

Lingvistica Matematica si Computationala

Liviu P. Dinu,

ldinu@fmi.unibuc.ro

University of Bucharest

Center for Computational Linguistics,

Faculty of Mathematics and Computer Science

nlp.unibuc.ro



Quantitative aspects of natural languages

History

- Mathematical linguistics, as the study of quantitative and formal aspects of language phenomena (Marcus, Nicolau, Stati 1971), has developed simultaneously in Europe and USA in the late fifties.
- Quantitative aspects of language were investigated long before the algebraic ones.

History

- There are records of all letters and diacritic symbols of Italian since the XIVth - XVIth century;
- The Morse alphabet was inspired by the different statistic behavior of letters;
- In the XIX-th century frequency dictionaries were edited
- The beginning of the XX-th century brings the first linguistically motivated studies which resulted in introducing the Markov models

Overview

The main goal of this presentation is to investigate the quantitative and formal behaviour of Romanian syllables

The results are compared with results of similar studies for different languages.

Syllable and Syllabification

- Syllable: the first linguistic units learned during the acquisition process.
- The children's first mental representation is syllabic in nature, the phonetic representation occurs later.
- Applications: poetics, logopedy, T2S, readability, text comprehension, speech production models, etc.

Motivation

- The formal, quantitative or cognitive study of syllable has various potential application in fields such as: speech recognition, automatic transcription of spoken language into written language, language acquisition, etc.
- A rigorous study of the structure and characteristics of the syllable is almost impossible without the help provided by a complete data base of the syllables in a given language.

Method (1)

- We used the DOOM dictionary, which contains $N_{words} = 74.276$ words
- We semi-automated syllabified their lexical (not phonological) form
- We extracted a series of quantitative and descriptive results for the Romanian syllables
- We investigate the behaviour of Romanian syllables w.r.t. the laws of Chebanow, Menzerath and Fenk.

Method (2)

- Recently, DOOM was completed with all inflectional forms. All these words were manually syllabified.
- However, in the spoken and literary language, the using of words is not equal.
- We will use a corpus of 5 Romanian writers to investigate the behavior of syllables: Mateiu Caragiale, Radu Albala, Ion Iovan, Stefan Agopian, Eugen Balan.
-

Quantitative and descriptive results(1)

- Total no. of type syllabals is $N_{\text{Stype}} = 6496$
- Total no. of token syllables is $N_{\text{stoken}} = 273261$
- Average length of a word measured in syllables is $L_{\text{wordssyl}} = N_{\text{stoken}} / N_{\text{words}} = 273261 / 74276 = 3.678$
- The total no. of letters is $N_{\text{letters}} = 32702$
- The average length of a word measured in letters is $L_{\text{wordslet}} = N_{\text{letters}} / N_{\text{words}} = 32702 / 74276 = 0.440$

Quantitative and descriptive results

- The average length of the token syllables measured in letters is:

$$L_{syltoken} = N_{letters} / N_{token} = 632706 / 273261 = 2.315$$

- The average length of a type syllable measured in letters is:

$$L_{syltype} = N_{letters} / N_{styp} = 24406 / 6496 = 3.757$$

Quantitative and descriptive results

- The number of consonant-vowel structures which appear in the syllables is 56.
- the most frequent consonant-vowel structures are
 - a) for the *type syllables*: *cvc* (22%), *ccvc* (14%), *cvcc* (10%)
 - b) for the *token-syllables*: *cv* (53%), *cvc* (17%), *v* (8%), *ccv* (6%), *vc* (4%), *cvv* (2%), *cvcc* (2%).
- It is remarkable that these last 7 structures (i.e. 12% of the 56 structures) cover approximately 95% of the total number of the existent syllables.

Quantitative and descriptive results

- The most frequent 50 syllables (i.e. 0,7% of the syllables number *NStype*) cover 50,03% of *NStoken*
- The most frequent 200 syllables cover 76% of *NStoken*
- The most frequent 400 cover 85% of *NStoken*
- The most frequent 500 syllables (i.e. 7,7 % of *NStype*) cover 87% of *NStoken*.
- Over this number, the percentage of covering rises slowly.

Quantitative and descriptive results

- The first 1200 syllables in their frequency order cover 95% of N_{Token} .
- 2651 syllables of N_{Type} occur only once (*hapax legomena*).
- 5060 syllables (i.e. 78%) of N_{Type} occur less than 10 times. These syllables represent 11960 syllables (4% of N_{Token}).

Quantitative and descriptive results

The results are similar to results for different languages:

- For Dutch the first 500 type syllables, ordered after their frequency, (5% of the total number of type syllables), cover approximately 85% of the total number of token syllables.
- For English, the result is similar, the first 500 syllables cover approximately 80% of the total number of the token syllables. This results support the mental syllabary thesis.

How many syllables are in Romanian?

How many are used? (Qualico 2012)

Name	Dict.	M. Car.	Agopian	Albala	Iovan	Eminescu
#words	525.528	6562	15.225	7089	24.627	11.029
#Syl_type	2.229.021	51.560	540.777	71.555	336.124	258.761
#Syl_token	8895	1929 (21%)	2688 (30%)	1945 (21%)	3456 (38%)	2653 (29%)
%Cov50syl	51.44	53.33	59.1	58.57	50.75	53.43
%Cov200syl	78.96	80.62	84.74	82.69	78.59	78.46
%Cov500syl	88.8	92.17	94.36	93.17	90.29	91.37
#Syl=1	2716	606	400	547	677	424
#Syl<10	5478	1414	1369	1350	2018	1441

The behaviour of Romanian syllables w.r.t. laws of minimum effort

- Chebanow's Poisson type law expresses the correlation between the words' length (in syllables) and their occurrence's probability.
- Denoting by $F(n)$ the frequency of a word having n syllables and by $i = \frac{\sum nF(n)}{\sum F(n)}$ the average length of words (measured in syllables), Chebanow proposed the following law between the average i and the probability of occurrences $P(n)$ of the words having n syllables:

$$P(n) = \frac{(i-1)^{n-1}}{(n-1)!} e^{1-i}$$

The behaviour of Romanian syllables w.r.t. laws of minimum effort

- We checked Chebanow's law on the data base of Romanian syllables, obtaining a strong similarity between the Poisson's distribution and the distribution of words length (in syllables):

$$P(n) = \frac{2.678^n}{n!} e^{-2.678}$$

Chebanow's law

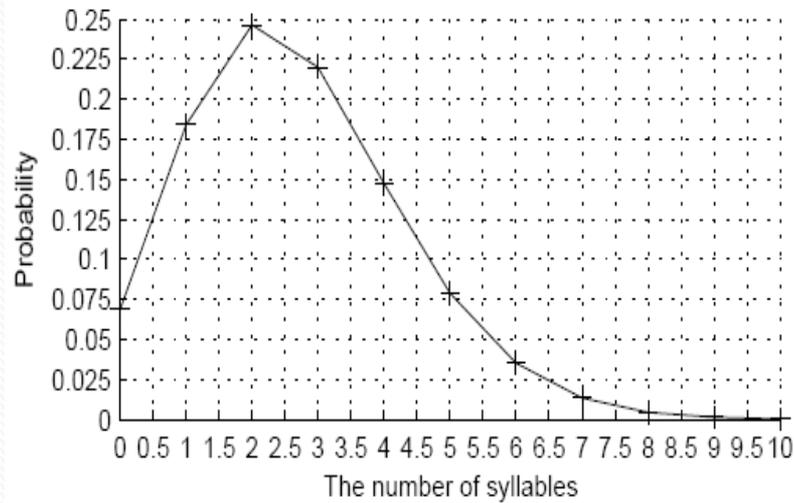


Fig. 1: *The Poisson distribution of length of words (parameter equal to 2.678)*

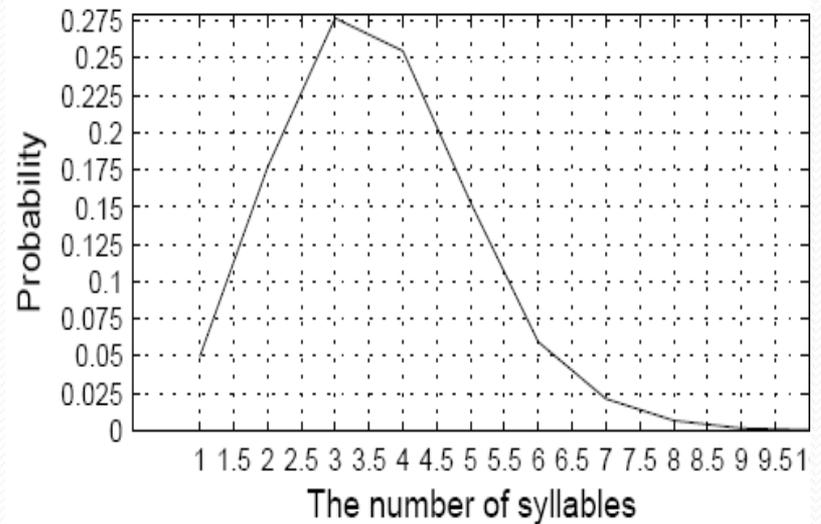


Fig. 2: *The probability distribution of the length of words*

Menzerath's law

- Menzerath's law expresses a negative correlation between the length of a word in syllables and the lengths in phonemes of its constitutive syllables. Fig. 3 shows that the law is satisfied.

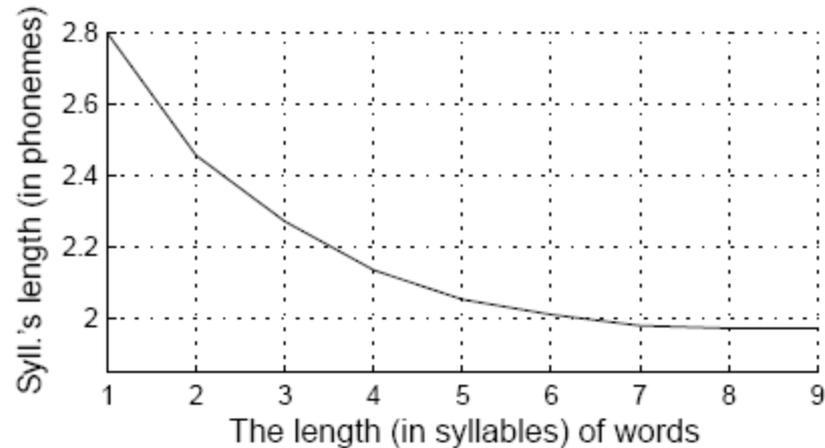


Fig. 3: *The Menzerath's law: The more syllables in a word, the smaller its syllables*

Fenk's law

- Fenk observed that the bigger the length of a word, measured in phonemes, the lesser the length of its constituent syllables, measured in phonemes. We checked this correlation and Fig. 4 confirms Fenk's law:

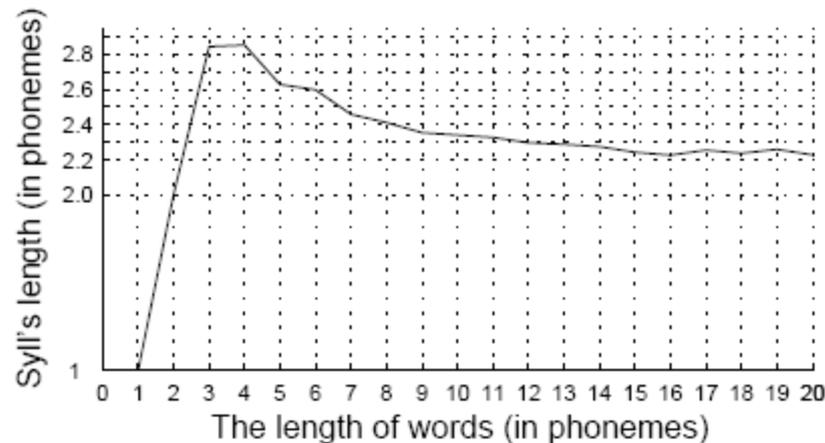


Fig. 4: *The Fenk's law: The more phonemes in a word, the lesser phonemes in its syllables*

Fenk's laws (2)

- The bigger the average length of sentences, measured in syllables, the lesser the average length of syllables, measured in phonemes.
- There is a negative correlation between the length of sentences, measured in words, and the length of the words, measured in syllables.

Optimal values

- Determining the optimal values of the length of sentences and of the words depending on the certain groups of readers may prove to be very useful in practical application.
- By optimum value we understand the value for which the level of comprehensibility is the biggest for a class of readers.

Optimal values

- Knowing this value should be especially important for the teachers and for publishers who print text books.
- The main conclusion of (Eltis and Mikk, 1996) is that, for a good understanding of a text, the length of sentences in the text must be around the average length of sentences

Optimal values

Table 1. Optimal length of words (Bamberge, Vanecek, 1984-cf. Elts and Mikk, 1996):

The length of words	The reader's level								
	4	5	6	7	8	9	10	11	12
in syllables	1.62	1.68	1.72	1.8	1.88	1.91	1.99	2.08	2.11
in letters	6.16	6.39	6.39	6.84	7.15	7.26	7.57	7.91	8.02

Another experiment on 98 students which were given 48 texts, produced the following optimal values (Table 2):

Table 2.

	Level 8	Level 10
Optimal length of words, measured in letters	8.53	8.67
Optimal length of sentences, measured in letters	71.5	76

MULTUMESC!

Lingvistica Matematica si Computationala

Liviu P. Dinu,

ldinu@fmi.unibuc.ro

University of Bucharest

Center for Computational Linguistics,

Faculty of Mathematics and Computer Science

nlp.unibuc.ro



FORMAL APPROACHES OF SYLLABLES

Syllabification formal approaches

- The linguists refused to accord to the syllable the status of structural unity of the language, as opposed to the phoneme and the morpheme.
- As a consequence, the formal models of the syllable failed to equal the complexity of upper units.
- Generally, based on rewriting:
 - Bird and Ellison (1994): based on automata,
 - Kaplan and Kay (1994): based on regular expressions,
 - *Karin Muller (2002): based on probabilistic CF.*

Contextual Approaches (Grammars, Cicing 2005, Fund. Inf.)

- In many languages, the syllabification of a word w depends on the partition of that word in three strings $w = x_1x_2x_3$ and all three strings affect the syllabification.
- Ex:
 - Rules like "if we have a consonant between two vowels then the syllabification is made before consonant": be\$re; a\$bi\$li\$ta\$re can be formalized:

$$xv_1cv_2y \Rightarrow xv_1$cv_2y$$

where \$ is the syllabification symbol, v_1 and v_2 are two vowels and c is a consonant.

A contextual-based approach

- Analogy between the syllabification of the words and the language generated by a contextual grammar.
- Formalization of the syllabification process, using an extension of total contextual grammars.
- Sequential manner (a derivation step implies only a cut; e.g., castravete -> castra\$vete).
- Restrictions which preserve the sequentiality, but determine a syllable at each derivation step (e.g., cas\$travete).

Contextual grammars. Definition

Definition 1. (Paun, 1997) *A total Marcus contextual grammar is a system $G = (V, A, C, \varphi)$, where V is an alphabet, A is a finite language over V (the axioms), C is a finite subset of $V \times V$ (the contexts) and $\varphi : V \times V \times V \rightarrow P(C)$ (the choice function)*

The language generated by G is:

$$L(G) = \{x \in V^* \mid w \xrightarrow{*} x, \text{ for } w \in A\},$$

where " $\xrightarrow{*}$ " is the reflexive and transitive closure of " \rightarrow ", given by:

$x \rightarrow y$ iff $x = x_1x_2x_3$, $y = x_1ux_2vx_3$ for $x_1, x_2, x_3 \in V^*$, and $\langle u, v \rangle \in C$ such that $\langle u, v \rangle \in \varphi(x_1, x_2, x_3)$.

A CONTEXTUAL APPROACH TO THE SYLLABLE

- Consider the Romanian alphabet $RO = \{a, \text{ă}, \text{â}, \text{ , } b, c, d, e, f, g, h, i, \text{î}, j, k, l, m, n, o, p, q, r, s, \text{ș}, t, \text{ț}, u, v, w, x, y, z\}$ and consider a nontrivial partition $RO = V_o \cup C_o$, where $V_o = \{a, \text{ă}, \text{â}, \text{ , } e, i, \text{î}, o, u, y\}$ and $C_o = \{b, c, d, f, g, h, j, k, l, m, n, p, q, r, s, \text{ș}, t, \text{ț}, v, w, x, z\}$, i.e., V_o and C_o are the Romanian vowels and the Romanian consonants, respectively.
- We will say that *a word over RO is regular if it contains no consecutive vowels.*

- In order to generate all the Romanian syllables which appear in regular words, and only them, we propose the grammar $G_{syl} = (V_{syl}, A_{syl}, ;C_{syl}, \varphi_{syl})$, whose components are:
- $V_{syl} = RO \{\$, \}$, where "\$" is a new symbol that is not in RO ; "\$" is the *syllable boundary marker*
- A_{syl} is the set of the regular words over RO . A_{syl} is finite since the set of all words in a natural language is finite.

$$3. C_{syl} = \{ \langle \lambda, \lambda \rangle, \langle \lambda, \$ \rangle, \langle \$, \lambda \rangle \}$$

4. φ_{syl} is defined based on the syllabification rules of the Romanian languages (DOOM, 1982).

- a. $\varphi_{syl}(\alpha v_1, c, v_2 \beta) = \{ \langle \$, \lambda \rangle \}$ if $\alpha, \beta \in V_{syl}^*$, $c \in Co$, $v_{1,2} \in Vo$ (i.e. in the case of a consonant between two vowels, the syllabification is done before the consonant)
- b. $\varphi_{syl}(\alpha v_1, c_1 c_2, v_2 \beta) = \{ \langle \$, \lambda \rangle \}$ if $\alpha, \beta \in V_{syl}^*$, $c_1 c_2 \in \{ch, gh\}$, or $(c_1, c_2) \in \{b, c, d, f, g, h, p, t\} \times \{l, r\}$
- c. $\varphi_{syl}(\alpha v_1 c_1, c_2, v_2 \beta) = \{ \langle \$, \lambda \rangle \}$ if $\alpha, \beta \in V_{syl}^*$, $c_1 c_2 \notin \{ch, gh\}$, and $(c_1, c_2) \notin \{b, c, d, f, g, h, p, t\} \times \{l, r\}$
- d. $\varphi_{syl}(\alpha v_1 c_1, c_2 c_3, v_2 \beta) = \{ \langle \$, \lambda \rangle \}$ if $\alpha, \beta \in V_{syl}^*$, $c_1 c_2 c_3 \notin \{lpt, mpt, mp t, nc s, nct, nc t, ndv, rct, rtf, stm\}$
- e. $\varphi_{syl}(\alpha v_1 c_1, c_2, c_3 v_2 \beta) = \{ \langle \lambda, \$ \rangle \}$ if $\alpha, \beta \in V_{syl}^*$, $c_1 c_2 c_3 \in \{lpt, mpt, mpt, nc s, nct, nct, ndv, rct, rtf, stm\}$
- f. $\varphi_{syl}(\alpha v_1 c_1, c_2 c_3 c_4, v_2 \beta) = \{ \langle \$, \lambda \rangle \}$ if $\alpha, \beta \in V_{syl}^*$, $c_2 c_3 c_4 \notin \{gst, nbl\}$
- g. $\varphi_{syl}(\alpha v_1 c_1 c_2, c_3 c_4, v_2 \beta) = \{ \langle \$, \lambda \rangle \}$ if $\alpha, \beta \in V_{syl}^*$, $c_2 c_3 c_4 \in \{gst, nbl\}$
- h. $\varphi_{syl}(\alpha v_1 c_1 c_2, c_3 c_4 c_5, v_2 \beta) = \{ \langle \$, \lambda \rangle \}$ if $\alpha, \beta \in V_{syl}^*$, $c_1 c_2 c_3 c_4 c_5 \in \{ptspr, stscr\}$
- i. $\varphi_{syl}(x_1, x_2, x_3) = \{ \langle \lambda, \lambda \rangle \}$, otherwise

The language generated by G_{syl} is:

$$L(G_{syl}) = \{ x \in V_{syl}^* \mid w \xrightarrow{*} x \text{ for } w \in A_{syl} \}$$

and it contains all possible ways of syllabification regular words (for example, the language contains the word *lingvistica* and all its possible syllabifications: *lin\$gvistica*, *lingvis\$tica*, *lingvisti\$ca*, *lingvis\$ti\$ca*, *lin\$gvisti\$ca*, *lin\$gvis\$tica*, *lin\$gvis\$ti\$ca*).

Syl

We introduce the set Syl as follows:

$$Syl = \{x \in (V_{syl} \setminus \$)^+ \mid \exists \alpha, \beta \in (V_{syl})^* \text{ such that } \alpha x \beta \in L(G_{syl}) \text{ and } x \Rightarrow y \text{ implies } x = y\}$$

This definition allows us to define the syllable as it follows:

Definition 2. A segment $syl \in \{Co \cup Vo\}^*$ is a syllable iff $syl \in Syl$.

Remark 3. In most of the natural languages there are words which have different syllabifications. For Romanian words, the only words which can have two different syllabifications are the words ending in "i" (e.g. *ochi* (noun) and *o\$chi* (verb)) (Petrovici, 1934). The syllabification of such a word depends on whether the final "i" is stressed or not. If the final "i" is stressed, the rules a)-i) are applied, else the final "i" is considered as a consonant and then the same rules are applied.

Vowel vs semivowel

- **Remark 4.** *Inside a graphical non regular word, in a sequence of 2, 3, 4 or 5 vowels it is difficult to distinguish between a vowel and a semivowel. In order to cut into syllables such a word we have tried to extract a set of rules based on the context in which the sequence appears.*



V vs SV

- Thus, we notice that the same group of vowels has an identical behavior (regarding the syllabification of words which contains it) depending on certain letters which precede and/or succeed it (Dinu, 1997).
- Once we have founded a set of rules which characterize the behavior of a sequence of vowels, we use it to extend the grammar G_{syl} . We have obtained a set of rules which characterize the behavior of some sequences of vowels, the rest of them being under construction.

- **Remark 5.** For a word w there may be two different decompositions of w , $w = x_1x_2x_3$ and $w = y_1y_2y_3$, such that using direct derivation we can obtain two different words, $w = x_1x_2x_3 \ x_1ux_2vx_3 = w_1$ and $w = y_1y_2y_3 \ y_1uy_2vy_3 = w_2$, with $w_1 \neq w_2$.
- In other words, the syllabification may be done anywhere inside the word, the only condition being that the cutting should be correct.

Example

- **Example 1.** Consider the word *lingvistica*. We may have the follow direct derivations:
 - *lingvistica* $lin\dot{s}gvistica$
 - *lingvistica* $lingvisti\dot{s}ca$
- To avoid these situations, we shall impose that the cutting to be always done at the leftmost position.

Leftmost derivation

- For this purpose we have considered a series of constraints of the derivation relation defined with respect to a total contextual grammar, called *total leftmost derivation*.
-
- By using it, contexts are introduced in the leftmost possible place.

Mental syllabary

- Junction with the mental syllabary model proposed by Levelt and Indefrey (an intermediate step in speech production is the syllabification).
- May the cost of syllabification operation be reduced?
- Mitchell's parallel metaphor ("Machine Learning",1997):
"many brain activities can be processed in a parallel manner"
- Is it possible to propose a parallel syllabification model?

Parallel approach via INS-DEL

- Insertion grammars: strings are inserted in a context (Galiukschov, 1981):

$tuvw \rightarrow tuxvw$ iff (u,x,v) is a production rule.

- Parallel derivation: we introduce a parallelism in a double sense:
 1. on one hand, we can insert **more than one** string and,

- On the other hand, a **context** that selects an inserted string **can interact** with a context that selects other inserted string, such that the **prefix** of one context can be the **suffix** of the other.

INS_pm (CICLING 2005, FI 2008)

- Maximum parallel derivation (INS_pM): in a derivation step we insert the **maximum** possible number of strings.
- INS_pM are incomparable but not disjoint to Context Free Languages, but are included in Context Sensitive Languages.
- INS_pM are incomparable but not disjoint to TC and ICC languages.
- Efficient syllabification of words: one step.
Rules: $(v, \$, cv)$, $(vc_1, \$, c_2v)$, etc...
E.g. : `lingvistica->lin$gvis$ti$ca`

Example

Example 2. Consider the word *lingvistica*. We may have the following parallel derivations:

- A parallel derivation: $lingvistica \Rightarrow lin\$gvis\$ti\ca , where:

a. $i=1: w_1x_1w_2 = \alpha_1u_1x_1v_1\beta_1$, with $(u_1, x_1, v_1) \in C_4$:

$\alpha_1 = l, u_1 = in, x_1 = \$, v_1 = gvi, \beta_1 = sti$

b. $i=2: w_2x_2w_3 = \alpha_2u_2x_2v_2\beta_2$, with $(u_2, x_2, v_2) \in C_3$:

$\alpha_2 = gvist, u_2 = i, x_2 = \$, v_2 = ca, \beta_2 = \lambda$

- Maximal Parallel derivation: $lingvistica \Rightarrow lin\$gvis\$ti\ca

a. $i=1: w_1x_1w_2 = \alpha_1u_1x_1v_1\beta_1$, with $(u_1, x_1, v_1) \in C_4$:

$\alpha_1 = l, u_1 = in, x_1 = \$, v_1 = gvi, \beta_1 = s$

b. $i=2: w_2x_2w_3 = \alpha_2u_2x_2v_2\beta_2$, with $(u_2, x_2, v_2) \in C_3$:

$\alpha_2 = gv, u_2 = is, x_2 = \$, v_2 = ti, \beta_2 = \lambda$

c. $i=3: w_3x_3w_4 = \alpha_3u_3x_3v_3\beta_3$, with $(u_3, x_3, v_3) \in C_1$:

$\alpha_3 = t, u_3 = i, x_3 = \$, v_3 = ca, \beta_3 = \lambda$

Conclusion and future work

- In first part of this presentation we have presented some quantitative observations obtained from the analyse of the first data base of Romanian syllables.
- In the second part of the paper we have investigated the contextual grammars as generative models for the natural language. We introduced some constraints to the derivation relation, obtaining new contextual grammars.

Conclusion

- Using the languages generated by these grammars we proposed a contextual model of the syllable.
- From the cognitive point of view, a model based on contextual grammar seems close to the way the brain operates when it produces speech.

MULTUMESC!

Lingvistica Matematica si Computationala

Liviu P. Dinu,

ldinu@fmi.unibuc.ro

University of Bucharest

Center for Computational Linguistics,

Faculty of Mathematics and Computer Science

nlp.unibuc.ro



Syllabification and stress prediction via machine learning

Can syllabification be learned?

- Formal approaches need almost an exhaustive set of syllabification rules.
- For vowel chains we need the detection of all contexts and corresponding rules extraction.
- Can we use the force of the machine learning methods?
- Yes, we can!

Two Linguistics decision problems

1. Syllabification.
2. Stress prediction: given a word, to determine its primary stress.

Syllable boundaries

- The task of finding syllable boundaries can be straightforward or challenging, depending on the language
- Text-to-speech applications have been shown to perform considerably better when syllabication, whether orthographic or phonetic, is employed as a means of breaking down the text into units below word level.

Syllable boundaries

- Romanian syllabication is non-trivial mainly but not exclusively due to its hiatus-diphthong ambiguity.
- This phenomenon affects both phonetic and orthographic syllabication.
- The most challenging aspect is that of distinguishing between hiatus and diphthongs, as well as between the letter *i* which can surface either as a non-vocalic element, or as a proper vowel, affecting thus the syllable boundary

Our approach

- We address the task of syllable boundary prediction for Romanian words (out-of-context) as a sequence tagging problem.

Methodology

- Requirement: electronic available **resources**.
 - Solved: RoMorphoDict (Barbu, LRECo8), dataset obtained from DOOM which contains the necessary information.
- The resource relevant to our task provides a long list of word forms along with their hyphenated form with accent marking. An online version of this second data resource is available for querying at <http://ilr.ro/silabisitor/>.

Syllabification. Features and classifiers

- Baseline: rule-based implementation.
- Classifiers:
 - Linear SVM with local binary decision; features: n-grams (optim for n=4) +labels
 - CRF; features: n-grams +labels
- Labels:
 - NB: mark the syllable boundary : di-a-mant->011000
 - #NB: mark the syllable boundary + distance from the last boundary: di-a-mant->100123

Features. Example

- We will consider $n = 3$, the word *dinosaur* and the split between *o* and *s*.
- The position induces two strings, *dino* and *saur* but we are only interested in the window of radius n around the split, so we are left with *ino* and *sau*.
- Since the bag-of- n -grams features we use for the SVM loses the order, we consider adding a special marker, obtaining *ino\$* and *\$sau*.
- The n -grams of length up to 3 are: *i*, *n*, *o*, *\$*, *in*, *no*, *o\$*, *ino*, *no\$* and the analogous for the right hand side

CRF Features

- For the CRF, the feature extraction is the same, but the sparse vectorized representation is replaced with an input like:
- $1\ c[-3]=i\ c[-2]=n\ c[-1]=o\ c[-3-2]=in\ c[-2-1]=no\ c[-3-2-1]=ino\ c[1]=s\ c[2]=a\ c[3]=u\ c[12]=sa\ c[23]=au\ c[123]=sau$
- The format above is the one accepted as input by CRFsuite.

