UNIVERSITATEA DIN BUCUREȘTI
VIRTUTE ET SAPIENTIA

Aspecte computaționale în producerea de cuvinte

Lucrare de Licență

Maria-Smaranda Pandele

22 Iunie, 2019

Coordonator: Prof. Dr. Liviu Dinu

Facultatea de Matematică și Informatică, UNIBUC

Rezumat

This example thesis briefly shows the main features of our thesis style, and how to use it for your purposes.

Cuprins

Capitolul 1

Introduction

This is version `v1.4` of the template.

We assume that you found this template on our institute's website, so we do not repeat everything stated there. Consult the website again for pointers to further reading about LaTeX. This chapter only gives a brief overview of the files you are looking at.

1.1 Features

The rest of this document shows off a few features of the template files. Look at the source code to see which macros we used!

The template is divided into TeX files as follows:

1. `thesis.tex` is the main file.
2. `extrapackages.tex` holds extra package includes.
3. `layoutsetup.tex` defines the style used in this document.
4. `theoremsetup.tex` declares the theorem-like environments.
5. `macrosetup.tex` defines extra macros that you may find useful.
6. `introduction.tex` contains this text.
7. `sections.tex` is a quick demo of each sectioning level available.
8. `refs.bib` is an example bibliography file. You can use BibTeX to quote references. For example, read if you can get a hold of it.

1.1.1 Extra package includes

The file `extrapackages.tex` lists some packages that usually come in handy. Simply have a look at the source code. We have added the following comments based on our experiences:

REC This package is recommended.

OPT This package is optional. It usually solves a specific problem in a clever way.

ADV This package is for the advanced user, but solves a problem frequent enough that we mention it. Consult the package's documentation.

As a small example, here is a reference to the Section *Features* typeset with the recommended *varioref* package:

> See Section 1.1 on the preceding page.

1.1.2 Layout setup

This defines the overall look of the document – for example, it changes the chapter and section heading appearance. We consider this a 'do not touch' area. Take a look at the excellent *Memoir* documentation before changing it.

In fact, take a look at the excellent *Memoir* documentation, full stop.

1.1.3 Theorem setup

This file defines a bunch of theorem-like environments.

Teoremă 1.1 *An example theorem.*

Proof Proof text goes here. □

Note that the q.e.d. symbol moves to the correct place automatically if you end the proof with an `enumerate` or `displaymath`. You do not need to use `\qedhere` as with *amsthm*.

Teoremă 1.2 (Some Famous Guy) *Another example theorem.*

Proof This proof

1. ends in an enumerate. □

Propoziție 1.3 *Note that all theorem-like environments are by default numbered on the same counter.*

Proof This proof ends in a display like so:

$$f(x) = x^2. \qquad \square$$

1.1.4 Macro setup

For now the macro setup only shows how to define some basic macros, and how to use a neat feature of the *mathtools* package:

$$\lvert a \rvert, \quad \left\lvert \frac{a}{b} \right\rvert, \quad \lvert \frac{a}{b} \rvert.$$

Capitolul 2

Introducere

Capitolul 3

Reconstrucție de cuvinte

Capitolul 4

Agregarea rezultatelor folosind distanța rank

Am văzut cum putem obține producții de cuvinte combinând câte o singură limbă romanică cu limba latină. Pentru a îmbunătății rezultatele vrem să folosim informația din mai multe limbi romanice moderne. Astfel, fiecare clasificator întoarce o listă ordonată de cuvinte latinești, pe prima poziție aflându-se perechea cognate cu cea mai mare probabilitate. Prin agregarea acestora cu o anumită metrică vom obține cele mai probabile perechi cognate.

4.1 Clasamente și distanța rank

Un *clasament* este o listă ordonată de obiecte după un anumit criteriu, pe prima poziție aflându-se cel cu cea mai mare importanță. În unele situații se pune problema găsirii unui clasament cât mai apropiat de o mulțime de mai multe clasamente. Pentru a rezolva această problemă trebuie sa definim mai întâi ce înseamnă distanța dintre două clasemente sau dintre un singur clasament și o mulțime de clasamente.

