Just-News

CADE-based research on newspapers language

Leonardo Alchieri
860624

Davide Badalotti
861354

Goal:

Determine the **Emotional Charge** of the language used in various American Newspapers

Goal:

Determine the **Emotional Charge** of the language used in various American Newspapers

"Efforts by states to expand access to mail-in voting have enlarged the pool of eligible mail voters."

- Describe factual realities
- Does not involve emotions
- Can be considered **objective** from our perspective

"*Police have been demonized in the days following the death of George Floyd."*

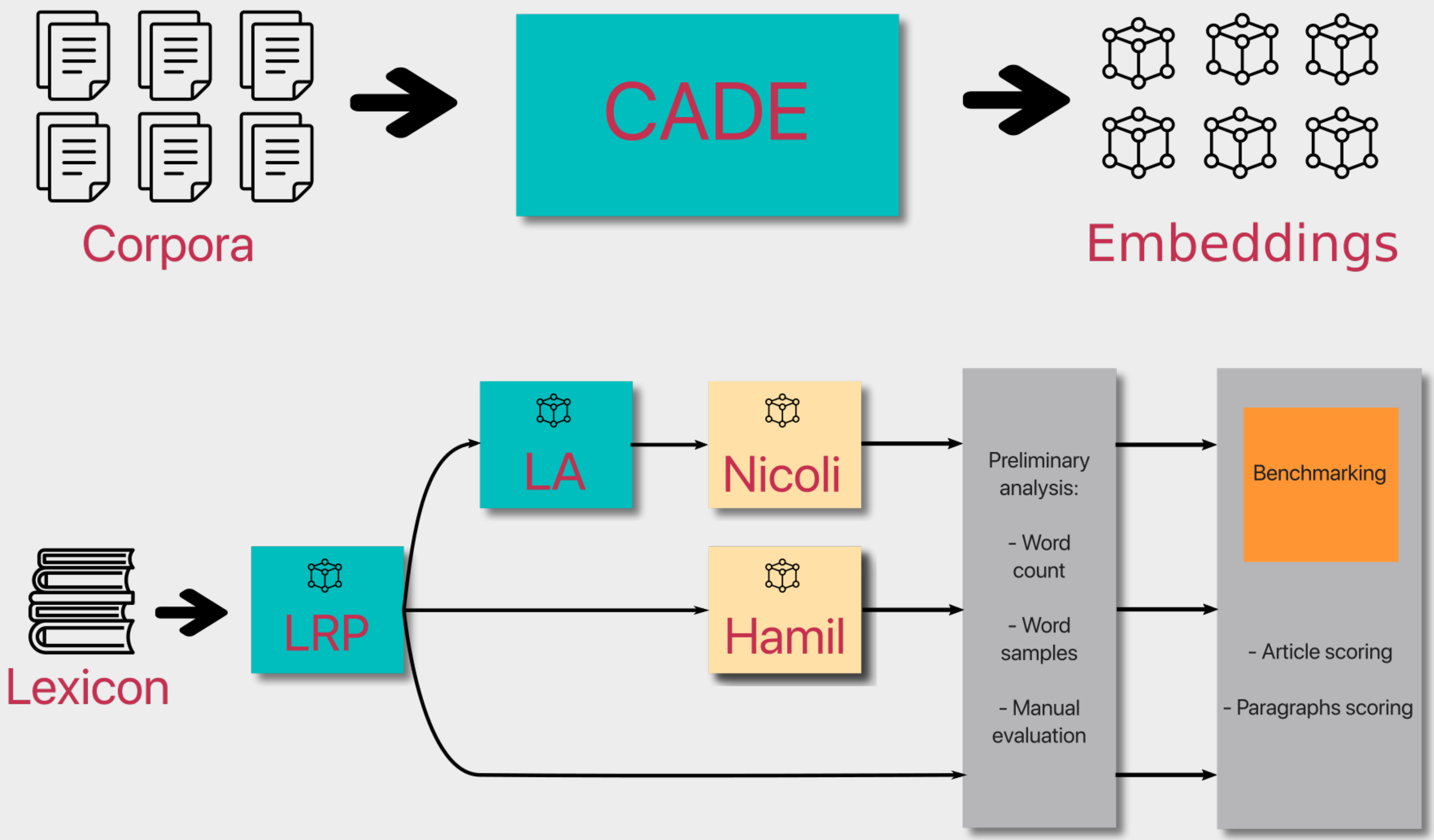
- Despite relying on facts, the author choses not to report them
- An emotion is presented
- Can be considered **subjective** from our point of view

Tools:

- Distributional methods (CADE).
- Annotated lexicons.
- Score induction methods:
  1. Dott. Nicoli
  2. Prof. Hamilton

  Are they comparable?

  How well do they perform?