CRF (cont)

- Because the feature names include the offset, the dollar marker would provide no useful information.
- The names could just as well be arbitrary: CRFsuite cannot understand that $c[-2-1]$ means the bigram just before the split, but the values that a certain feature tends to take carry the discriminative information.

Generating training samples

- The average word in our dictionary has 9.96 characters and 4.24 syllables.
- This means that each word generates around 9 training instances (possible splits), out of which we expect around 3 to be labeled as true, and the rest as false.
- Prior to generating training instances, we split the words into a training and test set, each consisting of 262,764 words.

Training

-
- For each word of length n we generate $n-1$ instances, corresponding to each position between two letters of the word.
- Instances are labeled as positive if a hyphen can be inserted there, or negative if not.
- This tagging method is called NB labelling [2], because we label each split as boundary (B or 1) or no boundary (N or 0).

Diamant

- For example, the word di-a-mant (diamond) would be encoded as:

d i a m a n t

0 1 1 0 0 0

- A slightly more informative way of assigning labels, introduced also in [2], is to use numbered NB (#NB) tags: each split is labeled with the distance from the last hyphen:

d i a m a n t

1 0 0 1 2 3

Software

- The software we use is the scikit-learn machine learning library for the Python scientific computing environment version 0.12.1 [8].
- The library provides efficient text n-gram feature extraction using the sparse matrix implementation in SciPy6.
- We use the SVM implementation by stochastic gradient descent.
- We also used CRFsuite version 0.12 [7] for its implementation of CRF inference and training.

Syllabification. Results (TSD 2013)

• Model	Hyphen_acc.	Hyphen_F1	Word acc.
• Rule	94.31%	92.12%	60.67%
• SVM NB	98.72%	98.24%	90.96%
• SVM #NB	98.82%	98.37%	91.46%
• CRF NB	99.15%	98.83%	94.67%
• CRF #NB	99.23%	98.94%	95.25%

Conclusion

- The rules in the rule-based system can take any form and they can model very complex interactions between features.
- This model has the largest predictive power, but the rules are written by hand, therefore limiting its practicality and its performance.
- At the opposite end of the spectrum is the SVM classifier, which applies a simple linear decision rule at each point within a word, looking only at its direct context.

Conclusion (2)

- This simple approach outperforms the rule-based system by being trained on large amounts of data.
- The sequence tagger is more successful because it exploits the data-driven advantage of the SVM, while having more modeling power.
- This comes at a cost in model complexity, which influences training and test times.



**Predicting Romanian Stress
Assignment
(EACL 2014, LREC 2014)**

Romanian Stress Assignment

- ▶ Romanian is a highly inflected language with a rich morphology.
- ▶ Most linguists claim that Romanian stress is not predictable.
- ▶ The first author to challenge this view is Chitoran.
- ▶ Stress placement strongly depends on the morphology of the language.

- ▶ *RoSyllabiDict* is a dataset of Romanian words.
- ▶ 525,528 inflected forms for \sim 65,000 lemmas.
- ▶ Contains annotations for:
 - ▶ Syllabication;
 - ▶ Stressed vowel;
 - ▶ Grammatical information/type of syllabication (in case of ambiguity).

Example: *copii* (children)

`<form w="copii" obs="s." > c o (-) p (i) </form>`

Annotations:

- word: `"copii"`
- part of speech: `"s."`
- syllabication: `(-)` and `(i)`
- stressed vowel: `i`

- ▶ We discard:
 - ▶ Words which do not have the stressed vowel marked (3,430 words);
 - ▶ Compound words having more than one stressed vowel (1,668 words);
 - ▶ Ambiguous words - POS/type of syllabication (20,123 words).
- ▶ The probability distribution of the n-syllabic lemmas in *RoSyllabiDict* follows a Poisson distribution.

Syllable	%words
1 st	5.59
2 nd	18.91
3 rd	39.23
4 th	23.68
5 th	8.52

(a) counting syllables
from left to right

Syllable	%words
1 st	28.16
2 nd	43.93
3 rd	24.14
4 th	3.08
5 th	0.24

(b) counting syllables
from right to left

Table: Stress placement for *RoSyllabiDict*.

I. Baseline

- ▶ We use a "majority class" type of baseline which employs the C/V structure of the words.
- ▶ For a word in the test set, the stress pattern which is most common in the training set for the C/V structure of the word is assigned.
- ▶ If the C/V structure of the word in the test set is not found in the training set, the stress is placed randomly on a vowel.

Example: *copii* (children)

Training set

CV-CVV

1) CV-CVV (283)

2) CV-CVV (309)

3) CV-CVV (67)

Test set

copii

↓
CV-CVV

↓
CV-CVV

II. Sequential Model

- ▶ We address stress prediction as a sequence tagging problem.
- ▶ Only primary stress is accounted for, but this approach allows further development (for secondary stress).
- ▶ The cascaded model consists of two sequential models:
 1. Model for predicting syllable boundaries;
 2. Model for predicting stress placement.
- ▶ The output of the first model is used as input for the second one.
- ▶ We use averaged perceptron for parameter estimation.

a) Syllabication

- ▶ Sequential model where each node corresponds to a position between two characters.
- ▶ Labels: integer denoting the distance from the previous boundary.
- ▶ Features: character n -grams up to $n = W$ in a window of radius W around the current position.

Example: copii (children)

c o - p i i

Labels: 1 0 1 2

Features ($w = 2$): $c[-2] = c$, $c[-1] = o$, $c[-2:-1] = co$
 $c[1] = p$, $c[2] = i$, $c[1:2] = pi$.

b) Stress Placement

- ▶ Sequential model where each node corresponds to a character.
- ▶ Labels:
 - ▶ 0 - characters before the stressed vowel;
 - ▶ 1 - stressed vowel;
 - ▶ 2 - characters after the stressed vowel.
- ▶ Features:
 - ▶ Character n -grams up to $n = W$ in a window of radius W around the current position;
 - ▶ Features regarding the C/V structure of the word (C/V n -grams);
 - ▶ Binary indicators regarding the position of the current character:
 - ▶ Exactly before/after a split;
 - ▶ In the 1st/2nd/3rd/4th syllable, from left to right;
 - ▶ In the 1st/2nd/3rd/4th syllable, from right to left.

Example: copii (children)

c o - p í i

Labels: 0 0 0 1 2

Features ($w = 2$):

- a) $c[-2] = o, c[-1] = p, c[0] = i, c[1] = i$
 $c[-2:-1] = op, c[-1:0] = pi, c[0:1] = ii.$
- b) $c[-2] = V, c[-1] = C, c[0] = V, c[1] = V$
 $c[-2:-1] = VC, c[-1:0] = CV, c[0:1] = VV.$
- c) exactly before a split: **false**
exactly after a split: **false**
in the 1st/2nd/3rd/4th syllable (left → right):
false/true/false/false
in the 1st/2nd/3rd/4th syllable (right → left):
true/false/false/false

Experiments

- ▶ We use averaged perceptron training from *CRFsuite*.
- ▶ We perform grid search to optimize the 3-fold CV F_1 score of:
 - ▶ Class 1 (stressed vowel), for the stress placement model;
 - ▶ Class 0 (syllable boundary), for the syllabication model.
- ▶ $W \in \{2, 3, 4\}$, *max. number of iterations* $\in \{1, 5, 10, 25, 50\}$.
- ▶ Optimal hyperparameters: $W = 4$, *max. number of iterations* = 50.

Model	Accuracy
Baseline	0.637
Cascaded (gold syllabication)	0.975
Cascaded (predicted syllabication)	0.973

Table: Accuracy for stress prediction

Further Experiments

- ▶ We perform an in-depth analysis of the sequential model's performance.
- ▶ We account for several fine-grained characteristics of the words:
 - ▶ Part of speech: verbs, nouns, adjectives;
 - ▶ Number of syllables: 2-8, 9+;
 - ▶ Number of consecutive vowels: none, at least 2.

Category	Subcategory	# words	Accuracy	
			G	P
POS	Verbs	167,193	0.995	0.991
	Nouns	266,987	0.979	0.979
	Adjectives	97,169	0.992	0.992
Syllables	2 syllables	34,810	0.921	0.920
	3 syllables	111,330	0.944	0.941
	4 syllables	154,341	0.966	0.964
	5 syllables	120,288	0.981	0.969
	6 syllables	54,918	0.985	0.985
	7 syllables	17,852	0.981	0.989
	8 syllables	5,278	0.992	0.984
9+ syllables	1,468	0.979	0.980	
Vowels	With VV	134,895	0.972	0.972
	Without VV	365,412	0.976	0.974

Table: Cascaded model with gold (G) and predicted (P) syllabication

Conclusion

- ▶ Romanian stress is predictable.
- ▶ Syllable structure is important and helps the task of stress prediction.
- ▶ The cascaded sequential model using gold syllabication outperforms systems with predicted syllabication by only very little.
- ▶ Future work
 - ▶ Using other features (e.g., syllable n-grams);
 - ▶ Adapting the learning model to finer-grained linguistic analysis.



MULTUMESC!

Aspecte computaționale și teoretice ale morfologiei limbii române

Octavia-Maria Șulea

mary.octavia@gmail.com

Liviu P. Dinu

liviu.p.dinu@gmail.com

Vlad Niculae

vlad@vene.ro

Centrul de Lingvistică Computațională din cadrul Universității București

Cuprins

- Domeniul verbal. Crearea unui conjugator pentru limba română
- Domeniul nominal. Clasificarea substantivelor din limba română după gen
- Concluzii
- Bibliografie

Cuprins

- Domeniul verbal. Crearea unui conjugator pentru limba română
- Domeniul nominal. Clasificarea substantivelor din limba română după gen
- Concluzii
- Bibliografie

Domeniul verbal. Conjugator

- Task: dându-se infinitivul unui verb nou și cunoștințe despre conjugarea unui set de verbe, vrem ca sistemul creat să poată oferi conjugarea corectă

Domeniul verbal. Conjugator

- Task: dându-se infinitivul unui verb nou și cunoștințe despre conjugarea unui set de verbe, vrem ca sistemul creat să poată oferi conjugarea corectă
- Utilitate: NLG, NLP

Domeniul verbal. Conjugator

- Task: dându-se infinitivul unui verb nou și cunoștințe despre conjugarea unui set de verbe, vrem ca sistemul creat să poată oferi conjugarea corectă
- Utilitate: NLG, NLP
- Probleme:

Domeniul verbal. Conjugator

- Task: dându-se infinitivul unui verb nou și cunoștințe despre conjugarea unui set de verbe, vrem ca sistemul creat să poată oferi conjugarea corectă
- Utilitate: NLG, NLP
- Probleme:
 - multe serii de flective

Domeniul verbal. Conjugator

- Task: dându-se infinitivul unui verb nou și cunoștințe despre conjugarea unui set de verbe, vrem ca sistemul creat să poată oferi conjugarea corectă
- Utilitate: NLG, NLP
- Probleme:
 - multe serii de flective => multe clase paradigmatică

Domeniul verbal. Conjugator

- Task: dându-se infinitivul unui verb nou și cunoștințe despre conjugarea unui set de verbe, vrem ca sistemul creat să poată oferi conjugarea corectă
- Utilitate: NLG, NLP
- Probleme:
 - multe serii de flective => multe clase paradigmatică
 - alternanțele din radical

Domeniul verbal. Clasificări

Clasificarea tradițională

- se bazează pe vocala tematică {a, e, é, i}
- determină 4 (sau 5) “conjugări”

Domeniul verbal. Clasificări

Clasificarea tradițională

- se bazează pe vocala tematică {a, e, é, i}
- determină 4 (sau 5) “conjugări”

	Terminația de infinitiv pe conjugări			
	I	II	III	IV
Latină	ĀRE	ĒRE	ERE	ĪRE
Română	a	ea	e	i

Domeniul verbal. Clasificări

Clasificarea tradițională

- se bazează pe vocala tematică {a, e, é, i}
- determină 4 (sau 5) “conjugări”
- nu reușește să surprindă aparenta multitudine de serii flexive (aceeași conjugare prezentând mai multe serii)

Domeniul verbal. Clasificări

Clasificarea tradițională

- se bazează pe vocala tematică {a, e, é, i}
- determină 4 (sau 5) “conjugări”
- nu reușește să surprindă aparenta multitudine de serii flexive (aceeași conjugare prezentând mai multe serii)

a dansa	a mânca	a afla	a continua
dansez	mănânc	aflu	continuu
dansezi	mănânci	afli	continui
dansează	mănâncă	află	continuă
dansăm	mâncăm	aflăm	continuăm
dansați	mâncați	aflați	continuați
dansează	mănâncă	află	continuă

Domeniul verbal. Clasificări

Clasificări moderne:

- Lombard (1955)
 - corpus de 667 de verbe
 - adaugă subclase pentru –ez și –esc și ajunge la 6 clase
- Felix (1964)
 - propune 12 conjugări
- Moșil (1960)
 - 5 clase regrupate cu numeroase subclase
 - introduce metoda literelor variabile
- Guțu-Romalo (1968)
 - corpus de peste 400 de verbe
 - identifică 38 de serii de flective pe care le restrânge pe bază de omonimii specifice la 10 clase conjugale
- Barbu (2009)
 - 41 de serii de flective pe un corpus de peste 7000 de verbe

Domeniul verbal. Alternanțe

Alternanțele din radical (apofoniile)

- apar la verbele (parțial) neregulate

Domeniul verbal. Alternanțe

Alternanțele din radical (apofoniile)

- apar la verbele (parțial) neregulate
- duc la îngreunarea învățării morfologiei limbii române

Domeniul verbal. Alternanțe

Alternanțele din radical (apofoniile)

- apar la verbele (parțial) neregulate
- duc la îngreunarea învățării morfologiei limbii române

a purta

eu port- ϕ

tu por τ -i

el poart-ă

noi purt-ăm

voi purt-ați

ei poart-ă

Domeniul verbal. Alternanțe

Alternanțele din radical (apofoniile)

- apar la verbele (parțial) neregulate
- duc la îngreunarea învățării morfologiei limbii române

a purta

eu port- ϕ
tu port- ɟ -i
el poart-ă
noi purt-ăm
voi purt-ați
ei poart-ă

vs.

a curta

eu curt-ez- ϕ
tu curt-ez-i
el curt-eaz-ă
noi curt- ϕ -ăm
voi curt- ϕ -ați
ei curt-eaz-ă

Domeniul verbal. Alternanțe

Alternanțele din radical (apofoniile)

- apar la verbele (parțial) neregulate
- duc la îngreunarea învățării morfologiei limbii române

⇒ pentru verbe parțial neregulate, nu e suficient să se învețe “seria” de flexive corespunzătoare

Conjugator. Modelare Alternanțe

Moisil (1960):

- litere cu valori variabile
- de ex.: $purta = pu_0rt_0a$, unde $u_0 = \{u, oa, o\}$, $t_0 = \{t, \text{ț}\}$

Conjugator. Modelare Alternanțe

Moisil (1960):

- litere cu valori variabile
- de ex.: purta = pu_0rt_0a , unde $u_0 = \{u, oa, o\}$, $t_0 = \{t, ț\}$

Dinu et al. (2011):

- 7 clase de conjugare pentru verbele care se termină în -ta la infinitiv (aprox. 700 de verbe din corpusul de 7295)

Conjugator. Modelare Alternanțe

Moisil (1960):

- litere cu valori variabile
- de ex.: purta = pu₀rt₀a, unde u₀={u, oa, o}, t₀={t, ț}

Dinu et al. (2011):

- 7 clase de conjugare pentru verbele care se termină în -ta la infinitiv (aprox. 700 de verbe din corpusul de 7295)

Dinu et al. (2012a):

- 30 de clase de conjugare care modelează 95% din același corpus

Conjugator. Modelare Alternanțe

Moisil (1960):

- litere cu valori variabile
- de ex.: purta = pu_0rt_0a , unde $u_0 = \{u, oa, o\}$, $t_0 = \{t, ț\}$

Dinu et al. (2011):

- 7 clase de conjugare pentru verbele care se termină în -ta la infinitiv (aprox. 700 de verbe din corpusul de 7295)

Dinu et al. (2012a):

- 30 de clase de conjugare care modelează 95% din același corpus
- ⇒ odată ce cunoști clasa cunoști modul în care verbul se conjugă (alternanțe + seria de flective) => e suficient să înveți clasa verbului

Conjugator. Modelarea Claselor

- O clasă corespunde unei reguli de conjugare
- O regulă de conjugare
 - = un set de 6 expresii regulate, fiecare recunoscând una din cele 6 forme ale unui verb conjugat la indicativ prezent.
 - părțile fixe ale expresiilor dintr-o regula reprezintă părțile verbului care nu alternează.

Conjugator. Modelarea Claselor

De exemplu, regula care recunoaște verbe precum
“omoară”:

Conjugator. Modelarea Claselor

De exemplu, regula care recunoaște verbe precum “omoară”:

1 sg:	$^{\wedge}(.*)o(.*)\$$	omor
2 sg:	$^{\wedge}(.*)o(.*)i\$$	omori
3 sg:	$^{\wedge}(.*)oa(.*)ă\$$	omoară
1 pl:	$^{\wedge}(.*)o(.*)âm\$$	omorâm
2 pl:	$^{\wedge}(.*)o(.*)âți\$$	omorâți
3 pl:	$^{\wedge}(.*)oa(.*)ă\$$	omoară

Conjugator. Rezultatul etichetării

regulă	dimensiune
1	547
2	8
3	18
4	5
5	8
6	16
7	3330
8	273
9	89
10	4

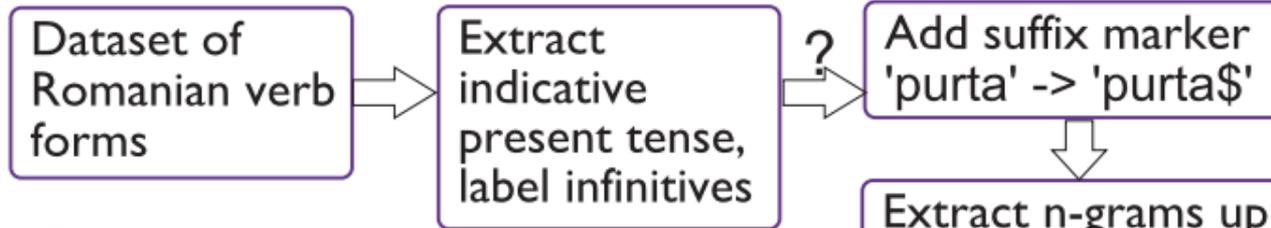
regulă	dimensiune
11	5
12	4
13	106
14	13
15	5
16	13
17	6
18	4
19	14
20	124

regulă	dimensiune
21	25
22	15
23	7
24	41
25	51
26	185
27	1554
28	486
29	5
30	27

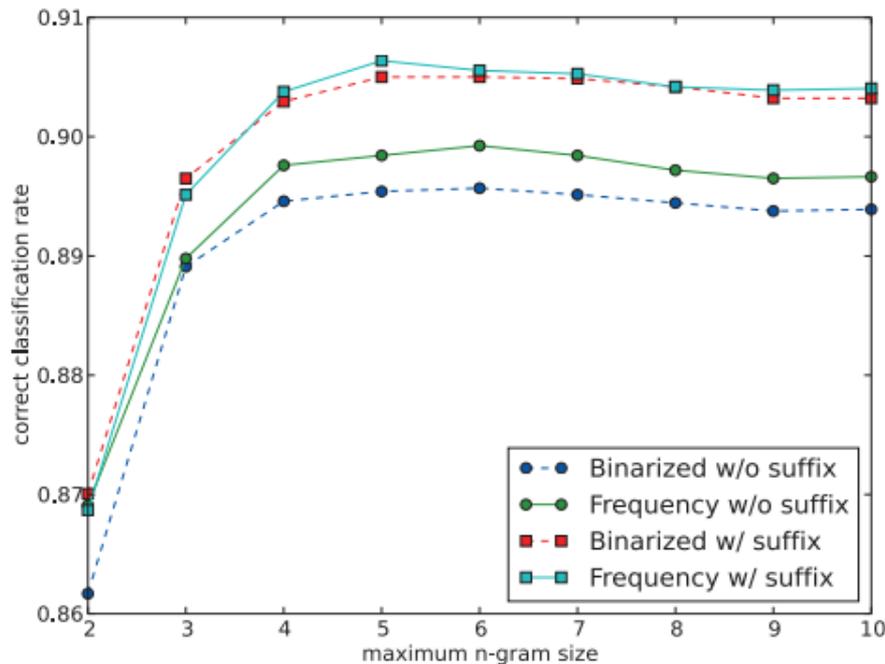
Conjugator. Sistemul de clasificare

- Clasificator folosind n-grame de caractere drept trăsături, unde $n=5$ e optim, + SVM
- Input: 'purta' => 'p', 'u', 'r', 't', 'a', 'pu', 'ur', 'rt', 'ta', 'pur', 'urt', 'rta', ...
- Output:
 - Dinu et al. (2011) etichetă în {1, 2, ..., 7}
 - Dinu et al. (2012a) etichetă în {1, 2, ..., 30}
- Rezultate:
 - Dinu et al. (2011): 82.71 % accuracy, 80% F-score
 - Dinu et al. (2012a): 90.64% accuracy, 89.89% F-score

Conjugator. Metodologia clasificării



Results:



Extract n-grams up to size n

Vectorize into n-gram frequency or occurrence (binarized) vectors

Classify using Linear SVC

Estimate scores using 10-fold cross validation

Conjugator. Interacțiune între reguli

- Unele reguli se “suprapun”, în sensul că modelează aceeași serie de flexive, dar alte alternanțe.

	regula 10 (a cânta)	regula 12 (a deștepta)	regula 13 (a deșerta)	regula 15 (a desfăta)
1sg	^(.*)t\$	^(.*)e(.*)t\$	^(.*)e(.*)t\$	^(.*)ăt\$
2sg	^(.*)ți\$	^(.*)e(.*)ți\$	^(.*)e(.*)ți\$	^(.*)eți\$
3sg	^(.*)tă\$	^(.*)ea(.*)tă\$	^(.*)a(.*)tă\$	^(.*)ată\$
1pl	^(.*)tăm\$	^(.*)e(.*)tăm\$	^(.*)e(.*)tăm\$	^(.*)ătăm\$
2pl	^(.*)tați\$	^(.*)e(.*)tați\$	^(.*)e(.*)tați\$	^(.*)ătați\$
3pl	^(.*)tă\$	^(.*)ea(.*)tă\$	^(.*)a(.*)tă\$	^(.*)ată\$

Conjugator. Concluzii. Viitor

- Conjugarea verbelor în Română poate fi învățată cu performanță ridicată, chiar și atunci când clasele nu interacționează
 - Clasele noastre sunt robuste \Leftrightarrow un model exhaustiv ar presupune, cel puțin pentru setul de antrenare, multe clase pentru conjugări unice sau aproape unice.
- \Rightarrow Pentru o mai bună generalizare vom avea nevoie de o modelare mai fină.

Domeniul verbal. Serii sau Serie?

- Feldstein (2004) propune o segmentare a flectivului verbal în 3 markeri: timp, număr, persoana.

	Indicativ prezent			Imperfect		
	Timp	Număr	Persoană	Timp	Număr	Persoană
1 sg	--	--	-u	-a	--	-u
2 sg	--	--	-i	-a	--	-i
3 sg	--	--	--	-a	--	--
1 pl	--	-m-	-u	-a	-m-	-u
2 pl	--	-t-	-i	-a	-t-	-i
3 pl	--	-u-	--	-a	-u-	--

Domeniul verbal. Serii sau Serie?

- Feldstein (2004) propune o segmentare a flectivului verbal în 3 markeri: timp, număr, persoana.
- Șulea (2012) argumentează pentru această segmentare, arătând că toate seriile de flective identificate până acum (i.e. de Guțu-Romalo) pot fi deduse prin procese fonologice din seria de flective fundamentală a limbii române dată de această segmentare

Sequence Tagging for Verb Conjugation in Romanian

Liviu P. Dinu Vlad Niculae Octavia-Maria Şulea

Center for Computational Linguistics
University of Bucharest
<http://nlp.unibuc.ro>

September 2013

Verbs in Romanian

Regularity is not black and white

		1 st	2 nd	3 rd
Regular	sg.	merg	mergi	merge
a merge (<i>to walk</i>)	pl.	mergem	mergeți	merg
Irregular	sg.	sunt	ești	este
a fi (<i>to be</i>)	pl.	suntem	sunteți	sunt

Verbs in Romanian

Regularity is not black and white

		1 st	2 nd	3 rd
Regular	sg.	merg	mergi	merge
a merge (<i>to walk</i>)	pl.	mergem	mergeți	merg
Irregular	sg.	sunt	ești	este
a fi (<i>to be</i>)	pl.	suntem	sunteți	sunt
Partially irregular	sg.	port	porți	poartă
a purta (<i>to wear</i>)	pl.	purțăm	purtați	poartă

Dinu et al, RANLP 2011, EACL 2012

- Hand-crafted sets of regular expressions fully describing conjugation of most verbs
- Predictive model $h(\text{infinitive}) = \text{regular expression set}$

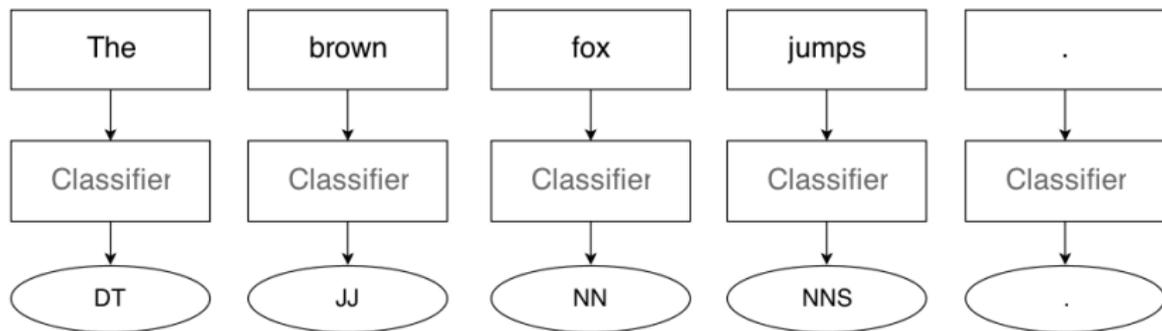
Running example

	sg.	port	porți	poartă
a purta (<i>to wear</i>)	pl.	purtăm	purtați	poartă

Regular expression set

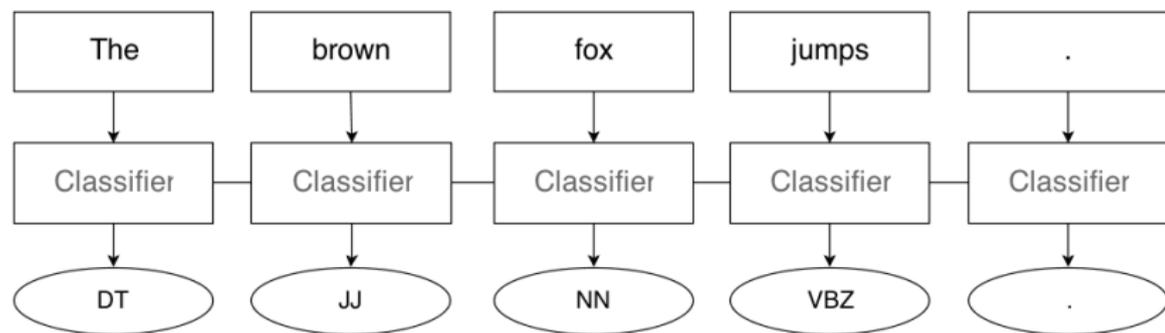
sg.	$\text{^\text{.}(.*)o(.*)t\$}$	$\text{^\text{.}(.*)o(.*)\text{ț}i\$}$	$\text{^\text{.}(.*)oa(.*)tă\$}$
pl.	$\text{^\text{.}(.*)u(.*)tăm\$}$	$\text{^\text{.}(.*)u(.*)ta\text{ț}i\$}$	$\text{^\text{.}(.*)oa(.*)tă\$}$

Sequence tagging: POS tagging example



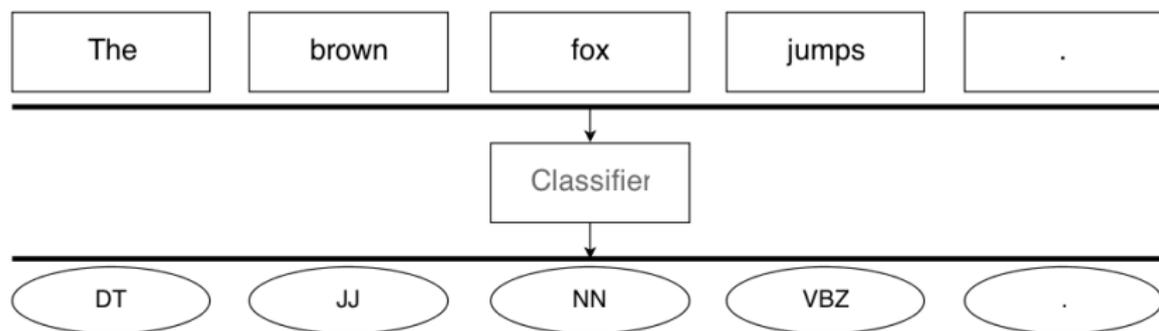
$$\prod \phi(y_i, x_i)$$

Sequence tagging: POS tagging example (better)



$$\prod \phi_1(y_i, x_i) \phi_2(y_i, y_{i+1})$$

Sequence tagging: POS tagging example (worse?)



