Inteligenta Artificiala - Tema 3
Clasificarea si rezumarea articolelor de stiri

Teodor-Stefan Dutu

Universitatea Politehnica Bucuresti
Facultatea de Automatica si Calculatoare
Grupa 341C3

Abstract. Tema propune implementarea algoritmului *Naive Bayes* atat pentru clasificarea unor articole de stiri in categoriile din care acestea fac parte, cat si pentru a crea cate un rezumat al fiecarui articol, format din anumite fraze din respectivul articol.

Cuprins

1 Cerinta 2 - Clasificarea articolelor

In vederea clasificarii, modelul ajunge sa invete *campul lexical* al fiecarei categorii de stiri. Astfel, cele mai des intalnite 10 cuvinte in functie de categorie se pot vedea in Tabelul 1. In acest tabel sunt trecute cele mai frecvente cuvinte atunci cand se aplica lematizare si se elimina cuvintele neinformative.

business	**entertainment**	**politics**	**sport**	**tech**
say	film	say	say	say
%	say	mr	win	people
year	year	labour	game	game
$	good	party	year	technology
company	award	government	play	mobile
bn	music	election	player	phone
mr	star	people	england	mr
firm	win	blair	time	year
market	include	minister	good	new
£	£	tory	world	user

Table 1: Cele mai frecvente cuvinte din fiecare clasa

Atunci cand textul este mai putin procesat, cele mai frecvente cuvinte sunt conjunctii, articole si prepozitii precum `"the"`, `"a"`, `"on"` etc., dar acestea sunt prezente in toate categoriile, drept care ele nu influenteaza acuratetea clasificatorului. Un aspect interesant este ca desi exista cuvinte comune mai multor categorii (`"say"`, `"year"`, `"game"`, `"mr"`), acestea ajung sa se anuleze reciproc, iar clasificarea propriu-zisa este facuta de fapt pe baza cuvintelor specifice fiecarei clase.

Asadar, preciziile si regasirile obtinute de model au valori de peste 95% indiferent daca se elimina cuvintele neinformative sau daca se foloseste lematizarea, deoarece, asa cum mentionam mai sus, atunci cand modelele invata cuvinte inutile, acestea fie sunt in numar mic, fie sunt prezente in toate clasele si nu mai influenteaza inferentele.

Pentru ca masura performantele, am rulat setul de teste pe parcursul antrenarii, o data la 100 de fisiere, pentru a observa evolutia preciziei si a regasirii. Din graficele prezentate in Figurile 1, 2, 3 si 4 se observa ca, indiferent daca se aplica sau nu lematizarea si eliminarea cuvintelor nesemnificative, modelul invata rapid campurile lexicale ale claselor si ajunge la performantele sale maxime dupa ce asimileaza informatia din 500-600 de fisiere. Mai mult decat atat, modelul se dovedeste a fi suficient de robust incat sa nu manifeste *overfitting* dupa o antrenare completa cu peste 1600 de fisiere. Totusi, nici performantele nu se imbunatatesc substantial dupa ce modelul este antrenat cu 1000 de fisiere. Unul dintre motive este acela ca inca din acest moment, performantele modelului atat in termeni de precizie cat si de regasire sunt de peste 95%.