Corpora
CADE
Embeddings
Lexicon
LRP
LA
Nicoli
Hamil
Preliminary analysis:
- Word count
- Word samples
- Manual evaluation
Benchmarking
- Article scoring
- Paragraphs scoring

Corpora:

Around ~60'000 articles and pages from 6 different sources:

- New York Times
- CNN
- ABC News
- Breitbart News
- The Federalist
- Wikipedia (used as control: hypothesis of neutral language)

Labeled Lexicons:

Two options:

- **MPQA Subjectivity Lexicon**. ~8000 words labeled as:
  - Strongly subjective (+1) [*fool, greatness, scary...*]
  - Weakly subjective (0) [*speculate, scheme, repute...*]

- **Harvard General Enquirer**. ~1000 words labeled as:
  - Over-stating (+1) [*bad, brutal, acute...*]
  - Under-stating (0) [*ambiguity, apparent, appear...*]

CADE Embeddings

Comparative Distributional Framework

$$\mathcal{F} = (D, V^*, \mathbf{C}, \Phi)$$

Set of slices:

$$D = \{D^1, \ldots, D^n\}$$

Set of vocabularies, including the shared one:

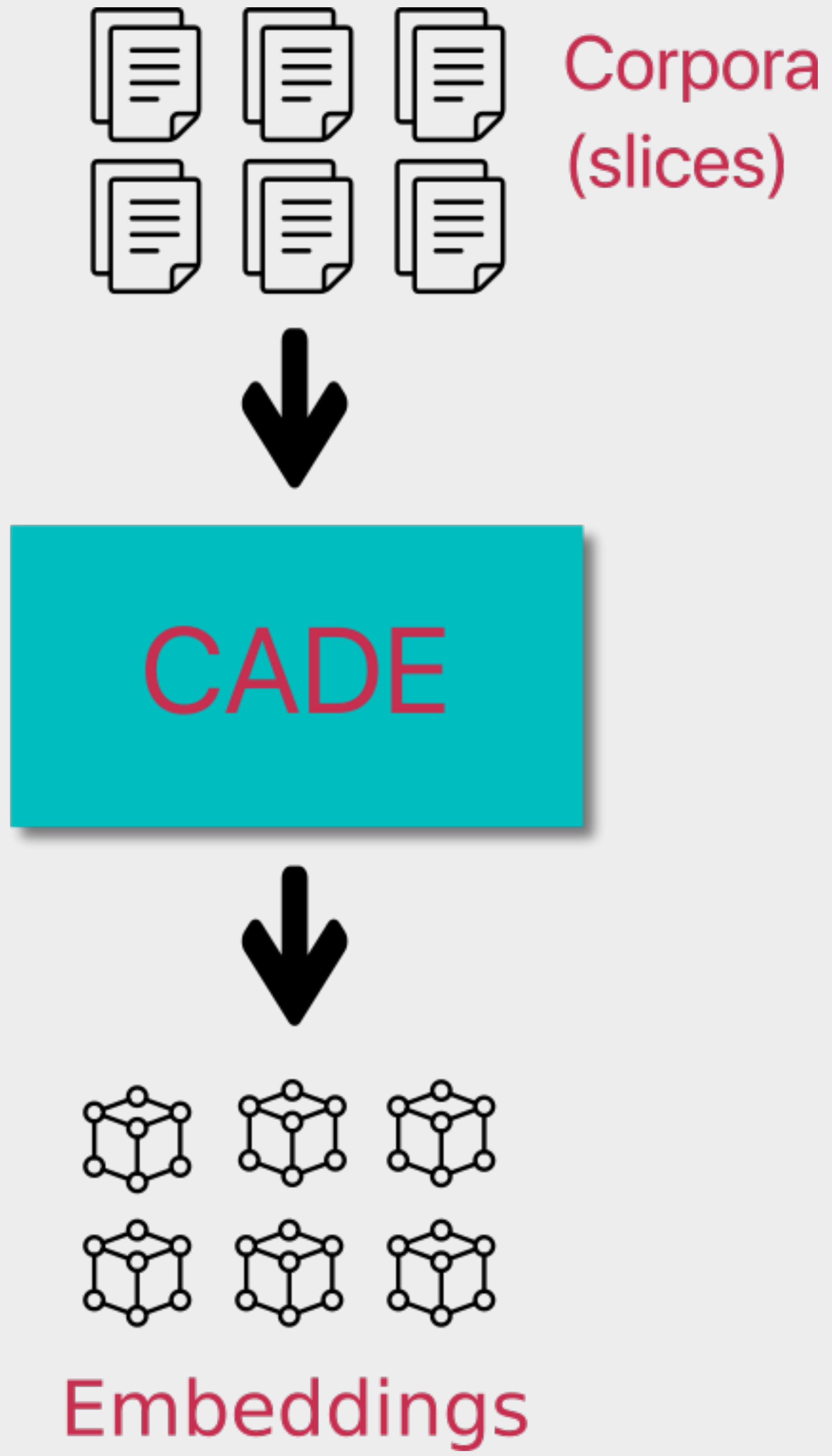
$$V^* = \{V, V^1, \ldots, V^n\} \qquad V = \cup_i^n V^i$$

Set of slice-specific embeddings:

$$\mathbf{C} = \{\mathbf{C}^1, \ldots, \mathbf{C}^n\}$$

Set of top-k nearest neighbours corresp. functions:

$$\Phi^k_{D^i \to D^j} : \mathbf{C}^i \to \{\mathbf{C}^j_{(1)}, \ldots, \mathbf{C}^j_{(k)}\}$$



Lexicon Refinement

Objective:

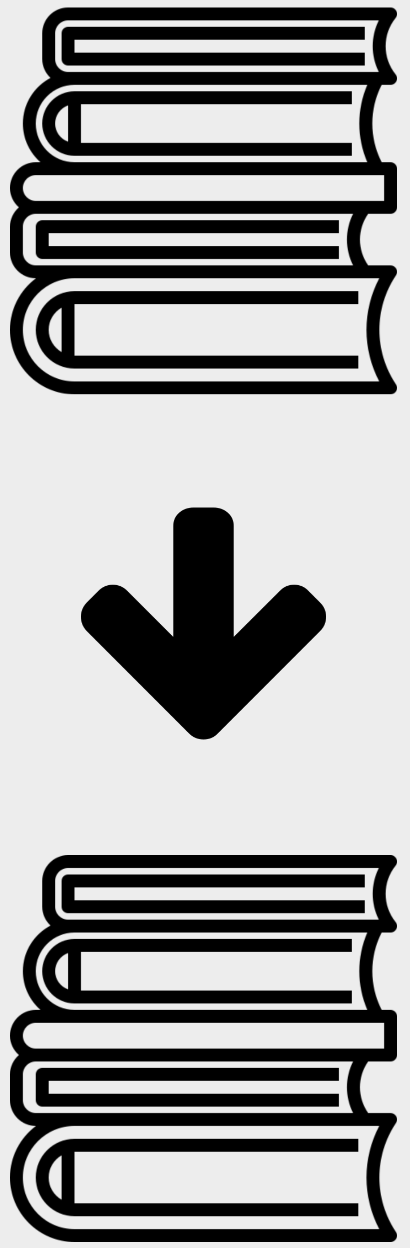
Determine a **subset of the initial lexicon** in which all the words have **stable vector-representation** across corpora.

$$\mathcal{L} \rightarrow \mathcal{L}_r$$

Where:

$$\mathcal{L}_r = \{w_j \in \cap_i V_i : \zeta_{D_i}(w_j) > 5 \; \forall i\}$$

$\zeta_{D_i}(w_j)$ is the Zipf measure of a word in corpus D_i.
High value means implies **stable and noise-free** word representation in corpus.
Standardized across corpora.



$\mathcal{L}$: 8222 words

$\mathcal{L}_r$: 64 words

Lexicon Augmentation

Objective:

Create new **artificial labeled word vectors** to increase the data quantity in the lexicon.

Procedure:

Given a word vector in the refined lexicon:

$$\mathbf{w_i} \in \mathcal{L}_r$$

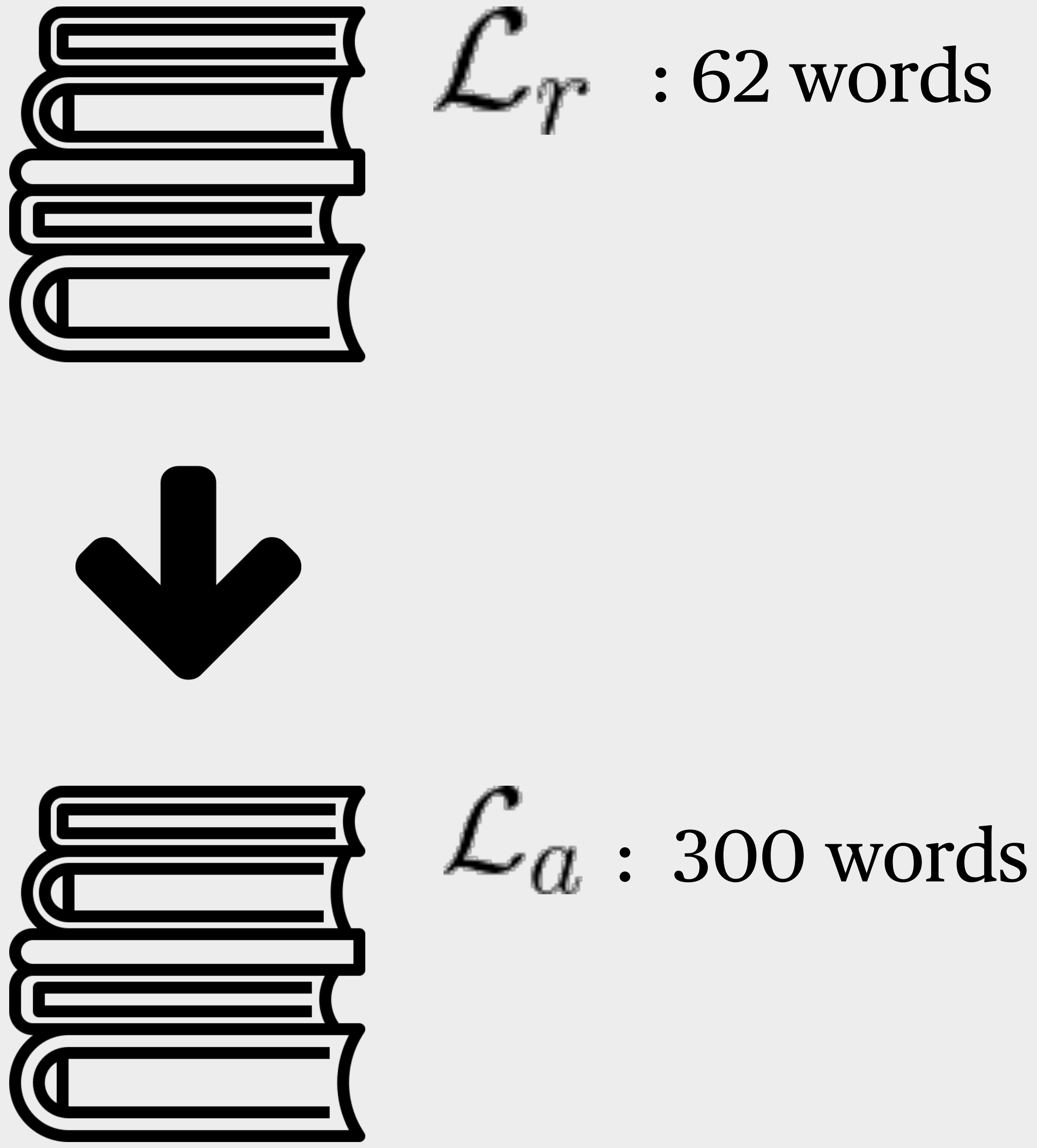
We apply a vector of norm 1, whose components are extracted randomly from a standardized **normal distribution**.
Thus obtaining:

$$\mathbf{w_{i,j}} = \mathbf{w_i} + \mathbf{n}$$

If the following condition is satisfied:

$$MostSimilar(\mathbf{w_{i,j}}) = \mathbf{w_i}$$

The vector is added to the lexicon, with the same label as the *parent*.



Score induction

Main focus of this project

Objective:

Propagate the labels in the lexicon to all the vectors inside each embedding $\mathbf{C}_i$.

Result:

For each embedding $\mathbf{C}_i$, we obtain a **labeled embedding:**

$$\mathbf{C}_i^\ell = \{(\mathbf{w_j}, L_j)\}_{j \in |V_i|}$$

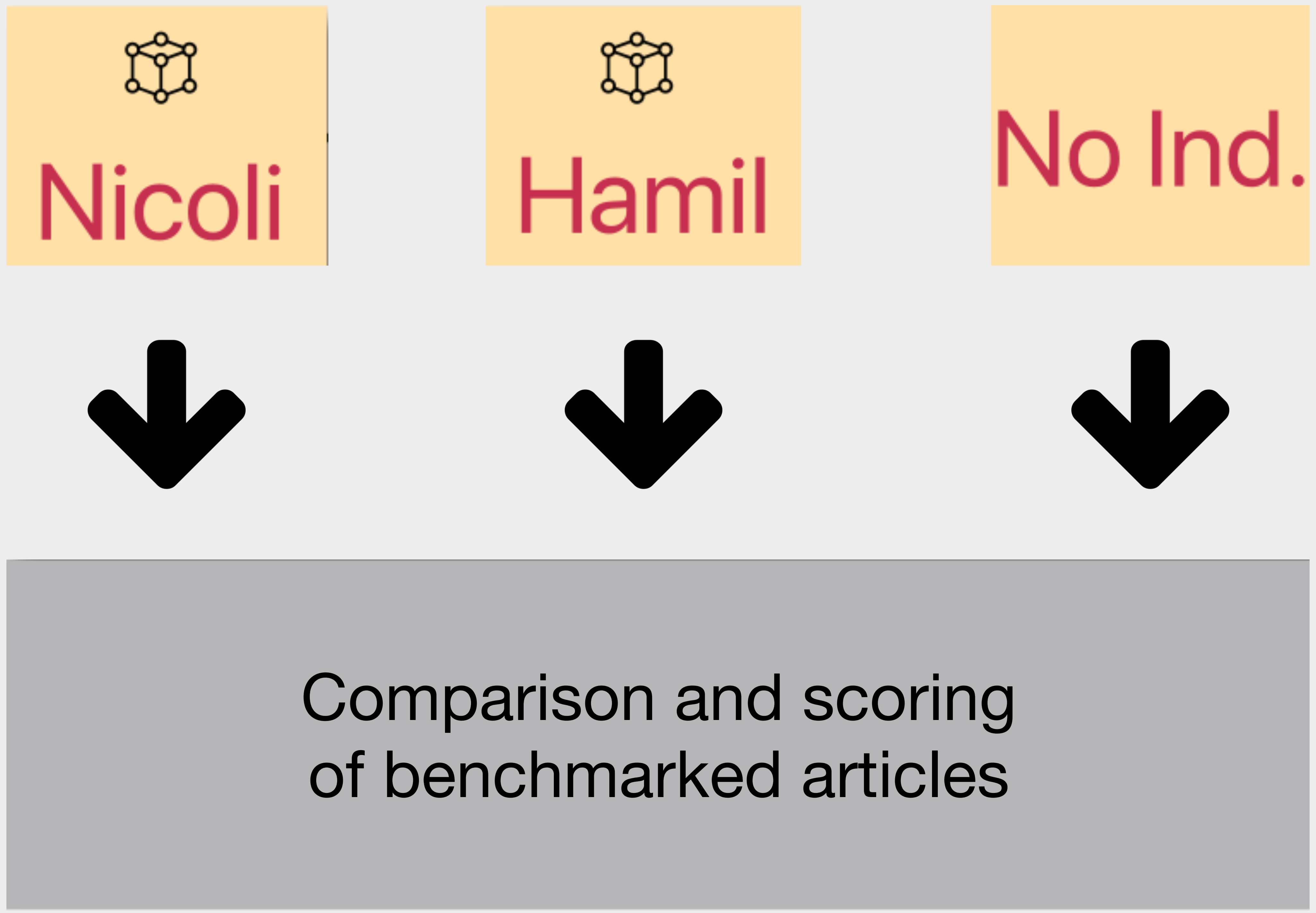
Where the label has value between 0 and 1, where **1 is for max subjectivity.**

We can also the define a **labeled vocabulary:**

$$V_i^\ell = \{(w_j, L_j)\}_{j \in |V_i|}$$

Three different methods:

Nicoli **Hamilton** **No-induction**



Score induction

Nicoli's Method

Overview:

The score induction process is framed as a **machine learning problem**.

