Specifikace ročníkového projektu

Doplňování tvaroslovné informace k chybějícím slovům textu

Autor: Jan Václ, e-mail: janvacl@centrum.cz
Vedoucí projektu: Barbora Vidová-Hladká

Popis problému

Český akademický korpus (http://ufal.mff.cuni.cz/rest/CAC/cac_10.html) je poměrně rozsáhlá banka českých textů.U každého slova korpusu je doplněna informace mj. o slovním druhu a jeho kategoriích (rod, číslo, pád, osoba aj.). V textech jsou věty, ve kterých chybí více či jedno slovo. Detekce těchto podezřelých míst byla provedena ručně. Cílem rp je implementace automatické procedury, která u chybějících a ručně lokalizovaných slov doplní informace o slovním druhu a dalších kategoriích.

Návrh řešení

Jako nástroj pro řešení problému se jeví nejvhodnější některý z algoritmů strojového učení. Vybral jsem tzv. case-based přístup, tj. rozhodování na základě analogie s uloženými případy. Je to jeden z přístupů tzv. líného učení, tedy případná abstrakce neprobíhá hned při trénování, ale až při výpočtu v případech, kdy jí je potřeba. „Případ“ (case) zde bude představovat vektor morfologických značek okolních slov (kontext). Výsledná značka bude chybějícímu slovu přiřazena právě na základě podobnosti tohoto kontextu s některým s naučených případů. Míra podobnosti je právě klíčovým faktorem v tomto přístupu (viz také dále).

Výhodami tohoto přístupu je vedle celkově jednoduchého principu (tedy i poměrně přímočaré implementace) také rychlost a inkrementalita učení (doučovatelnost – i později lze jednoduše přidávat další případy).

Vzhledem k tomu, že je tento projekt spíše experimentální, budu se snažit udělat program co nejtvárnější – co nejvíce voleb souvisejících s algoritmem bude umístěno v externím konfiguračním souboru.

Trénování

Prvním krokem učícího algoritmu je trénování, učení. Jelikož je k dispozici dostatek označkovaných dat, lze je použít jako trénovací množinu (dají se využít jakákoliv označkovaná data bez chybějících slov). U case-based přístupu je trénovací proces velmi přímočarý – pro každé slovo se vytvoří jeden stav (case base) obsahující vektor morfologických značek okolních slov, značku daného slova (řešení) a násobnost tohoto případu. Tyto stavy se ukládají v paměti do tabulky, kde mohou být řazeny s ohledem na další fázi – hledání podobnosti.

Míra podobnosti (similarity metric)

Klíčovým okamžikem výpočtu je právě porovnávání stavů, na jehož základě se vybírá z uložených kontextů ten „nejpodobnější“ zkoumanému. Ukazuje se, že se vyplatí vzít v úvahu důležitost jednotlivých hodnot (feature relevance problem) - např. slovní druh slova o dvě pozice dále je zřejmě méně důležitý než pád slova těsně předcházejícího slovo zkoumané. Stanovení těchto důležitostí pak významně ovlivňuje výsledky značkovacího algoritmu.

Implementace

Struktura programu

1. Hlavní modul – ovládání, vstup, výstup. Rozhraní bude pouze konzolové, bez grafiky.
2. Trénovací modul – bude zajišťovat vytvoření datové struktury (tabulky) stavů na trénovacích datech
3. Značkovací (tagovací) modul – algoritmus značkování – využití tabulky stavů na přiřazení značek chybějícím slovům
4. Třídy datových struktur
5. Konfigurační soubor – bude obsahovat všemožná nastavení ovlivňující jednak komunikaci s uživatelem, ale především i samotný výpočet – velikost kontextu, hodnoty důležitostí.

Jazyk: C++

Cílové OS: Windows, Linux

Vývojové prostředí: MS Visual Studio, GCC