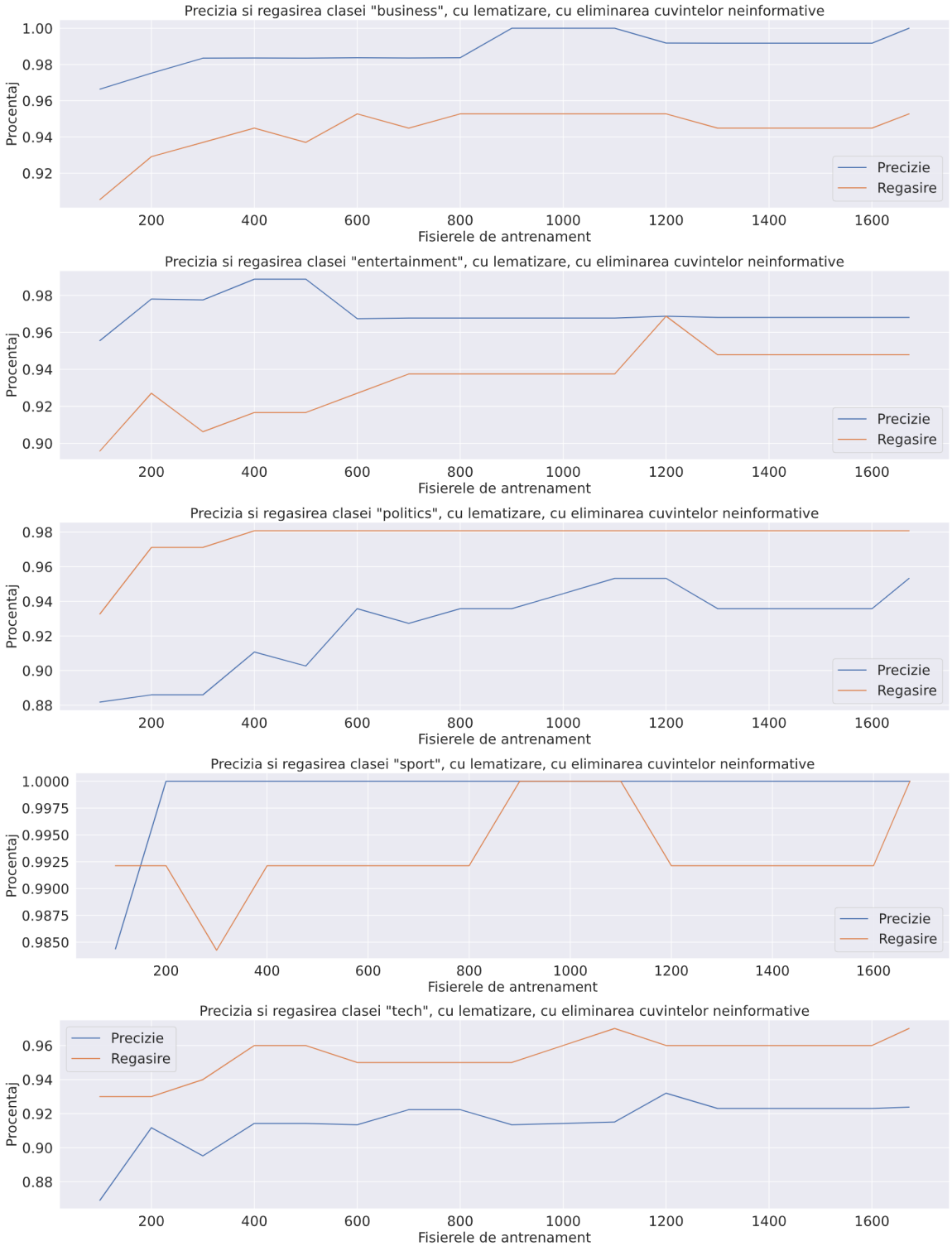

Fig. 1: Performantele modelului cand se aplica lematizare si eliminarea cuvintelor neinformative

