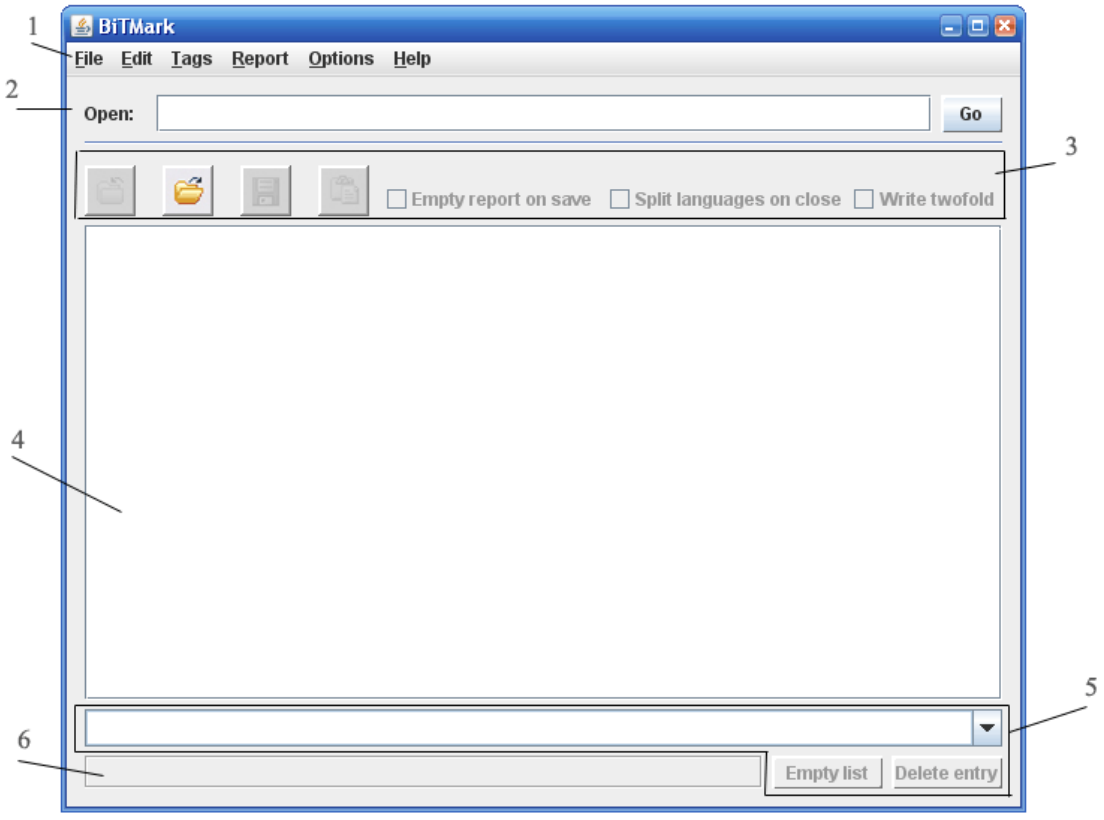
{N cat="NDN" fs="mp"} problèmes d'actualité {/N} , {N cat="NA" fs="mp" N='problèmes'} urgents {/N} et {N cat="NDN" fs="mp" N='problèmes'} d'importance majeure {/N}

BiTMark
(Binary Text Marker)

BiTMark je program za rad sa paralelnim tekstom (bitekstom) u HTML formatu. BiTMark ispunjava dve funkcije: obeležavanje fraza u paralelnom tekstu i generisanje “rečnika” od odgovarajućih fraza.

Radna površina

Osnovna radna površina se sastoji od nekoliko celina (Slika 1.):



Slika 1.

1. Menu bar
2. Address bar

3. Command bar
4. Editor
5. Report bar i
6. Status bar.

1. Menu Bar

Pored standardnih menija File, Edit i Help, Menu Bar sadrži i menije: Tag (za rad sa tagovima), Report (za rad sa “rečnikom”) i Options.

2. Address bar

Address bar sadrži polje u koje može da se unese naziv HTML fajla sa kojim želimo da radimo, i dugme Go, koje taj fajl otvara.

3. Command bar

Command bar sadrži dugmad Close (za zatvaranje HTML fajla), Open (za biranje i otvaranje HTML fajla), Save (za snimanje otvorenog HTML fajla pod istim imenom) i Report (za generisanje rečnika). Tu se takođe nalaze i tri chackbox-a: Empty_report_on_save (posle snimanja rečnika briše snimljene fraze iz memorije programa), Split_language_on_Close (pri izlasku iz programa deli označeni tekst na sastavne jezike i snima ih u source.txt i destination.txt fajlove) i Write_twofold (dozvoljava da se u rečnik unose ponovljene fraze).

4. Editor

Glavna radna površina u kojoj se otvara HTML fajl koji se obrađuje.

5. Report bar

Služi za pregled i kontrolisanje unosa u rečnik. Sadrži drop-down listu u kojoj se vide svi unosi koji su napravljeni u tekućoj sesiji (i ranijim ako je učitan raniji rečnik). Pojedinačne unose možemo ukloniti dugmetom Delete_entry, a celu listu praznimo dugmetom Empty_list.

6. Status bar

Kada se miš nalazi iznad taga, Status bar pokazuje podatke o tom tagu. Ako je u toku povezivanje fraza (bind stanje), prikazuje frazu koja je već unešena.

Osnovne operacije nad HTML fajlom

Otvaranje fajla

Da bi se počelo sa radom, prvo otvaramo HTML fajl koji sadrži paralelni tekst. To možemo da učinimo na više načina:

- u Address bar upišemo putanju i ime fajla, i pritisnemo dugme Go;
- klikom na Open dugme u Command bar-u otvara se dijalog u kome biramo potrebni fajl;
- u File meniju biramo Open …;
- pritiskom na tastere CTRL+O.

Otvaranjem fajla, sve ostale komande postaju dostupne.

Snimanje fajla

Fajl snimamo na sledeće načine:

- pritiskom na dugme Save u Command bar-u snimamo fajl pod istim imenom pod kojim je i otvoren. Bićemo upozoreni da fajl sa tim imenom već postoji, i biće nam ponuđeno da ga presnimimo, ili da otvoreni fajl snimimo pod drugim imenom;
- isto se događa i ako u File meniju izaberemo opciju Save;
- takođe, za isti efekat možemo da pritisnemo tastere ALT+S;
- ako u File meniju izaberemo opciju Save_as…, odmah nam se otvara dijalog za snimanje fajla;
- isto postižemo pritiskom na tastere CTRL+S.

Zatvaranje fajla

Fajl se zatvara :

- pritiskom na Close dugme u Command bar-u ili
- biranjem Close opcije u File meniju.

Prilikom zatvaranja, ako u HTML fajlu postoje promene koje nisu ranije snimljene, biće nam ponuđeno da snimimo fajl. Takođe, ako rečnik nije prazan, biće nam ponuđeno da snimimo i njega.