Procedure:

We used a logistic regression.

- $\mathbf{w_i}$, word-vector in a certain embedding space, $i < m$
- *y,* its subjectivity score
- **W,** vector of weights

We optimize the cross-entropy loss function:

$$L(\mathbf{W}) = \sum_{j=1}^{m} \log(1 + e^{\mathbf{W} \cdot \mathbf{w_j}})$$

Perks:

- Fairly easy to set-up
- Fast-training
- Flexible

Disadvantages:

- Requires a consistent amount of labeled words (lexicon)
- Requires manual tuning

Score induction

Hamilton's Method

Overview:

Based on **random-walks** on proximity graphs.

Procedure basics:

- $\mathbf{p}^{(i)} \in \mathbb{R}^{|V_i|}$ vector of labels, initialized as: $\mathbf{p}^{(0)} = (\dots, \frac{1}{|V|}, \dots)$
- $E \in \mathbb{R}^{|V_i| \times |V_i|}$ matrix of distances between word-vectors.
- $\mathbf{s} \in \mathbb{R}^{|V_i|}$ lexicon labels vector.
- β parameter that controls local/global consistency

The vector **p** is updated iteratively until convergence, as:

$$\mathbf{p}^{(i)} = f(E, \beta, \mathbf{s}, \mathbf{p}^{(i-1)})$$

Perks:

- Very robust
- Can work with a small lexicon (20 words)
- Only one parameter

Disadvantages:

- Heavy on computation resources and time

Score induction

Hamilton's Method

Overview:

Based on **random-walks** on proximity graphs.

Procedure basics:

- $\mathbf{p}^{(i)} \in \mathbb{R}^{|V_i|}$ vector of labels, initialized as: $\mathbf{p}^{(0)} = (\dots, \frac{1}{|V|}, \dots)$
- $E \in \mathbb{R}^{|V_i| \times |V_i|}$ matrix of distances between word-vectors.
- $\mathbf{s} \in \mathbb{R}^{|V_i|}$ lexicon labels vector.
- β parameter that controls local/global consistency

The vector **p** is updated iteratively until convergence, as:

$$\mathbf{p}^{(i)} = f(E, \beta, \mathbf{s}, \mathbf{p}^{(i-1)})$$

Perks:

- Very robust
- Can work with a small lexicon (20 words)
- Only one parameter

Disadvantages:

- Heavy on computation resources and time

(VERY HEAVY!!!)

Process Name	Memory
python3.7	180.77 GB

Score induction

Notes on implementations

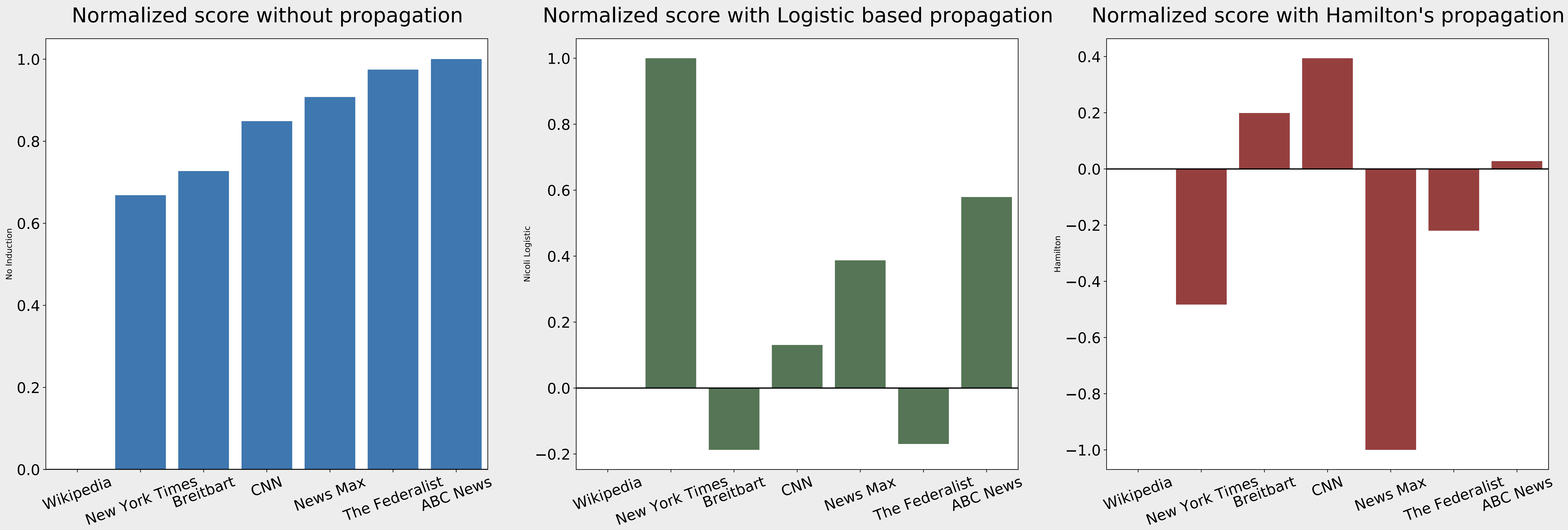
Both implementations needed to be adapted:

- Dott. Nicoli's code only worked **for two models.** (Fork, modify, merge)
- Prof. Hamilton's code was written in python2, many parts were **deprecated,** also:
    - Implementation **not agnostic to words**
    - Did not support the **lexicon augmentation** process

Score induction

Results

Score: mean subjectivity score on each V^l_i, normalized to the Wikipedia one



Without induction, Wikipedia is the most objective of all.