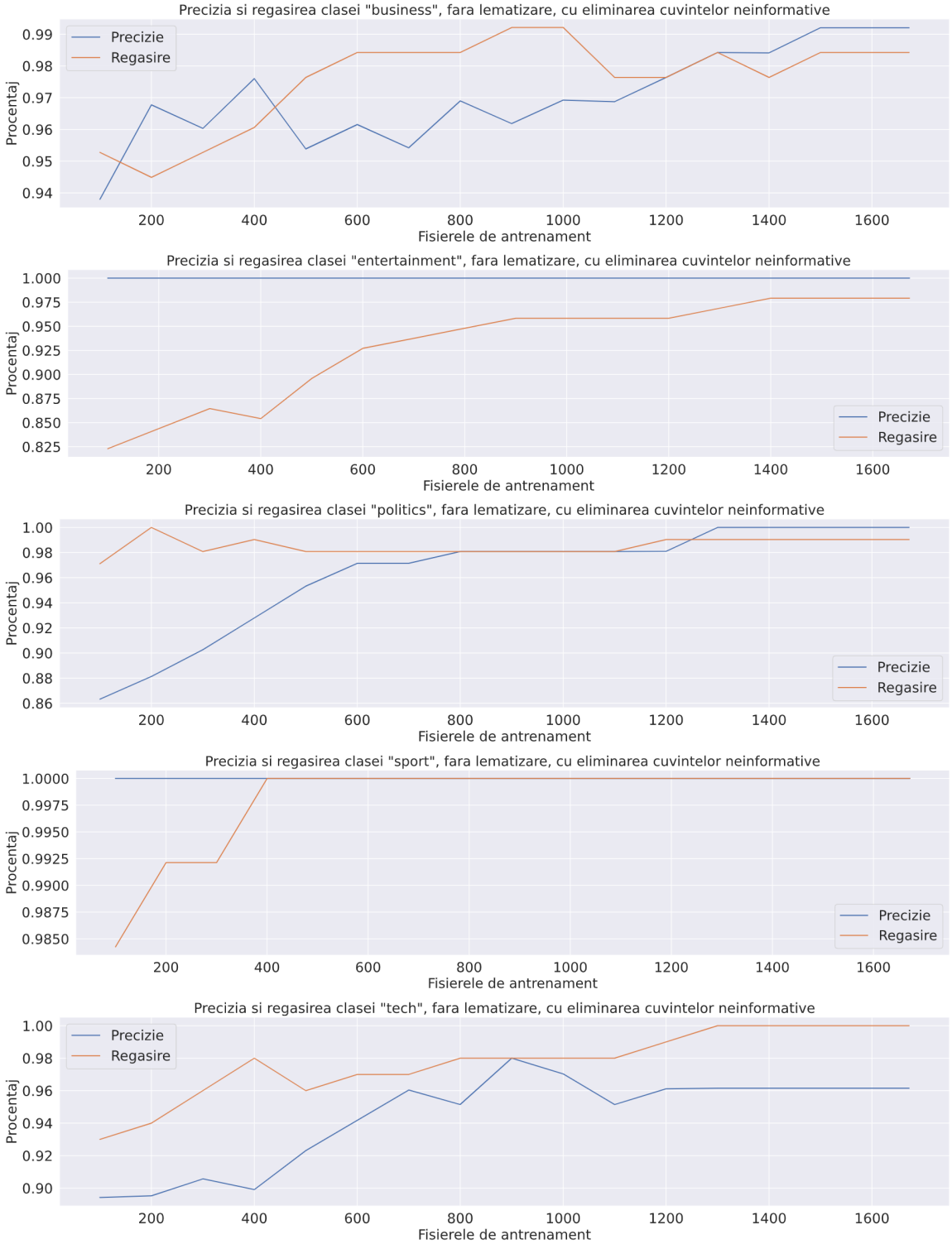

Fig. 2: Performantele modelului cand nu se aplica lematizare, dar se elimina cuvintele neinformative

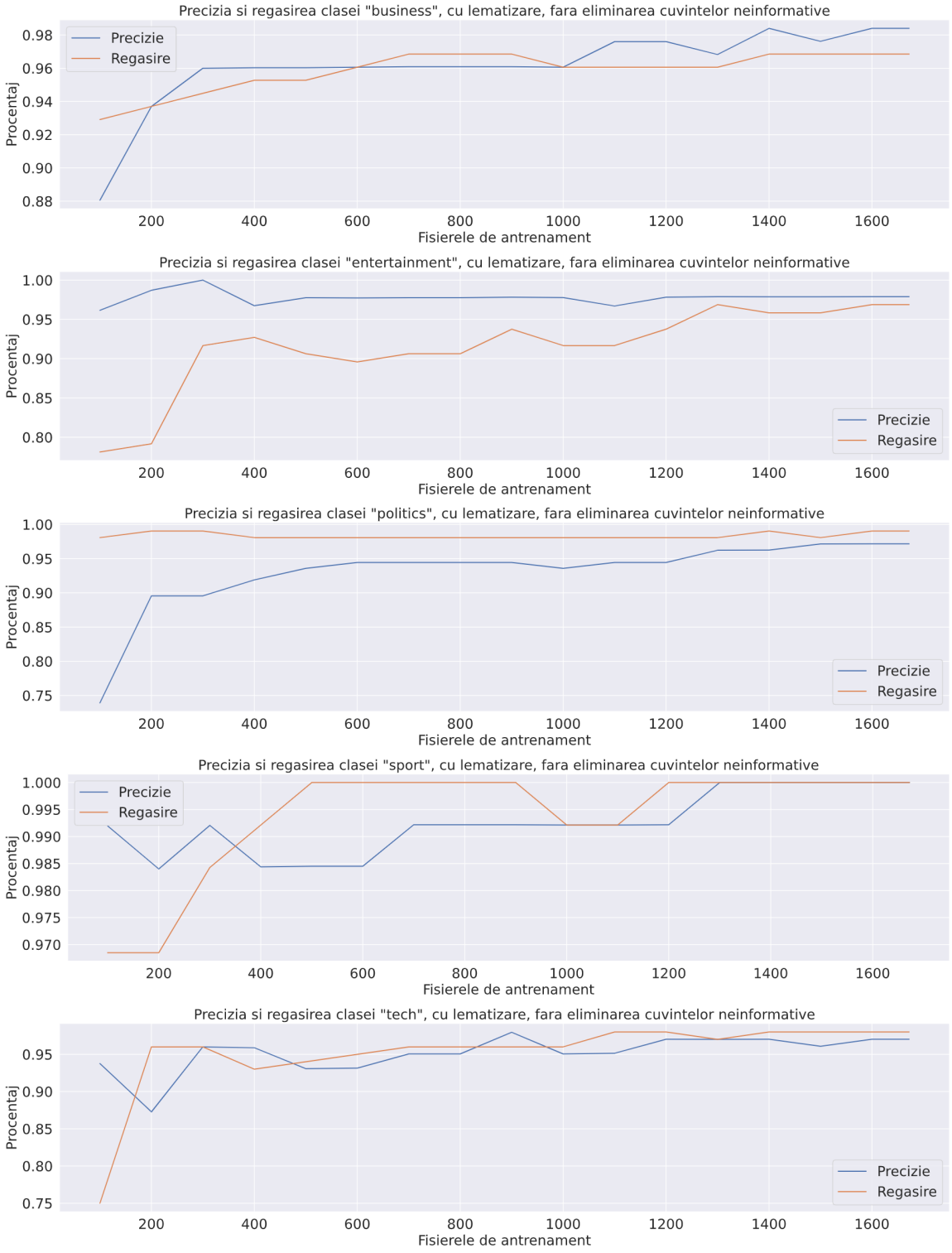

Fig. 3: Performantele modelului cand se aplica lematizare, dar nu se elimina cuvintele neinformative