Izlazak iz programa

Iz programa se izlazi biranjem Exit opcije u File meniju.

Ova opcija izaziva gubitak svih podataka koji predhodno nisu snimljeni.

Kopiranje teksta

Kada se fajl otvori, možemo da manipulišemo sa tekstom u njemu. Selektovani tekst možemo da iskopiramo iz programa pritiskom na tastere CTRL+C ili izborom Copy opcije u Edit meniju.

Osnovno editovanje teksta

Selektovani tekst u fajlu možemo da editujemo:

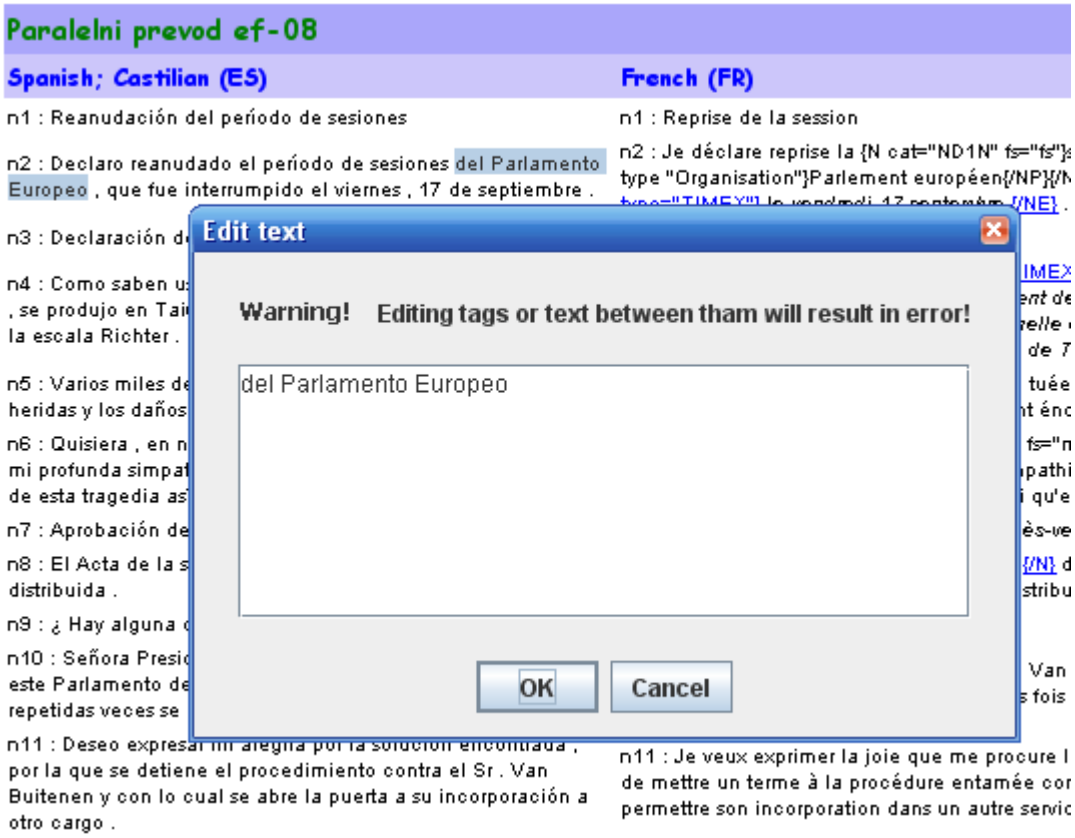
- pritiskom na tastere CRTL+E;
- izborom Edit_text … opcije u Edit meniju ili
- izborom Edit_text … opcije u Popup meniju u Editoru

otvara se dijalog u kome je moguće menjati selektovani tekst (Slika 2).


Da bi se tekst editovao, on predhodno **mora** da bude selektovan.

Menjanje teksta koji se nalazi između tagova, ili koji sadrži tagove, dovešće do greške.

Slika 2.



Označavanje teksta

Učitavanje tagova u program

Pre otvaranja fajla sa paralelnim tekstom, formira se lista tagova.

Ako postoji fajl koji je u konfiguracionom fajlu definisan da sadrži tagove, oni se učitavaju iz njega.

Moguće je i pre učitavanja HTML fajla izabrati tag-fajl pritiskom na tastere SHIFT+L, ili izborom opcije Load_tag_file… iz Tag menija.

Ako tag-fajl nije određen, formiraće se lista defaulta tagova.

Prilikom učitavanja HTML fajla, tagovi u tekstu se prepoznaju i označavaju kao linkovi, a ako se naiđe na tag koji se ne nalazi u već formiranoj listi tagova, on se dodaje u listu.

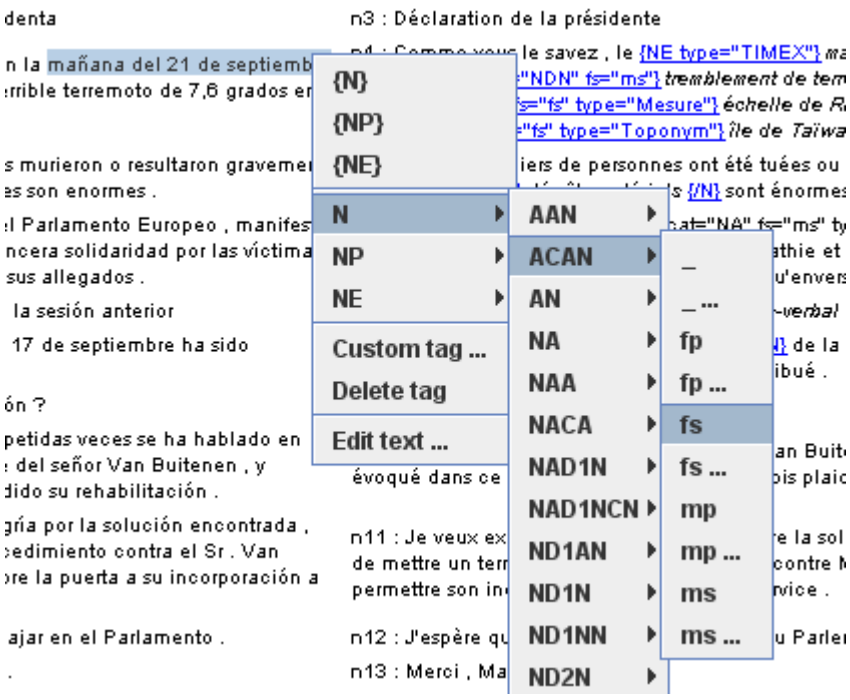
Posle učitavanja HTML fajla, moguće je kompletnu listu tagova zameniti tagovima učitanim iz izabranog fajla. Tag-fajl se bira na predhodno opisani način.