Both other methods, **Nicoli's Logistic Propagation** and **Hamilton's Propagation**, do not match the initial structure.

Score induction

Results

Base hypothesis: Wikipedia has the most objective language.

True before score induction, **False** after.

The propagation might have some *undesired* effects.

friend

Newssite	Hamilton	Nicoli - Logistic
Wikipedia	0.57	0.0000
Breitbart	0.21	0.0353
New York Times	0.47	0.0002
News Max	**0.94**	**0.3172**
CNN	0.18	0.0088
The Federalist	0.16	0.1715
ABC News	0.43	0.0005

snow

Newssite	Hamilton	Nicoli - Logistic
Wikipedia	0.11	0.17
Breitbart	0.50	0.22
New York Times	0.11	0.07
News Max	0.90	0.16
CNN	0.50	0.24
The Federalist	0.35	0.65
ABC News	0.53	0.87

Performance on benchmark articles

Can it spot subjective/objective texts?

Benchmarking:

Manually classified articles (~50).

Manually classified paragraphs (~200).

Values:

- **1** : subjective
- **0** : objective
- **-1** : uncertain

Scoring:

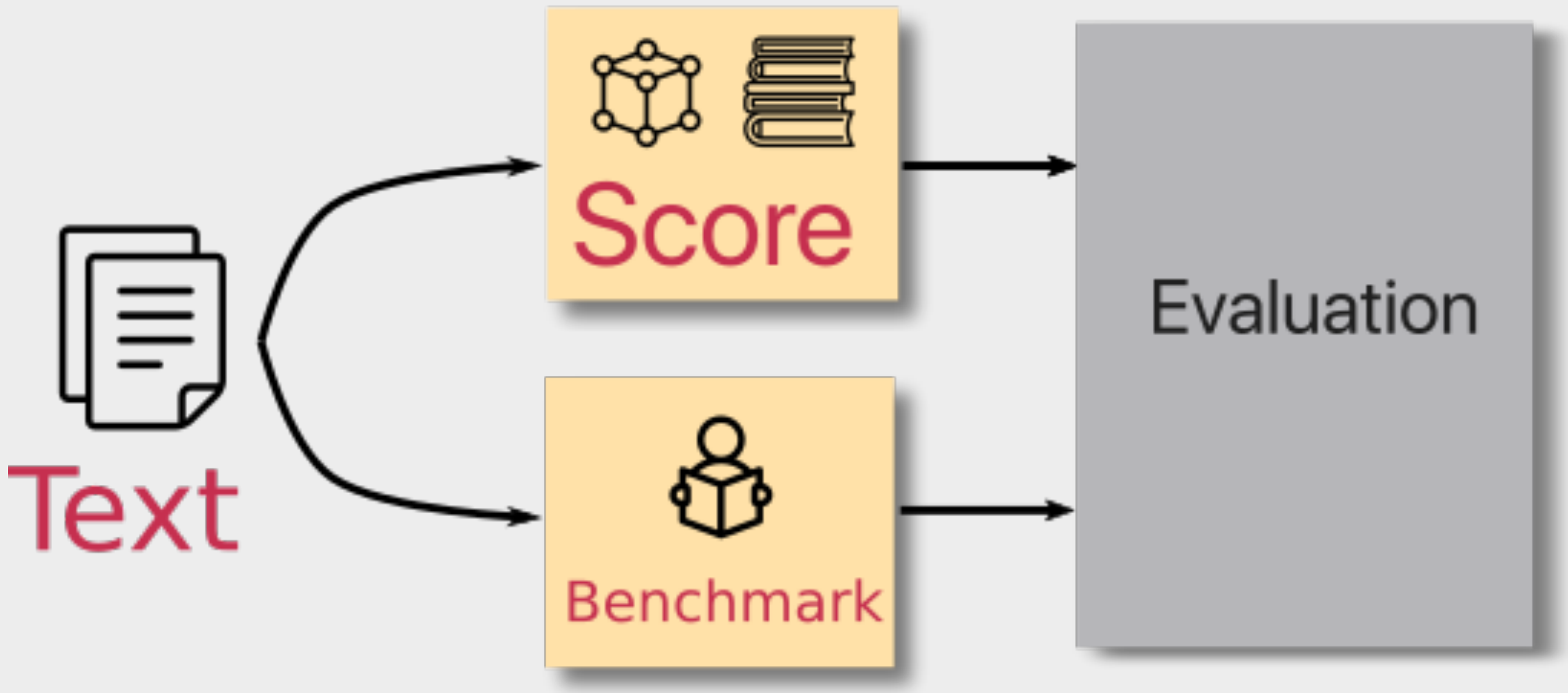
Collection of word-items T: $T = \{w_i, w_j, \ldots, w_h\}$