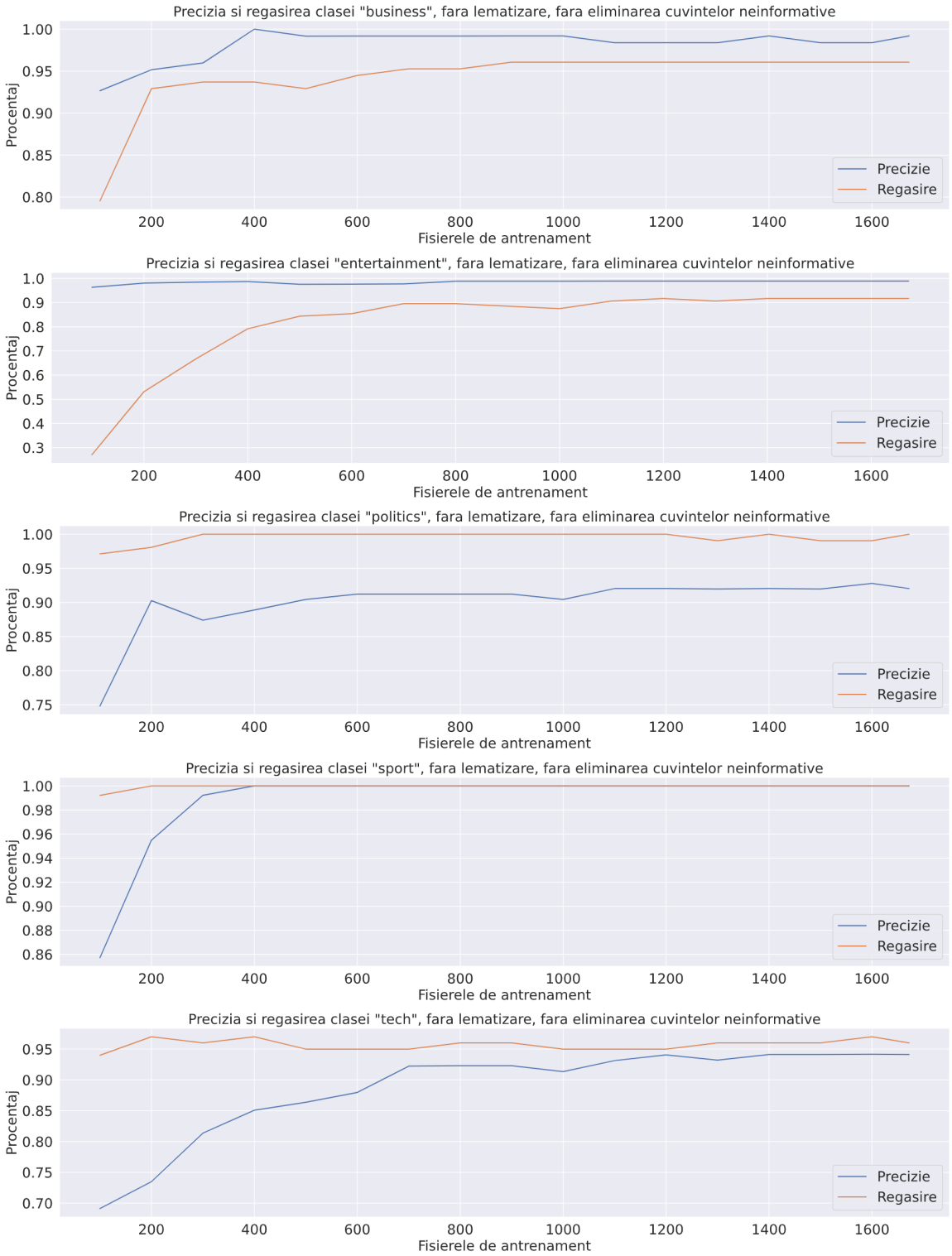

Fig. 4: Performantele modelului cand nu se aplica nici lematizare, nici eliminarea

Analizand matricele de confuzie din Figurile 5 si 6, observam in primul rand ca domeniul **sport** este foarte usor de recunoscut. Acest lucru se datoreaza unor cuvinte specifice acestui domeniu care nu se gasesc in celelalte (nume de sporturi, termeni tehnici din acestea, jucatori faimosi). In al doilea rand, se poate vedea ca de multe ori domeniile **business** si in special **entertainment** sunt confundate cu celelalte. Acest lucru poate fi cauzat de faptul ca articolele despre acestea vizeaza si alte zone, precum politica sau tehnologia, ceea ce poate induce modelul in erorare.