Dodavanje tagova u tekst

Tagovi se u tekst dodaju na sledeći način:

- selektuje se željeni tekst
- desnim klikom u Editoru dobijemo popup meni
- u meniju izaberemo potrebni tag.

Slika 3.



Slika 3. pokazuje osnovni popup meni.

Ponuđene su dve grupe opcija: osnovni tagovi ({N}, {NP} i {NE}) i prošireni tagovi.

Ako u proširenim tagovima želimo da izostavimo neki atribut, na mestu tog atributa biramo opciju "_".

Ako označavamo frazu u kojoj je izostavljena imenica, biramo odgovarajući tag koji se završava sa "…". Tada nam se otvara dijalog za kreiranje tagova, koji omogućava da se u odgovarajuće polje upiše potrebna imenica.

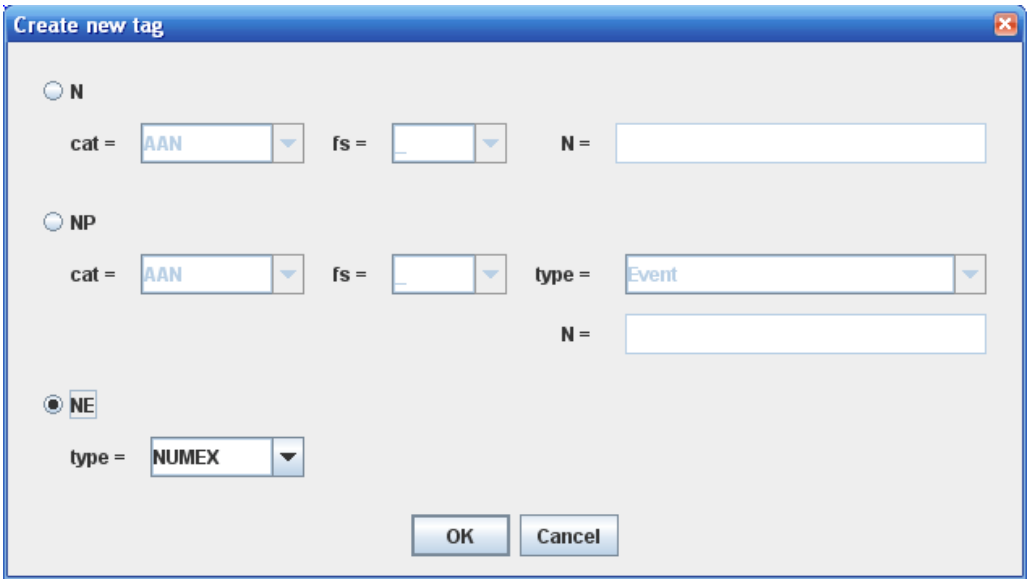
Ako smo u toku povezivanja dve fraze u rečnik, na vrhu popup menija biće takođe ponuđen tag predhodno izabrane fraze.

 Dozvoljeno je ugnježdavanje tagova spolja ka unutra.

 Selektovani tekst ne sme da obuhvata već postojeći tag ili njegov deo.

Kreiranje novih tagova

Ako se među ponuđenim tagovima ne nalazi tag koji nam je potreban, možemo da ga kreiramo i direktno ubacimo u tekst. To činimo izborom opcije Custom_tag… u popup meniju. Time nam se otvara dijalog za kreiranje tagova (Slika 4). U ponuđenim poljima možemo da izaberemo potrebne vrednosti, ili, ako ih već nema, upišemo nove.



Slika 4.

Novokreirani tag se ovim ubacuje u listu ponuđenih tagova.

Ako izaberemo opciju Create_tag… iz Tag menija u Menu bar-u, ili tastere CTRL+N, postižemo sličan efekat. Razlika je u tome što se u ovom slučaju novokreirani tag neće ubaciti direktno u tekst.

Brisanje tagova iz teksta

Ako želimo da izbrišemo ranije ubačeni tag iz teksta, to činimo na sledeći način: selektujemo kompletan željeni tag (početni tag, tekst na koji se odnosi i završni tag) i izaberemo:

- Delete_tag iz popup menija;
- Delete_tag iz Tag menija ili
- tastere CTRL+D.

Brisanje neće uspeti ako tekst nije pravilno selektovan.

Kod ugnježdenih tagova nije moguće izbrisati spoljašnji tag pre brisanja unutrašnjeg.

Kreiranje tag-fajla

Moguće je napraviti skup potrebnih tagova i snimiti ih u fajl sa ekstenzijom *.tag kako bi se kasnije po potrebi koristili. To se radi u dijalogu Create-tag_file (Slika 5), koji se pokreće tasterima SHIFT+C ili izborom Create_tag_file… opcije u Tag meniju.

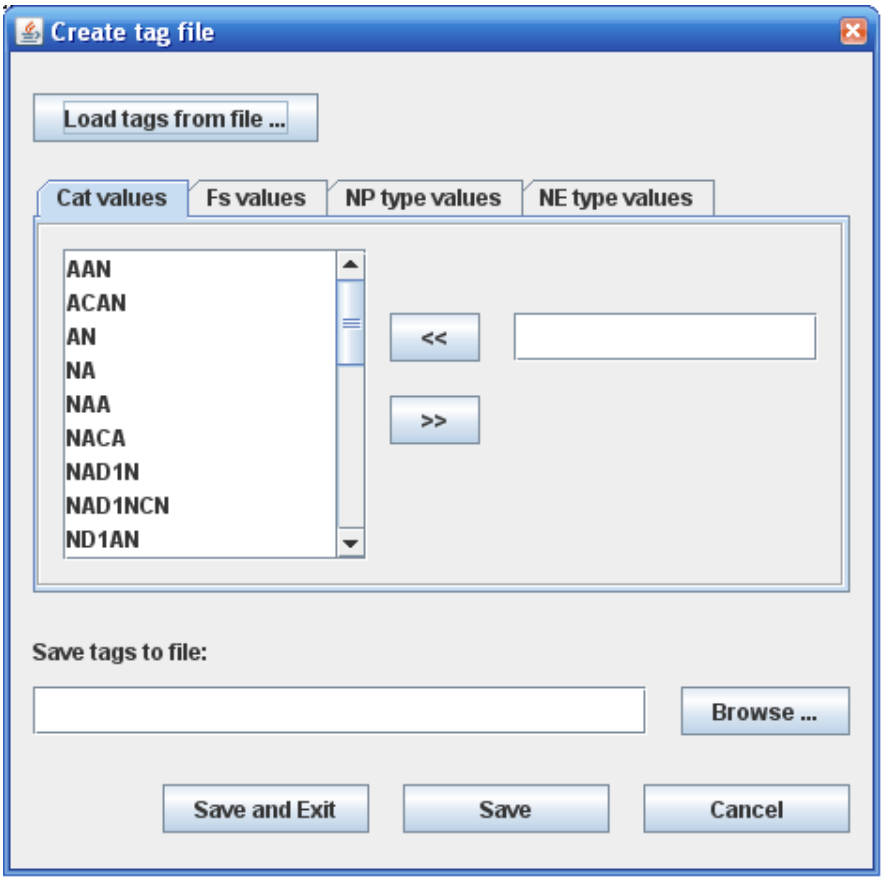
Kao baza, mogu se koristiti tagovi učitani iz tekuće liste tagova u glavnom delu programa, ili se mogu učitati tagovi iz ranije napravljenog tag-fajla pritiskom na dugme Load_tags_from_file…

Klikom na odgovarajući tab biramo atribut koji hoćemo da promenimo.