Labeled vocabulary V_i: $V_i = \{(w_j, L_j)\}_{j \in |V_i|}$

Mean **subjectivity score**: $< L_T >= \frac{1}{|T|} \sum_{j}^{w_j \in T} L_j$

Performance on benchmark articles

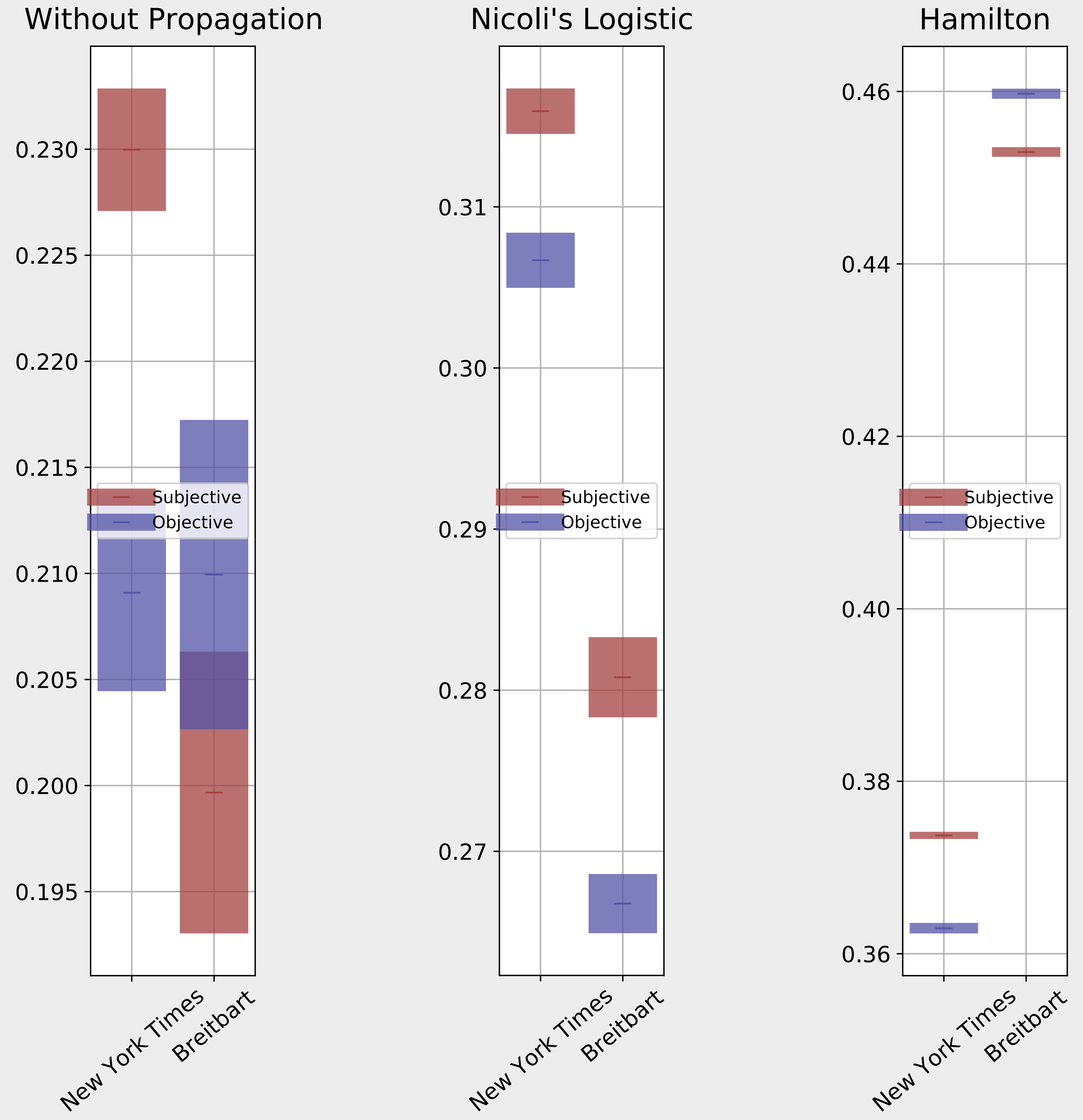
Can it spot subjective/objective texts?



Performance on benchmark articles

Results

Mean subjectivity score for benchmark articles, classified as subjective and objective, for 3 score induction method.



On bechmarked text classified as *subjective*, both Hamilton's and Nicoli's method attribute (in mean) a slightly higher score, with Nicoli's method being more consistent. The distinction is fairly weak, **however present**.

Conclusions

- Some undesired effects during the propagation, probably due to lexicon composition.
- Propagation methods catches some aspects of subjectivity inside articles, as mean scores suggest.
- Nicoli's method seems consistent in results, despite primitive and simple scoring method.
- Hamilton's method result's are more robust, however it fails to recognize subjectivity in some context.
- Meaningful baseline for future improvements.

How to Improve:

- New, ad-hoc **lexicon**: CROWD-SOURCING, SOCIAL AGREEMENTS
- Refined **benchmarking**: higher number of scorers, attenuate personal biases.
- **Contextual** word embedding. Example: *Good*.

  Good as an adjective, *Good* as a noun

  Inside the SubjectivityLexicon, Good is labeled as objective.
- Re-implementation of **Hamilton's framework**: word-agnostic
- Evaluating robustness: **bootstrapping** procedure
- The problem of direct and indirect quotes.