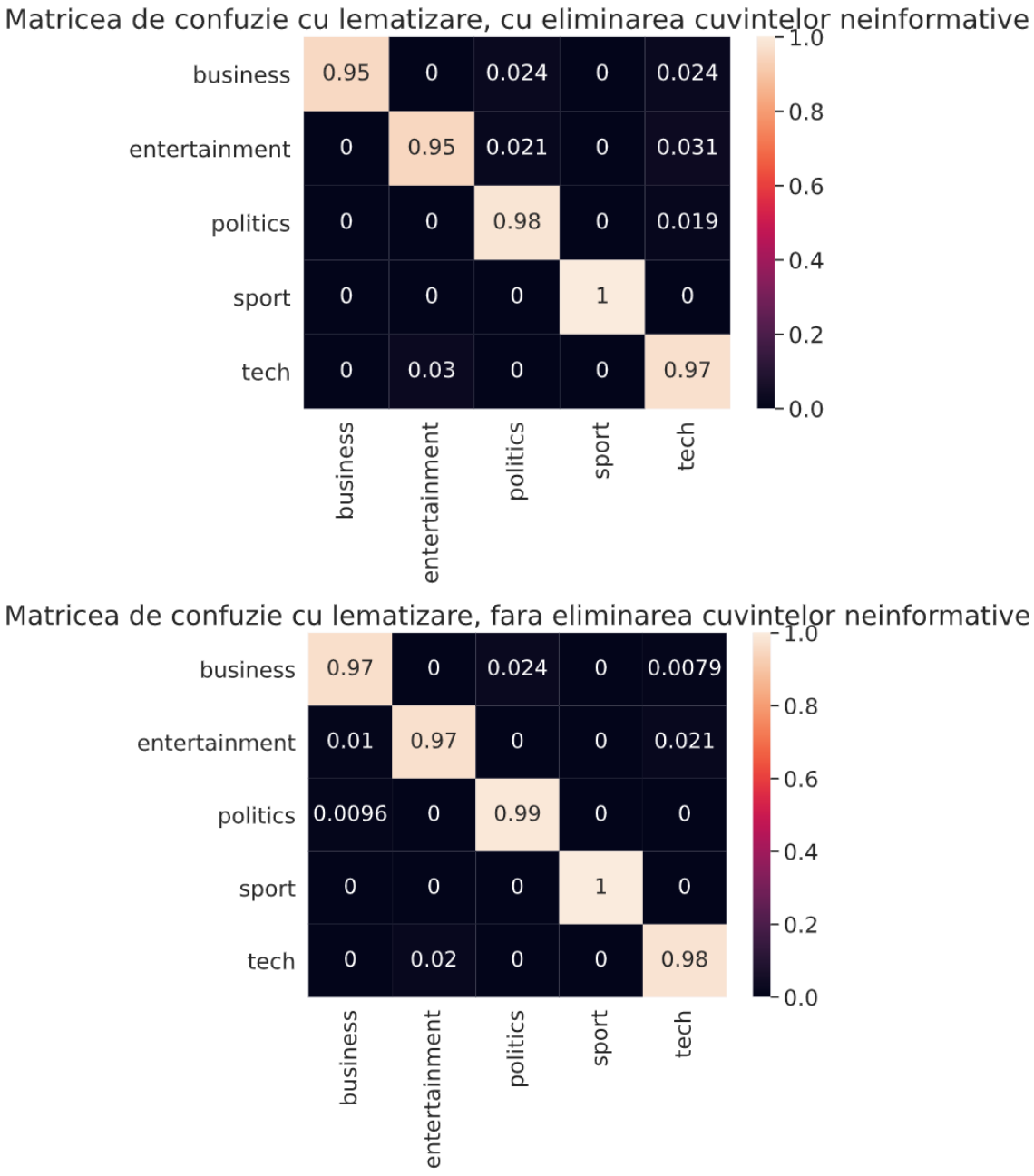

Fig. 5: Matricele de confuzie ale modelelor ce elimina cuvintele neinformative

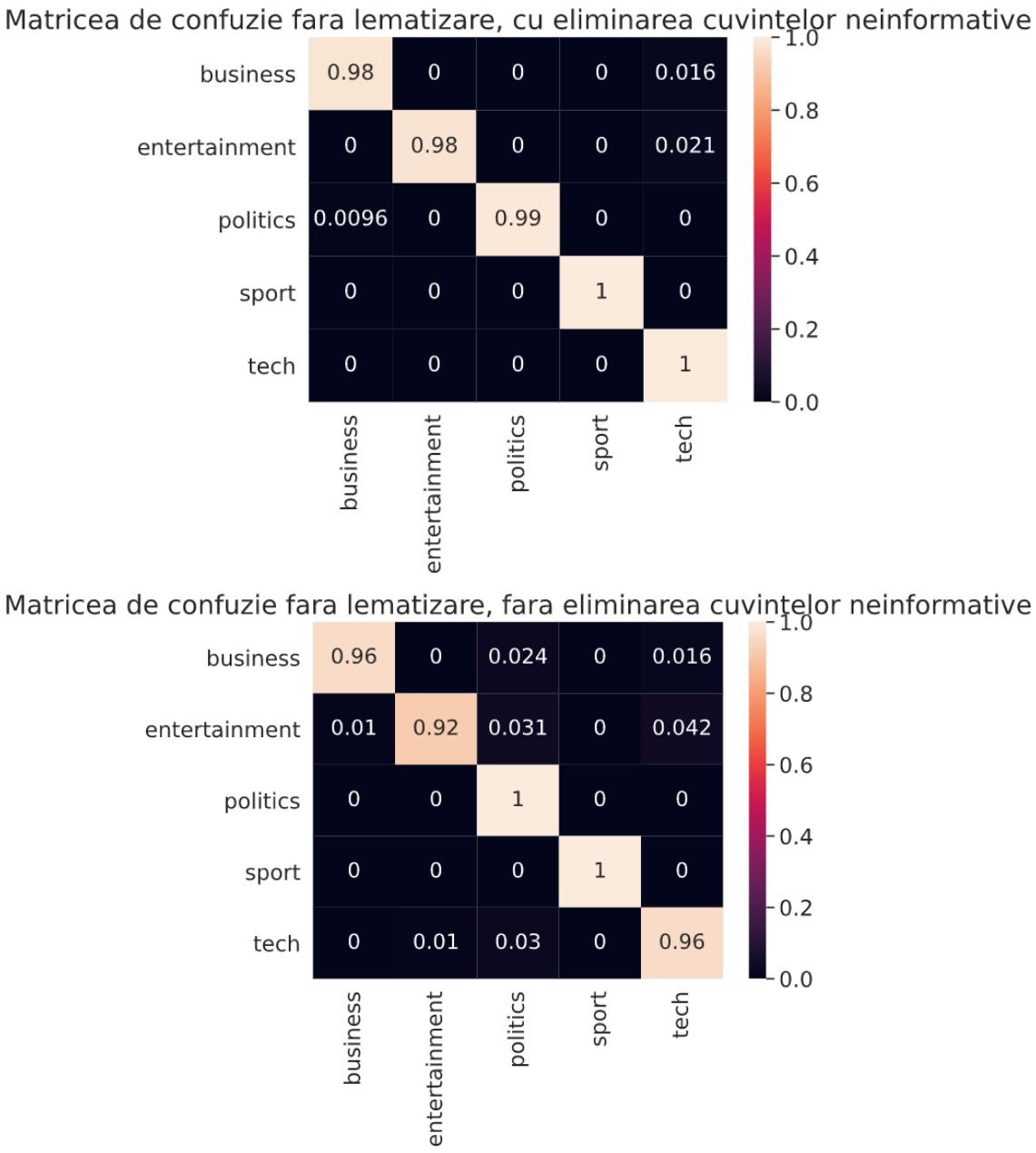

Fig. 6: Matricele de confuzie ale modelelor ce nu elimina cuvintele neinformative

2 Cerinta 3 - Rezumarea articolelor

Pentru rezumare, modelul clasifica enunturi in doua clase: una ce contine enunturi pastrate in rezumat si alta care le contine pe cele eliminate. In mod similar cerintei anterioare, am reprezentat grafic evolutia performantelor modelului pe parcursul antrenarii, folosind metricile *ROUGE-1* si *ROUGE-2*. De mentionat e ca pentru cacularea scorului *ROUGE-1* am folosit monograme, iar pentru *ROUGE-2*, bigrame

Scorurile *ROUGE-1* se gasesc in Figura 7. Date fiind regasirile mici si preciziile relativ mari din aceste grafice, inseamna ca modelul produce rezumate alcatuite din prea putine enunturi, dar aceste enunturi sunt in mare parte corecte. Se observa de asemenea ca regasirile scad, iar precizia creste pe parcursul antrenarii. Acest lucru inseamna ca modelul devine foarte restrictiv si considera ca tot mai putine enunturi trebuie sa faca parte din rezumat.

Atunci cand se folosesc bigrame si se monitorizeaza scorul *ROUGE-2*, comportamentul modelului pe parcursul antrenarii este acelasi, preciziile crescand, in timp ce regasirile scad, dar valorile acestora acum sunt aproape