Există mai multe distanțe folosite cu succes în diverse aplicații: distanța *Kedall tau*, *Spearman footrule*, *Levenshtein*, dar noi vom folosi distanța *rank* [1], pe care o vom defini folosindu-ne de următoarele noțiuni. Fie $U = \{1, 2, ..., n\}$ o mulțime finită de obiecte numită univers. Un clasament peste universul U este lista ordonată $\tau = (x_1 > x_2 > ... > x_d)$, unde $\{x_1, x_2, ..., x_d\} \subseteq U$. $>$ este o relație de ordine strictă reprezentând criteriul de ordonare. Notăm cu $\tau(x)$, poziția elementului $x \in U$ în clasamentul τ dacă $x \in \tau$, numerotând pozițiile de la 1 începând cu cel mai important obiect din clasament. Dacă un clasament conține toate elementele din univers, atunci el se va numi *clasament total*. Asemănător, dacă conține doar o submulțime de obiecte din univers, atunci îl vom numi *clasament parțial*.

Notăm ordinea elementului x în τ astfel:

$$ord(\tau, x) = \begin{cases} |n + 1 - \tau(x)| & , x \in \tau \\ 0 & , x \in U \setminus \tau \end{cases}$$

Definiție 4.1 *Fie τ si σ două clasamente parțiale peste același univers U. Atunci distanța rank va fi*

$$\Delta(\tau, \sigma) = \sum_{x \in \tau \cup \sigma} |ord(\tau, x) - ord(\sigma, x)|$$

Se observă faptul că, în calculul distanței rank, se ia în considerare ordinea definită mai sus și nu poziția. În primul rând, cum primele poziții sunt cele mai importante, distanța dintre două clasamente trebuie sa fie cu atât mai mare cu cât diferă mai mult începutul lor[3]. În al doilea rând, definiția funcției *ord* pune accentul pe lungimea clasamentelor întrucăt putem presupune că un clasament mai lung a fost obținut în urma comparării mai multor obiecte din univers. Deci ordinea elementelor este mai solidă. Spre exemplu, dacă două clasamente de lungimi diferite au același element pe prima poziție, există totuși o diferență a ordinii obiectului în cele două liste, diferență ce contribuie la calculul distanței rank totale.[2]

4.1.1 Agregări cu distanța rank

O *agregare de clasamente* reprezintă un singur clasament σ astfel încât o anumită metrică de la acesta la mulțimea de liste de agregat T este minimă. Raportându-ne la distanța rank, putem defini formal o agregare astfel:

Definiție 4.2 *Fie un set de clasamente $T = \{\tau_1, \tau_2, ..., \tau_m\}$ dintr-un univers U și $\sigma = (\sigma_1 > \sigma_2 > ... > \sigma_k)$ un clasament astfel încât $\sigma_i \in U, \forall 1 \leqslant i \leqslant k$. Definim distanța rank de la σ la T astfel:*

$$\Delta(\sigma, T) = \sum_{i=1}^{m} \Delta(\sigma, \tau_i)$$

Definiție 4.3 *Fie un set de clasamente $T = \{\tau_1, \tau_2, ..., \tau_m\}$ dintr-un univers $U = \{x_1, x_2, ..., x_n\}$. Se numește agregare a mulțimii T folosind distanța rank clasamentul σ, $\sigma_i \in U, \forall 1 \leqslant i \leqslant k$, cu propietatea că $\Delta(\sigma, T)$ este minim posibilă.*

Definim următoarele matrici bidimensionale $W^l(i, j)$ cu n linii și n coloane. Fiecare celulă din fiecare matrice va semnifica costul total din distanța rank de la un clasament σ, de lungime l, către o mulțime $T = \{\tau_1, \tau_2, ..., \tau_m\}$ fixată indus de plasarea elementului $x_i \in U$ pe poziția j în σ [2]. Se observă faptul că un clasament peste universul U definit mai sus poate avea lungimea maxim n. Rezultă că numărul de coloane al matricilor W^t este egal cu n.

$$W^l(i,j) = \begin{cases} \sum_{p=1}^{m} |ord(p,i) - l + j| & ,j \leqslant l \\ \sum_{p=1}^{m} |ord(p,i)| & ,j > l \end{cases}$$

Propoziție 4.4 *Distanța de la un clasament $\sigma = (\sigma_1 > \sigma_2 > ...\sigma_k)$ la mulțimea T este*

$$\Delta(\sigma, T) = \sum_{x_i \in U \cap \sigma} W^k(i, \sigma(x_i)) + \sum_{x_i \in U \setminus \sigma} W^k(i, k+1)$$

unde n este reprezintă numărul de obiecte din univers, iar $k < n$.