Postojeće vrednosti atributa brišemo tako što ih selektujemo (može i više njih odjednom), a zatim pritisnemo dugme >>.

Nove atribute upisujemo u tekst-polje pored liste, a zatim ih ubacujemo pritiskom na dugme <<.

U Save_tags_to_file polje upisujemo put i ime fajla koji hoćemo da snimimo, ili možemo da otvorimo Save dijalog pritiskom na dugme Browse…



Slika 5.

Pritiskom na dugme Save snimamo izabrani fajl, ali ostajemo u dijalogu gde možemo da pravimo dalje izmene.

Pritiskom na dugme Save_and_Exit snimamo fajl i izlazimo iz dijaloga.

Pritiskom na dugme Cancel izlazimo iz dijloga bez snimanja. Predhodno napravljene promene biće izgubljene.

Rečnik (report)

Popunjavanje rečnika

Rečnik se popunjava povezivanjem po dve fraze.

Klikom na tag prve faze započinjemo proces povezivanja (Bind state). U Status baru se pojavljuje obaveštenje da je proces počeo (vidi se “Bind:” i tekst na koji se izabrani tag odnosi).

 Ako se izabrani tekst odranije nalazi u rečniku, proces povezivanja neće početi. Izuzetak je ako je selektovan chackbox Write_twofold u Command baru. Tada se ne vrši nikakva provera.

Povezivanje se završava na neki od sledećih načina:

- klikom na već postojeći tag druge fraze;
- dodavanjem novog taga u tekst, na način koji je već opisan;
- klikom na Cancel opciju u Popup meniju, čime se povezivanje prekida. Cancel opcija se javlja u Popup meniju isključivo ako je u toku proces povezivanja.

Po završetku procesa povezivanja, u drop-down listi u Report bar-u javlja se novi red koji sadrzi tag i tekst prve fraze, za kojima slede tag i tekst druge fraze.

U slučaju da želimo da izbrišemo iz rečnika postojeći unos, u drop-down listi selektujemo željeni red i pritisnemo dugme Delete_Entry.

Ako želimo da izbrišemo sve dosadašnje unose u rečnik, to činimo:

- pritiskom na dugme Empty_list u Report bar-u;
- opcijom Empty_report u Report meniju u Menu bar-u ili
- tasterima CTRL+M.

Učitavanje rečnika

Ako želimo da nastavimo rad sa rečnikom koji smo ranije snimili, možemo da ga učitamo na sledeći način:

- biranjem u konfiguraciji fajla iz koga će se po default-u učitavati rečnik;
- opcijom Load_report u Report meniju biramo željeni fajl; ili
- tasterima CTRL+L.

Sadržaj učitanog rečnika će se pojaviti u drop-down listi u Report bar-u.

Snimanje rečnika

Snimanje rečnika vršimo:

- pritiskom na Report dugme u Command bar-u;
- biranjem Generate_Report… opcije u Report meniju ili
- tasterima CTRL+G.

Ovim se otvara Save dijalog u koji unosimo naziv fajla u koji ćemo da snimimo rečnik. Ako izaberemo naziv već postojećeg fajla, biće nam ponuđen izbor da dopunimo taj fajl, da ga presnimimo ili da se vratimo u Save dijalog.

Ako je selektovan chackbox Empty_report_on_save u Commana bar-u, posle snimanja rečnika u fajl, rečnik će se isprazniti. U suprotnom, dosadašnji unosi u rečnik su još prisutni.

Prilikom zatvaranja HTML fajla (Close), ako rečnik nije prazan, biće ponuđeno da se snimi.

Ova opcija nije ponuđena prilikom izlaska iz programa (Exit).

Poređenje fraza

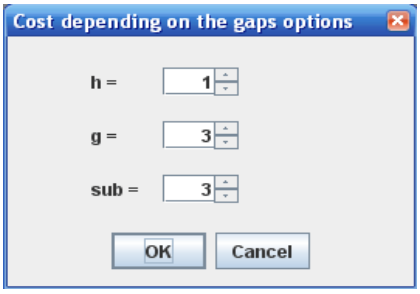
BiTMark omogućava ispitivanje sličnosti između dve fraze. Omogućen je rad sa četiti metode poređenja:

- Levenshtein distance;
- Longest common subsequance;
- Cost depending on the gap i
- Local similarity.

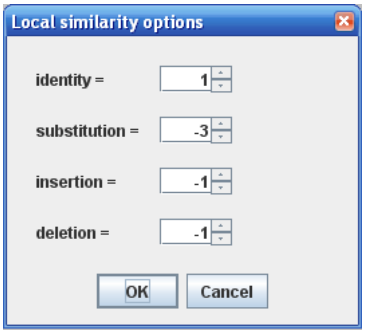
Dodatno, za poslednje dve metode omogućeno je i menjanje parametara izračunavanja.

Poređenje fraza u rečniku

Prilikom snimanja rečnika, moguće je izvršiti poređenje povezanih fraza po zadatim kriterijumima. Metode poređenja biraju se chackbox-ovima u Report meniju. Ako su izabrane Cost_depending_on_the_gap ili Local_similarity metode, omogućava se prilagođavanje njihovih parametara u odgovarajućim dijalozima (Slike 6.a i 6.b).



Slika 6.a



Slika 6.b

Dijalozi se otvaraju biranjem opcija Cost_options i Local_similarity_options u Report meniju.

Prilikom kasnijeg učitavanja snimljenog rečnika, rezultati poređenja fraza se ne učitavaju.

Poređenje fraza u toku rada

Ako u toku rada želimo da uporedimo dve fraze, to je omogućeno pokretanjem prozora Phrase Comparison (Slika 7.), koje se vrši opcijom Compare_phrases… u Options meniju.

U polja First_phrase i Second_phrase se unose fraze koje želimo da uporedimo. Moguće ih je iskopirati iz Editora pomoću Copy opcije u Edit meniju, ili tasterima CTRL+C.