inversate. Astfel, folosindu-se bigrame, inferentele produc rezumate prea ample, printre care bineinteles ca se gasesc si propozitiile din rezumat (un rezumat ce contine toate enunturile ar avea regasirea 1). Graficele acestei implementari se pot vedea in Figura 8.

Preciziile si regasirile finale ale modelului in toate scenariile descrise mai sus sunt trecute in Tabelele 2 si 3. Din cel din urma reiese impactul pozitiv al lematizarii asupra rezumarii atunci cand antrarea foloseste bigrame, valorile regasirii crescand de la 0.55-0.6 chiar pana la 0.77, fara a se reduce semnificativ precizia. Acest lucru se datoreaza faptului ca prin intermediu lematizarii modelul poate invata mai eficient acele cuvinte-cheie care fac ca o propozitie sa trebuiasca sa fie adaugata intr-un rezumat. Faptul ca acest fenomen se intampla doar atunci cand antrenarea foloseste bigrame este cauzat de numarul de bigrame rezultate atunci cand nu se foloseste lematizare, un numar mult mai mare in comparatie cu cel de monograme rezultate in acelasi scenariu. Astfel, luand exemplul verbului `"play"`, cand modelul foloseste monograme, acesta mai poate aparea doar in formele `"plays"` si `"played"`, pe cand bigramele introduc introduc variante ce cuprind si declinarile sau conjugarile cuvantului ce precede sau succeda verbul `"play"`. Din acest motiv, nu intalnim variatiuni prea mari in Tabelul 2.