$$\phi(y_1, y_2, \dots, y_n, x_1, x_2, \dots, x_n)$$

Ignored structure: interaction between classes

a cânta	a deștepta	a deșerta
<i>to sing</i>	<i>to rise</i>	<i>to empty</i>
$\text{^}(\text{.}*)\text{t}\text{\$}$	$\text{^}(\text{.}*)\text{e}(\text{.}*)\text{t}\text{\$}$	$\text{^}(\text{.}*)\text{e}(\text{.}*)\text{t}\text{\$}$
$\text{^}(\text{.}*)\text{ț}\text{i}\text{\$}$	$\text{^}(\text{.}*)\text{e}(\text{.}*)\text{ț}\text{i}\text{\$}$	$\text{^}(\text{.}*)\text{e}(\text{.}*)\text{ț}\text{i}\text{\$}$
$\text{^}(\text{.}*)\text{t}\text{ă}\text{\$}$	$\text{^}(\text{.}*)\text{ea}(\text{.}*)\text{t}\text{ă}\text{\$}$	$\text{^}(\text{.}*)\text{a}(\text{.}*)\text{t}\text{ă}\text{\$}$
$\text{^}(\text{.}*)\text{t}\text{ă}\text{m}\text{\$}$	$\text{^}(\text{.}*)\text{e}(\text{.}*)\text{t}\text{ă}\text{m}\text{\$}$	$\text{^}(\text{.}*)\text{e}(\text{.}*)\text{t}\text{ă}\text{m}\text{\$}$
$\text{^}(\text{.}*)\text{ta}\text{ț}\text{i}\text{\$}$	$\text{^}(\text{.}*)\text{e}(\text{.}*)\text{ta}\text{ț}\text{i}\text{\$}$	$\text{^}(\text{.}*)\text{e}(\text{.}*)\text{ta}\text{ț}\text{i}\text{\$}$
$\text{^}(\text{.}*)\text{t}\text{ă}\text{\$}$	$\text{^}(\text{.}*)\text{ea}(\text{.}*)\text{t}\text{ă}\text{\$}$	$\text{^}(\text{.}*)\text{a}(\text{.}*)\text{t}\text{ă}\text{\$}$

Conjugation as sequence tagging

Running example

	sg.	port	porți	poartă
a purta (<i>to wear</i>)	pl.	purtăm	purtați	poartă

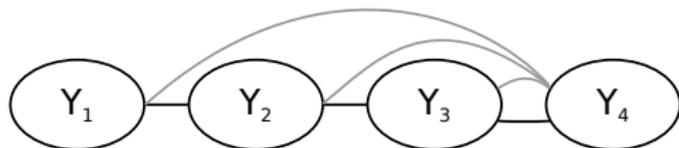
Variable letters (Moisil)

$\text{form}(u_0 1sg) =$	o	$\text{form}(t_0 1sg) =$	t
$\text{form}(u_0 3sg) =$	oa	$\text{form}(t_0 2sg) =$	ț
$\text{form}(u_0 1pl) =$	u		

Tagging example

p	u	r	t	a
0	u_0	0	t_0	T_4

- Features: character n-grams to the left and right size up to n
- Dataset: RoMorphoDict (lemmas and forms) labeled using the RegEx sets
16 ending patterns, 17 variable letters
4,699 train / 2,257 test / 339 unlabeled
- Grid search, 10-fold cross validation



- An extra factor template allowing the ending to influence all positions
- Inference becomes more complex
- Out-of-the-box sequence tagging no longer appropriate

method	Cross-val. accuracy			Test accuracy		
	word	char	char'	word	char	char'
SVM	0.886	-	-	0.896	-	-
ML	0.924	0.987	0.913	0.914	0.985	0.900
AP	0.923	0.987	0.917	0.912	0.985	0.900
PA	0.925	0.987	0.917	0.912	0.984	0.900
AROW	0.916	0.986	0.912	0.908	0.984	0.895
SKIP	-	0.984	-	0.906	0.983	0.896

Generalization on 105 of the unlabeled verbs:

- many termination patterns are correctly found (30)
- some alternations are found (3)

Cuprins

- Domeniul verbal. Crearea unui conjugator pentru limba română
- Domeniul nominal. Clasificarea substantivelor din limba română după gen
- Concluzii
- Bibliografie

Domeniul Nominal. Clasă Nominală

- Task: dându-se forma nearticulată de nominativ-acuzativ a unui substantiv nou și cunoștințe referitoare la genul substantivelor în română, vrem ca sistemul să spună genul corect al substantivului

Domeniul Nominal. Clasă Nominală

- Task: dându-se forma nearticulată de nominativ-acuzativ a unui substantiv nou și cunoștințe referitoare la genul substantivelor în română, vrem ca sistemul să spună genul corect al substantivului
- Motivație: clasificatoarele anterioare ale substantivelor din Română după gen ori:
 - eșuau în a distinge neutru de masculin (Năstase și Popescu, 2009)
 - nu se oboseau să îl identifice (Cucerzan și Yarowski, 2003)

Domeniul Nominal. 2 sau 3 clase?

- Română:
 - în dicționar: 3 genuri (masculin, feminin, neutru)
 - pe adjective, pronume, etc.: doar 2 markeri de acord

	Singular	Plural
Masculin	<u>un</u> băiat	doi băieți
Neutru	<u>un</u> stilou	<u>două</u> stilouri
Feminin	o fată	<u>două</u> fete

Domeniul Nominal. 2 sau 3 clase?

- Română:
 - în dicționar: 3 genuri (masculin, feminin, neutru)
 - pe adjective, pronume, etc.: doar 2 markeri de acord

	Singular	Plural
Masculin	<u>un</u> băiat	doi băieți
Neutru	<u>un</u> stilou	<u>două</u> stilouri
Feminin	o fată	<u>două</u> fete

- neutrul, în termeni de acord, urmează sistematic masculinul la singular și femininul la plural.

Domeniul Nominal. 2 sau 3 clase?

- Sistemul tradițional, trinitar (Graur et al., 1966)
 - 3 genuri marcate în lexicon/ clase nominale
 - modul în care substantivele sunt distribuite într-una din clase și legătura dintre ele și sistemul acordului sunt apoi schițate de Corbett (1991), Farkas (1990), însă Bateman și Polinsky (2010) argumentează împotriva abordării lor

Domeniul Nominal. 2 sau 3 clase?

- Sistemul tradițional, trinitar (Graur et al., 1966)
 - 3 genuri marcate în lexicon/ clase nominale
 - modul în care substantivele sunt distribuite într-una din clase și legătura dintre ele și sistemul acordului sunt apoi schițate de Corbett (1991), Farkas (1990), însă Bateman și Polinsky (2010) argumentează împotriva abordării lor
- Sistemul modern, dual (Bateman și Polinsky, 2010)
 - 2 clase nominale (nemarcate în lexicon) la singular (m/f) și alte 2 clase la plural (tot m/f);
 - asignarea genului la singular și plural se face separat și bazat pe trăsături semantice (gen natural) și fonologice.
 - neutrul presupune asignare diferită la singular și plural

Clasificatori anteriori

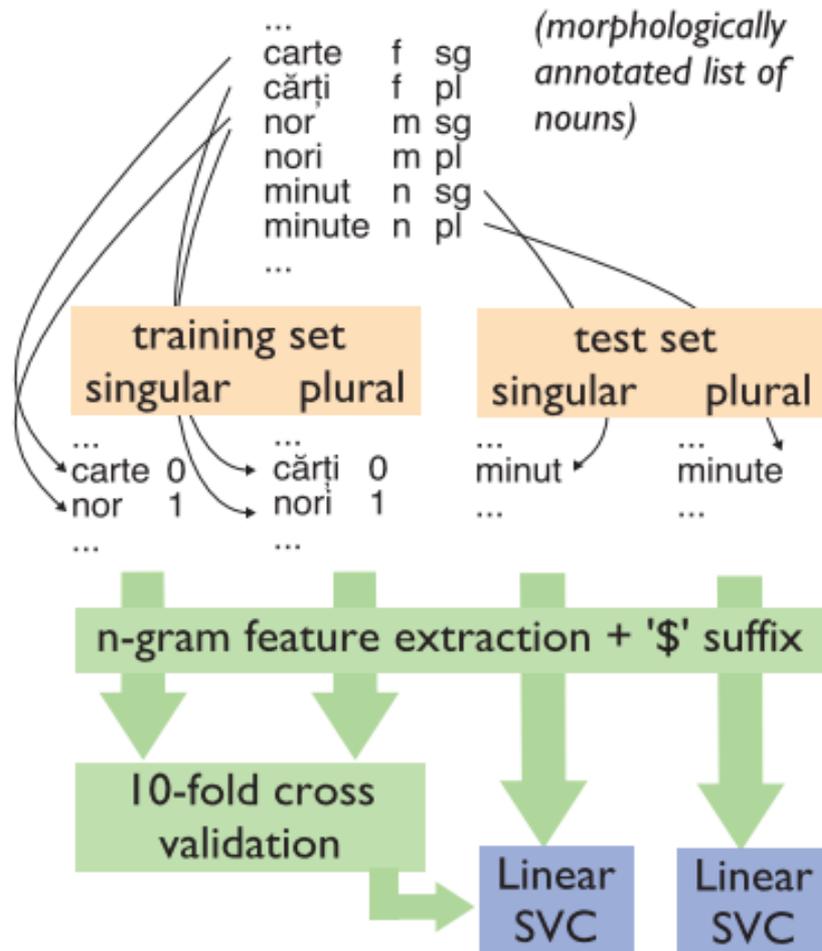
- Nastase și Popescu (2009)
 - presupun sistemul trinitar
 - folosesc doar formele de singular, nominativ-acuzativ, neutru.
 - au probleme (firește) în a distinge neutru de masculin (la singular)
- Cucerzan și Yarovsky (2003)
 - presupun sistemul dual
 - se uită doar la singular, dar folosesc informații din context (i.e. articolul, acordul cu adjective, pronume, etc.)
 - nu diferențiază între neutru și masculin

Clasificatorul nostru

Dinu et al. (2012b)

- presupun și verifică sistemul modern, dual
- împart problema clasificării neutrului în două probleme de clasificare binară (singular / plural).
- folosesc ca input atât formele (N-A, neart.) de singular cât și de plural
- pentru a verifica ipoteza duală, testează dacă neutrul se clasifică drept masculin la singular și feminin la plural folosind ca trăsături n-grame de caractere (trăsături fonologice).

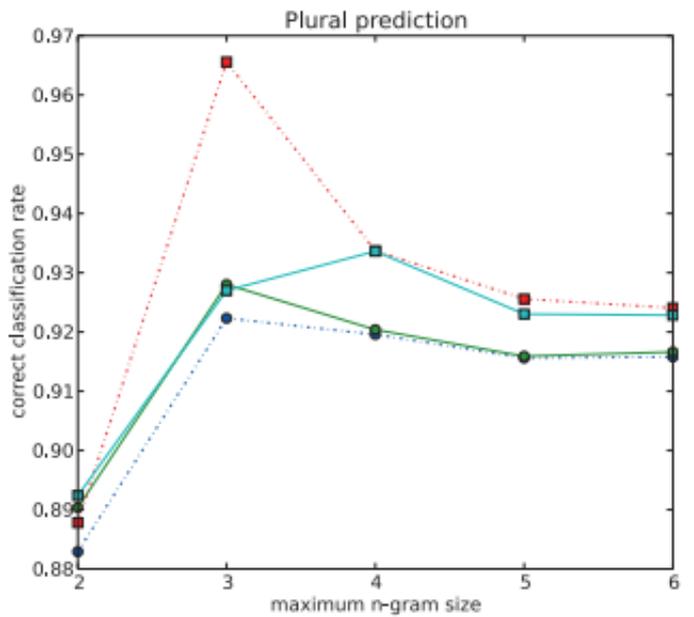
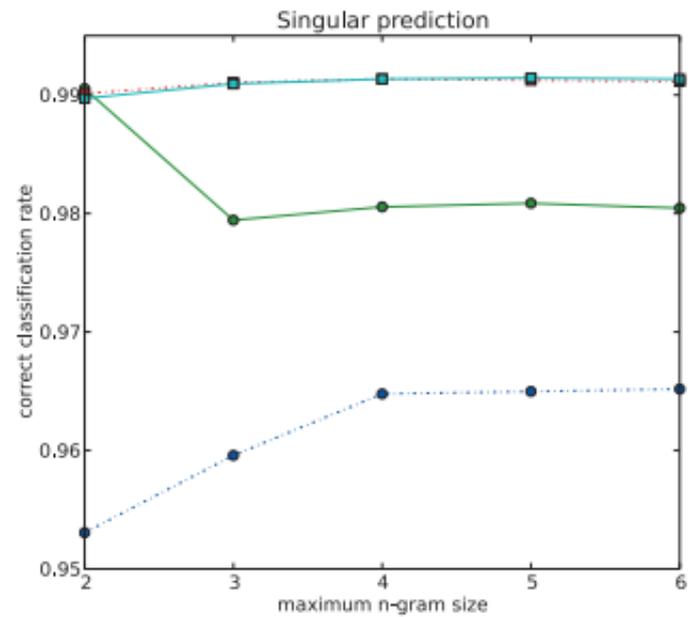
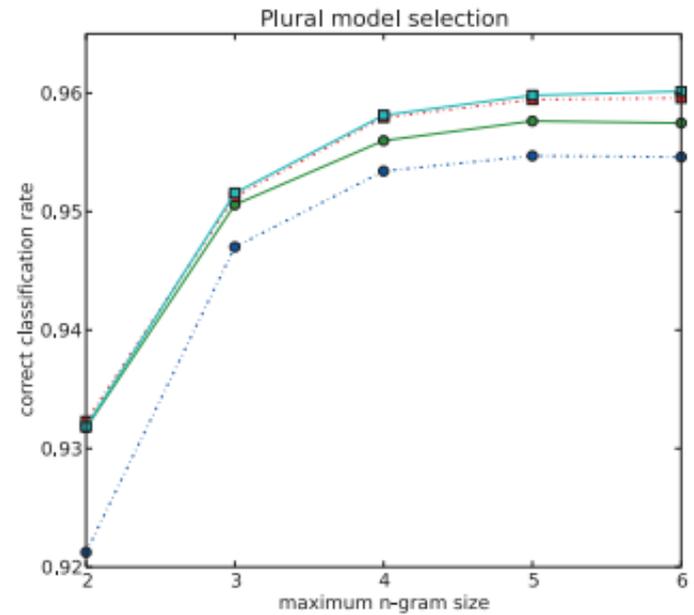
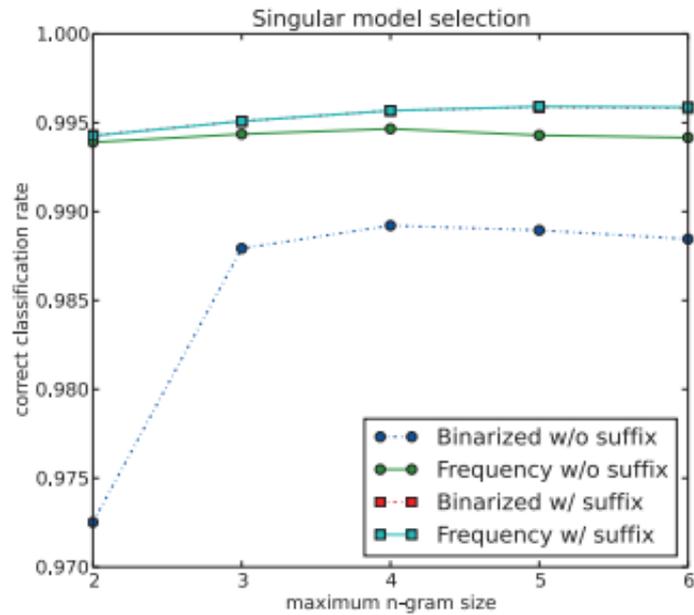
Antrenarea modelelor



- Antrenare pe masculine și feminine, la singular și plural
- Testare pe neutre, la singular și plural

Domeniul Nominal. Rezultate

- Parametrii aleși pentru sistem: 5-grame, fără binarizare și adaugă sufixul '\$'.
- Scorurile estimate prin validare încrucișată:
 - singular: accuracy 99.59%, precision 99.63%, recall 99.80%, F1 99.71 %
 - plural: accuracy 95.98%, precision 97.32%, recall 97.05%, F1 97.18%
- Evaluarea neutrelor:
 - performanță 99.14% la singular și 92.30% la plural.
 - la plural e mai scăzută din cauza substantivelor compuse și terminațiilor derutante:
balaur/ balaur-i vs. bord/ bord-uri.



Cuprins

- Domeniul verbal. Crearea unui conjugator pentru limba română
- Domeniul nominal. Clasificarea substantivelor din limba română după gen
- Concluzii
- Bibliografie

Concluzii

- Problemele prezentate se modelează foarte bine ca probleme de clasificare, folosind n-grame de caractere drept trăsături (unde $n=5$ e optim).
- Analiza teoretică a problemei lingvistice este esențială și poate afecta eficiența clasificatorului.
- În ambele cazuri, adăugând sufixul artificial '\$' pentru a oferi greutate mai mare terminațiilor duce la un rezultat mai bun.
- SVM-urile se pretează bine pe aceste task-uri

Bibliografie

- Nicoleta Bateman and Maria Polinsky, 2010. Romanian as a two-gender language, chapter 3, pages 41–78. MIT Press, Cambridge, MA.
- Greville G. Corbett. 1991. *Gender*. Cambridge University Press.
- S. Cucerzan and D. Yarowsky. 2003. Minimally supervised induction of grammatical gender. In *HLT-NAACL 2003*, pages 40–47.
- Liviu P. Dinu, Emil Ionescu, Vlad Niculae, and Octavia-Maria Şulea. Can alternations be learned? a machine learning approach to verb alternations. *In Recent Advances in Natural Language Processing 2011*, September 2011.
- Liviu P. Dinu, Vlad Niculae, Octavia-Maria Şulea. Learning How to Conjugate the Romanian Verb. Rules for Regular and Partially Irregular Verbs. In: *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL 2012)*. April 2012a.
- Liviu P. Dinu, Vlad Niculae, Octavia-Maria Şulea. The Romanian Neuter Examined Through A Two-Gender N-Gram Classification System. In: *Proceedings of the International Conference on Language Resources (LREC 2012)*. May 2012b.

Bibliografie

- Graur, A., Avram, M., and Vasiliu, L. (1966). *Gramatica Limbii Române, volume 1. Academy of the Socialist Republic of Romania*, 2nd edition.
- Feldstein, R. F. 2004. On the structure of syncretism in Romanian conjugation. In J. Auger, J. C. Clements and B. Vance (eds.), *Contemporary Approaches to Romance Linguistics. Selected Papers from the 33rd linguistic symposium on Romance Languages*, 177-195. Amsterdam/ Philadelphia: John Benjamins.
- Valeria Guțu-Romalo. *Morfologie Structurală a limbii române*. Editura Academiei Republicii Socialiste România, 1968.
- Alf Lombard. *Le verbe roumain. Etude morphologique, volume 1*. Lund, C. W. K. Gleerup, 1955.
- Grigore C. Moisil. Probleme puse de traducerea automată. Conjugarea verbelor în limba română. *Studii și cercetări lingvistice*, XI(1):7–29, 1960.
- Vivi Nastase and Marius Popescu. 2009. What's in a name? in some languages, grammatical gender. In EMNLP, pages 1368–1377. ACL.
- Octavia-Maria Șulea. *Alternations in the Romanian verb paradigm. Analyzing the indicative present*. MA thesis. University of Bucharest.
<http://ling.auf.net/lingBuzz/001562>

Vă mulțumesc!

Lingvistica Matematica si Computationala

Liviu P. Dinu,

ldinu@fmi.unibuc.ro

University of Bucharest

Center for Computational Linguistics,

Faculty of Mathematics and Computer Science

nlp.unibuc.ro



Rank distance, rank aggregation and applications

Rank Distance. Motivation

- Often, the main information of a message is placed in its first part.

The length of the phrase	Percentage of memorized words from:		
	the whole phrase	the first half of the phrase	the second half of the phrase
12	100 %	100 %	100 %
13	90 %	95 %	85 %
17	70 %	90%	50%
24	50 %	70 %	30 %
40	30 %	50 %	10 %

- Given a set of messages (usually rankings), one faces two problems: how to compute their distance and how to aggregate them?

Rank Distance

- To measure the distance between two rankings, we proceed as follows:
 - assign a position (in Borda order) to each letter ;
 - scan (top-down) both rankings, and for each letter from the first ranking count the number of elements between its position in the first ranking and its position in the second ranking;
 - for unmatched letters, add their position;
 - finally, sum all these scores and obtain the **rank distance**.

Extension to strings. Efficient computation

- Given two strings x and y , the RD is defined through the following algorithmic process:
 - both strings are scanned (from left to right) and for each character a in the first string, and for each of its k -th occurrence in x , the algorithm sums up the absolute difference between the position of its k -th occurrences in x and y .

Extension to strings

- for each of the non-matched occurrences of a in one of the two strings, the algorithm adds to the sum the arithmetic mean of $|x|$ and $|y|$.
- **RD** =The total sum computed by this algorithm

Mathematical results (selection)

- P1. (**collinearity** problem) Given two strings f and g over U , how many strings h over U are there, such that
$$\Delta(f, g) + \Delta(g, h) = \Delta(f, h) \text{ (or } \Delta(f, h) + \Delta(h, g) = \Delta(f, g)\text{)}?$$
- P2. (**diameter** over binary strings). Let $T_{m,n}$ be the set of all words over V with m 0's and n 1's. The diameter of the set $T_{m,n}$ is given by the computing of RD between strings $p_{01} = 00\dots011\dots1$ and $p_{10} = 11\dots100\dots0$.

max RD on strings

- Theorem (**max** RD on binary strings). Let u in $T_{m,n}$ be a string. $\Delta(u,v) \leq \max \{\Delta(u, \mathbf{p}_{01}), \Delta(u, \mathbf{p}_{10})\}$, for any string v from $T_{m,n}$.

Open problems

- Diameter over **arbitrary** strings
- Cardinality of ball (sphere) of center **u** and radius **r**: given a string **u**, how many strings are at a given rank distance **r** of it?

Rank aggregation(s)

- How do we aggregate the voters?
 1. The rank distance aggregation (RDA, **Median string**): given n rankings (voters), **RDA** is that ranking (voter) whose sum of the distances (via rank distance) to all rankings (voters) is minimum.
 2. **Closest string problem**: given a set of strings, the goal is to find a string with the property that it is the centre of a ball with minimum radius such that all the other strings are inside the ball.

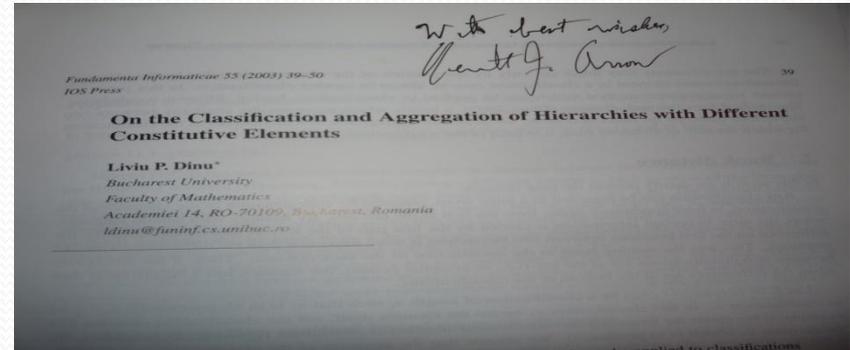
• ...

Computational Properties (TCS 2006, CPM 2012)

- **Median string:** NP hard problem for edit and Kendall distance.
- **Closest string:** non-polynomial solution for Hamming, Levenshtein, or Kendall distances.
- Rank distance approaches:
 - **MSRD:** a polynomial time solution
 - **CSRD:** no polynomial solution

Rationality (Arrow's) properties

1. Pareto optimality: if all voters prefer a to b in all initial rankings, there is an aggregation in RDA in which a is preferred to b .
2. RDA does not satisfy the independence condition.



Rationality (Arrow's) properties

1. RDA is “reasonable”: if we apply RDA to rankings with two elements, the result is the same as when the majority rule is applied.
2. RDA is stable, free order, loyal and invertible.

Loose stability

(A last voter, voting manipulation)

- We are in the following situation:
 - we have many voters, and we are interested in their aggregation. We compute RDA, and we obtain a set of aggregation.
- What if a last minute voter comes and wants to vote?
 - Intuitively, if there are many voters, if we add only one more voter, the aggregating result must be more or less the same.

Manipulation

- Our results show that, if the voter is *a special one*, the result is completely changed:
 - if we add to the initial voters a voter which is in their aggregation set (RDA), and we aggregate again, the result is formed *only from this voter*. In other words, it eliminates all the competitors.

Open problems

- RDA produces a set of aggregations. How many are there? Which is the best one?
- An efficient heuristic for determining the closest string.
- How many closest strings are between two strings? (a closed formula).
- Given two strings x and y , at least a closest string is on the $[x, y]$ segment (proved at least once!).
- Relation between $\#RDA$ and Genocchi number (thanks to C. Zara).
- ...

Rank Categorization

- RDA produces a set of rankings.
- If we are interested only in the *winner*, not by the *full preferences*, we need one more step: we have to transform the RDA of the voters in a categorization.
- The procedure is simple: we actually apply the voting methods on voters' aggregations (RDA);
 - in other words, we count who is on the first position most of the times and we choose it as the winner.

RDC Properties: Is half enough?

- If all voters prefer A on the first position, then A will be the winner.
- If *half plus one* from the voters vote for the candidate A to be on the first position, then this candidate will also be the winner.
- However, if “*only*” *half* of voters vote the candidate A to be on the first position, then this candidate is not necessarily the winner (he can lose).

Applications

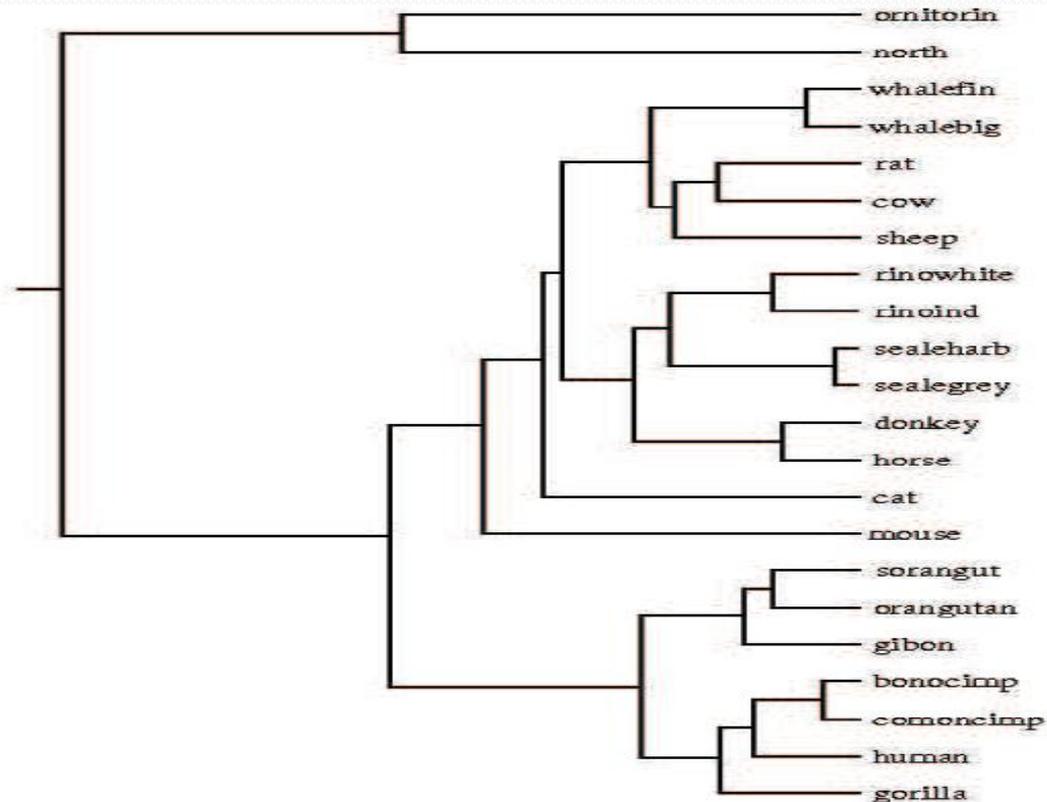
- Handwritten digit classification (Fund. Inf., 2008)
 - Better than other multicriterial categorization methods applied on a Dutch database (the best rate 98.2).
- Text categorization (CiCling 2010)
 - Good behaviour on Reuters database, on instances with more than 20 classes.

Applications

- Clustering methods based on rank distance (Iconip 2012):
 - K-Means-type algorithms based on rank distance.
 - Hierarchical clustering based on rank distance.
- Others: meta-search engine aggregation, collocation detection.