References:

- *Bianchi, F., Di Carlo, V., Nicoli, P. and Palmonari, M., 2020. Compass-Aligned Distributional Embeddings For Studying Semantic Differences Across Corpora. [online] arXiv.org. Available at: <https://arxiv.org/abs/2004.06519> [Accessed 10 September 2020].*
- *Hamilton, W., Clark, K., Leskovec, J. and Jurafsky, D., 2020. Inducing Domain-Specific Sentiment Lexicons From Unlabeled Corpora.*
- *Nicoli, P., Palmonari, M., Bianchi, F., 2019. Framework for Comparison of Corpus-Specific Models*

Thanks for the attention

Zipf measure:

Definition

:

$$\zeta_D(w) = \log_{10}\left(\frac{\#w + 1}{|V|_M + |D|_M}\right) + 3$$

- *#w* is the frequency of the word inside the corpus D
- |V| is the dimension of the vocabulary ($|_M$ indicates the unit of a million words)
- |D| is the dimension of the corpus (or slice)

Features:

- Widely used in literaure
- Standardized across various corpora and vocabularies of different dimension

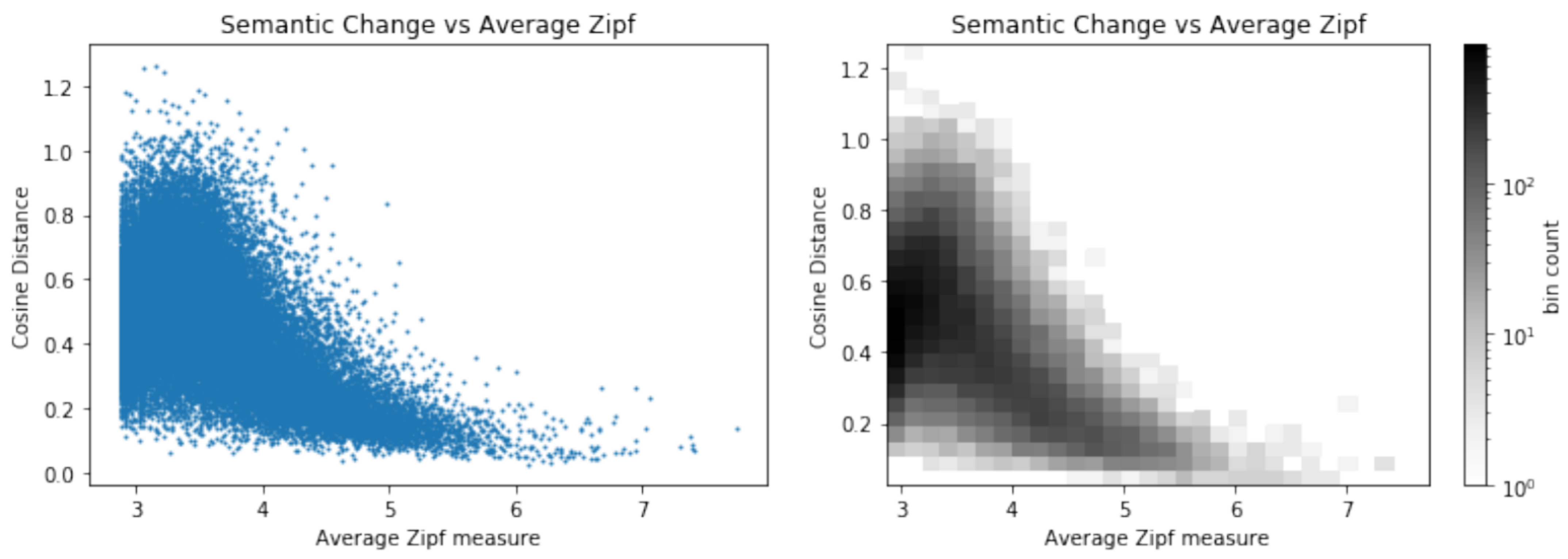


Figure 3.11: Scatterplot and 2D Histogram of frequency-change relation for `CADE` slices

Ragionamenti Just News

- Wikipedia sì: permette un ottimo confronto alla fine
- Slate no: troppi pochi articoli
- I processi sono tutti model dependent
- Wikipedia potrebbe modificare in direzione indesiderata tutto l'embedding, ma è utile per il confronto finale quindi lo teniamo.
- Nicoli è molto veloce da addestrare, ma dipende dalla Data Augmentation e dall'algoritmo di ML scelto.
- Hamilton è molto dispendioso computazionalmente sia di tempo che memoria. Per come è implementato, non è agnostico alle parole, e quindi non supporta un lessico arricchito (data augmentation).
- Dire che abbiamo modificato codice di Nicoli
- Dire che abbiamo tradotto Hamilton da Python2 a Python3
- Per i benchmark, sarebbe meglio fare media di score su tante persone diverse, per eliminare bias
- Abbiamo cercato per Politica, e abbiamo notato come il termine sia bello diverso tra giornali
- Far presente il bias personale
- Determinare miglior lessico, magari facendo scan a partire da paragrafo classificati
- Usiamo sia i valori "certi" (1 o 0) sia le probabilità per confrontare Nicoli e Hamilton
- Abbiamo provato diversi thresholds per Hamilton, ma difatti non cambia nulla nell'ordine.
- Fare esempi per CADE
- Fare esempi sul lessico (magari condivisa da entrambi). Dire che non tutte le parole sono condivise dai due lessici annotati.
- Fare esempi sulle propagazioni, sia positivi che negativi. Far vedere che parole uguali hanno classificazioni diversa in diversi giornali.