Run dugme vrši poređenje i upisuje rezultate u odgovarajuća tekst polja. Rezultati imaju oblik:

- za metode Levenshtein distance i Cost depending on the gap: cena_pretvaranja_jedne_fraze_u_drugu**:**jedno_od_najboljih_mogućih_poravnanja_za_ovaj_metod;
- za metodu Longest common subsequance: jedan_od_najvećih_mogućih_uređenih_podskupova_obe_fraze**:** jedno_od_najboljih_mogućih_poravnanja_za_ovaj_metod;
- za Local similarity: stepen_najveće_sličnosti_delova_dve_fraze**:** jedno_od_najboljih_mogućih_poravnanja_za_ovaj_metod.

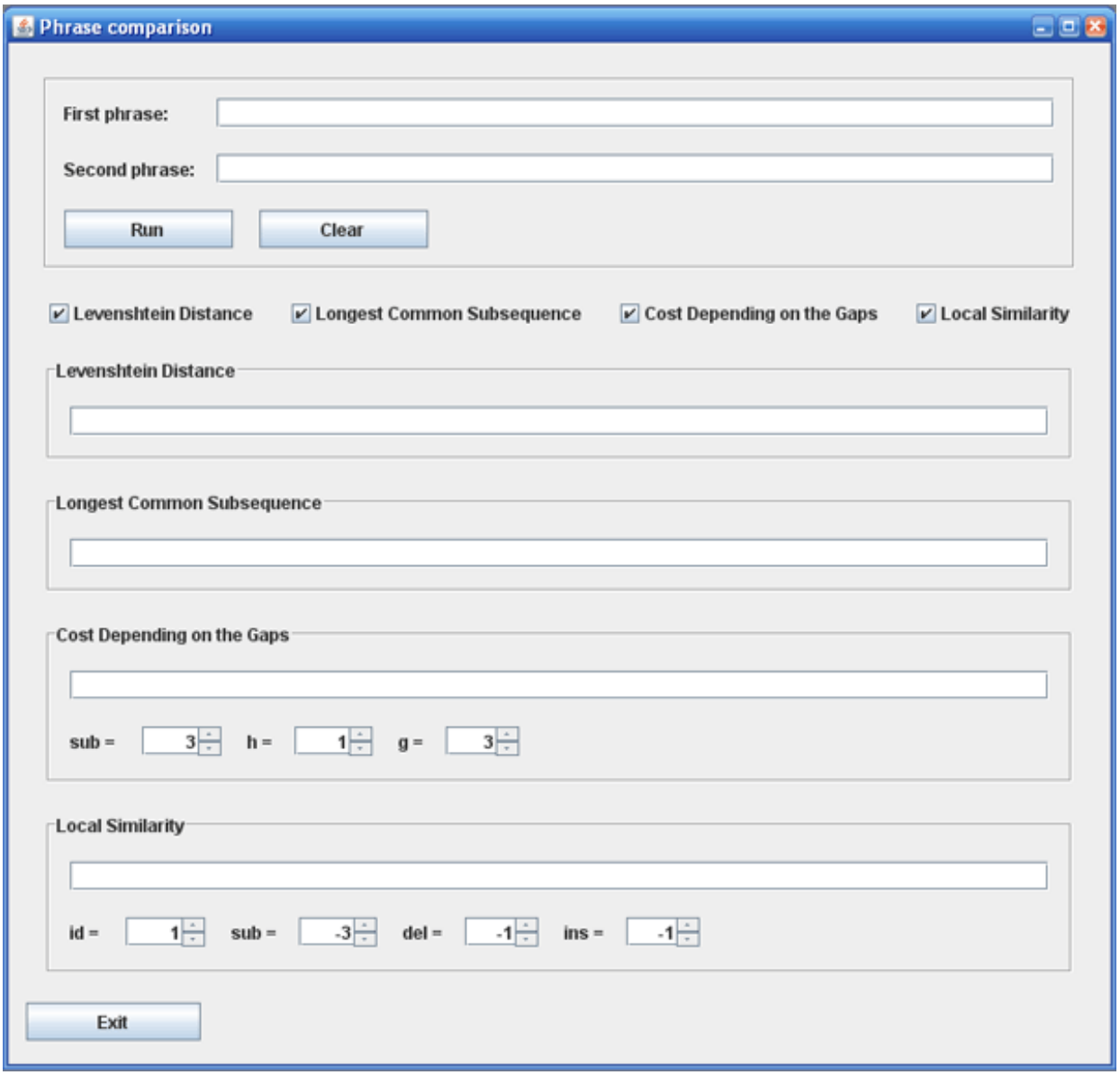
Clear dugme briše predhodno poređenje i priprema prozor za novo.

Biranjem chackbox-ova određujemo koje metode poređenja nas zanimaju.

Moguće je menjanje parametara za metode Cost_depending_on_the_gap i Local_similarity.

Iz prozora se izlazi pritiskom na dugme Exit.

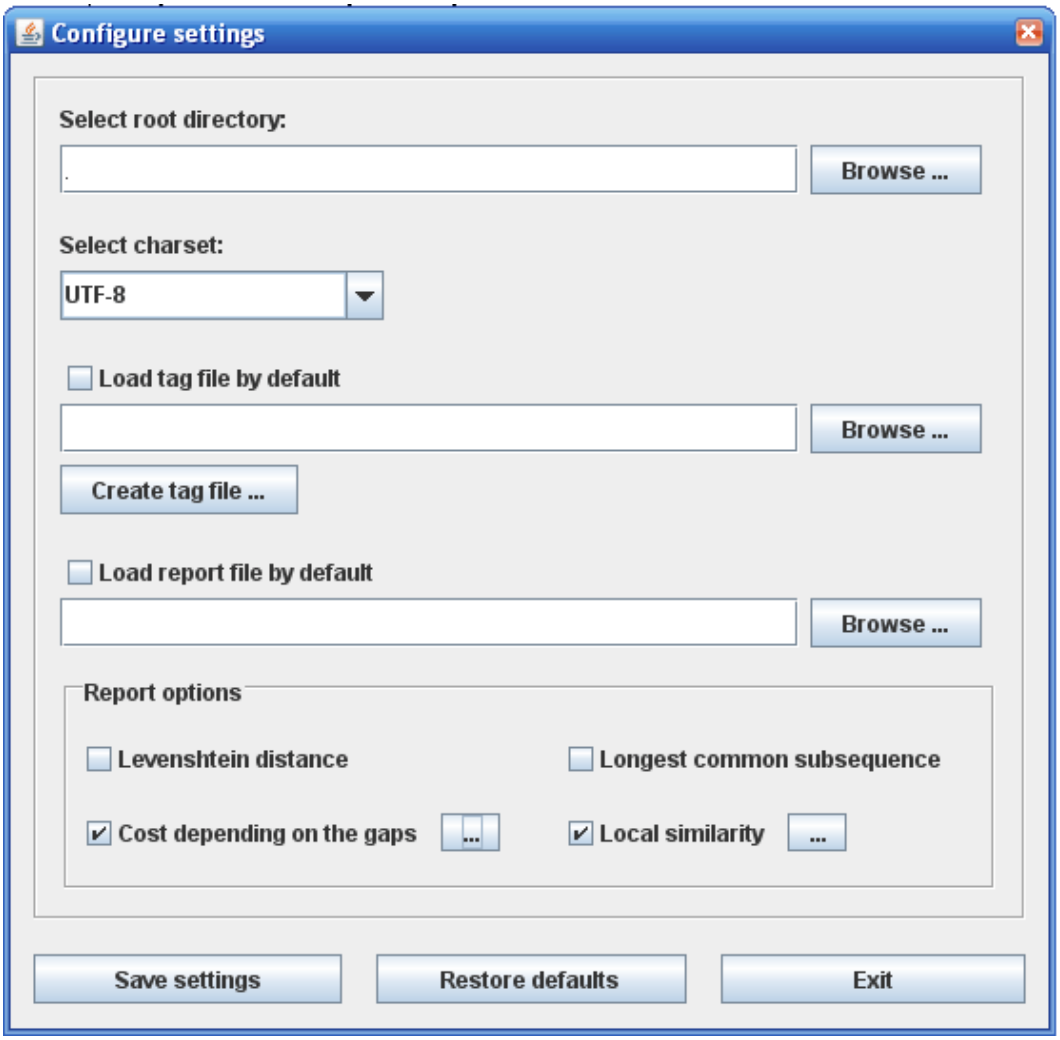
 U toku rada sa Phrase comparison prozorom, moguć je pristup i rad sa glavnim prozorom. Takođe je moguć rad sa više od jednog Phrase Comparison prozora u isto vreme.