Applications in Bioinformatics: DNA similarity

(PlosOne'12&'14, ICONIP12, SYNASC12)

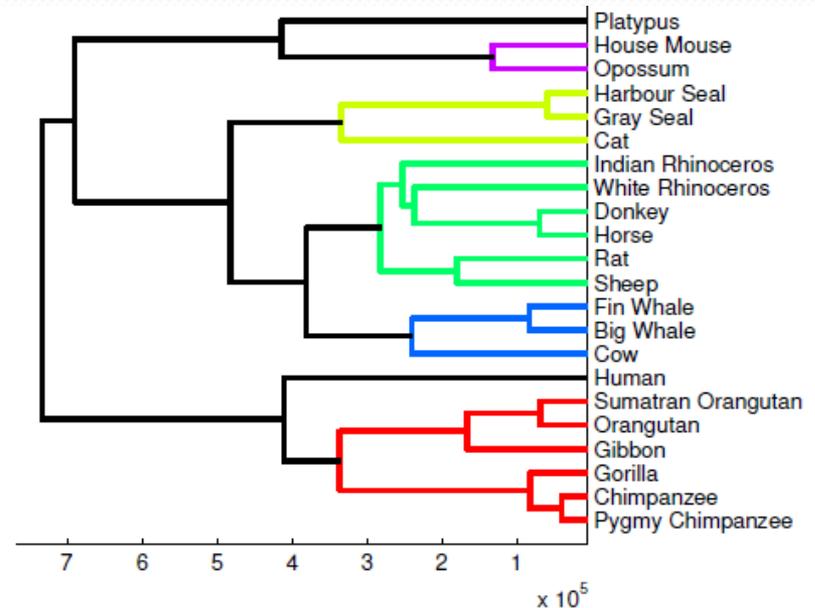


The mammals phylogenies build from complete mammalian mtDNA sequences using rank distance

Phylogenetic (clustering) analysis

String Alignment analysis

Family	K-median	K-closest
Cetartiodactylae	75%	75%
Carnivora	100%	0%
Metatheria	100%	100%
Monotremata	100%	100%
Rodentia	100%	100%
Primates	100%	85%
Perissodactylae	50%	100%
Overall	86%(19/22)	77%(17/22)



Tree obtained with median string

Experiment: Handwritten digit classification

- We make a comparative study regarding the behavior of six combining schema on the same input data set.
- The input dataset consists of handwritten numerals (000,...,090) extracted from a collection of Dutch utility maps.
- We compare the RDC results with reported results obtained by other five combining methods.

Experiment

- The experiments are done on a data set which consists of six different feature sets for the same set of objects.
- The six feature sets are:
 - Fourier: 76 Fourier coefficients of the character shapes.
 - Profiles: 216 profile correlations.
 - KL-coef: 64 Karhunen-Love coefficients.
 - Pixel: 240 pixel averages in 2 x 3 windows.
 - Zernike: 47 Zernike moments.
 - Morph: 6 morphological features.

Details

- The 12 individual classifiers for a single feature set were combined using the five combining rules.
- On each combining rule line (R_1, \dots, R_5), its success rate (in percent) is given for each feature from the corresponding column.
- We combined the 12 classifiers for each feature using the RDC method.

Details (2)

- The results are shown in the table 1, last line.
- For each feature set, the best result over the classifiers is printed in bold;
- The underlined results indicate that combination result is better than the performance of individual classifiers for this feature set.

12 classifiers (c1,...,c12) and 6 combining rules

Table 1. The success rate (%) of different classifiers and combining method (1)

Classifiers		Feature					
		No.1	No.2	No.3	No.4	No.5	No.6
c1	Bayes-NQ	74.3	94.2	87.2	93.8	78.8	69.0
c2	Bayes-NL	78.7	96.6	94.3	90.1	82.0	70.9
c3	Nrst-Mean	77.5	81.9	90.1	90.4	72.2	46.0
c4	1-NN	80.8	91.0	95.6	96.3	80.2	40.4
c5	k-NN	81.1	90.8	95.6	96.3	80.7	49
c6	Parzen	82.9	92.1	96.3	96.3	81.5	47.9
c7	Fisher	75.2	95.2	91.8	85.1	79.0	71.8
c8	Dec. Tree	54.6	59.7	60.0	45.1	40.2	67.1
c9	ANN-20	10.0	95.4	85.4	14.8	10.0	67.2
c10	ANN-50	75.5	87.0	17.7	19.0	73.5	28.3
c11	SVC-L	64.9	81.6	94.2	92.3	57.4	16.5
c12	SVC-Q	72.3	92.6	96.1	94.0	63.2	16.3
Combining rule:							
R1	Median	81	95.7	<u>96.4</u>	<u>95.5</u>	<u>82.6</u>	71.3
R2	Product	70.6	86.9	95.6	91.8	59.9	58.8
R3	Voting	82.5	96.5	<u>96.8</u>	<u>96.3</u>	<u>83.1</u>	68.2
R4	Nrst-Mean	80.2	96.3	95.4	92.7	81.9	<u>73.4</u>
R5	1-NN	81.4	96.2	95.9	92.8	<u>83</u>	67.2
R6	RDC	<u>83.6</u>	<u>96.6</u>	<u>96.7</u>	<u>96.6</u>	<u>83.2</u>	70.8

RDC =the best

- The RDC method gives better results in 5 out of 6 cases than all the individual classifiers (which is the best performance within the 6 methods).
- Out of all other methods RDC gives the best results in 4 out of 6 cases
- Combining rules are applied on the 6 features for a single classification rule.
- In table 2 we present the success rate for each combining rule, including the RDC method in the last column

Table 2. The success rate (%) of different classifiers and combining methods (2)

Classifiers	Combination rule					
	Med.	Prod.	Voting	Nrst-Mean	1-NN	RDC
c1	<u>97.2</u>	93.7	93.2	93.3	<u>95</u>	<u>96.5</u>
c2	96.3	<u>96.9</u>	94.9	96.1	95.8	<u>97.1</u>
c3	<u>93.8</u>	<u>95.4</u>	<u>92.5</u>	89.7	<u>95.4</u>	<u>93.9</u>
c4	<u>97.4</u>	<u>98.3</u>	96	88.7	<u>97</u>	<u>97.3</u>
c5	94.6	95.8	94.9	<u>96.4</u>	<u>97.4</u>	<u>97.2</u>
c6	<u>97.1</u>	<u>97.3</u>	94.9	<u>96.9</u>	<u>96.9</u>	<u>97.5</u>
c7	<u>96.8</u>	94.8	94.3	<u>96.5</u>	<u>96.4</u>	<u>97.2</u>
c8	<u>82.6</u>	<u>89</u>	<u>78.2</u>	<u>89.8</u>	<u>89.2</u>	<u>85</u>
c9	82.3	10	67.3	<u>97.4</u>	<u>97.9</u>	74.5
c10	75.6	19.3	83.7	<u>94.5</u>	<u>96.7</u>	81.9
c11	89.2	89.9	<u>95.3</u>	<u>94</u>	<u>94.2</u>	<u>95</u>
c12	<u>96.4</u>	<u>96.2</u>	<u>96.2</u>	96	96	95.8

Table 3. The average success rate (%) of c_1, \dots, c_{12} classifiers

	c1	c2	c3	c4	c5	c6
Average	82.88	85.43	76.35	80.71	80.71	82.83
	c7	c8	c9	c10	c11	c12
Average	71.35	54.45	47.13	50.16	67.81	72.41

Analyse

- For each of the 12 classifiers, the underlined results indicate that this combination result is better than each of the six individual results of the current classifier.
- The RDC method gives better results in 9 out of 12 cases than all the individual classifiers (only 1-NN method gives 10 of 12).
- In table 3 we present the average success rate of each of the 12 classifiers over the six features.

Analyse

- We can see that the classifiers c9 and c10 have average success rate around 50%, much less than all other classifiers .
- If we have ignored these 2 classifiers, the RDC method would have overcome the 1-NN method, becoming first in all competition between the combining methods.

The winner is... RDC

- We applied all 12 classifiers to all six features and obtained for each document a multiset of 72 rankings.
- We combined these rankings by using RDC combining schema and compare the results to real results.
- The success rate of RDC is in this case 98.2%. This is the best result of our experiment.

Conclusion

- Our analysis reveals that the classification based on RDC combining schema gives one of the best possible results.
- We have to say that RDC can be computed in a polynomial time and it is based on a nontrivial, linear-time metric.
- This facts, corroborated with the remark that RDC is a fixed combiner, make RDC a serious candidate in the very competitive field of multi-agent classification.



MULTUMESC!

Rank Distance Aggregation as a Fixed Classifier Combining Rule for Text Categorization

Liviu P. Dinu¹, Andrei A. Rusu²

¹Faculty of Mathematics and Computer Science
University of Bucharest, Romania,

²(Student) Faculty of Exact Sciences
Vrije Universiteit Amsterdam, The Netherlands.

March 26 / CICLing 2010

Outline

- 1 Introduction
 - Motivation
 - Background
 - New to our Approach
- 2 Rank Distance Aggregation
 - Rankings
 - Rank Distance
 - Rank Distance Aggregation
 - Rank Distance Categorization
- 3 Experiments in Text Categorization
 - Tools and Setting
- 4 Summary and Conclusions

Outline

- 1 Introduction
 - Motivation
 - Background
 - New to our Approach
- 2 Rank Distance Aggregation
 - Rankings
 - Rank Distance
 - Rank Distance Aggregation
 - Rank Distance Categorization
- 3 Experiments in Text Categorization
 - Tools and Setting
- 4 Summary and Conclusions

Motivation

Why another fixed combining rule?

- Benefits of using more than one classifier:
 - learning more complex decision boundaries (e.g. more than *circles* or *lines*)
 - theoretical advantage shown for some combining methods: *boosting*
 - many classifiers *already implemented*, showing different accuracies
- Ensembles of classifiers are a well researched Machine Learning topic. However, ...
 - achieving the theoretical advantage of *trained combining rules* proves to be a very difficult task
 - fixed combining rules are widely used as the final decision maker, even within other combination schemes (bagging and boosting)

So, a better fixed combining rule can't hurt!...

Motivation

Why another fixed combining rule?

- Benefits of using more than one classifier:
 - learning more complex decision boundaries (e.g. more than *circles* or *lines*)
 - theoretical advantage shown for some combining methods: *boosting*
 - many classifiers *already implemented*, showing different accuracies
- Ensembles of classifiers are a well researched Machine Learning topic. However, ...
 - achieving the theoretical advantage of *trained combining rules* proves to be a very difficult task
 - fixed combining rules are widely used as the final decision maker, even within other combination schemes (bagging and boosting)

So, a better fixed combining rule can't hurt!...

Motivation

Why another fixed combining rule?

- Benefits of using more than one classifier:
 - learning more complex decision boundaries (e.g. more than *circles* or *lines*)
 - theoretical advantage shown for some combining methods: *boosting*
 - many classifiers *already implemented*, showing different accuracies
- Ensembles of classifiers are a well researched Machine Learning topic. However, ...
 - achieving the theoretical advantage of *trained combining rules* proves to be a very difficult task
 - fixed combining rules are widely used as the final decision maker, even within other combination schemes (bagging and boosting)

So, a better fixed combining rule can't hurt!...

Motivation

Why another fixed combining rule?

- Benefits of using more than one classifier:
 - learning more complex decision boundaries (e.g. more than *circles* or *lines*)
 - theoretical advantage shown for some combining methods: *boosting*
 - many classifiers *already implemented*, showing different accuracies
- Ensembles of classifiers are a well researched Machine Learning topic. However, ...
 - achieving the theoretical advantage of *trained combining rules* proves to be a very difficult task
 - fixed combining rules are widely used as the final decision maker, even within other combination schemes (bagging and boosting)

So, a better fixed combining rule can't hurt!...

Motivation

Why another fixed combining rule?

- Benefits of using more than one classifier:
 - learning more complex decision boundaries (e.g. more than *circles* or *lines*)
 - theoretical advantage shown for some combining methods: *boosting*
 - many classifiers *already implemented*, showing different accuracies
- Ensembles of classifiers are a well researched Machine Learning topic. However, ...
 - achieving the theoretical advantage of *trained combining rules* proves to be a very difficult task
 - fixed combining rules are widely used as the final decision maker, even within other combination schemes (bagging and boosting)

So, a better fixed combining rule can't hurt!...

Motivation

Why another fixed combining rule?

- Benefits of using more than one classifier:
 - learning more complex decision boundaries (e.g. more than *circles* or *lines*)
 - theoretical advantage shown for some combining methods: *boosting*
 - many classifiers *already implemented*, showing different accuracies
- Ensembles of classifiers are a well researched Machine Learning topic. However, ...
 - achieving the theoretical advantage of *trained combining rules* proves to be a very difficult task
 - fixed combining rules are widely used as the final decision maker, even within other combination schemes (bagging and boosting)

So, a better fixed combining rule can't hurt!...

Motivation

Why another fixed combining rule?

- Benefits of using more than one classifier:
 - learning more complex decision boundaries (e.g. more than *circles* or *lines*)
 - theoretical advantage shown for some combining methods: *boosting*
 - many classifiers *already implemented*, showing different accuracies
- Ensembles of classifiers are a well researched Machine Learning topic. However, ...
 - achieving the theoretical advantage of *trained combining rules* proves to be a very difficult task
 - fixed combining rules are widely used as the final decision maker, even within other combination schemes (bagging and boosting)

So, a better fixed combining rule can't hurt!...

Motivation

Why another fixed combining rule?

- Benefits of using more than one classifier:
 - learning more complex decision boundaries (e.g. more than *circles* or *lines*)
 - theoretical advantage shown for some combining methods: *boosting*
 - many classifiers *already implemented*, showing different accuracies
- Ensembles of classifiers are a well researched Machine Learning topic. However, ...
 - achieving the theoretical advantage of *trained combining rules* proves to be a very difficult task
 - fixed combining rules are widely used as the final decision maker, even within other combination schemes (bagging and boosting)

So, a better fixed combining rule can't hurt!...

Outline

- 1 Introduction
 - Motivation
 - Background
 - New to our Approach
- 2 Rank Distance Aggregation
 - Rankings
 - Rank Distance
 - Rank Distance Aggregation
 - Rank Distance Categorization
- 3 Experiments in Text Categorization
 - Tools and Setting
- 4 Summary and Conclusions

Background

What's available out there. . .

- Many *feature extraction* techniques do exist for nearly all applications
- Many *classifiers* readily available, so which is the “best” feature–classifier pair?
- Options:
 - Choose wisely (but don't optimize for one dataset)
 - Use more than one pair, thus combine different features with different classifiers.

It all boils down to what's good enough for you! Would you trust your bank account to a 99% accurate fingerprint classifier. . .

P.S: We can produce a fake fingerprint from your cup of coffee this morning. How about now?

Background

What's available out there. . .

- Many *feature extraction* techniques do exist for nearly all applications
- Many *classifiers* readily available, so which is the “best” feature–classifier pair?
- Options:
 - Choose wisely (but don't optimize for one dataset)
 - Use more than one pair, thus combine different features with different classifiers.

It all boils down to what's good enough for you! Would you trust your bank account to a 99% accurate fingerprint classifier. . .

P.S: We can produce a fake fingerprint from your cup of coffee this morning. How about now?

Background

What's available out there. . .

- Many *feature extraction* techniques do exist for nearly all applications
- Many *classifiers* readily available, so which is the “best” feature–classifier pair?
- Options:
 - Choose wisely (but don't optimize for one dataset)
 - Use more than one pair, thus combine different features with different classifiers.

It all boils down to what's good enough for you! Would you trust your bank account to a 99% accurate fingerprint classifier. . .

P.S: We can produce a fake fingerprint from your cup of coffee this morning. How about now?

Background

What's available out there. . .

- Many *feature extraction* techniques do exist for nearly all applications
- Many *classifiers* readily available, so which is the “best” feature–classifier pair?
- Options:
 - Choose wisely (but don't optimize for one dataset)
 - Use more than one pair, thus combine different features with different classifiers.

It all boils down to what's good enough for you! Would you trust your bank account to a 99% accurate fingerprint classifier. . .

P.S: We can produce a fake fingerprint from your cup of coffee this morning. How about now?

Background

What's available out there. . .

- Many *feature extraction* techniques do exist for nearly all applications
- Many *classifiers* readily available, so which is the “best” feature–classifier pair?
- Options:
 - Choose wisely (but don't optimize for one dataset)
 - Use more than one pair, thus combine different features with different classifiers.

It all boils down to what's good enough for you! Would you trust your bank account to a 99% accurate fingerprint classifier. . .

P.S: We can produce a fake fingerprint from your cup of coffee this morning. How about now?

Background

What's available out there. . .

- Many *feature extraction* techniques do exist for nearly all applications
- Many *classifiers* readily available, so which is the “best” feature–classifier pair?
- Options:
 - Choose wisely (but don't optimize for one dataset)
 - Use more than one pair, thus combine different features with different classifiers.

It all boils down to what's good enough for you! Would you trust your bank account to a 99% accurate fingerprint classifier. . .

P.S: We can produce a fake fingerprint from your cup of coffee this morning. How about now?

Background

What's available out there. . .

- Many *feature extraction* techniques do exist for nearly all applications
- Many *classifiers* readily available, so which is the “best” feature–classifier pair?
- Options:
 - Choose wisely (but don't optimize for one dataset)
 - Use more than one pair, thus combine different features with different classifiers.

It all boils down to what's good enough for you! Would you trust your bank account to a 99% accurate fingerprint classifier. . .

P.S: We can produce a fake fingerprint from your cup of coffee this morning. How about now?

Outline

- 1 Introduction
 - Motivation
 - Background
 - **New to our Approach**
- 2 Rank Distance Aggregation
 - Rankings
 - Rank Distance
 - Rank Distance Aggregation
 - Rank Distance Categorization
- 3 Experiments in Text Categorization
 - Tools and Setting
- 4 Summary and Conclusions

New to our Approach

How much do you trust your models?

- Models of your data (i.e. classifier decision boundaries) are intrinsically biased (lines, circles, etc)
- ... and many times are simply *wrong*
- In a classical setting like text classification they associate probabilities or confidences to the set of possible topics (classes). How much should you “trust” these values?
- **NEW:** We build **rankings** out of the classifier outputs and discard the values.
- **NEW:** We use these **rankings** to assign documents to one (or a few) of the topics

How? Bare with me!

New to our Approach

How much do you trust your models?

- Models of your data (i.e. classifier decision boundaries) are intrinsically biased (lines, circles, etc)
- ...and many times are simply *wrong*
- In a classical setting like text classification they associate probabilities or confidences to the set of possible topics (classes). How much should you “trust” these values?
- **NEW:** We build **rankings** out of the classifier outputs and discard the values.
- **NEW:** We use these **rankings** to assign documents to one (or a few) of the topics

How? Bare with me!

New to our Approach

How much do you trust your models?

- Models of your data (i.e. classifier decision boundaries) are intrinsically biased (lines, circles, etc)
- ... and many times are simply *wrong*
- In a classical setting like text classification they associate probabilities or confidences to the set of possible topics (classes). How much should you “trust” these values?
- **NEW:** We build **rankings** out of the classifier outputs and discard the values.
- **NEW:** We use these **rankings** to assign documents to one (or a few) of the topics

How? Bare with me!

New to our Approach

How much do you trust your models?

- Models of your data (i.e. classifier decision boundaries) are intrinsically biased (lines, circles, etc)
- ... and many times are simply *wrong*
- In a classical setting like text classification they associate probabilities or confidences to the set of possible topics (classes). How much should you “trust” these values?
- **NEW:** We build **rankings** out of the classifier outputs and discard the values.
- **NEW:** We use these **rankings** to assign documents to one (or a few) of the topics

How? Bare with me!

New to our Approach

How much do you trust your models?

- Models of your data (i.e. classifier decision boundaries) are intrinsically biased (lines, circles, etc)
- ... and many times are simply *wrong*
- In a classical setting like text classification they associate probabilities or confidences to the set of possible topics (classes). How much should you “trust” these values?
- **NEW:** We build **rankings** out of the classifier outputs and discard the values.
- **NEW:** We use these **rankings** to assign documents to one (or a few) of the topics

How? Bare with me!

New to our Approach

How much do you trust your models?

- Models of your data (i.e. classifier decision boundaries) are intrinsically biased (lines, circles, etc)
- ... and many times are simply *wrong*
- In a classical setting like text classification they associate probabilities or confidences to the set of possible topics (classes). How much should you “trust” these values?
- **NEW:** We build **rankings** out of the classifier outputs and discard the values.
- **NEW:** We use these **rankings** to assign documents to one (or a few) of the topics

How? Bare with me!

Outline

- 1 Introduction
 - Motivation
 - Background
 - New to our Approach
- 2 Rank Distance Aggregation
 - Rankings
 - Rank Distance
 - Rank Distance Aggregation
 - Rank Distance Categorization
- 3 Experiments in Text Categorization
 - Tools and Setting
- 4 Summary and Conclusions

Raking

- Rankings express a subjective order of preference; they are very natural to us (competitions, public opinion surveys).
- The underlying subjective criteria for creating rankings can be very different, and not even applicable to all the contenders.
- Usually they account for a small number of the rank-able objects.
- A longer ranking usually suggests a more thorough criterion.
- **Formally:** for a set of document topics $\mathcal{U} = \{1, 2, \dots, \#\mathcal{U}\}$, a ranking over \mathcal{U} is an ordered list: $\tau = (x_1 > x_2 > \dots > x_d)$, where $x_i \in \mathcal{U}$ for all $1 \leq i \leq d$, $x_i \neq x_j$ for all $1 \leq i \neq j \leq d$, and $>$ a strict ordering relation on the set $\{x_1, x_2, \dots, x_d\}$.

Raking

- Rankings express a subjective order of preference; they are very natural to us (competitions, public opinion surveys).
- The underlying subjective criteria for creating rankings can be very different, and not even applicable to all the contenders.
- Usually they account for a small number of the rank-able objects.
- A longer ranking usually suggests a more thorough criterion.
- **Formally:** for a set of document topics $\mathcal{U} = \{1, 2, \dots, \#\mathcal{U}\}$, a ranking over \mathcal{U} is an ordered list: $\tau = (x_1 > x_2 > \dots > x_d)$, where $x_i \in \mathcal{U}$ for all $1 \leq i \leq d$, $x_i \neq x_j$ for all $1 \leq i \neq j \leq d$, and $>$ a strict ordering relation on the set $\{x_1, x_2, \dots, x_d\}$.

Raking

- Rankings express a subjective order of preference; they are very natural to us (competitions, public opinion surveys).
- The underlying subjective criteria for creating rankings can be very different, and not even applicable to all the contenders.
- Usually they account for a small number of the rank-able objects.
- A longer ranking usually suggests a more thorough criterion.
- **Formally:** for a set of document topics $\mathcal{U} = \{1, 2, \dots, \#\mathcal{U}\}$, a ranking over \mathcal{U} is an ordered list: $\tau = (x_1 > x_2 > \dots > x_d)$, where $x_i \in \mathcal{U}$ for all $1 \leq i \leq d$, $x_i \neq x_j$ for all $1 \leq i \neq j \leq d$, and $>$ a strict ordering relation on the set $\{x_1, x_2, \dots, x_d\}$.

Raking

- Rankings express a subjective order of preference; they are very natural to us (competitions, public opinion surveys).
- The underlying subjective criteria for creating rankings can be very different, and not even applicable to all the contenders.
- Usually they account for a small number of the rank-able objects.
- A longer ranking usually suggests a more thorough criterion.
- **Formally:** for a set of document topics $\mathcal{U} = \{1, 2, \dots, \#\mathcal{U}\}$, a ranking over \mathcal{U} is an ordered list: $\tau = (x_1 > x_2 > \dots > x_d)$, where $x_i \in \mathcal{U}$ for all $1 \leq i \leq d$, $x_i \neq x_j$ for all $1 \leq i \neq j \leq d$, and $>$ a strict ordering relation on the set $\{x_1, x_2, \dots, x_d\}$.

Raking

- Rankings express a subjective order of preference; they are very natural to us (competitions, public opinion surveys).
- The underlying subjective criteria for creating rankings can be very different, and not even applicable to all the contenders.
- Usually they account for a small number of the rank-able objects.
- A longer ranking usually suggests a more thorough criterion.
- **Formally:** for a set of document topics $\mathcal{U} = \{1, 2, \dots, \#\mathcal{U}\}$, a ranking over \mathcal{U} is an ordered list: $\tau = (x_1 > x_2 > \dots > x_d)$, where $x_i \in \mathcal{U}$ for all $1 \leq i \leq d$, $x_i \neq x_j$ for all $1 \leq i \neq j \leq d$, and $>$ a strict ordering relation on the set $\{x_1, x_2, \dots, x_d\}$.

Outline

- 1 Introduction
 - Motivation
 - Background
 - New to our Approach
- 2 Rank Distance Aggregation
 - Rankings
 - Rank Distance
 - Rank Distance Aggregation
 - Rank Distance Categorization
- 3 Experiments in Text Categorization
 - Tools and Setting
- 4 Summary and Conclusions

Order of a topic in a ranking

Context:

- $\mathcal{U} = \{1, 2, \dots, \#\mathcal{U}\}$ (document topics)
- $\sigma = (x_1 > x_2 > \dots > x_n), x_i \in \mathcal{U}$ (one opinion, e.g. one classifier output)
- Order of topic x in ranking σ is:

$$\text{ord}(\sigma, x) = |\text{length}(\sigma) - \sigma(x)| = |n - \sigma(x)|$$

E.g. for $\sigma = (x_1 > x_2 > x_3)$, $\sigma(x_2) = 2$ and
 $\text{ord}(\sigma, x_2) = |3 - 2| = 1$

- By convention, if $x \in \mathcal{U} \setminus \sigma$, we have $\text{ord}(\sigma, x) = 0$.

Order of a topic in a ranking

Context:

- $\mathcal{U} = \{1, 2, \dots, \#\mathcal{U}\}$ (document topics)
- $\sigma = (x_1 > x_2 > \dots > x_n), x_i \in \mathcal{U}$ (one opinion, e.g. one classifier output)
- Order of topic x in ranking σ is:

$$\text{ord}(\sigma, x) = |\text{length}(\sigma) - \sigma(x)| = |n - \sigma(x)|$$

E.g. for $\sigma = (x_1 > x_2 > x_3)$, $\sigma(x_2) = 2$ and
 $\text{ord}(\sigma, x_2) = |3 - 2| = 1$

- By convention, if $x \in \mathcal{U} \setminus \sigma$, we have $\text{ord}(\sigma, x) = 0$.

Order of a topic in a ranking

Context:

- $\mathcal{U} = \{1, 2, \dots, \#\mathcal{U}\}$ (document topics)
- $\sigma = (x_1 > x_2 > \dots > x_n), x_i \in \mathcal{U}$ (one opinion, e.g. one classifier output)
- Order of topic x in ranking σ is:

$$\text{ord}(\sigma, x) = |\text{length}(\sigma) - \sigma(x)| = |n - \sigma(x)|$$

E.g. for $\sigma = (x_1 > x_2 > x_3)$, $\sigma(x_2) = 2$ and
 $\text{ord}(\sigma, x_2) = |3 - 2| = 1$

- By convention, if $x \in \mathcal{U} \setminus \sigma$, we have $\text{ord}(\sigma, x) = 0$.

Rank Distance

Then: For two rankings σ and τ over the same set of topics \mathcal{U} , we define the *Rank Distance* between them as:

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

Remember:

$$\text{ord}(\sigma, x) = |\text{length}(\sigma) - \sigma(x)|$$

Theorem

Δ is a distance function.

Rank Distance

Then: For two rankings σ and τ over the same set of topics \mathcal{U} , we define the *Rank Distance* between them as:

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

Rationale:

- Ranking differences on the highly ranked objects should have a larger impact than disagreements on the lower ranked objects
- Longer rankings should be justified (tricky, with the benefit of extra expressibility)
- Computing is straight-forward and linear in the number of objects of the two rankings (usually much lower than the total number of universe objects)

Rank Distance

Then: For two rankings σ and τ over the same set of topics \mathcal{U} , we define the *Rank Distance* between them as:

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

Rationale:

- Ranking differences on the highly ranked objects should have a larger impact than disagreements on the lower ranked objects
- Longer rankings should be justified (tricky, with the benefit of extra expressibility)
- Computing is straight-forward and linear in the number of objects of the two rankings (usually much lower than the total number of universe objects)

Rank Distance

Then: For two rankings σ and τ over the same set of topics \mathcal{U} , we define the *Rank Distance* between them as:

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

Rationale:

- Ranking differences on the highly ranked objects should have a larger impact than disagreements on the lower ranked objects
- Longer rankings should be justified (tricky, with the benefit of extra expressibility)
- Computing is straight-forward and linear in the number of objects of the two rankings (usually much lower than the total number of universe objects)

Outline

- 1 Introduction
 - Motivation
 - Background
 - New to our Approach
- 2 Rank Distance Aggregation
 - Rankings
 - Rank Distance
 - Rank Distance Aggregation
 - Rank Distance Categorization
- 3 Experiments in Text Categorization
 - Tools and Setting
- 4 Summary and Conclusions

Rank Distance Aggregation

How to be fair to all the rankings?

- From the k classifiers outputs we compute a **multiset of rankings**:

$$\mathcal{T} = \{\tau_1, \tau_2, \dots, \tau_k\}$$

E.g. in text classification we used 4 classifiers, which produced 4 rankings per evaluated document.

- The rank-distance from a ranking σ to multiset \mathcal{T} is:

$$\Delta(\sigma, \mathcal{T}) = \sum_{\tau \in \mathcal{T}} \Delta(\sigma, \tau).$$

Rank Distance Aggregation

How to be fair to all the rankings?