	Lematizare si eliminare	**Lematizare fara eliminare**	**Eliminare fara lematizare**	**Nici eliminare nici lematizare**
Precizie	0.697	0.704	0.717	0.717
Regasire	0.474	0.508	0.523	0.515

Table 2: Scorurile *ROUGE-1* obtinute de model

	Lematizare si eliminare	**Lematizare fara eliminare**	**Eliminare fara lematizare**	**Nici eliminare nici lematizare**
Precizie	0.532	0.550	0.515	0.540
Regasire	0.706	0.553	0.773	0.613

Table 3: Scorurile *ROUGE-2* obtinute de model

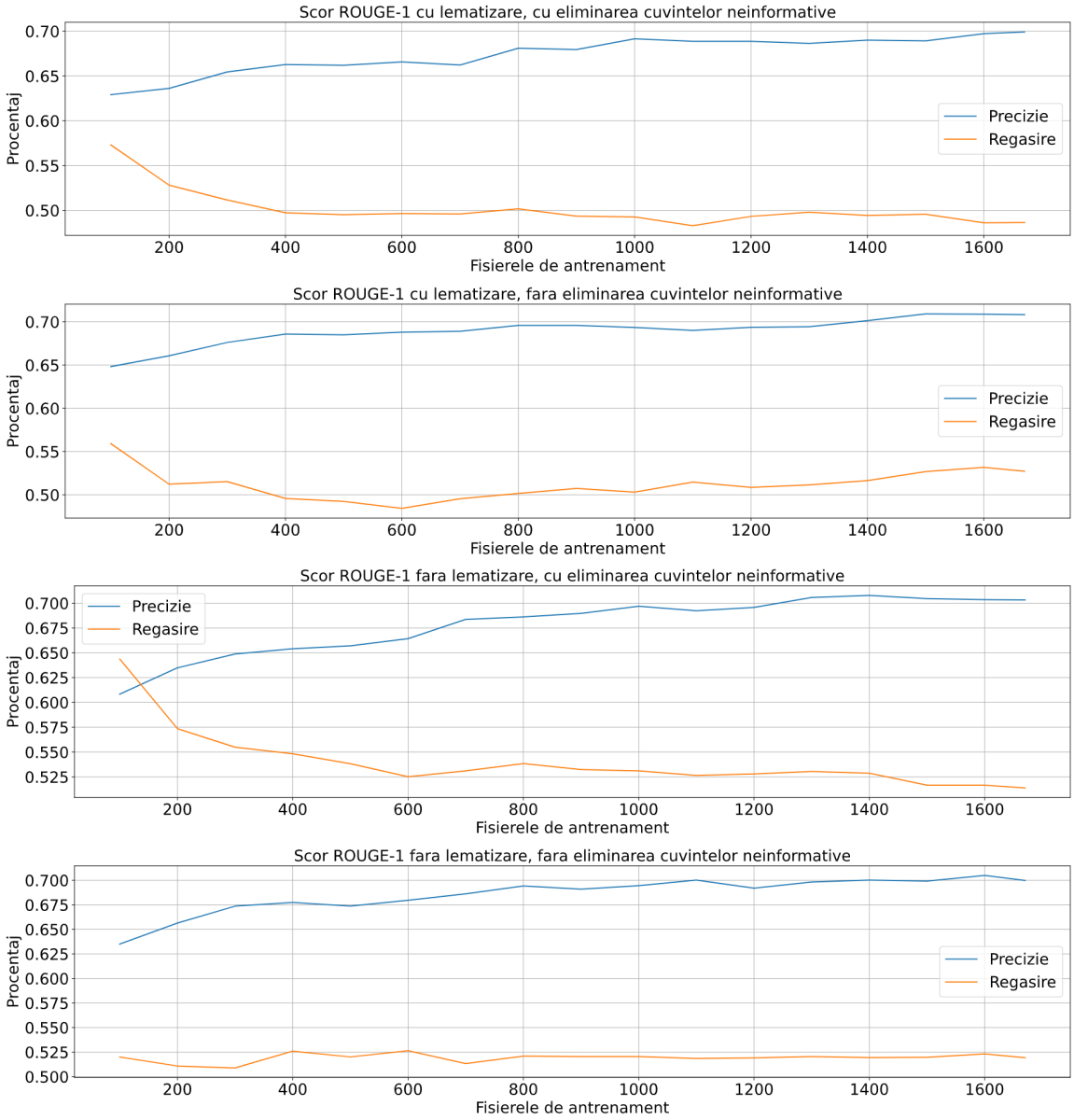

Fig. 7: Scorurile *ROUGE-1* obtinute de model

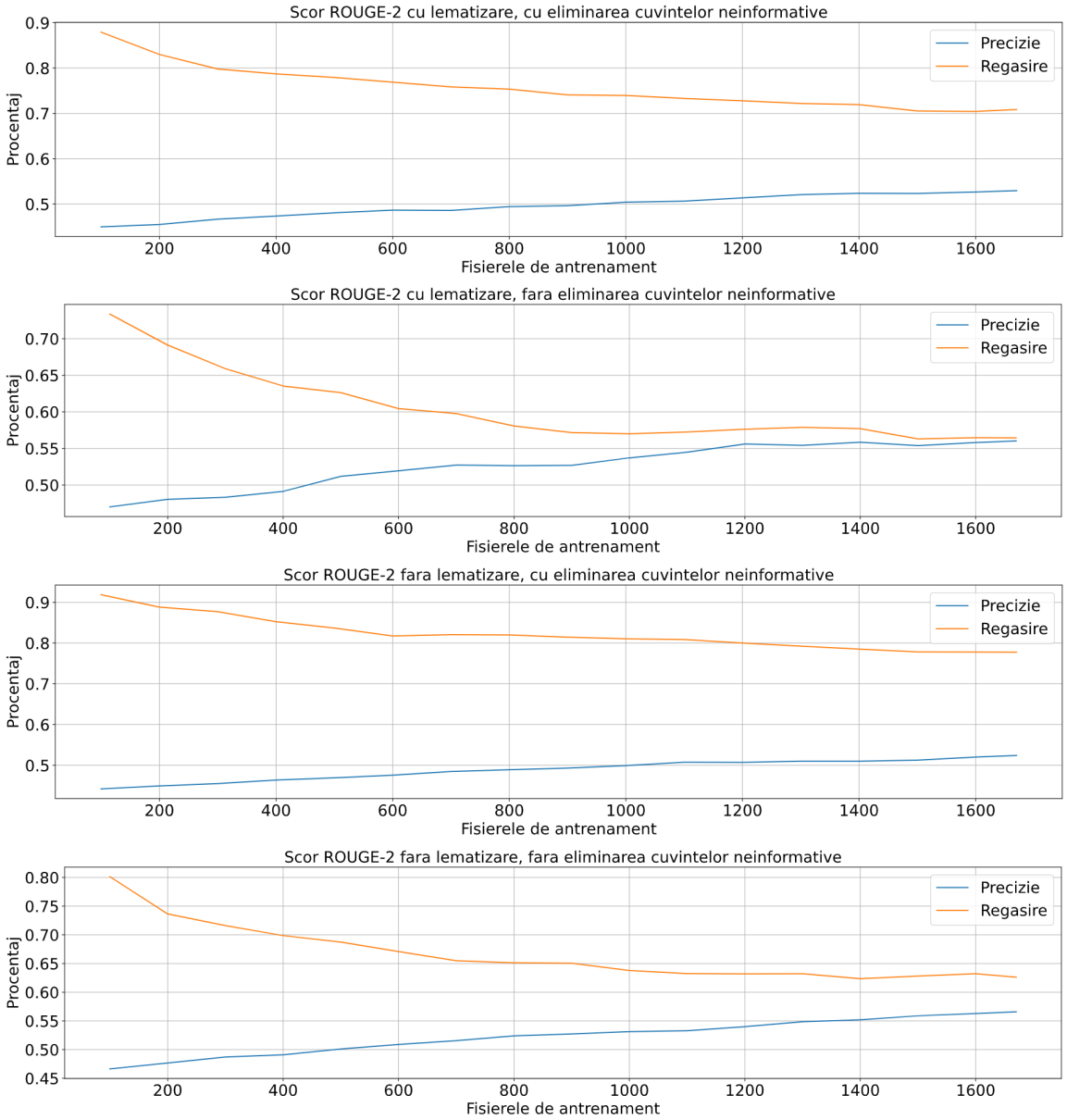

Fig. 8: Scorurile *ROUGE-2* obtinute de model

Bibliografie

1. *Netezire Laplace*
   `https://towardsdatascience.com/laplace-smoothing-in-na%C3%AFve-bayes-algorithm-9c237a8bdece`
   Data ultimei accesari: 30 Dec 2021
2. *Matrice de confuzie*
   `https://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html`
   Data ultimei accesari: 31 Dec 2021
3. *Scorul ROUGE*
   `http://www.ccs.neu.edu/home/vip/teach/DMcourse/5_topicmodel_summ/notes_slides/What-is-ROUGE.pdf`
   Data ultimei accesari: 2 Ian 2021
4. *Calculearea scorului ROUGE-N in Python*
   `https://pypi.org/project/rouge-score/`
   Data ultimei accesari: 2 Ian 2021