Slika 7.

Konfiguracija

Radi lakšeg rada, za većinu opcija je moguće da se snime u konfiguracioni fajl **config.ini** i učitaju prilikom pokretanja programa. To se postiže iz dijaloga Configure settings, koji se pokreće opcijom Configue… u Options meniju (Slika 8).



Slika 8.

Select root directory – definiše početni direktorijum za sve Open i Save dijaloge.

Select charset – definiše charset koji se koristi prilikom rada sa HTML- i tag- fajlovima.

Load tag file by default – omogućava učitavanje tagova iz zadatog tag fajla pre učitavanja HTML fajla. Listu tagova će u ovom slučaju da čine tagovi iz tag-fajla i tagovi prepoznati u HTML fajlu.

Create tag file – otvara dijalog za pravljenje i snimanje novog tag fajla.

Load report file by default – omogućava učitavanje sadržaja zadatog report fajla prilikom otvaranja HTML fajla.

Report options – omogućava biranje metoda poređenja fraza koje će biti izvršeno prilikom snimanja rečnika. Dugmad "…" omogućavaju određivanje parametara za odgovarajuće metode.

Save settings – vrši snimanje podataka u fajl **config.ini** i izlazi iz dijaloga.

Restore defaults – vraća sve parametre na vrednosti koje su predefinisane programom.

Exit – izlazi iz dijaloga bez čuvanja bilo kakvih promena.

 Snimljena konfiguracija će da važi tek od sledećeg pokretanja programa.

Ako fajl **config.ini** ne postoji, program će se pokrenuti sa predefinisanim parametrima.

Poređenje delova teksta

Do sada smo pominjali poređenje delova teksta (reči, kompozita…) na dva mesta: kao način prepoznavanja kognata u tekstu i kao mogućnost u BiTMark-u. Recimo nešto više o tome.

Dynamic-Programming algoritam

Umesto da pri poređenju dve reči (kompozite…) tražimo njihove sličnosti, do rezultata možemo da dođemo posmatrajući njihove razlike. Drugim rečima, posmatramo rastojanje između njih.

Neka je Σ alfabet datog jezika, i N_0 skup prirodnih brojeva i nule. Nad tim alfabetom definišemo funkciju

$$d: \Sigma \rightarrow N_0,$$

za koju važe osobine:

1) $\forall$ x, y $\in$ Σ : d(x,x) = 0 i d(x, y) > 0 za x ≠ y
2) $\forall$ x, y $\in$ Σ : d(x, y) = d(y, x)
3) $\forall$ x, y, z $\in$ Σ : d(x, y) ≤ d(x, z) + d(z, y).

Tada za funkciju d kažemo da je metričko rastojanje na prostoru Σ.

Neka je sada $x = x_0x_1\ldots x_{m-1}$ reč dužine m sastavljena od slova iz alfabeta Σ, i $y = y_0y_1\ldots y_{n-1}$ reč sastavljena od slova istog alfabeta, dužine n. Smatramo da prazna reč, ε, ima dužinu 0. Sada nam je porebna funkcija rastojanja koja definiše rastojanje između ovih reči.

Zanimaju nas ona rastojanja koja omogućavaju prevođenje jedne reči u drugu korišćenjem substitucije (menjanje jednog slova drugim), brisanja i ubacivanja slova. Svakoj od tih operacija dodeljuje se brojčana vrednost koju nazivamo cenom operacije. Naš cilj je nalaženja takvog niza substitucija, brisanja i ubacivanja koje će transformisati reč *x* u reč *y*, pri čemu će zbir pojedinačnih cena tih transformacija, ukupna cena, da bude minimalna. Neka je *Sub(a, b)* cena substitucije slova *a* slovom *b*, *Del(a)* cena brisanja slova *a*, a *Ins(b)* cana ubacivanja slova *b*. Ako je *Sub(a, b)* metričko rastojanje na prostoru Σ, tj. rastojanje između slova, tada je ovako definisana ukupna cena rastojanje između reči.

Niz transformacija sa minimalnom cenom nije uvek jedinstven.

Ove transformacije možemo da prikažemo poravnavanjem reči. Poravnavanje obeležavamo ovako: označimo sa *(a→b)* substituciju slova *a* sa slovom *b*, *(a→ε)* brisanje slova *a*, i *(ε→b)* dodavanje slova *b*.

Drugi način zapisa poravnanja (alignment) je pomoću grafa.

Neka je G = (V, E) označen težinski graf takav da se svakoj grani pridružuju funkcije

- label: $E \rightarrow \Sigma \cup \{\varepsilon\} \times \Sigma \cup \{\varepsilon\}$, koja svakoj grani dodeljuje poravnati par slova i
- cost: $E \rightarrow R$, koja svakoj grani dodeljuje cenu odgovarajućeg poravnanja slova.