- From the k classifiers outputs we compute a **multiset of rankings**:

$$\mathcal{T} = \{\tau_1, \tau_2, \dots, \tau_k\}$$

E.g. in text classification we used 4 classifiers, which produced 4 rankings per evaluated document.

- The rank-distance from a ranking σ to multiset \mathcal{T} is:

$$\Delta(\sigma, \mathcal{T}) = \sum_{\tau \in \mathcal{T}} \Delta(\sigma, \tau).$$

Rank Distance Aggregation

How to be fair to all the rankings?

To aggregate \mathcal{T} into a single ranking:

- Look for a ranking σ of minimal rank-distance to \mathcal{T}
- i.e. minimize the sum:

$$\Delta(\sigma, \mathcal{T}) = \sum_{\tau \in \mathcal{T}} \Delta(\sigma, \tau).$$

- We call such a σ a Rank Distance Aggregation of \mathcal{T} , and we call the set of such rankings: $agr(\mathcal{T})$
- Computing $agr(\mathcal{T})$ is polynomial in the number of objects that appear at least once in the multiset \mathcal{T} , with complexity $\mathcal{O}((2x + 2)n^4)$, where x is the size of $agr(\mathcal{T})$ and n is usually very small (say less than 10)

Rank Distance Aggregation

How to be fair to all the rankings?

To aggregate \mathcal{T} into a single ranking:

- Look for a ranking σ of minimal rank-distance to \mathcal{T}
- i.e. minimize the sum:

$$\Delta(\sigma, \mathcal{T}) = \sum_{\tau \in \mathcal{T}} \Delta(\sigma, \tau).$$

- We call such a σ a Rank Distance Aggregation of \mathcal{T} , and we call the set of such rankings: $agr(\mathcal{T})$
- Computing $agr(\mathcal{T})$ is polynomial in the number of objects that appear at least once in the multiset \mathcal{T} , with complexity $\mathcal{O}((2x + 2)n^4)$, where x is the size of $agr(\mathcal{T})$ and n is usually very small (say less than 10)

Rank Distance Aggregation

How to be fair to all the rankings?

To aggregate \mathcal{T} into a single ranking:

- Look for a ranking σ of minimal rank-distance to \mathcal{T}
- i.e. minimize the sum:

$$\Delta(\sigma, \mathcal{T}) = \sum_{\tau \in \mathcal{T}} \Delta(\sigma, \tau).$$

- We call such a σ a Rank Distance Aggregation of \mathcal{T} , and we call the set of such rankings: $agr(\mathcal{T})$
- Computing $agr(\mathcal{T})$ is polynomial in the number of objects that appear at least once in the multiset \mathcal{T} , with complexity $\mathcal{O}((2x + 2)n^4)$, where x is the size of $agr(\mathcal{T})$ and n is usually very small (say less than 10)

Rank Distance Aggregation

How to be fair to all the rankings?

To aggregate \mathcal{T} into a single ranking:

- Look for a ranking σ of minimal rank-distance to \mathcal{T}
- i.e. minimize the sum:

$$\Delta(\sigma, \mathcal{T}) = \sum_{\tau \in \mathcal{T}} \Delta(\sigma, \tau).$$

- We call such a σ a Rank Distance Aggregation of \mathcal{T} , and we call the set of such rankings: $agr(\mathcal{T})$
- Computing $agr(\mathcal{T})$ is polynomial in the number of objects that appear at least once in the multiset \mathcal{T} , with complexity $\mathcal{O}((2x + 2)n^4)$, where x is the size of $agr(\mathcal{T})$ and n is usually very small (say less than 10)

Outline

- 1 Introduction
 - Motivation
 - Background
 - New to our Approach
- 2 Rank Distance Aggregation
 - Rankings
 - Rank Distance
 - Rank Distance Aggregation
 - Rank Distance Categorization
- 3 Experiments in Text Categorization
 - Tools and Setting
- 4 Summary and Conclusions

Rank Distance Categorization (RDC)

Let:

- $\mathcal{U} = \{1, 2, \dots, \#\mathcal{U}\}$ be a set of document topics
- $\mathcal{T} = \{\tau_1, \tau_2, \dots, \tau_k\}$ be a multiset of rankings computed from classifier outputs for a certain document
- $agr(\mathcal{T})$ be the set of all rankings with minimal distance to \mathcal{T}

Then:

- The topic predicted by the RDC method for that particular document is the one that occupies most frequently the first position in the rankings of $agr(\mathcal{T})$
- RDC is Voting on $agr(\mathcal{T})$

Rank Distance Categorization (RDC)

Let:

- $\mathcal{U} = \{1, 2, \dots, \#\mathcal{U}\}$ be a set of document topics
- $\mathcal{T} = \{\tau_1, \tau_2, \dots, \tau_k\}$ be a multiset of rankings computed from classifier outputs for a certain document
- $agr(\mathcal{T})$ be the set of all rankings with minimal distance to \mathcal{T}

Then:

- The topic predicted by the RDC method for that particular document is the one that occupies most frequently the first position in the rankings of $agr(\mathcal{T})$
- RDC is Voting on $agr(\mathcal{T})$

Rank Distance Categorization (RDC)

Let:

- $\mathcal{U} = \{1, 2, \dots, \#\mathcal{U}\}$ be a set of document topics
- $\mathcal{T} = \{\tau_1, \tau_2, \dots, \tau_k\}$ be a multiset of rankings computed from classifier outputs for a certain document
- $agr(\mathcal{T})$ be the set of all rankings with minimal distance to \mathcal{T}

Then:

- The topic predicted by the RDC method for that particular document is the one that occupies most frequently the first position in the rankings of $agr(\mathcal{T})$
- RDC is Voting on $agr(\mathcal{T})$

Outline

- 1 Introduction
 - Motivation
 - Background
 - New to our Approach
- 2 Rank Distance Aggregation
 - Rankings
 - Rank Distance
 - Rank Distance Aggregation
 - Rank Distance Categorization
- 3 Experiments in Text Categorization
 - Tools and Setting
- 4 Summary and Conclusions

Categorization Task

- We used the *rainbow* text classification tool, which is available for most Linux systems (e.g. in default repositories of Ubuntu)
- Corpus: a collection of 20,000 texts covering 20 topics
- Classifiers: Naive Bayes, TF-IDF/Rocchio, Probabilistic Indexing, K-Nearest Neighbor
- Since the number of *training* documents greatly influences the accuracy of most classifiers, we chose 7 different training scenarios: N random documents, per class, where:

$$N \in \{2, 5, 10, 20, 50, 100, 500\}$$

- *Testing*: 500 new documents, per class

Categorization Task

- We used the *rainbow* text classification tool, which is available for most Linux systems (e.g. in default repositories of Ubuntu)
- Corpus: a collection of 20,000 texts covering 20 topics
- Classifiers: Naive Bayes, TF-IDF/Rocchio, Probabilistic Indexing, K-Nearest Neighbor
- Since the number of *training* documents greatly influences the accuracy of most classifiers, we chose 7 different training scenarios: N random documents, per class, where:

$$N \in \{2, 5, 10, 20, 50, 100, 500\}$$

- *Testing*: 500 new documents, per class

Categorization Task

- We used the *rainbow* text classification tool, which is available for most Linux systems (e.g. in default repositories of Ubuntu)
- Corpus: a collection of 20,000 texts covering 20 topics
- Classifiers: Naive Bayes, TF-IDF/Rocchio, Probabilistic Indexing, K-Nearest Neighbor
- Since the number of *training* documents greatly influences the accuracy of most classifiers, we chose 7 different training scenarios: N random documents, per class, where:

$$N \in \{2, 5, 10, 20, 50, 100, 500\}$$

- *Testing*: 500 new documents, per class

Categorization Task

- We used the *rainbow* text classification tool, which is available for most Linux systems (e.g. in default repositories of Ubuntu)
- Corpus: a collection of 20,000 texts covering 20 topics
- Classifiers: Naive Bayes, TF-IDF/Rocchio, Probabilistic Indexing, K-Nearest Neighbor
- Since the number of *training* documents greatly influences the accuracy of most classifiers, we chose 7 different training scenarios: N random documents, per class, where:

$$N \in \{2, 5, 10, 20, 50, 100, 500\}$$

- *Testing*: 500 new documents, per class

Categorization Task

- We used the *rainbow* text classification tool, which is available for most Linux systems (e.g. in default repositories of Ubuntu)
- Corpus: a collection of 20,000 texts covering 20 topics
- Classifiers: Naive Bayes, TF-IDF/Rocchio, Probabilistic Indexing, K-Nearest Neighbor
- Since the number of *training* documents greatly influences the accuracy of most classifiers, we chose 7 different training scenarios: N random documents, per class, where:

$$N \in \{2, 5, 10, 20, 50, 100, 500\}$$

- *Testing*: 500 new documents, per class

Results: at most 10 training docs per class

Classifiers	2pc	5pc	10pc
TFIDF	<u>79.23</u>	70.46	<u>93.10</u>
PRIND	42.56	56.76	71.30
KNN	71.90	74.86	75.36
NBAYES	75.23	76.26	92.53
Voting	75.50	77.96	91.69
Product	75.50	77.00	92.73
Sum	74.90	<u>81.09</u>	92.66
Max	75.06	80.79	92.56
Min	74.13	72.80	85.60
Median	76.96	76.13	92.76
Voting on RDA	76.23	77.06	91.86

Precision (%). **Underlined** is the maximum, **bold** is everything closer than 0.50% to the maximum.

Results: 20 training docs per class and above

Classifiers	20pc	50pc	100pc	500pc
TFIDF	<u>92.83</u>	91.53	91.63	91.76
PRIND	77.19	82.86	83.86	86.86
KNN	81.83	89.16	89.83	88.96
NBAYES	91.63	91.19	91.03	92.00
Voting	92.00	92.09	91.93	92.16
Product	92.26	92.06	91.56	91.40
Sum	92.46	91.66	91.33	92.30
Max	91.36	91.40	91.00	91.96
Min	86.36	88.93	90.60	91.70
Median	91.96	91.39	90.96	92.23
Voting on RDA	<u>92.66</u>	<u>92.56</u>	<u>92.16</u>	<u>92.40</u>

Precision (%). **Underlined** is the maximum, **bold** is everything closer than 0.50% to the maximum.

Conclusions

- If the number of training documents is relatively small, the base classifiers produce unreliable results (as expected), and aggregations have lower precision than some of the classifiers. However, . . .
- If the training set is sufficiently large, aggregations usually do better than individual classifiers, and at least as well as the local best (which may be different for different classes)

Conclusions

- If the number of training documents is relatively small, the base classifiers produce unreliable results (as expected), and aggregations have lower precision than some of the classifiers. However, . . .
- If the training set is sufficiently large, aggregations usually do better than individual classifiers, and at least as well as the local best (which may be different for different classes)

Summary

- This article presents a series of experiments with text categorization methods, combined by the common, fixed, classifier fusion rules and by the new Voting on the Rank-Distance Aggregation set.
- We use the *rainbow* document classification tool to output the results of 4 different text categorization methods, and we aggregate by 6 established fixed fusion rules.
- We compare these results with Voting on the Rank Distance Aggregation set, which demonstrates robust performance.

MULTUMESC!

Lingvistica Matematica si Computationala

Liviu P. Dinu,

ldinu@fmi.unibuc.ro

University of Bucharest

Center for Computational Linguistics,

Faculty of Mathematics and Computer Science

nlp.unibuc.ro



Cognate Analysis

Introduction

- Cognates are words in different languages having the same etymology and a common ancestor.
- Investigating pairs of cognates is very useful
 - in historical and comparative linguistics,
 - in the study of language relatedness (Ng et al., 2010), phylogenetic inference (Atkinson et al., 2005)
 - in identifying how and to what extent languages change over time.

Introduction (2)

- In other several research areas, such as language acquisition, bilingual word recognition (Dijkstra et al., 2012), corpus linguistics (Simard et al., 1992), cross-lingual information retrieval (Buckley et al., 1997) and machine translation (Kondrak et al., 2003), the condition of common etymology is usually not essential and cognates are regarded as words with high cross-lingual meaning and orthographic or phonetic similarity.



An Assessment of String Similarity Methods for Cognate Identification (Qualico 2012)

Overview

The main goal of this part is to investigate and compare the performance of several manually-designed procedures (the *Manhattan*, *Jaro*, *Jaro/Winkler* distances and the *ALINE* phonetic aligner) and data-driven models (*Pair Hidden Markov Model*, *Dynamic Bayesian Networks*, and a measuring string similarity system, inspired by biological sequence alignment) in the task of cognate identification.

Motivation (1)

- The study of language relatedness has been historically based on the detection of strict or genetic cognates (words deriving “vertically” from the same predecessor)
- Cognate identification has been successfully applied to a multitude of areas of computational linguistics and NLP: dialectology, proto-language reconstruction, phylogenetic inference, machine translation, semantic word clustering, lexical induction

Motivation (2)

- Approaches to the cognate identification problem include *static procedures* and *learning systems*
- Our results in comparing the two suggest that learning algorithms outperform static procedures and that orthographic learning methods may outperform static learning methods, accurately detecting traces of sound change left in the orthography.

Static methods (1)

- I. Manhattan distance - metric that calculates the distance between two points in an n-dimensional space:

if $\mathbf{p}=(p_1,p_2,\dots,p_n)$ and $\mathbf{q}=(q_1,q_2,\dots,q_n)$, then

$$M(\mathbf{p},\mathbf{q})=\sum |p_i-q_i|$$

We have computed this distance on the vectorial representation of each word (written in Roman alphabet) through the computation of the occurrences of each letter (0 = no occurrence)

Static methods (2)

II. The **Jaro** and **Jaro/Winkler** distances calculate the similarity between short strings. Given 2 strings, $S_1=(a_1, \dots, a_m)$ and $S_2=(b_1, \dots, b_n)$, c =the no. of common characters between them, and t =the no. of char. transposition, then :

$$\mathbf{JD(S_1, S_2) = 1/3 * (c/m + c/n + (c-t)/c)}$$

and

$$\mathbf{JWD(S_1, S_2) = JD(S_1, S_2) + L * P * (1 - JD(S_1, S_2))}$$

Static methods (3)

where L is the length of the longest common prefix of S_1 and S_2 , $L < 5$, and $P = 0.1$ is a scaling factor

III. **ALINE** is a manually-designed algorithm developed by Kondrak for sequence alignment. It works on phonetic segments and calculates their similarity through local alignment. Twelve phonetic features are considered and weighted according to their manually-established saliency.

Data-driven models (1)

I. Pair Hidden Markov Model (PHMM)

- A suite of PHMM's utilising alignment and log-odds Viterbi algorithms to calculate word-pair similarity
- Training dataset of 120,000 word pairs from the "*Comparative Indo-European corpus by Dyen et al.*"
- Test dataset of 10 language pairs extracted from the 200-word Swadesh lists

Data-driven models (2)

II. Dynamic Bayesian Network (DBN)

- Training dataset of 180,000 word pairs from the "Comparative Indo-European... "
- The authors set parameters for their model by building a development dataset of 3 language pairs representing distant (*Italian-Croatian*), medium (*Spanish-Romanian*) and close (*Polish-Russian*) relatedness.

Data-driven models (3)

III. A string similarity measuring system

- Training dataset of 650 word pairs from the "*Comparative...*" classified as certain cognates.
- Pairwise global alignment applied to cognate pairs, with the aid of a *novel linguistic-inspired substitution matrix*.
- Increasingly complex scoring matrices inferred by several learning techniques:

Data-driven models (4)

Absolute Frequency Ratio, Pointwise Mutual Information, PAM-like matrices

- The produced substitution matrices were used to measure word similarity, employing global and local alignment algorithms and a novel family of parametrised string similarity measures
- The test dataset was the same as the one utilised by PHMM

Experimental design

- Training dataset extracted from the "*Comparative Indoeuropean Database by Dyen*" – 84 Swadesh lists, each containing 200 universal words(no diacritics,Roman alphabet,clustered by meaning and cognateness)
- Test dataset: Swadesh lists for *English, German, French, Latin and Albanian*,(ortographic format with phonetic transcription) + cognateness info.
- Evaluation methodology: the *11-point interpolated average precision* method

Experimental results

- We compared the results achieved by the manually-designed procedures and the data-driven models.

Languages		Cognate proportion	MD	NEDIT	JD	JWD	ALINE	AFR	PHMM	DBN	PMI	PAM like
English	German	0.590	0.883	0.907	0.912	0.912	0.912	0.909	0.930	0.927	0.925	0.934
French	Latin	0.560	0.866	0.921	0.908	0.912	0.862	0.924	0.934	0.923	0.925	0.924
English	Latin	0.290	0.605	0.703	0.676	0.676	0.732	0.776	0.803	0.822	0.795	0.826
German	Latin	0.290	0.564	0.591	0.568	0.564	0.705	0.706	0.730	0.772	0.745	0.772
English	French	0.275	0.676	0.659	0.693	0.693	0.623	0.768	0.812	0.802	0.790	0.830
French	German	0.245	0.545	0.498	0.567	0.551	0.534	0.700	0.734	0.645	0.757	0.788
Albanian	Latin	0.195	0.440	0.561	0.566	0.566	0.630	0.584	0.680	0.676	0.676	0.721
Albanian	French	0.165	0.369	0.499	0.526	0.538	0.610	0.557	0.653	0.658	0.621	0.625
Albanian	German	0.125	0.244	0.207	0.233	0.242	0.369	0.486	0.379	0.420	0.470	0.552
Albanian	English	0.100	0.229	0.289	0.272	0.272	0.302	0.280	0.382	0.446	0.404	0.518
AVERAGE		0.284	0.542	0.584	0.592	0.593	0.628	0.669	0.704	0.709	0.711	0.749
Standard deviation		0.168	0.229	0.231	0.225	0.224	0.193	0.197	0.194	0.176	0.173	0.144
Variance		0.028	0.052	0.054	0.051	0.050	0.037	0.039	0.038	0.031	0.030	0.021
Median		0.260	0.555	0.576	0.568	0.565	0.627	0.703	0.732	0.724	0.751	0.780

Table 1: 11-point interpolated average precision for several methods

Experimental results

- The table above shows an assessment in cognate identification of the comparable orthographic and phonetic methods in terms of 11-point interpolated average precision over 10 language pairs.
- The baseline we used was the edit-distance with unitary costs normalised by the length of the longer string (NEDIT)

Experimental results

- We see that the results obtained for ALINE, PHMM, DBN and PAM-like are as reported in the literature.
- PAM-like shows the best results achieved with the first training dataset, of about 650 cognate pairs

Experimental results

- The Manhattan distance produces a negative outcome, showing a performance lower than NEDIT
- The Jaro and Jaro/Winkler distances generate results only slightly higher than NEDIT
- The Absolute Frequency Ratio (AFR) performs a little better and ALINE

Experimental results

- The Pairwise Mutual Information (PMI) reaches good results, comparable to those by the best PHMM and DBN
- The PAM-like method achieves the highest accuracy in cognate identification, with an average precision approx. 5% higher than PHMM, DBN and PMI, 18% higher than ALINE, and 28% higher than NEDIT.

Conclusion and future work

- Results suggest that the performance of the PAM-like system is more stable than the other methods analysed.
- Even though in cognate identification, a phonetic approach is supposed to be more accurate than an ortographic one, recent studies have shown the contrary.

Conclusion and future work

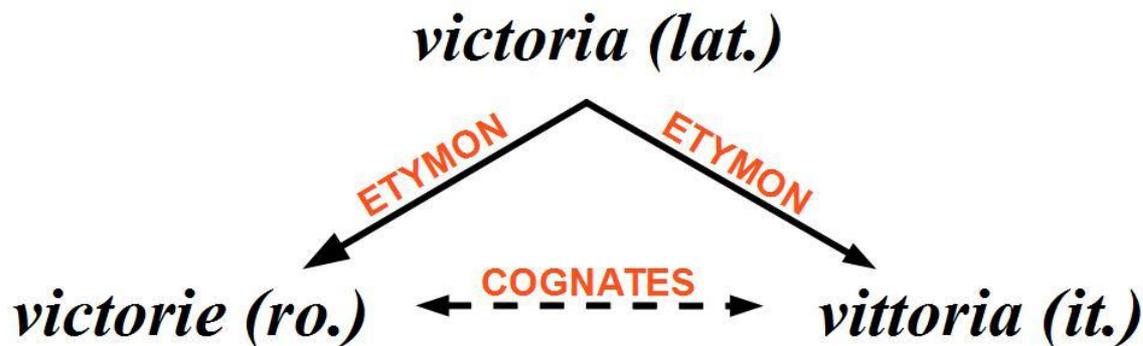
- Our investigation has confirmed this tendency, suggesting that phonetic changes can leave enough traces in the word orthography, to be used by orthographic learning systems.
- Our future plans include the investigation of other learning techniques developed for biological sequence analysis and their application to cognate identification. We are particularly interested in training BLOSUM-like matrices



**Cognate detection (ACL 2014,
LREC 2014, RANLP 2013)**

Cognate detection (ACL 2014, LREC 2014, RANLP 2013)

- Cognates are words in different languages having the same etymology and a common ancestor.
- We identify cognates using electronic dictionaries.
- We build a dataset of multilingual cognates for the Romanian lexicon.



Romanian cognate

- We focus on the Romanian language and first we investigate its cognate pairs with two other Romance languages, French and Italian.
- We believe this comparison is interesting for the following reason:
 - the two related languages differ significantly with respect to their orthographic depth: the mapping rules between graphemes and phonemes are more complex for French, which has a deep orthography, than for Italian, which has a highly phonemic orthography.

Ethymologies

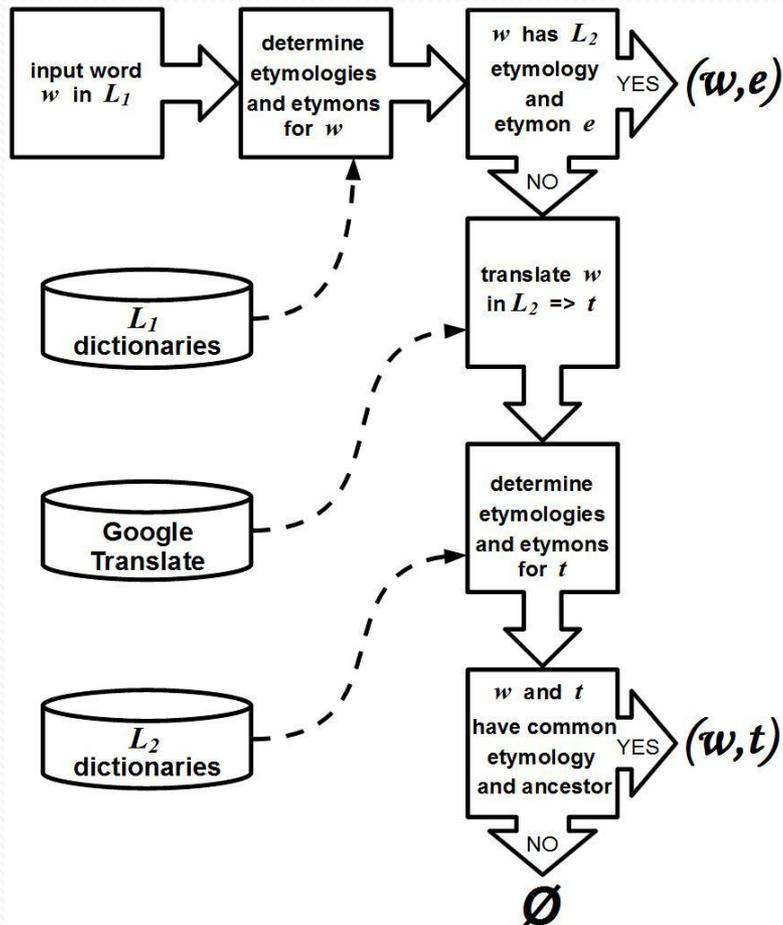
- We identify the etymologies and etymons of the Romanian words using dexonline 1 machine-readable dictionary, which is an aggregator for over 30 Romanian dictionaries.
- By parsing its definitions, we are able to automatically extract information regarding words' etymologies and etymons.

Method

- After determining the etymologies of the Romanian words, we translate in French all words without French etymology and in Italian all words without Italian etymology using Google Translate.
- We consider cognate candidates pairs formed of Romanian words and their translations.

- Using French and Italian dictionaries, we extract etymology-related information for French and Italian words.
- To identify cognates we compare, for each pair of candidates, the etymologies and the etymons. If they match, we identify the words as being cognates

Algorithm and data



- We use the lexicon provided by DexOnline (<http://dexonline.ro>).
 - ~ 137,000 lexemes
- We identify cognate pairs between Romanian and five other languages:
 - French, Italian, Spanish, Portuguese, Turkish.

The Corpus

- We apply our method on a high-quality Romanian corpus comprising of the transcription of the parliamentary debates held between 1996 and 2007 in the Romanian Parliament, recently proposed in (Grozea, 2012)
- For preprocessing this corpus, we removed words that are irrelevant for our investigation, such as dates and numbers and all the transcribers' descriptions of the parliamentary sessions (such as “The session began at 8:40.”), as we focus on the spoken language.

Processing

- We performed word segmentation, using whitespace and punctuation marks as delimiters, we lower-cased all words and we removed stop words, using a list of Romanian stop words provided by Apache Lucene 5 text search engine library .
- We lemmatized the words using dexonline, which provides information regarding the words' inflected forms and enables us to correctly identify lemmas where no part-of-speech or semantic ambiguities arise (in this case we consider the first occurred lemma).

Orthographic Approaches

- We chose some standard distances, another distance that was successfully employed for record linkage and also an original metric in the field of cognates identification, rank distance.
- • Levenshtein distance
- • Rank distance
- Longest common subsequence ratio
- Xdice
- Jaro distance

Evaluation and Results Analysis

	Nwords	Ncognates	
		French	Italian
Type	162,399	77,029	35,581
Token	22,469,290	15,858,140	10,895,298
Lemmas	40,065	17,929	6,768

Table 1: Statistics for the Romanian corpus: the total number of type words, token words and lemmas (in column 1) and the number of type words, token words and lemmas having an etymon or a cognate pair in French (column 2) or in Italian (column 3). It can be noticed that the sum of token words with cognate pairs or etymons in French and Italian is higher than the total number of token words after preprocessing the corpus, due to the fact that many of these words have cognate pairs or etymons in both languages

Evaluation and Results Analysis

- We excerpt from the corpus, for each of the two languages, random samples of 5,000 words which have a cognate pair in the related language and 5,000 which do not have such matching pair.
- We match these latter words with their translations.
- Thus, we obtain a sample of 10,000 pairs of words for Romanian and Italian, 5,000 pairs of cognates and 5,000 pairs of non-cognates.
-

Evaluation

- We obtain a similar set for Romanian and French.
- For each dataset we also consider the version without diacritics.
- We compute the lexical distances for each pair of words, setting various thresholds for identifying cognates.