Se observă faptul că, în cazul în care σ conține toate elementele din U, deci cazul $k = n$, formula se reduce la

$$\Delta(\sigma, T) = \sum_{x_i \in U \cap \sigma} W^k(i, \sigma(x_i)) \tag{4.1}$$

4.1.2 Reducerea la o problemă de cuplaj perfect de cost minim

Fiecare matrice W^l din secțiunea precedentă este caluculată în mod independent de celelalte. Deci putem determina doar o singură matrice pentru o anumită lungime fixată l. Vom presupune în continuare ca vrem să calculăm agregările unui set de clasamente având lungimea egală cu numărul total de elemente din univers. Astfel, problema se reduce la găsirea unu clasament σ ce minimizează formula (4.1). Formal:

Problemă 4.5 *Fiind dată o matrice pătratică W, $W = (w_{i,j})_{1 \leqslant i,j \leqslant n}$ vrem să determinăm următoarea mulțime:*

$$S = \{(i_1, i_2, ..., i_n) | (i_k \neq i_j \forall k \neq j), (i_j \in U) \text{ și } \sum_{j=1}^{n} w_{i_j,j} \text{ este minim}\}$$

Problema de mai sus se aseamănă cu o problemă de cuplaj perfect de cost minim întrucât vrem să formăm perechi între obiectele dintr-un univers și pozițiile unui clasament de tip agregare, iar fiecare combinație are un anumit cost. Practic $(i_1, i_2, ..., i_n)$ reprezintă o permutare a elementelor din U.

4.1.3 Calcularea tuturor agregărilor optime

4.2 Determinarea tuturor agregărilor producțiilor de cuvinte

În capitolul precedent am prezentat o metoda de a combina o limbă romanică modernă si limba latină pentru a automatiza procesul de determinare a etimonului latinesc. Metoda returna primele n cuvinte posibile ordonate de la cel cu probabilitatea cea mai mare la cel cu probabilitatea cea mai mică. Vom considera aceste liste de cuvinte ca fiind clasamente. Pentru fiecare cuvânt latinesc vom agrega clasamentele produse din fiecare limbă romanică modernă *(ro, it, fr, es, pt)*. Se observă faptul că pot exista mai multe astfel de agregări așa că ne propunem să le aflăm pe toate într-un mod eficient din punct de vedere al complexității timp. Alegem să luăm în considerare doar primele 5 cele mai bune cuvinte din fiecare set. Astfel, pentru un singur cuvânt latinesc, vom avea:

$$R = \{r_1, r_2, ..., r_k\}, \text{clasamentele produse din ro, it, fr, es, pt}$$
$$k = |R|, 1 \leqslant k \leqslant 5$$
$$U = \bigcup_{i=1}^{k} r_i \text{ universul de cuvinte}$$
$$n = |U|$$

Definim o matrice bidimensională de k linii și n coloane în care calculăm ordinea fiecărui cuvânt din univers în fiecare clasament dat:

$$ord[i][j] = \begin{cases} |6 - r_i(x_j)| & , x_j \in r_i \\ 0 & , x_j \in U \setminus r_i \end{cases}$$

Capitolul 5

Rezultate

Capitolul 6

Concluzii

Anexa A

Dummy Appendix

You can defer lengthy calculations that would otherwise only interrupt the flow of your thesis to an appendix.

Bibliografie

[1] L.P. Dinu. On the classification and aggregation of hierarchies with different constitutive elements. *Fund. Inform. 55 (1)*, pages 39–50, 2003.

[2] F. Manea L.P. Dinu. An efficient approach for the rank aggregation problem. *Theoretical Computer Science vol 359(1-3)*, pages 455–461, 2006.

[3] S. Marcus. Linguistic structures and generative devices in molecular genetics. *Cahiers Ling. Theor. Appl.*, pages 77–101, 1974.