Graf definišemo na sledeći način:

- V je skup čvorova:

  $V = \{(i, j) \mid i \in [-1, m-1], j \in [-1, n-1]\}$

- Ako je $i \in [-1, m-1]$, $j \in [-1, n-1]$, definišemo skup grana E:
  - $((i-1, j-1), (i, j)) \in E$ ako
    - $label((i-1, j-1), (i, j)) = (x_i \rightarrow y_j)$ i
    - $cost((i-1, j-1), (i, j)) = Sub(x_i, y_j)$
  - $((i-1, j), (i, j)) \in E$ ako
    - $label((i-1, j), (i, j)) = (x_i \rightarrow \varepsilon)$ i
    - $cost((i-1, j), (i, j)) = Del(x_i)$
  - $((i, j-1), (i, j)) \in E$ ako
    - $label((i, j-1), (i, j)) = (\varepsilon \rightarrow y_j)$ i
    - $cost((i, j-1), (i, j)) = Ins(y_j)$

Sada je potrebno naći najkraći put između čvora (-1, -1) i (m-1, n-1).

Na osnovu ovog grafa, možemo da napravimo dvodimenzionalnu tabelu T sa m+1-im redom i n+1-om kolonom. Iz grafa vidimo da vrednost u ćeliji (T[i, j]) zavisi samo od tri njoj susedne ćelije: gornje (T[i-1, j]), leve (T[i, j-1]) i gornje leve (T[i-1, j-1]). Vrednosti računamo Dynamic-programming algoritmom:

```
Dynamic-programming (x, m, y, n)
    T[-1, -1] = 0;
    FOR j = 0, n-1
        T[-1, j] = T[-1, j-1] + Ins(y_j);
```

```
FOR i = 0, m-1
{
        T[-1, j] = T[-1, j-1] + Ins(y_j);
        FOR j = 1, n-1
                T[i, j] = min {T[i-1, j-1] + Sub(x_i, y_j), T[i-1, j] + Del(x_i),
                T[i, j-1] + Ins(y_j)};
}
RETURN T[m-1, n-1];
```

Ovaj algoritam vraća najmanju cenu poravnanja reči. Da bi se dobio i potreban niz transformacija, potrebno se vratiti kroz tabelu od ćelije T[m-1, n-1] do T[-1, -1] birajući najpogodniji put. Sledeći algoritam daje jedan od najkraćih puteva:

```
One-Alignment (x, m, y, n, T)
        z = ();
        i = m-1;
        j = n-1;
        WHILE i ≠ -1 AND j ≠ -1
                IF T[i, j] = T[i-1, j-1] + Sub(x_i, y_j)
                THEN
                        z = (x_i → y_j) z;
                        i = i - 1;
                        j = j – 1;
                ELSE IF T[i, j] = T[i-1, j] + Del(x_i)
                THEN
                        z = (x_i →ε) z;
                        i = i – 1;
                ELSE
                        z = (ε → y_j) z
                        j = j – 1;
```

WHILE $i \neq -1$

$z = (x_i \rightarrow \varepsilon)\ z;$

$i = i - 1;$

WHILE $j \neq -1$

$z = (\varepsilon \rightarrow y_j)\ z$

$j = j - 1;$

RETURN z;

Levenshtein distance

Levenštajnovo rastojanje se dobija kada se u predhodnim algoritmima uzmu vrednosti:

$Sub(x, x) = 0$, za $x \in \Sigma$

$Sub(x, y) = 1$, za $x, y \in \Sigma$ i $x \neq y$

$Del(x) = Ins(x) = 1$, za $x \in \Sigma$.

Longest common subsequences

Za razliku od Levenštajnovog rastojanja, koje isto tretira substituciju, brisanje i ubacivanje, ovaj algoritam dodeljuje najveću cenu substituciji. Cena kombinacije po jednog brisanja i ubacivanja je uvek manja od cene jedne substitucije. Dakle, za $x, y \in \Sigma$ važi

$Sub(x, x) = 0;$

$Sub(x, y) > Del(a) + Ins(b)$, za $x \neq y;$

Ovakvo rastojanje nam vraća najdužu moguću zajedničku podsekvencu. Podsekvenca reči *x* se dobija brisanjem nula ili više slova iz reči *x*. Dve reči mogu da imaju više različitih najdužih zajedničkih podsekvenci, ali je dužina svih njih ista. Dužina najduže moguće podsekvence označava se sa lcs(x, y), dok se skup takvih podsekvenci označava sa Lcs(x, y).

Za računanje dužine najduže moguć podsekvence ponovo definišemo tabelu T, veličine (m+1) x (n+1). Vrednosti u tabeli T su definišemo kao $T[i, j] = lsc(x_0 \ldots x_i, y_0 \ldots y_j)$, iz čega možemo da izvučemo formulu za $i \in [-1, m-1]$ i $j \in [-1, n-1]$

$T[i, -1] = T[-1, j] = 0$

$T[i, j] = T[i-1, j-1] +1$, za $x_i = y_j$

$T[i, j] = \max\{T[i-1, j], T[i, j-1]\}$, za $x_i \neq y_j$

Računanje dužine tražene podsekvence vrši se algoritmom Longest-Common-Subsequence, a traženje same podsekvence algoritmom Trace-Back.

```
Longest-Common-Subsequence (x, m, y, n)
    FOR i = -1, m-1
        T[i, -1] = 0;
    FOR j = -1, n-1
        T[-1, j] = 0;
    FOR i = 0, m-1
        FOR j = 0, n-1
            IF x_i = y_j
            THEN
                T[i, j] = T[i-1, j-1] + 1;
            ELSE
                T[i, j] = max{T[i-1, j], T[i, j-1]};
    RETURN T;
```

```
Trace-Back (x, m, y, n, T)
    i = m-1;
    j = n-1;
    k = T[m-1, n-1] – 1;
    WHILE i > 0 AND j > 0
        IF x_i = y_j AND T[i, j] = T[i-1, j-1]
        THEN
            w_k = x_i;
            i = i – 1;
            j = j – 1;
```

```
            k = k – 1
        ELSE IF T[i-1, j] > T[i, j-1]
        THEN
            i = i – 1;
        ELSE
            j = j – 1;
    RETURN w;
```

Cost depending on the length of gaps

Ovaj algoritam računa rastojanje između reči na osnovu broja i dužine rupa nastalih prilikom poravnanja reči.

Posmatrajmo funkciju λ: N $\rightarrow$ R. Neka je $\lambda(k)$ otvor dužine k. Možemo da definišemo vrednosti

$D(i, j) = \min \{T[k, j] + \lambda(i - k) \mid k \in [0, i-1]\};$

$I(i, j) = \min \{T[i, k] + \lambda(j - k) \mid k \in [0, j-1]\};$

$T[i, j] = \min \{T[i-1, j-1] + Sub(x_i, y_j), Del(i, j), Ins(i, j)\};$

gde je D(i, j) rezultat najboljeg poravnanja reči $x_0 \ldots x_i$ i $y_0 \ldots y_j$ koje se završava brisanjem slova iz *x*, I(i, j) rezultat najboljeg poravnanja koje se završava ubacivanjem slova iz *y*, i T[i, j] rezultat najboljeg poravnanja.