French																				
th	EDIT				LCSR				RD				JW				XDICE			
	R	P	A	F	R	P	A	F	R	P	A	F	R	P	A	F	R	P	A	F
0.0	06.4	100.0	53.2	12.0	06.4	100.0	53.2	12.0	06.4	100.0	53.2	12.0	06.4	100.0	53.2	12.0	06.4	100.0	53.2	12.0
0.1	08.9	94.3	54.2	16.3	09.3	93.8	54.4	17.0	15.2	87.6	56.5	26.0	41.9	81.1	66.1	55.3	09.4	92.5	54.3	17.0
0.2	24.9	83.2	60.0	38.4	26.4	82.5	60.4	40.0	40.6	83.4	66.3	54.7	71.8	78.6	76.1	75.1	18.1	83.1	57.2	29.8
0.3	47.6	83.1	68.9	60.5	50.3	82.3	69.7	62.4	63.3	81.1	74.3	71.1	88.2	75.9	80.1	81.6	34.0	81.8	63.2	48.0
0.4	68.7	80.6	76.1	74.2	71.8	79.4	76.6	75.4	79.7	78.5	78.9	79.1	95.6	71.1	78.3	81.5	49.1	80.6	68.7	61.0
0.5	84.9	78.2	80.6	81.4	87.1	76.4	80.1	81.4	89.9	75.5	80.3	82.0	98.2	62.7	69.8	76.5	65.4	79.5	74.3	71.8
0.6	91.3	76.0	81.3	83.0	93.2	73.1	79.4	81.9	94.4	71.3	78.2	81.2	99.4	54.3	57.9	70.2	74.7	78.4	77.1	76.5
0.7	94.8	72.9	79.8	82.4	96.4	67.4	74.9	79.3	97.2	65.3	72.7	78.1	99.4	53.3	56.1	69.4	81.8	77.1	78.8	79.4
0.8	98.2	65.1	72.8	78.3	98.8	57.5	63.0	72.7	98.5	58.7	64.6	73.6	99.4	53.2	56.1	69.3	89.9	74.3	79.4	81.4
0.9	99.4	57.1	62.4	72.6	99.7	52.2	54.1	68.5	99.5	54.0	57.3	70.0	99.4	53.2	56.1	69.3	94.5	69.2	76.3	79.9
1.0	100.0	50.0	50.0	66.7	100.0	50.0	50.0	66.7	100.0	50.0	50.0	66.7	100.0	50.0	50.0	66.7	100.0	50.0	50.0	66.7

Italian																				
th	EDIT				LCSR				RD				JW				XDICE			
	R	P	A	F	R	P	A	F	R	P	A	F	R	P	A	F	R	P	A	F
0.0	03.8	100.0	51.9	07.2	03.8	100.0	51.9	07.2	03.8	100.0	51.9	07.2	03.8	100.0	51.9	07.2	03.8	100.0	51.9	07.2
0.1	08.5	71.3	52.5	15.3	08.6	70.0	52.5	15.4	15.7	72.7	54.9	25.9	58.3	70.8	67.1	64.0	15.4	72.4	54.8	25.4
0.2	35.7	70.6	60.4	47.4	36.3	69.1	60.0	47.6	40.8	68.9	61.2	51.2	80.5	67.8	71.1	73.6	33.4	72.9	60.5	45.8
0.3	60.3	70.6	67.6	65.0	61.9	69.7	67.5	65.6	64.1	68.0	67.0	66.0	91.5	66.4	72.6	77.0	47.8	70.6	64.0	57.0
0.4	76.0	68.5	70.6	72.1	77.7	67.6	70.2	72.3	79.6	66.8	70.0	72.6	96.7	63.5	70.5	76.7	61.1	69.2	66.9	64.9
0.5	88.5	67.4	72.8	76.5	90.1	66.1	72.0	76.3	88.5	65.1	70.6	75.0	99.4	58.2	64.0	73.4	72.6	67.7	69.0	70.1
0.6	93.1	66.0	72.6	77.3	94.6	64.0	70.7	76.4	94.2	63.0	69.5	75.5	99.8	52.5	54.7	68.8	80.0	66.9	70.2	72.9
0.7	96.5	64.4	71.6	77.3	97.7	61.0	67.7	75.1	98.0	59.7	66.0	74.2	99.8	51.8	53.4	68.2	85.8	65.9	70.7	74.5
0.8	99.1	59.4	65.7	74.3	99.7	54.4	58.1	70.4	99.3	55.5	59.8	71.2	99.8	51.7	53.3	68.1	92.6	64.4	70.6	76.0
0.9	99.8	54.5	58.2	70.5	99.9	51.3	52.6	67.8	99.7	52.3	54.4	68.6	99.8	51.7	53.3	68.1	96.5	61.5	68.0	75.1
1.0	100.0	50.0	50.0	66.7	100.0	50.0	50.0	66.7	100.0	50.0	50.0	66.7	100.0	50.0	50.0	66.7	100.0	50.0	50.0	66.7

Table 2: Recall (R), precision (P), accuracy (A) and f-score (F) values (computed as percentages) for orthographic measures in the task of cognates identification when diacritics are accounted for

French																				
th	EDIT				LCSR				RD				JW				XDICE			
	R	P	A	F	R	P	A	F	R	P	A	F	R	P	A	F	R	P	A	F
0.0	08.9	100.0	54.4	16.3	08.9	100.0	54.4	16.3	08.9	100.0	54.4	16.3	08.9	100.0	54.4	16.3	08.9	100.0	54.4	16.3
0.1	12.3	94.0	55.8	21.7	12.9	93.2	56.0	22.6	21.4	87.7	59.2	34.4	58.1	80.6	72.0	67.5	13.4	90.3	56.0	23.3
0.2	34.1	81.2	63.1	48.0	35.9	80.6	63.6	49.7	54.6	82.5	71.5	65.7	82.6	77.9	79.6	80.2	28.3	81.8	61.0	42.1
0.3	60.5	82.0	73.6	69.6	62.9	81.0	74.1	70.8	73.4	79.9	77.4	76.5	92.5	74.5	80.4	82.5	48.8	80.6	68.5	60.8
0.4	77.1	79.8	78.8	78.4	79.3	78.2	78.6	78.8	85.4	77.1	80.0	81.1	96.7	69.4	77.0	80.8	63.8	79.5	73.7	70.8
0.5	89.1	77.1	81.4	82.7	90.9	75.0	80.3	82.2	92.3	73.4	79.4	81.8	98.8	60.6	67.3	75.1	76.4	78.5	77.7	77.4
0.6	93.9	74.8	81.1	83.3	95.3	71.2	78.4	81.5	95.5	68.9	76.2	80.0	99.5	53.6	56.7	69.7	82.5	77.3	79.1	79.8
0.7	96.5	71.4	78.9	82.1	97.6	65.3	72.9	78.3	97.8	62.7	69.9	76.4	99.6	52.6	55.0	68.9	87.5	75.6	79.6	81.1
0.8	98.5	63.1	70.5	76.9	99.1	55.8	60.3	71.4	98.9	56.7	61.8	72.1	99.6	52.6	54.9	68.8	93.0	72.2	78.6	81.3
0.9	99.6	55.6	60.0	71.3	99.8	51.6	53.0	68.0	99.7	52.9	55.4	69.1	99.6	52.6	54.9	68.8	96.7	66.6	74.1	78.9
1.0	100.0	50.0	50.0	66.7	100.0	50.0	50.0	66.7	100.0	50.0	50.0	66.7	100.0	50.0	50.0	66.7	100.0	50.0	50.0	66.7

Italian																				
th	EDIT				LCSR				RD				JW				XDICE			
	R	P	A	F	R	P	A	F	R	P	A	F	R	P	A	F	R	P	A	F
0.0	06.7	100.0	53.4	12.6	06.7	100.0	53.4	12.6	06.7	100.0	53.4	12.6	06.7	100.0	53.4	12.6	06.7	100.0	53.4	12.6
0.1	12.2	77.0	54.3	21.0	12.3	75.7	54.2	21.2	17.5	73.8	55.7	28.3	63.8	70.9	68.8	67.1	19.1	74.4	56.2	30.4
0.2	41.4	70.9	62.2	52.3	42.3	69.5	61.9	52.6	43.5	68.6	61.8	53.2	84.9	68.0	72.5	75.5	38.6	72.8	62.1	50.5
0.3	64.6	70.3	68.6	67.3	66.3	69.4	68.6	67.9	66.8	67.9	67.6	67.4	94.0	66.2	73.0	77.7	52.6	70.6	65.3	60.2
0.4	80.1	68.9	72.0	74.1	82.0	67.8	71.5	74.2	82.9	66.7	70.8	74.0	97.7	62.7	69.8	76.4	65.9	69.4	68.4	67.6
0.5	91.8	67.5	73.8	77.8	93.3	66.1	72.7	77.4	91.3	64.9	70.9	75.8	99.6	57.1	62.3	72.6	76.9	68.1	70.4	72.2
0.6	95.4	65.7	72.9	77.8	96.7	63.4	70.5	76.6	95.9	62.2	68.8	75.5	99.9	52.0	53.9	68.4	84.1	67.2	71.6	74.7
0.7	97.8	63.7	71.0	77.1	98.6	59.8	66.2	74.5	98.5	58.5	64.3	73.4	99.9	51.4	52.6	67.8	90.0	66.0	71.9	76.2
0.8	99.4	58.1	63.9	73.4	99.7	53.3	56.2	69.5	99.3	54.2	57.7	70.2	99.9	51.3	52.6	67.8	95.1	63.9	70.7	76.4
0.9	99.9	53.6	56.7	69.7	99.9	50.8	51.6	67.4	99.8	51.7	53.4	68.1	99.9	51.3	52.6	67.8	97.7	60.4	66.8	74.6
1.0	100.0	50.0	50.0	66.7	100.0	50.0	50.0	66.7	100.0	50.0	50.0	66.7	100.0	50.0	50.0	66.7	100.0	50.0	50.0	66.7

Table 3: Recall (R), precision (P), accuracy (A) and f-score (F) values (computed as percentages) for orthographic measures in the task of cognates identification when diacritics are not accounted for

Results for Romanian

Lang	#words	#etymons	#cognates	Et. det. acc.
Fr.	53,347	52,868	479	.966
It.	13,377	9,874	3,503	.980
Sp.	7,780	2,181	5,599	.982
Pt.	10,972	1,318	9,654	.998
Tr.	4,608	2,307	2,301	.996

Results for Romanian

	FR	IT	SP	PT	TR
American	1	-	-	-	-
Arabian	-	10	15	13	4
English	2	57	94	195	158
French	-	547	455	1,925	1,157
German	-	16	14	10	-
Greek	-	221	-	1,366	410
Hebrew	-	-	1	-	-
Italian	1	-	143	238	-
Latin	475	2,606	4,874	5,815	572
Persian	-	1	-	2	-
Polish	-	-	-	2	-
Portuguese	-	3	-	-	-
Provencal	-	1	3	4	-
Russian	-	4	-	6	-
Spanish	-	34	-	72	-
Turkish	-	3	-	6	-
<i>Total</i>	479	3,503	5,599	9,654	2,301

Statistics regarding the common ancestors of the identified cognate pairs

Evaluation

- According to the outcome of our investigation, the edit distance identifies Romanian-French and Romanian-Italian cognates with the highest degree of accuracy, reaching its maximum for a threshold value of 0.5 (and 0.6 for French, when diacritics are accounted for), followed closely by JaroWinkler distance and the longest common subsequence ratio.



MULTUMESC!

Lingvistica Matematica si Computationala

Liviu P. Dinu,

ldinu@fmi.unibuc.ro

University of Bucharest

Center for Computational Linguistics,

Faculty of Mathematics and Computer Science

nlp.unibuc.ro



**Cognates detection &
discrimination, word production
(ACL 2014, 2015)**

Introduction

- Words undergo various changes when entering new languages.
- We assume that rules for adapting foreign words to the orthographic system of the target languages might not have been very well defined in their period of early development, but they may have since become complex and probably language-specific.

Our intuition

- We employed orthographic alignment for identifying pairs of cognates, not only to compute similarity scores, as was previously done, but to use aligned subsequences as features for machine learning algorithms.
- Our intuition is that inferring language-specific rules for aligning words will lead to better performance in the task of cognate identification.

Pairwise similarity

- Word-cognate vs. word-etymon overall pairwise similarity:

Language	Word-etymon pairs		Cognate pairs	
	w/ diacritics	w/o diacritics	w/ diacritics	w/o diacritics
French	.72	.77	.62	.69
Italian	.73	.76	.75	.77
Spanish	.53	.57	.76	.79
Portuguese	.49	.53	.77	.81
Turkish	.63	.69	.74	.76

The orthographic approach

- Over time, sound changes leave traces in the orthography of the words (Delmestri and Cristianini, 2010).
- Orthographic changes undergone by words when entering new languages follow specific patterns.
- We use the Needleman-Wunsch algorithm for sequence alignment, used in computational biology (Needleman and Wunsch, 1970).

Feature extraction

- Features are n-grams of characters around mismatches in the aligned words, n in {1, 2, 3}.
- For a given n, using all i-grams, where i {1, 2, ..., n} leads to better results.
- Word boundaries are marked by \$ symbols

EXAMPLE

x	h	-
s	-	o

EXAMPLE

>	>	>
>	>	>
>	>	>

The alignment

- There are three types of mismatches, corresponding to the following operations: insertion, deletion and substitution.
- For example, for the Romanian word *exhaustiv* and its Italian cognate pair *esaustivo*, the alignment is as follows:

e x h a u s t i v -
e s - a u s t i v o

Types of features

- We experiment with two types of features:
 - n-grams around gaps, i.e., we account only for insertions and deletions;
 - n-grams around any type of mismatch, i.e., we account for all three types of mismatches.
- The second alternative leads to better performance, so we account for all mismatches. As for the length of the grams, we experiment with $n \in \{1, 2, 3\}$

Example

- In order to provide information regarding the position of the features, we mark the beginning and the end of the word with a \$ symbol.
- Thus, for the above-mentioned pair of cognates, (exhaustiv, esaustivo), we extract the following features when $n = 2$:
 - $x>s$ $ex>es$ $xh>s$
 - $h>-$ $xh>s-$ $ha>-a$
 - $->o$ $v->vo$ $-$>o$$

Algorithms and setup

- Naïve Bayes and Support Vector Machines (SVM).
- Training/test sets - 3:1 ratio.
- Grid search & 3-fold CV over the training set to optimize hyperparameters for SVM.
- The system was implemented using the Weka machine learning toolkit (Hall et al, 2009).

Experiment & Data

- We apply our method on an automatically extracted dataset of cognates for four pairs of languages: Romanian-French, Romanian-Italian, Romanian-Spanish and Romanian-Portuguese.
- We discard pairs of words for which the forms across languages are identical (i.e., the Romanian word *matrice* and its Italian cognate *matrice*, having the same form), because these pairs do not provide any orthographic changes to be learned.

Relevant cues

	1 st	2 nd	3 rd	4 th	5 th
IT	iu>io	un>on	l->le	t\$>-\$	-\$>e\$
FR	un>on	ne>n-	iu>io	ti>ti	e\$>-\$
ES	-\$>o\$	ti>ci	->ón	ie>ió	at>ad
PT	ie>ão	aç>aç	ti>çã	i\$>-\$	ã\$>a\$

Table 1: The most relevant orthographic cues for each pair of languages determined on the entire datasets using the χ^2 attribute evaluation method implemented in Weka.

	1 st	2 nd	3 rd	4 th	5 th
IT	-\$>e\$	-\$>o\$	ã\$>a\$	->re	ti>zi
FR	e\$>-\$	un>on	ne>n-	iu>io	ti>ti
ES	-\$>o\$	e\$>-\$	ti>ci	ã\$>a\$	at>ad
PT	-\$>o\$	ã\$>a\$	e\$>-\$	-\$>r\$	-\$>a\$

Table 2: The most frequent orthographic cues for each pair of languages determined on the cognate lists using the raw frequencies.

Experiments and results

- We used the dataset of cognates extracted from DexOnline for the Romance languages.
- 400 pairs of cognates and 400 pairs of non-cognates for each pair of languages.

Language	Naïve Bayes				SVM			
	Precision	Recall	Accuracy	n-grams	Precision	Recall	Accuracy	n-grams
Italian	.727	.930	.790	1	.760	.920	.815	1
French	.813	.910	.820	2	.849	.890	.870	2
Spanish	.793	.920	.840	1	.854	.880	.865	2
Portuguese	.677	.880	.730	2	.709	.780	.730	2

	Naive Bayes				SVM					
	P	R	A	n	P	R	A	n	c	γ
IT	0.72	0.93	79.0	1	0.76	0.92	81.5	1	1	0.10
FR	0.81	0.91	82.0	2	0.84	0.89	87.0	2	10	0.01
ES	0.79	0.92	84.0	1	0.85	0.88	86.5	2	4	0.01
PT	0.67	0.88	73.0	2	0.70	0.78	73.0	2	10	0.01

Table 3: Results for automatic detection of cognates using orthographic alignment. We report the precision (P), recall (R) and accuracy (A) obtained on the test sets and the optimal n -gram values. For SVM we also report the optimal hyperparameters c and γ obtained during cross-validation on the training sets.

	EDIT				LCSR				XDICE				SPSIM			
	P	R	A	t	P	R	A	t	P	R	A	t	P	R	A	t
IT	0.67	0.97	75.0	0.43	0.68	0.91	75.0	0.51	0.66	0.98	74.0	0.21	0.66	0.98	74.5	0.44
FR	0.76	0.93	82.0	0.30	0.76	0.90	81.5	0.42	0.77	0.79	78.0	0.26	0.86	0.83	85.0	0.59
ES	0.77	0.91	82.0	0.56	0.72	0.97	80.0	0.47	0.72	0.99	80.5	0.19	0.81	0.90	85.0	0.64
PT	0.62	0.99	69.5	0.34	0.59	0.99	65.5	0.34	0.57	0.99	63.5	0.10	0.62	0.97	69.0	0.39

Table 4: Comparison with previous methods for automatic detection of cognate pairs based on orthography. We report the precision (P), recall (R) and accuracy (A) obtained on the test sets and the optimal threshold t for discriminating between cognates and non-cognates.

Result Analyse

- The best results are obtained for French and Spanish, while the lowest accuracy is obtained for Portuguese.
- The SVM produces better results for all languages except Portuguese, where the accuracy is equal.
- For Portuguese, both Naive Bayes and SVM misclassify more non-cognates as cognates than viceversa. A possible explanation might be the occurrence, in the dataset, of more remotely related words, which are not labeled as cognates.

Final remarks and conclusion

- We investigate the performance of the method we propose in comparison to previous approaches for automatic detection of cognate pairs based on orthographic similarity.
- Our method performs better than the orthographic metrics considered as individual features.
- Out of the four similarity metrics, SpSim obtains, overall, the best performance. These results support the relevance of accounting for orthographic cues in cognate identification.



**Automatic Discrimination
between Cognates and
Borrowings (ACL 2015)**

Automatic Discrimination between Cognates and Borrowings_(ACL 2015)

- Identifying the type of relationship between words provides a deeper insight into the history of a language and allows a better characterization of language relatedness.
- Natural languages are living eco-systems. They are subject to continuous change due, in part, to the natural phenomena of language contact and borrowing (Campbell, 1998).

Motivation

- According to Hall (1960), there is no such thing as a “pure language” – a language “without any borrowing from a foreign language”.
- Although admittedly regarded as relevant factors in the history of a language (McMahon et al., 2005), borrowings bias the genetic classification of the languages, characterizing them as being closer than they actually are (Minett and Wang, 2003).

“Computerized approaches”

- Thus, the need for discriminating between cognates and borrowings emerges.
- Heggarty (2012) acknowledges the necessity and difficulty of the task, emphasizing the role of the “computerized approaches”
- A borrowing (loanword), is defined by Campbell (1998) as a “lexical item (a word) which has been ‘borrowed’ from another language, a word which originally was not part of the vocabulary of the recipient language but was adopted from some other language and made part of the borrowing language’s vocabulary”

Our approach

- We address here the task of automatically distinguishing between borrowings and cognates:
 - given a pair of words, the task is to determine whether one is a historical descendant of the other, or whether they both share a common ancestor
- To our knowledge, this is the first attempt of this kind.

Strategy

- Input:
 - a pair of words in two different languages (x, y)
- Output:
 - we want to determine whether x and y are cognates or if y is borrowed from x (in other words, x is the etymon of y).

Strategy and parameters

- Aligning the pairs of related words using a string alignment algorithm (Needleman-Wunsch);
- Extracting orthographic features from the aligned words;
- Training a binary classifier to discriminate between the two types of relationship

Features

- Features: n-grams (n=1,2,3) + Linguistics parameters (POS + syllabification + STEM + diacritics + consonants).
- Classifiers: Naive Bayes and Support Vector Machines with Radial Basis Function Kernel

Lang.	Borrowings	Cognates
IT-RO	baletto → balet (ballet)	vittoria - victorie (victory) ↑ victoria (LAT)
PT-RO	selva → selvă(selva)	instinto - instinct (instinct) ↑ instinctus (LAT)
ES-RO	machete → macetă (machete)	castillo - castel (castle) ↑ castellum (LAT)
TR-RO	tütün → tutun (tobacco)	aranjman - aranjament (arrangement) ↑ arrangement (FR)

Table 1: Examples of borrowings and cognates. For cognates we also report the common ancestor.

Results

Lang.	Base #1	Base #2	Naive Bayes		SVM			
	acc.	acc.	acc.	n	acc.	n	c	γ
RO-IT	52.0	50.8	67.38	3	67.38	2	2	0.10
RO-ES	48.2	78.5	80.00	2	83.69	2	2	0.10
RO-PT	48.6	78.8	83.23	2	86.00	2	2	0.10
RO-TR	48.5	60.3	83.38	2	87.38	3	10	0.01

Table 2: Results for automatic discrimination between cognates & word-etymon pairs using orthographic alignment. For the baselines we report the accuracy obtained on the test sets. For Naive Bayes we report the accuracy and the optimal n -gram values. For SVM we report the accuracy, the n -gram values and the optimal hyperparameters c and γ .

Lang.	Baseline #1			Baseline #2		
	P	R	A	P	R	A
RO-IT			52.0			50.8
RO-ES			48.2			78.5
RO-PT			48.6			78.8
RO-TR			48.5			60.3

Table 3: Results for automatic discrimination between cognates & word-etymon pairs using orthographic alignment. For the baselines we report the precision (P), recall (R) and accuracy (A) obtained on the test sets.

Result analysis

- The two baselines produce comparable results.
- For all pairs of languages, our method significantly improves over the baselines (99% confidence level) with values between 7% and 29% for the F1 score, suggesting that the n-grams extracted from the alignment of the words are better indicators of the type of relationship than the edit distance between them.

Result analysis

- The best results are obtained for TR-RO, with an F1 score of 92.1, followed closely by PT-RO with 90.1 and ES-RO with 85.5.
- These results show that, for these pairs of languages, the orthographic cues are different with regard to the relationship between the words.
- For IT-RO we obtain the lowest F1 score, 69.0.

Conclusion

- We propose a computational method for discriminating between cognates and borrowings based on their orthography.
- Our results show that it is possible to identify the type of relationship with fairly good performance (over 85.0 F1 score) for 3 out of the 4 pairs of languages we investigate.
- The method we propose is language-independent, but we believe that incorporating language-specific knowledge might improve the system's performance.



Word production

Producere de cuvinte (submitted, work in progress)

- Putem determina forma in care cuvinte viitoare vor intra intr-o limba tinta din alte limbi sursa?
- Rezultate preliminare pe Romana ca limbă tinta si 20 de limbi sursa.
- Comportament mai bun al limbilor cu influenta culturala, nu neaparat genetica.
- Diferente semnificative de predictie pentru producerea de cognates vs producerea de etimons.

Rezultate preliminare

Language	Baseline				This work			
	EDIT	COV ₁	COV ₅	COV ₁₀	EDIT	COV ₁	COV ₅	COV ₁₀
English	2.04 (0.23)	0.02	0.16	0.25	1.33 (0.15)	0.36	0.56	0.61
French	2.16 (0.24)	0.06	0.25	0.35	1.42 (0.15)	0.32	0.63	0.70
Italian	2.60 (0.32)	0.00	0.17	0.23	1.62 (0.23)	0.35	0.47	0.53
Latin	2.75 (0.34)	0.00	0.08	0.17	1.76 (0.22)	0.28	0.48	0.55
Neo-Greek	2.39 (0.29)	0.08	0.17	0.25	1.82 (0.24)	0.25	0.53	0.58
Old Slavic	2.34 (0.33)	0.08	0.18	0.23	1.84 (0.27)	0.17	0.39	0.47
German	2.36 (0.32)	0.07	0.23	0.26	2.00 (0.29)	0.26	0.41	0.45
Turkish	1.88 (0.27)	0.11	0.17	0.21	2.01 (0.29)	0.23	0.37	0.41
Bulgarian	2.33 (0.34)	0.06	0.20	0.21	2.22 (0.33)	0.15	0.23	0.28
Ruthenian	2.33 (0.35)	0.09	0.19	0.25	2.31 (0.35)	0.11	0.18	0.21
Russian	2.24 (0.33)	0.09	0.19	0.23	2.33 (0.33)	0.13	0.20	0.25
Albanian	2.60 (0.42)	0.06	0.11	0.12	2.35 (0.38)	0.08	0.20	0.25
Serbian	2.43 (0.37)	0.01	0.19	0.21	2.38 (0.36)	0.11	0.23	0.27
Polish	2.49 (0.38)	0.04	0.12	0.15	2.43 (0.36)	0.08	0.13	0.19
Portuguese	2.95 (0.52)	0.00	0.03	0.08	2.50 (0.43)	0.07	0.30	0.33
Slavic	2.88 (0.42)	0.05	0.11	0.17	2.66 (0.41)	0.12	0.27	0.31
Provençal	3.01 (0.49)	0.01	0.04	0.07	2.70 (0.44)	0.05	0.17	0.21
Hungarian	2.80 (0.43)	0.05	0.16	0.21	2.73 (0.42)	0.05	0.19	0.21
Spanish	3.22 (0.53)	0.02	0.06	0.11	3.06 (0.50)	0.05	0.12	0.15
Greek	4.36 (0.49)	0.01	0.08	0.08	4.28 (0.48)	0.05	0.15	0.15

Table 1: **Exp. #1.1:** Word form production for borrowings, using lemmas as input. The **Language** column indicates the source language. The target language is, in all cases, Romanian. We report the average edit distance between the produced form and the correct form of the borrowing (**EDIT**) un-normalized (and between parantheses the normalized version) and the coverage (**COV** for $n \in \{1, 5, 10\}$) for the baseline and for the method presented in this paper.

MULTUMESC!

On The Natural Languages Similarity. An Orthographic Perspective with a Focus on Romanian

Liviu P. Dinu

University of Bucharest
Center for Computational Linguistics
<http://nlp.unibuc.ro>

2015

Overview

- Natural languages similarity: motivation and approaches
- Romance syllabic similarity: motivation, approach, results
- Orthographic similarity: motivation and approach
- Computing degrees of similarity
 - Results on 3 Romanian corpora from different historical periods
 - Results on Europarl (Romanian subcorpus)
- Conclusions and future work

Language similarity

- The similarity of natural languages is a fairly vague notion, both linguists and non-linguists having intuitions about which languages are more similar to which others [McMahon and McMahon, 2003].
- Four types of similarity: typological, morphological, syntactic, lexical [Homola and Kubon, 2006].
- It is necessary to develop quantitative and computational methods in this field [McMahon and McMahon, 2003].

Syllabic similarity

- The more alike the languages sound, the more similar they are.
- When listeners hear a language for the first time, it is plausible that they can distinguish and individualize syllables.
- We investigate the syllabic similarities of the Romance languages based on the syllables excerpted from the representative vocabularies of seven Romance languages:
 - Latin, Romanian, Italian, Spanish, Catalan, French and Portuguese.

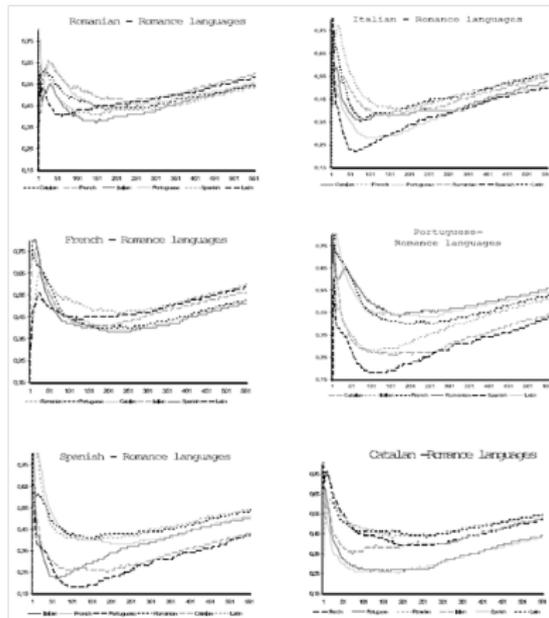
Strategy

- The representative vocabularies of seven Romance languages are syllabified.
- For each vocabulary, a ranking of syllables is constructed: the most frequent syllable of the vocabulary is placed on the first position, the next frequent syllable is placed on the second position , and so on.
- Then each of the seven Romance languages is compared to the other six (using the rank distance), each comparison having a graphic as a result.

Number of syllables

Language	% covered by the first ... syllables						# syllables	
	100	200	300	400	500	561	type	token
Latin	72%	86%	92%	95%	98%	100%	561	3922
Romanian	63%	74%	80%	84%	87%	90%	1243	6591
Italian	75%	85%	91%	94%	96%	97%	803	7937
Portuguese	69%	84%	91%	95%	97%	98%	693	6152
Spanish	73%	87%	93%	96%	98%	99%	672	7477
Catalan	62%	77%	84%	88%	92%	93%	967	5624
French	48%	61%	67%	72%	76%	78%	1738	5691

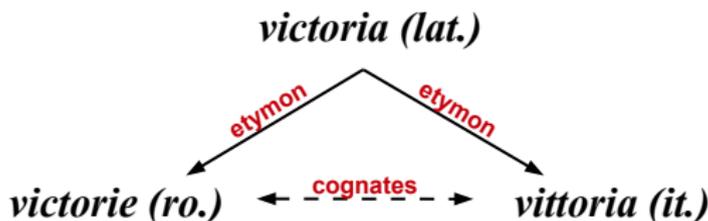
Results



- If we look at the first 300 syllables, Romanian is closest to Italian, followed by Spanish, Catalan and Portuguese.
- It can be observed that almost every time Romanian finds itself at the biggest distance from the other languages.

Ortographic approach

- A language $L1$ is closer to a language $L2$ when texts written in $L2$ are easier understood by speakers of $L1$ without prior knowledge of $L2$.
- When people read a text in a foreign language, they first identify the words which resemble words from their native language.
- Two types of related words:
 - Word-etymon pairs
 - Cognate pairs



Orthographic similarity

- Some pairs of related words are closer than others.
- Word-etymon pairs:

lună (ro.), ***luna*** (lat.) vs. ***bătrân*** (ro.), ***veteranus*** (lat.)

- Cognate pairs:

vânt (ro.), ***vent*** (fr.) vs. ***castel*** (ro.), ***château*** (fr.)

Algorithm and methodology

Input: corpus C in L_1

1. Text processing
 - 1.1. Remove stop words
 - 1.2. Lemmatize
2. Language relationships identification
 - 2.1. Detect etymologies
 - 2.2. Identify cognates
 - 2.3. Cluster by language families
3. Language similarity computation
 - 3.1. Measure word distances
 - 3.2. Compute degrees of similarity

Output: similarity hierarchy for L_1

Similarity method

Definition

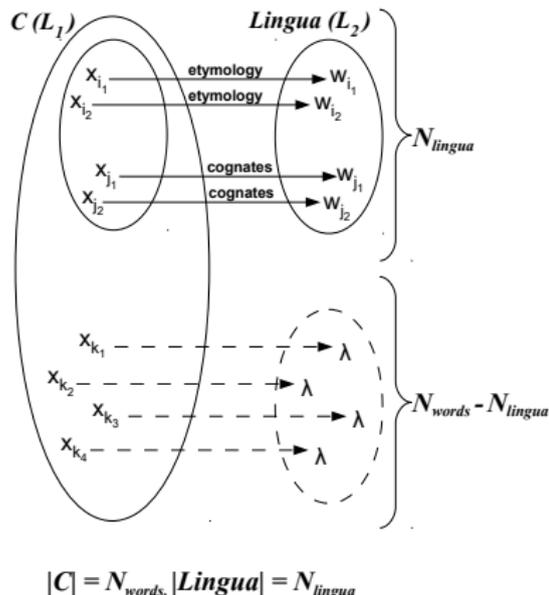
Given a string distance Δ , we define the distance between languages L_1 and L_2 (with frequency support from corpus C in L_1) as follows:

$$\Delta(L_1, L_2) = 1 - \frac{N_{lingua}}{N_{words}} + \frac{\sum_{i=1}^{N_{lingua}} \Delta(w_i, x_i)}{N_{words}} \quad (1)$$

Definition

The similarity between L_1 and L_2 is:

$$Sim(L_1, L_2) = 1 - \Delta(L_1, L_2) \quad (2)$$



Etymology detection

- We extract etymologies from electronic dictionaries.

Pattern

```
<abbr class="abbrev" title="limba language_name">  
  language_abbreviation  
</abbr>  
<b> etymon </b>
```

Entry

```
<b> capitol </b>  
  
<abbr class="abbrev" title="limba italiana">  
  it.  
</abbr>  
<b> capitolo </b>  
<abbr class="abbrev" title="limba latina">  
  lat.  
</abbr>  
<b> capitulum </b>
```

Etymology detection

- We extract etymologies from electronic dictionaries.

Pattern

```
<abbr class="abbrev" title="limba language_name">  
  language_abbreviation  
</abbr>  
<b> etymon </b>
```

Entry

```
<b> capitol </b>  
  
<abbr class="abbrev" title="limba italiana">  
  it.  
</abbr>  
<b> capitolo </b>  
<abbr class="abbrev" title="limba latina">  
  lat.  
</abbr>  
<b> capitulum </b>
```

Etymology detection

- We extract etymologies from electronic dictionaries.

Pattern

```
<abbr class="abbrev" title="limba language_name">  
  language_abbreviation  
</abbr>  
<b> etymon </b>
```

Entry

```
<b> capitol </b>  
  
<abbr class="abbrev" title="limba italiana">  
  it.  
</abbr>  
<b> capitolo </b>  
<abbr class="abbrev" title="limba latina">  
  lat.  
</abbr>  
<b> capitulum </b>
```

Etymology detection

- We extract etymologies from electronic dictionaries.

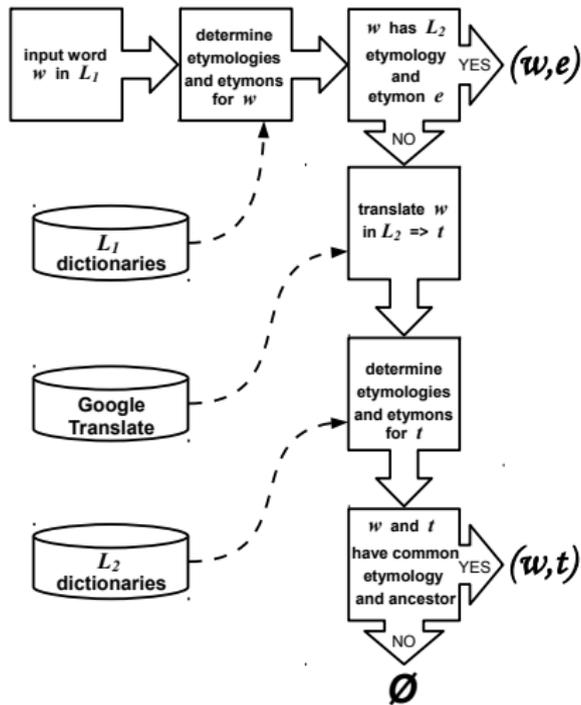
Pattern

```
<abbr class="abbrev" title="limba language_name">  
  language_abbreviation  
</abbr>  
<b> etymon </b>
```

Entry

```
<b> capitol </b>  
  
<abbr class="abbrev" title="limba italiana">  
  it.  
</abbr>  
<b> capitolo </b>  
<abbr class="abbrev" title="limba latina">  
  lat.  
</abbr>  
<b> capitulum </b>
```

Cognate identification



Orthographic metrics

- We use string similarity metrics to compute the orthographic similarity between related words.
- Many methods have been used so far, but we cannot say which is the most appropriate for a given task.
- We use three orthographic metrics and compare their results.

Orthographic metrics

The edit distance

$$\Delta(w_i, w_j) = \frac{LD(w_i, w_j)}{\max(|w_i|, |w_j|)} \quad (3)$$

where $LD(w_i, w_j)$ is the number of operations required to transform w_i in w_j .

The longest common subsequence ratio

$$\Delta(w_i, w_j) = \frac{LCS(w_i, w_j)}{\max(|w_i|, |w_j|)} \quad (4)$$

where $LCS(w_i, w_j)$ is the longest common subsequence of w_i and w_j .

The rank distance

Given two rankings $L_1 = (x_1, x_2, \dots, x_n)$ and $L_2 = (y_1, y_2, \dots, y_n)$, and $V(L_1), V(L_2)$ their alphabets, the rank distance is defined as follows:

$$\Delta(L_1, L_2) = \sum_{x \in V(L_1) \cap V(L_2)} |\text{ord}(x|L_1) - \text{ord}(x|L_2)| + \sum_{x \in V(L_1) \setminus V(L_2)} \text{ord}(x|L_1) + \sum_{x \in V(L_2) \setminus V(L_1)} \text{ord}(x|L_2) \quad (5)$$

where $\text{ord}(x|L)$ is the rank of x in ranking L , in a Borda sense. To extend the distance to words, we index each character with a number equal to the number of its previous occurrences in the given word. For normalization, we divide the rank distance by the maximum possible value between w_i and w_j : $|w_i|(|w_i| + 1)/2 + |w_j|(|w_j| + 1)/2$.

Application: Romanian

- Romanian is a Romance language, surrounded by Slavic languages.
- Its communication with the Romance kernel was difficult.
- Its position in the Romance family is controversial, either isolated or more integrated within the group [McMahon and McMahon, 2003].



Common ancestors

	FR	IT	ES	PT	TR
Arabic	-	10	15	13	4
English	3	57	94	195	158
French	-	547	455	1,925	1,157
German	-	16	14	10	-
Greek	-	221	-	1,366	410
Hebrew	-	-	1	-	-
Italian	1	-	143	238	-
Latin	475	2,606	4,874	5,815	572
Persian	-	1	-	2	-
Polish	-	-	-	2	-
Portuguese	-	3	-	-	-
Provençal	-	1	3	4	-
Russian	-	4	-	6	-
Spanish	-	34	-	72	-
Turkish	-	3	-	6	-
Total	479	3,503	5,599	9,654	2,301

Datasets

- 17th and 18th century: Romanian chronicles. (**Chronicles**)
- 19th century: the publishing works of the Romanian poet Mihai Eminescu. (**Eminescu**)
- 21st century: the parliamentary debates held in the Romanian Parliament. (**Parliament**)

- The basic Romanian lexicon. (**RVR**)

Dataset	#words		#stop words		#lemmas
	token	type	token	type	type
Parliament	22,469,290	162,399	14,451,178	214	40,065
Eminescu	870,828	65,742	565,396	212	21,456
Chronicles	253,786	28,936	170,582	193	8,189
RVR	2,464	2,464	124	124	2,252

Etymology detection evaluation

- We compare the manually determined etymologies with the automatically obtained etymologies on samples of 500 words.
- We evaluate the languages for which we determine both etymologies and cognate pairs:
 - Romanian 95.8%
 - Spanish 96.6%
 - Turkish 96.0%
 - French 96.8%
 - Portuguese 97.0%
 - English 97.2%
 - Italian 97.8%

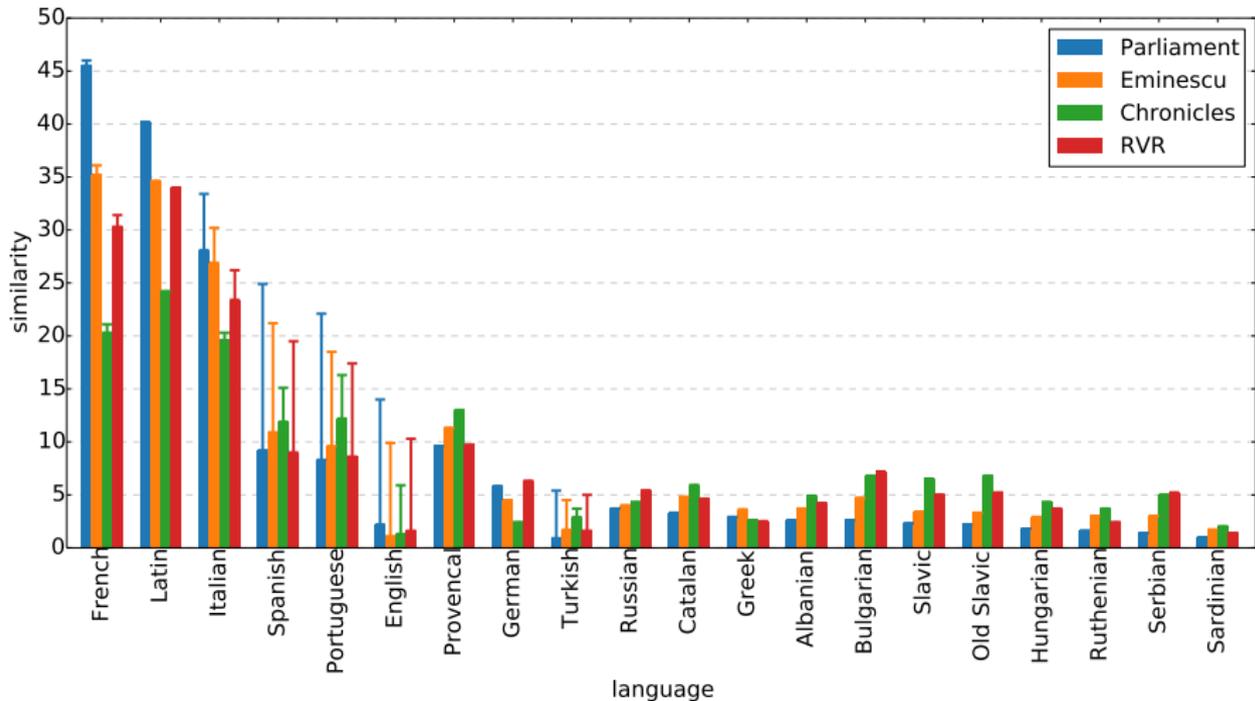
Diacritics

- Many words have undergone transformations by the augmentation of language-specific diacritics when entering a new language.
- From an orthographic perspective, the resemblance of words is higher between words without diacritics.

amiciție (ro.), *amitié* (fr.) vs. *amicitie* (ro.), *amitie* (fr.)

- In Romanian, five diacritics are used today: *ă*, *â*, *î*, *ș*, *ț*.
- We create two versions of each dataset: with and without diacritics.

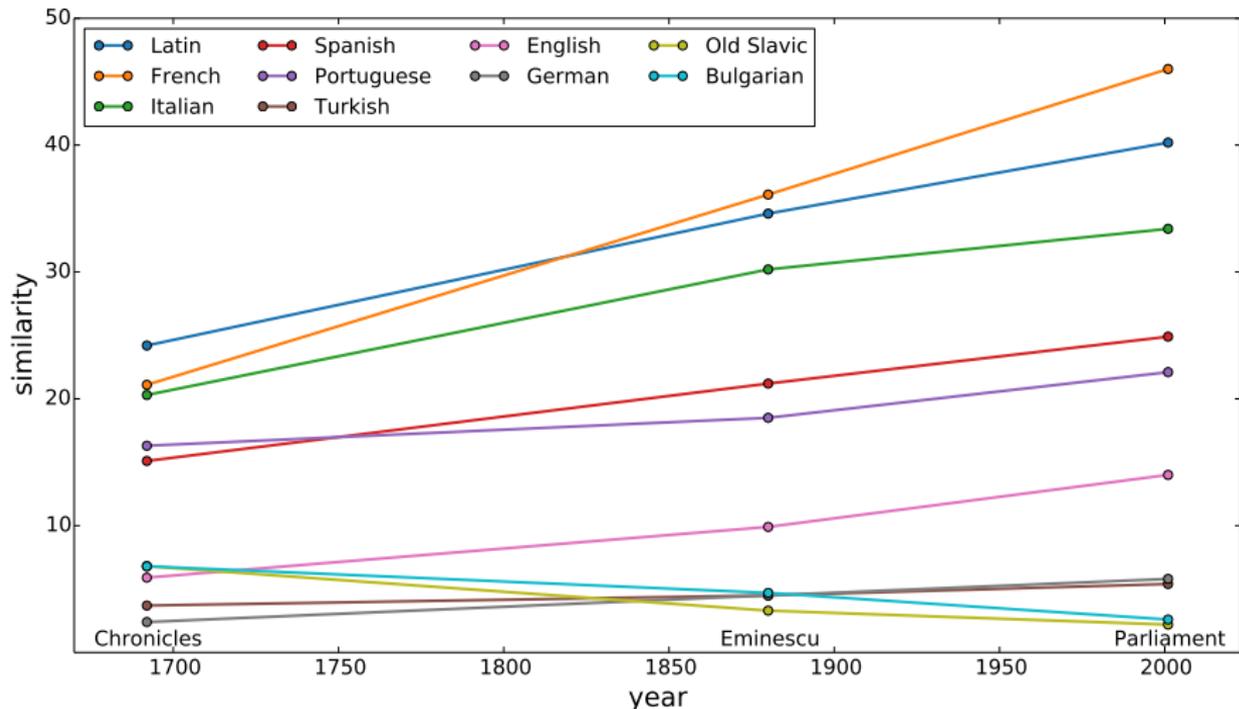
Results for the Romanian datasets



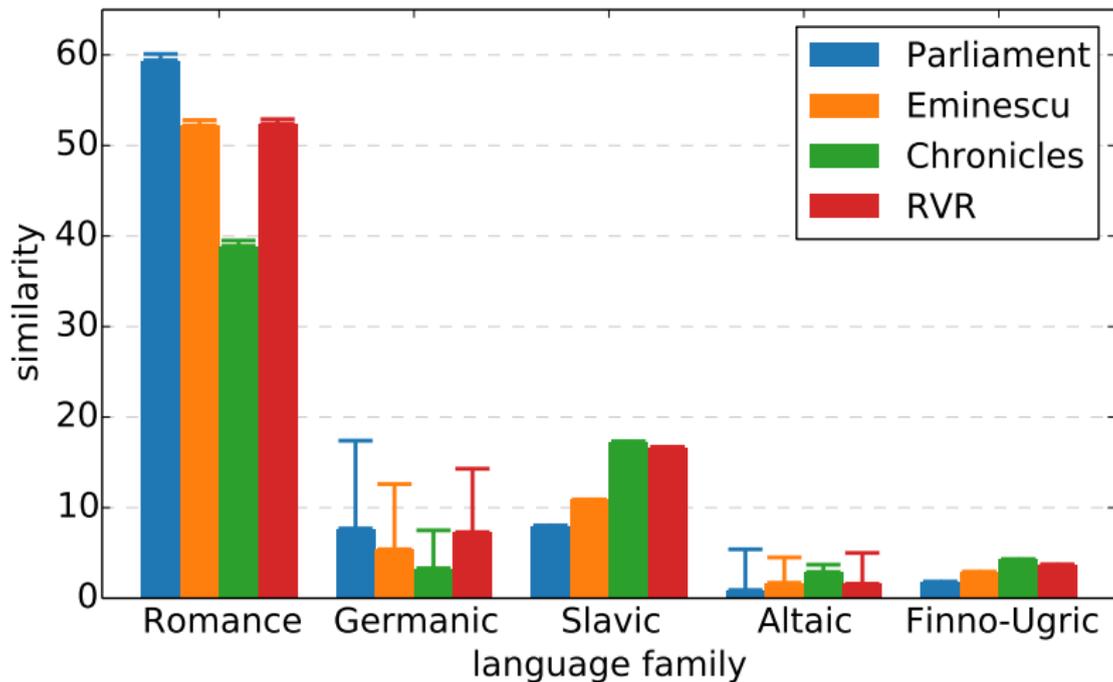
Ranking of similarity

Language	Parliament			Eminescu			Chronicles			RVR		
	%w	e	e+c	%w	e	e+c	%w	e	e+c	%w	e	e+c
French	70.6	45.5	46.0	57.2	35.2	36.1	36.7	20.3	21.1	50.6	30.3	31.4
Latin	63.7	40.2	—	59.9	34.6	—	44.9	24.2	—	56.5	34.0	—
Italian	48.5	28.1	33.4	44.7	26.9	30.2	31.7	19.6	20.3	41.4	23.4	26.2
Spanish	40.2	9.2	24.9	38.1	10.9	21.2	29.7	11.9	15.1	32.5	9.0	19.5
Portuguese	35.0	8.3	22.1	31.3	9.6	18.5	28.3	12.2	16.3	29.3	8.6	17.4
English	22.1	2.2	14.0	18.8	1.1	9.9	11.3	1.3	5.9	14.3	1.6	10.3
Provencal	17.7	9.6	—	20.7	11.3	—	21.8	13.0	—	16.8	9.7	—
German	9.2	5.8	—	6.9	4.5	—	4.9	2.4	—	10.2	6.3	—
Turkish	7.7	0.9	5.4	6.6	1.7	4.5	5.6	2.9	3.7	7.4	1.6	5.0
Russian	5.9	3.7	—	6.5	4.0	—	7.5	4.3	—	9.0	5.4	—

Romanian evolution



Language families



Surrounding languages

Language	Parliament			Eminescu			Chronicles			RVR		
	%w	d	nd	%w	d	nd	%w	d	nd	%w	d	nd
Turkish	7.7	5.4	5.6	6.6	4.5	4.7	5.6	3.7	3.9	7.4	5.0	5.3
Russian	5.9	3.7	4.0	6.5	4.0	4.4	7.5	4.3	4.9	9.0	5.4	6.2
Albanian	4.8	2.6	3.0	6.7	3.7	4.0	9.1	4.9	5.3	8.4	4.2	4.8
Bulgarian	4	2.6	3.0	7.4	4.7	5.5	10.6	6.8	7.8	11.8	7.2	8.4
Slavic	4.9	2.3	2.5	6.6	3.4	3.8	12.1	6.5	7.7	9.8	5.0	5.7
Old Slavic	3.8	2.2	2.7	6.1	3.3	4.3	11.9	6.8	8.7	9.5	5.2	6.0
Hungarian	2.9	1.8	2.0	5.1	2.9	3.3	7.5	4.3	4.7	7.4	3.7	4.6
Serbian	2.6	1.4	1.6	5.8	3.0	3.4	8.9	5.0	5.5	8.6	5.2	6.0
Polish	1.3	0.7	0.8	2.2	1.2	1.5	4.3	2.2	2.6	4.3	2.5	2.8
Serbo-Croatian	0.3	0.1	0.1	0.6	0.3	0.3	1.1	0.5	0.5	1.6	0.8	0.9
Ukrainian	0.0	0.0	0.0	0.1	0.0	0.0	0.6	0.3	0.3	0.4	0.3	0.3

Orthographic metrics

- Are the differences between the results obtained with each metric statistically significant?
- ANOVA hypothesis tests on samples of 5,000 words.
 - The mean computed values for the three metrics are not all equal.
- Pairwise t-tests with Bonferonni correction for the p-value.
 - The differences between the metrics are statistically significant, but they are small.
- There is a high correlation between the similarity rankings ($\rho > 0.98$ for each pair of metrics).

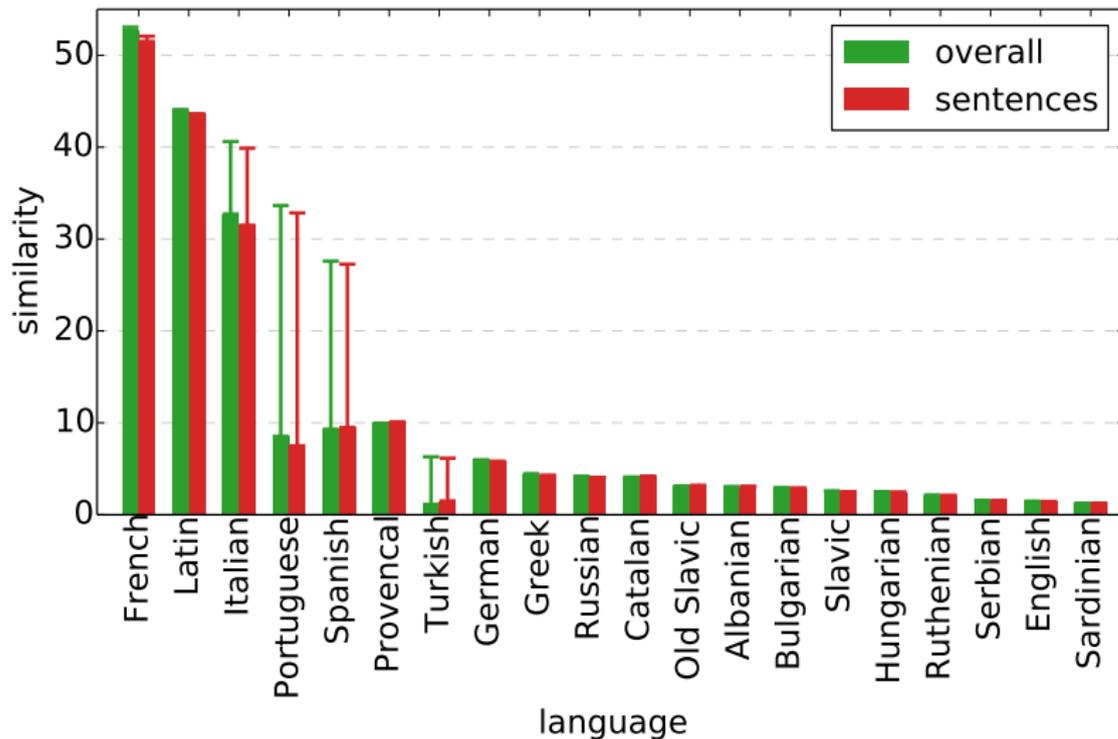
Further experiments

- We use Europarl [Koehn, 2005] - the Romanian subcorpus.
- We investigate two questions:
 - Are degrees of similarity between Romanian and other languages consistent across different corpora from the same period?
 - Are there differences between the overall degrees of similarity (the bag-of-words model) and those obtained at sentence level?

Further experiments

- We conduct four experiments:
 - **Exp. #1:** we use the bag-of-words model on Europarl.
 - **Exp. #2:** we aggregate sentence-level rankings of similarity.
 - **Exp. #3:** we remove outliers (regarding the sentence length).
 - **Exp. #4:** we remove outliers (regarding the degrees of similarity).

Results for Europarl



Results for Europarl

Language	Parl.	Exp. #1	Exp. #2	Exp. #3	Exp. #4
French	45.5	53.1	52.1	52.1	52.8
Latin	40.2	44.1	43.6	43.6	44.0
Italian	33.4	40.6	39.9	39.9	40.2
Portuguese	22.1	33.6	32.9	32.8	33.2
Spanish	24.9	27.6	27.3	27.3	26.8
English	14.0	16.0	15.7	15.7	15.1
Provençal	9.6	10.0	10.1	10.1	9.3
Turkish	5.4	6.3	6.2	6.1	5.7
German	5.8	5.9	5.8	5.8	5.3
Greek	2.9	4.4	4.3	4.3	3.8

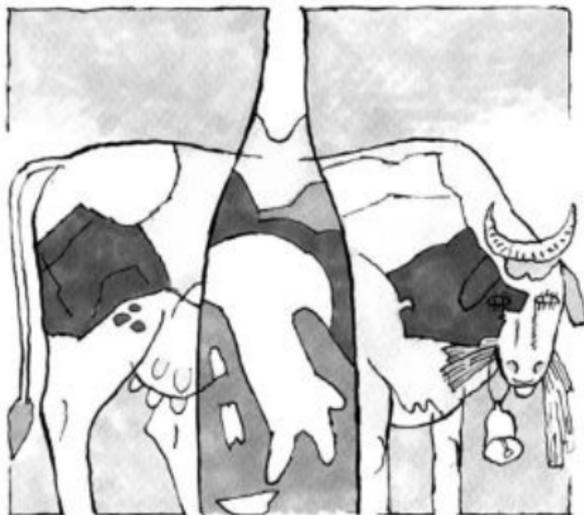
Language similarity

Cu un kil de carne de vacă nu mori de foame, cu un litru de vin nu mori de sete¹. **(ro)**

Con un chilo di carne di vaca non morire di fame, con un litro di vino non morire di sete. **(it)**

Com um quilo de carne de vaca não morrer de fome, com um litro de vinho não morrer de sede. **(pt)**

Con un kilo de carne de vacuno no morirse de hambre, con un litro de vino no morir de sed. **(es)**



¹With a kilo of beef one does not starve, with a liter of wine one does not die of thirst. **(en)**

Conclusions

- We proposed a computational method for determining cross-language orthographic similarity.
- We applied the method on Romanian corpora from different historical periods.
- We plan to extend our analysis to other languages as well, as we gain access to resources.
- We plan to combine the orthographic approach with syntactic and semantic evidence for a wider perspective on language similarity.

References

- Anca Dinu, Liviu P. Dinu, 2005. On the syllabic similarities of Romance languages. In Proc. CICLing 2005, p. 785-788, Mexico City, Mexico, February 13-19.
- Alina Ciobanu, Liviu P. Dinu, 2014. On the Romance Languages Mutual Intelligibility. In Proc. LREC 2014, p. 3313-3318, Reykjavik, Iceland, May 26-31.
- Alina Maria Ciobanu, Liviu P. Dinu, 2014. An Etymological Approach to Cross-Language Orthographic Similarity. Application on Romanian. In Proc. EMNLP 2014, p. 1047-1058, Doha, Qatar, October 25-29.

Thank you!

Corpus Linguistics

Formal vs. Distributional

Semantics

Liviu P. Dinu, ldinu@fmi.unibuc.ro

University of Bucharest

Center for Computational Linguistics,

Faculty of Mathematics and Computer Science

nlp.unibuc.ro

Corpus Linguistics

- Corpus linguistics is a study of language and a method of linguistic analysis which uses a collection of natural or “real word” texts known as corpus.
- **What Corpus Linguistics Does:**
 - Gives an access to naturalistic linguistic information, “real word” texts which are mostly a product of real life situations. This makes corpora a valuable research source for grammar, semantics, dialectology, sociolinguistics, stylistics, etc.

Corpus Linguistics (2)

- Facilitates linguistic research.
- Electronically readable corpora have dramatically reduced the time needed to find particular words or phrases.
- A research that would take days or even years to complete manually can be done in a matter of seconds with the highest degree of accuracy.

Corpus Linguistics (3)

- Enables the study of wider patterns and collocation of words.
- Before the advent of computers, corpus linguistics was studying only single words and their frequency.
- Modern technology allowed the study of wider patterns and collocation of words.

Corpus Linguistics (4)

- Allows analysis of multiple parameters at the same time.
- Various corpus linguistics software programmes and analytical tools allow the researchers to analyse a larger number of parameters simultaneously.
- In addition, many corpora are enriched with various linguistic information such as annotation.

CL...

- Facilitates the study of the second language.
- Study of the second language with the use of natural language allows the students to get a better “feeling” for the language and learn the language like it is used in real rather than “invented” situations.

Corpus Linguistics

- **What Corpus Linguistics Does Not:**
 - Does not explain why.
 - The study of corpora tells us what and how happened but it does not tell us why the frequency of a particular word has increased over time for instance.
 - Does not represent the entire language.
 - Corpus linguistics studies the language by using randomly or systematically selected corpora.

CL...

- They typically consist of a large number of naturally occurring texts, however, they do not represent the entire language.
- Linguistic analyses that use the methods and tools of corpus linguistics thus do not represent the entire language.

Corpus Linguistics

- **Application of Corpus Linguistics:**
 - **Lexicography.** Corpus linguistics plays an important role in compiling, writing and revising dictionaries as within a few seconds, the linguist can get examples of words or phrases from millions of spoken and written texts.
 - **Grammar.** The huge amount of texts offers a reliable representation of the language to be able to conduct grammatical research as well as to test theoretical hypotheses.

CL. Applications

- **Sociolinguistics.** Corpus Linguistics offers a valuable insight into how language varies from place to place and between different social groups.
- **Translation studies.** Corpora which contain texts in several different languages are a valuable tool for translators as they make it easy to determine how particular words and their synonyms collocate and differ in practical use.

Corpus Linguistics. Applications

- **Language learning/teaching.** A growing number of textbooks which are used for language learning/teaching contain texts from corpora rather than “invented” situations because they expose the students to real life situations.
- **Stylistics.** For genres such as the language used by politicians, pop culture, advertising industry, etc., corpora as an important source of information.