Ako uzmemo da je λ afina funkcija, oblika $\lambda(k) = g + h(k-1)$, možemo da smatramo da je cena otvaranja praznine *g*, a cena njenog proširenja *h*. Pod tim uslovima, definišimo vrednosti

$D(i, j) = \min \{D(i-1, j) + h, T[i-1, j] + g\};$

$I(i, j) = \min \{I(i, j-1) + h, T[i, j-1] + g\};$

$T[i, j] = \min \{T[i-1, j-1] + Sub(x_i, y_j), Del(i, j), Ins(i, j)\};$

Sledi algoritam

```
Gap((x, m, y, n)

FOR i = -1, m-1
        D[i, -1] = ∞;
```

I[i, -1] = ∞;

FOR j = -1, n-1

D[-1, i] = ∞;

I[-1, i] = ∞;

T[-1, -1] = 0;

T[-1, 0] = g;

T[0, -1] = g;

FOR i = 1, m-1

T[i, -1] = T[i-1, -1] + h;

FOR i = 1, n-1

T[-1, i] = T[-1, i-1] + h;

FOR i = 0, m-1

FOR j = 0, n-1

D[i, j] = min {D[i-1, j] + h, T[i-1, j] + g};

I[i, j] = min {I[i, j-1] +h, T[i, j-1] + g};

T[i, j] = min {T[i-1, j-1] + Sub(x_i, y_j), Del[i, j], Ins[i, j]};

Tražena vrednost se nalazi u polju T[m-1, n-1].

Menjanjem vrednosti *g* i *h* moguće je menjati i krajnji rezultat.

Local similarity

Ovde pokušavamo da nađemo najbolje poravnanje između delova reči *x* i *y*. Drugim rečima, tražimo delove reči koji su najsličniji. U tu svrhu svim transformacijama dodeljujemo negativne vrednosti. Prvo definišemo tabelu T:

T[i, j] = max {0,

T[i-1, j-1] + Sub(x_i, y_j),

T[i-1, j] + Del(x_i),

T[i, j-1] + Ins(y_j)},

a zatim se u tabeli nađe najveća vrednost i krene unazad.

Ponovo, na konačan rezultat je moguće uticati menjanjem vrednosti transformacija.

U ovom radu je opisan način formiranja biteksta, sa posebnim naglaskom na označavanje kompozita, njihovo prenošenje iz jednog teksta u drugi i formiranje rečnika koji sadrži ove kompozite.

Postojanje ovakvih rečnika, kojih je za sada veoma mali broj, značajno olakšava i pridonosi tačnosti kako mašinskog, tako i polu-mašinskog prevođenja.

Dodajući mogućnosti BiTMark-a već zavidnim mogućnostima WS4LR-a, formiranje ovakvih rečnika iz novih bitekstova je svedeno na jednostavno označavanje željenih kompozita mišom i izbor potrebnih oznaka.

Nadajmo se da će ovako olakšan način označavanja dati podstrek bržem porastu broja ovakvih rečnika, a samim time razvoju nadasve tačnijih automatskih prevodilaca.

Bibliografija

1. **Simard, Michel** (1998.): The BAF: A Corpus of English-French Bitext. In First International Conference on Language Resources and Evaluation, Granada, Spain. [Online] http://www.iro.umontreal.ca/~simardm/lrec98/lrec98.html.

2. **Vukosavljević, Milan:** *Algoritmi za paralelizovanje tekstova i njihova implementacija.* Magistarski rad, Matematički fakultet, Univerzitet u Beogradu. Mentor: Vitas, Duško.

3. **Laporte, Éric and Vitas, Duško and Krstev, Cvetana** (2006.): *Preparation and Exploitation of Bilingual Text.* Matematički fakultet, Univerzitet u Beogradu, Filozofski fakultet, Univerzitet u Beogradu i Institut Gaspard-Monge, Université de Marne-la-Vallée.

4. **Simoes, Alberto Manuel Brandao** (2004): *Parallel Corpora Word Alignment and Applications.* Escola de Engenharia, Universidade do Minho.

5. **OSCAR – LISA** (2005.): *TMX 1.4b Specification.* [Online] *http://www.lisa.org/standards/tmx/tmx.html.*

6. **TEI P5** (2006): Guidelines for Electronic Text Encoding and Interchange. [Online] http://www.tei-c.org/release/doc/tei-p5-doc/html/

7. **Bekavac, Božo** (2001.): *Primjena Računalnojezikoslovnih Alata na Hrvatske Korpuse.* Magistarski rad, Filozofski fakultet, Zagreb. Mentor: Tadić, Marko.

8. **Vitas, Duško and Krstov, Cvetana** (2006.): *Literature and Aligned Text,*. in *Readings in Multilinguality*, eds. Milena Slavcheva, Galia Angelova and Kiril Simov, pp. 148-155, Institute for Parallel Processing, Bulgarian Academy of Sciences, Sofia, Bulgaria,

9. **Ridings, Daniel and Danielsson, Pernilla** (1997.): *Practical Presentation of "Vanilla" Aligner,* in *Workshop on Alignment and Exploitation of Multilingual Texts,* at the Jozef Stefan Institute in Ljubljana, Slovenia.

10. **Krstev, Cvetana and Stanković, Ranka and Vitas, Duško and Obradović, Ivan** (2006.): *WS4LR: A Workstation for Lexical Resources.* Proceedings of the 5th International Conference on Language Resources and Evaluation, LREC 2006, Genoa, Italy.

11. **Laporte, Éric and Nakamura, Takuya and Voyatzi, Stavroula** (2008.): *A French Corpus Annotated for Multiword Nouns.* Proceedings of the Language Resources and Evaluation

Conference (LREC), Workshop Towards a Shared Task on Multiword Expressions, Marrakech, Morocco.

12. **Charras, Christian and Lecroq, Thierry** (1996.): *Sequence Comparison.* Universit'e De Rouen , Mont-saint-aignan Cedex.

Reference

1. *http://poincare.matf.bg.ac.rs/~vitas//Stavra/fr-tag/ef-01-TMX.tmx.* [Online]
2. **Foo, Jody:** *An Overview of Bitext Alignment Algorithms.* [Online] *http://www.ida.liu.se/~jodfo/gslt/bitext-alignment-jody.pdf*
3. *Wikipedia* [Online]:
    - Natural Language Processing: *http://en.wikipedia.org/wiki/Natural_language_processing*.
    - Text Corpus: *http://en.wikipedia.org/wiki/Text_corpus.*
    - Parallel Text: *http://en.wikipedia.org/wiki/Parallel_text.*
    - Translation Memory: *http://en.wikipedia.org/wiki/Translation_memory.*
    - Translation Memory eXchange: *http://en.wikipedia.org/wiki/Translation_Memory_eXchange.*
    - Text Encoding Initiative: *http://en.wikipedia.org/wiki/Text_Encoding_Initiative*.