CL. Applications...

- **Dialectology.** The texts included in corpora are in their original form, including dialect which gives the linguists a priceless insight into geographical variation of a language.
- **Historical linguistics.** Historical corpora offer an easy access to virtually all known historic books and manuscripts in electronic form.

Corpus Linguistics

- **Notable Corpora:**
 - **Brown Corpus (the Brown Standard Corpus of Present-Day American English).** It contains about 500 English texts that total about 1 million words compiled in the 1960s. It is rather small, but it is the first modern and electronically readable corpus.

Notable corpora

- **British National Corpus.** It consists of a wide range of written and spoken texts in English language, totalling 100 million words. Since 1994.
- **Oxford English Corpus.** It is a huge corpus of English language totalling over 2 billion words. The texts included in the corpus are taken from all sorts of sources, ranging from literary works to the language in forums and chatrooms.

Notable corpora

- **American National Corpus.** It is the American English equivalent to the British National Corpus, however, it only contains about 22 million words of American English spoken and written texts., but it is richly annotated. It is being developed since 1990.
- **International Corpus of English.** It consists of a set of corpora which contain variations of English language from countries where English is either the first or official second language. Each set of the International Corpus of English contains 1 million word texts that have been created after the year 1989.

Notable corpora

- **Scottish Corpus of Texts and Speech.** The collection of written and spoken texts in Scottish English and Scots after 1940 is available online for free since 2004. In 2007, the corpus reached a total of 4 million words.
- **WaCky** 2 billion words

Corpus Linguistics

- Out of the many possible applications of Corpus Linguistics, we will chose lexical semantics (Generative Lexicon, Pustejovsky 1995) and Distributional Semantics (Baroni 2010).
- The course will focuse on Formal vs. Distributional Semantics

Reference/Sense distinction

- Frege: Linguistic signs have a reference and a sense:
 - (i) “Mark Twain is Mark Twain” vs. (ii) “Mark Twain is Samuel Clemens”.
 - (i) same sense and same reference vs. (ii) different sense and same reference.
- Both the sense and reference of a sentence are built compositionally.
- **Formal Semantics** studies “meaning” as “reference”.
- **Distributional semantics** focuses on “meaning” as “sense” leading to the “language as use” view.

Formal vs. Distributional Semantics

Focus of FS:

Grammatical words:

- prepositions,
- articles, quantifiers,
- coordination,
- auxiliary verbs,
- Pronouns,
- negation

Focus of DS:

Content words:

- nouns,
- adjectives,
- verbs.

Formal Semantics

- Formal semantics gives an elaborate and elegant account of the productive and systematic nature of language.
- The formal account of compositionality relies on:
 - words (the minimal parts of language, with an assigned meaning)
 - syntax (the theory which explains how to make complex expressions out of words)
 - semantics (the theory which explains how meanings are combined in the process of particular syntactic compositions).

Formal Semantics

Theory of Meaning

A theory of meaning is understood as providing a detailed specification of the knowledge that a native speaker has about his/her own language. [Dummett, 91]

In doing this, a theory of meaning has to provide a way to assign meaning to all the different words in the language and then a mechanism by means of which all these meanings can be combined into larger expressions to form the meaning of phrases, sentences, discourses, and so on.

Formal Semantics

Truth-conditional semantics program

To state the meaning of a sentence we should state which conditions must be true in the world for this sentence to be true.

e.g. Every man loves a woman.

Truth-conditions:

For each member “x” of the set of men, there should be at least one member “y” of the set of women, in such a way that the pair $\langle x, y \rangle$ is in the relation loves.

Logic:

$$\forall x.(\text{man}(x) \rightarrow \exists y.(\text{woman}(y) \ \& \ \text{loves}(x,y)))$$

Formal Semantics

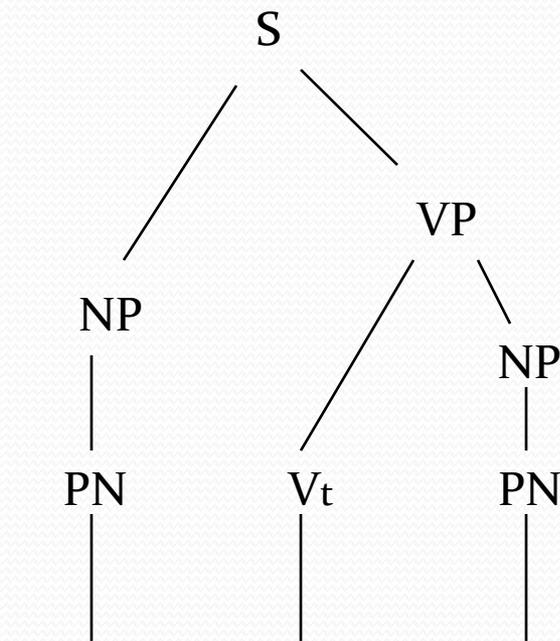
Frege's Compositional Semantics

The meaning of the sentence is determined by the meaning of the words of which it is composed, and the way in which these are put together.

The linear order of the words in a sentence hide the role that different kinds of words play in the building of the meaning of the whole.

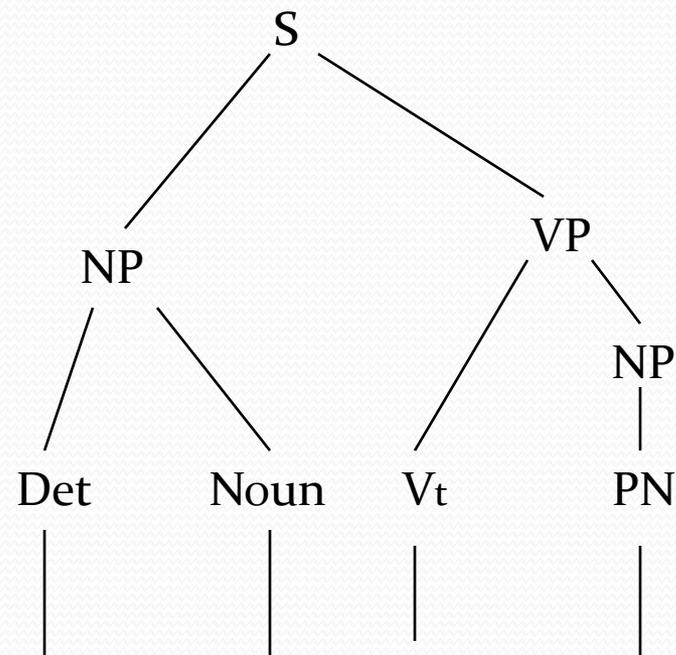
Formal Semantics

Syntactic structure



John likes

Mary
 $\text{like}(\text{john}, \text{mary})$



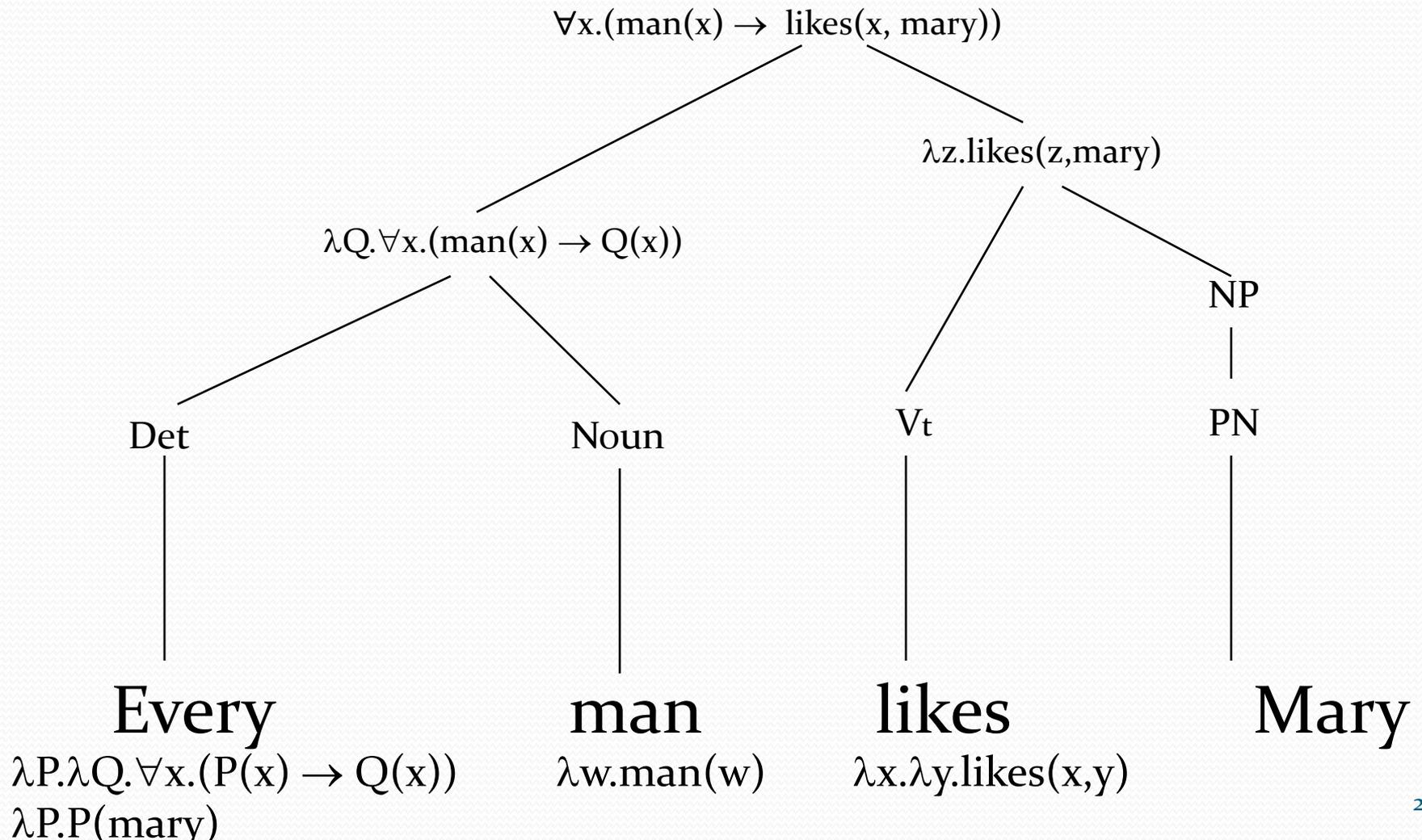
Every man likes Mary

$\forall x.(\text{man}(x) \rightarrow \text{likes}(x, \text{mary}))$

Formal Semantics

- Semantic Structure
- Formal Semantics uses Lambda Calculus as a means of combining meaning guided by the syntactic operations.

Formal Semantics



Distributional Semantics

- You shall know a word by the company it keeps (Firth);
- The meaning of a word is defined by the way it is used (Wittgenstein).
- This leads to the distributional hypothesis about word meaning:
 - the context surrounding a given word provides information about its meaning;
 - words are similar if they share similar linguistic contexts;
 - semantic similarity = distributional similarity.

Distributional Semantics

- Examples of similar words:
 - “astronaut” and “cosmonaut”
 - “car” and “automobile”
 - “banana” and “apple” (these two are less similar)
 - “huge” and “large”,
 - “eat” and “devour”
- Not similar:
 - “car” and “flower”,
 - “car” and “pope”

Distributional Semantics

- For example, if one word describes a given situation
 - “I’m on the *highway*”
- then it is very likely that the other word also describes this situation
 - “I’m in a *car*”
- **Distributional semantics** is an approach to semantics that is based on the contexts of words and linguistic expressions in **large corpora**.

Distributional Semantics

- Take a word and its contexts. By looking at a word's context, one can infer its meaning
 - tasty *tnassiorc*
 - greasy *tnassiorc*
 - *tnassiorc* with butter
 - *tnassiorc* for breakfast
- FOOD

Distributional Semantics

- He filled the *wampimuk*, passed it around and we all drunk some



DRINK

- We found a little, hairy *wampimuk* sleeping behind the tree



ANIMAL

Distributional Semantics

DS accounts for different uses of words (like in Generative Lexicon). Take “brown” for example. Each adjective acts on nouns in a different way:

“In order for a cow to be brown most of its body's surface should be brown, though not its udders, eyes, or internal organs. A brown crystal, on the other hand, needs to be brown both inside and outside. A book is brown if its cover, but not necessarily its inner pages, are mostly brown, while a newspaper is brown only if all its pages are brown. For a potato to be brown it needs to be brown only outside. . . Furthermore, in order for a cow or a bird to be brown the brown color should be the animal's natural color, since it is regarded as being ‘really’ brown even if it is painted white all over. A table, on the other hand, is brown even if it is only painted brown and its ‘natural’ color underneath the paint is, say, yellow. But while a table or a bird are not brown if covered with brown sugar, a cookie is. In short, what is to be brown is different for different types of objects. To be sure, brown objects do have something in common: a salient part that is wholly brownish. But this hardly suffices for an object to count as brown. A significant component of the applicability condition of the predicate ‘brown’ varies from one linguistic context to another.” (Lahav 1993:76)

Distributional Semantics

- What happens with brown is replicated by the large majority of adjective-noun combinations. Treating them all like 'idioms' would mean to turn the exception into the rule.
- As it is easy to see, many of the problems come from the lexicon of content words, such as nouns, verbs and adjectives, and not from grammatical terms.

Distributional Semantics

- Of course, there have been important attempts to tackle the lexicon problem from the point of view of formal semantics, like Pustejovsky's (1995) theory of the Generative Lexicon.
- More recently, Asher (2011) has approached lexical semantics with a theory of predication that uses a sophisticated system of semantic types, plus a mechanism of type coercion.

Distributional Semantics

- However, the problem of lexical semantics is primarily a problem of size: even considering the many subregularities found in the content lexicon, a hand-by-hand analysis is simply not feasible.
- The problem of assigning reasonable (if not exhaustive) syntactic structure to arbitrary, real-life sentences is perhaps equally hard. Here, however, technology has been an important part of the answer: Natural language **parsers**, that automatically assign a syntactic structure to sentences, have made great advances in recent years by exploiting probabilistic information about parts of speech (POS tags) and syntactic attachment preferences.

Distributional Semantics

- Tasks where DS has been successful:
 - Word similarity,
 - Information retrieval,
 - Question Answering,
 - Entailment, etc.

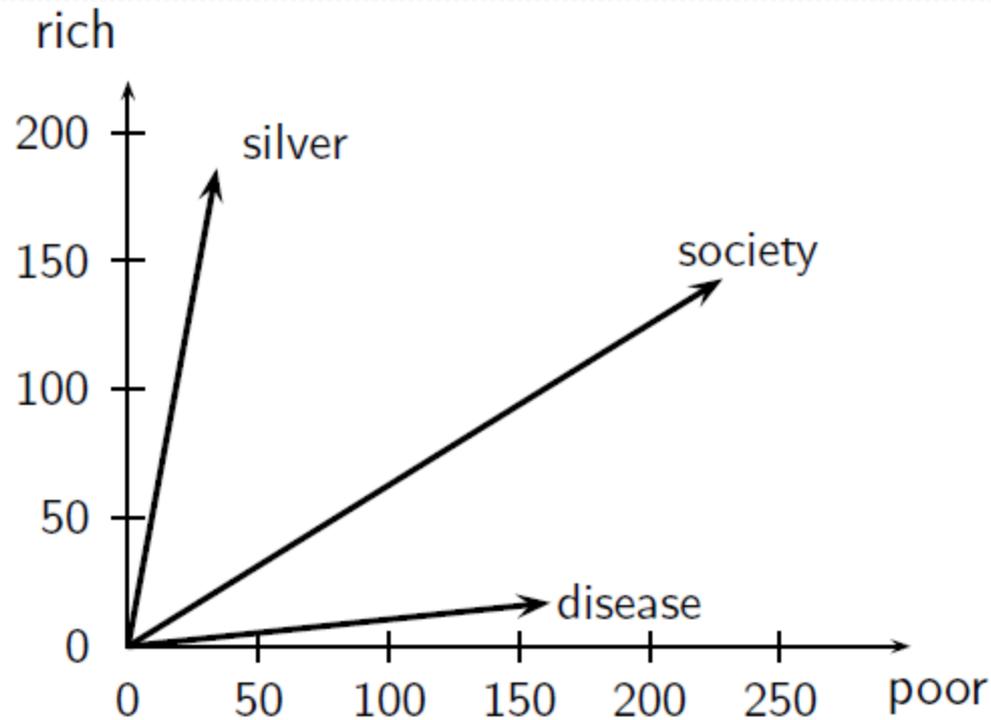
Distributional Semantics

- Two words are neighbors if they cooccur.
- The cooccurrence count of words w_1 and w_2 in corpus G is the number of times that w_1 and w_2 occur:
 - in a linguistic relationship with each other (e.g., w_1 is a modifier of w_2) or
 - in the same sentence or
 - in the same document or
 - within a distance of at most k words (where k is a parameter)

Distributional Semantics

- corpus = English Wikipedia
- cooccurrence defined as occurrence within $k = 10$ words of each other:
 - $\text{cooc.}(\text{rich}, \text{silver}) = 186$
 - $\text{cooc.}(\text{poor}, \text{silver}) = 34$
 - $\text{cooc.}(\text{rich}, \text{disease}) = 17$
 - $\text{cooc.}(\text{poor}, \text{disease}) = 162$
 - $\text{cooc.}(\text{rich}, \text{society}) = 143$
 - $\text{cooc.}(\text{poor}, \text{society}) = 228$

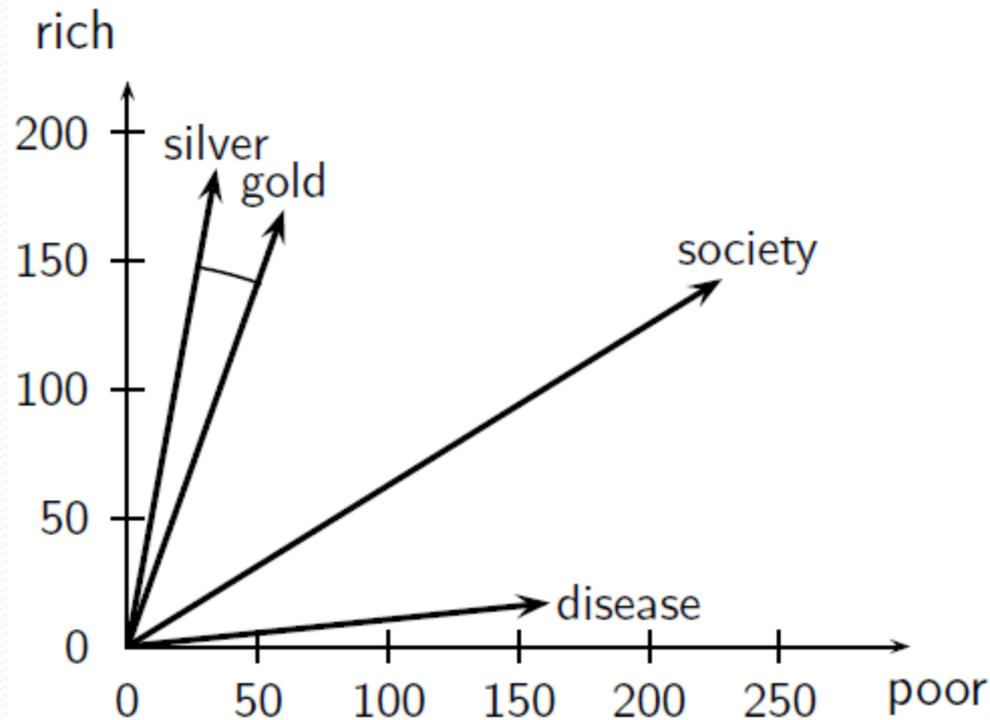
Distributional Semantics



- $\text{cooc.}(\text{poor}, \text{silver})=34$, $\text{cooc.}(\text{rich}, \text{silver})=186$,
- $\text{cooc.}(\text{poor}, \text{disease})=162$, $\text{cooc.}(\text{rich}, \text{disease})=17$,
- $\text{cooc.}(\text{poor}, \text{society})=228$, $\text{cooc.}(\text{rich}, \text{society})=143$

Distributional Semantics

- The similarity between two words is usually measured with the cosine of the angle between them.
- Small angle: silver and gold are similar.



Distributional Semantics

- Up to now we've only used two dimension words: rich and poor.
- Now do this for a very large number of dimension words: hundreds or thousands.
- This is now a very high-dimensional space with a large number of vectors represented in it.
- Note: a word can have a dual role in word space.
 - Each word can, in principle, be a dimension word, an axis of the space.
 - But each word is also a vector in that space.

Distributional Semantics

- We can compute now the nearest neighbors of any word in this in word space.
- Nearest neighbors of “silver”:
1.000 silver / 0.865 bronze / 0.842 gold / 0.836 medal /
0.826 medals / 0.761 relay / 0.740 medalist / 0.737
coins / 0.724 freestyle / 0.720 metre / 0.716 coin / 0.714
copper / 0.712 golden / 0.706 event / 0.701 won / 0.700
foil / 0.698 Winter / 0.684 Pan / 0.680 vault / 0.675
jump

Distributional Semantics

- Nearest neighbors of “disease”:

1.000 disease / 0.858 Alzheimer / 0.852 chronic /
0.846 infectious / 0.843 diseases / 0.823 diabetes /
0.814 cardiovascular / 0.810 infection / 0.807
symptoms / 0.805 syndrome / 0.801 kidney / 0.796
liver / 0.788 Parkinson / 0.787 disorders / 0.787
coronary / 0.779 complications / 0.778 cure / 0.778
disorder / 0.778 Crohn / 0.773 bowel

Distributional Semantics

- Cases where simple word space models fail:
 - Antonyms are judged to be similar: “disease” and “cure”
 - Ambiguity: “Cambridge”
 - Homonymy: ”bank”
 - Non-specificity (occurs in a large variety of different contexts and has few/no specific semantic associations): “person”

Distributional Semantics

- The vectors in our space have been words so far.
- But we can also represent other entities like: phrases, sentences, paragraphs, documents, even entire books.
- Compositionality problem: how to obtain the distribution vector of a phrase?

Distributional Semantics- from words to phrases

- Option 1: The distribution of phrases – even sentences – can be obtained from corpora, but...
 - those distributions are very sparse;
 - observing them does not account for productivity in language.
- Option 2: Use vector product of two or more words to compute the phrase distribution, but...
 - Multiplication is commutative in a word-based model:
 - $[[\text{The cat chases the mouse}]] = [[\text{The mouse chases the cat}]]$.
 - Multiplication is intersective – problem for non-intersective adjectives:

Distributional Semantics

- Adjective types, Partee (1995)
- **Intersective:** carnivorous mammal
- $||\text{carnivorous mammal}|| = ||\text{carnivorous}|| \cap ||\text{mammal}||$
- **Subsective:** skilful surgeon
- $||\text{skilful surgeon}|| \subset ||\text{surgeon}||$
- **Non-subsective:** former senator
- $||\text{former senator}|| \neq ||\text{former}|| \cap ||\text{senator}||$
- $||\text{former senator}|| \subset ||\text{senator}||$

Distributional Semantics

- DS Strengths:
 - fully automatic construction;
 - representationally simple: all we need is a corpus and some notion of what counts as a word;
 - language-independent, cognitively plausible.
- DS Weaknesses:
 - no generative model
 - many ad-hoc parameters
 - ambiguous words: their meaning is the average of all senses
 - context words contribute indiscriminately to meaning;
[[The cat chases the mouse]] = [[The mouse chases the cat]].

Example

"Light: a multilingual distributional analysis"

Abordare

- Lumina? Analiza distributionala in texte religioase si texte politice.
- Limbi: romana, engleza, franceza.
- Corpus: Vechiul Testament, Noul Testament, Europarl (discursurile din parlamentul european)

Metoda

- Determinam si analizam sinonimele cuvantului lumina
- Determinam si analizam antonimele cuvantului lumina
- Analizam si comparam contextele in care apar acestea

Lumină / light / lumière - frecvența în Biblie

- *Romana: Lumină* – 308 apariții
- Frecvența medie a cuvintelor în Biblie în limba română: 37,41
- *Lumină* apare de 8,23 ori mai frecvent decât media

- *Engleza: Light* – 307 apariții
- Frecvența medie a cuvintelor în Biblie în limba engleză: 44,38
- *Light* apare de 6,91 ori mai frecvent decât media

- *Franceza: Lumière* – 193 apariții
- Frecvența medie a cuvintelor în Biblie în limba franceză:
38,93
- *Lumière* apare de 4,95 ori mai frecvent decât media

Lumină / light / lumière - frecvența în Europarl

- *Lumină* – 591 apariții
- Frecvența medie a cuvintelor în Europarl în limba română: 140,55
- *Lumină* apare de 4,20 ori mai frecvent decât media
- *Light* – 1312 apariții (frecvent ca “in light of..”)
- Frecvența medie a cuvintelor în Europarl în limba engleză: 163,39
- *Light* apare de 8,02 ori mai frecvent decât media
- *Lumière* – 872 apariții
- Frecvența medie a cuvintelor în Europarl în limba franceză: 163,6
- *Lumière* apare de 5,33 ori mai frecvent decât media

întuneric / dark(ness) / ténèbres - frecvența în Biblie

- *Întuneric/întunerec* – 154 apariții
- *Întunecat* – 46 apariții
- Frecvența medie a cuvintelor în Biblie română: 37,41
- *Întuneric/întunerec/întunecat* apare de 5,34 ori mai frecvent decât media

- *Dark(ness)* – 206 apariții
- Frecvența medie a cuvintelor în Biblie în engleză: 44,38
- *Dark(ness)* apare de 4,58 ori mai frecvent decât media

- *Ténèbres* – 151 apariții
- Frecvența medie a cuvintelor în Biblie în franceză: 38,93
- *Ténèbres* apare de 3,87 ori mai frecvent decât media

întuneric / dark(ness) / obscurité - frecvența în Europarl

- *Întuneric/întunecat* – 96 apariții
- Frecvența medie a cuvintelor în Europarl în română: 140,55
- *Întuneric/întunecat* apare de 1,46 ori mai rar decât media
- *Dark(ness)* – 100 apariții; *darker*: 5; *darkest*: 10
- Frecvența medie a cuvintelor în Europarl în limba engleză: 163,39
- *Dark(ness)* apare de 1,63 ori mai rar decât media
- *Obscure/obscurité* – 41 apariții
- Frecvența medie a cuvintelor în Europarl în franceză: 163,6
- *Obscure/obscurité* apare de 3,99 ori mai rar decât media

Cele mai frecvente sinonime în Biblie

Sinonim	Frecvența
străluci	137
lume	270
vedere	93
strălucire	5
lumânare	1

Sinonim	Frecvența
fall	267
clean	138
faint	54
loose	65
bright	51

Sinonim	Frecvența
jour	1927
vie	619
gloire	428
feu	512
éclat	102

Totalul frecvențelor sinonimelor:

- Lumină: 508
- Light: 812
- Lumière: 5097

Cele mai frecvente sinonime în Europarl

Sinonim	Frecvența
străluci	60
lume	6156
vedere	13168
watt	1
lumânare	5

Sinonim	Frecvența
clear	10430
clean	414
enlighten	52
short	1482
weak	743

Sinonim	Frecvența
raison	8327
vérité	3148
jour	5774
évident	3392
vie	5791

Totalul frecvențelor sinonimelor:

- Lumină: 19391
- Light: 14793
- Lumière: 36636

Lumină vs întuneric

	Noul Testament	Vechiul Testament
freq(lumină) / freq(întuneric)	1,92	1,35
freq(light)/ freq(dark)	1,85	1,35
freq(lumière)/ freq(ténèbres)	1,43	1,19

Future work

- Noi interpretari.
- Dezambiguizarea sensurilor.
- Analiza contextelor.
- Analiza polaritatilor textelor: sunt unele texte mai optimiste decat altele?

MULTUMESC!

Lingvistica Matematica si Computationala

Liviu P. Dinu,

ldinu@fmi.unibuc.ro

University of Bucharest

Center for Computational Linguistics,

Faculty of Mathematics and Computer Science

nlp.unibuc.ro



Does Translation Influence the Readability of Political Speeches?

Readability. Definition

- **Readability** is the ease with which a written text can be understood by a reader.
- The problem that we address here is whether human translation has impact on readability.
- We investigate the main shallow, lexical and morpho-syntactic features.

Methodology

- Given a text T_1 in a target language L_1 and the texts in source languages L_2, \dots, L_n , how does the readability level vary from a text written in the native language of a speaker and a text translated into the same language?
- Is the original text more comprehensible?
- We consider English as the target language, i.e., we investigate texts written (or translated) in English

Flesch-Kincaid

- We employ the Flesch-Kincaid measure, which assesses readability based on the average number of syllables per word and the average number of words per sentence:

$0.39 \cdot \frac{\text{total words}}{\text{total sentences}} + 11.8 \cdot \frac{\text{total syllables}}{\text{total words}}$

- 15.59

- The Flesch-Kincaid formula produces values which correspond with U.S. Grade levels.

Approach

- We run our experiments on Europarl, a multilingual parallel corpus extracted from the proceedings of the European Parliament
- To obtain the dataset for our experiments, we extract segments of text written in English, we identify their source languages, and we group them based on the language of the speaker.

- 
- We compute the Flesch-Kicaid formula for each collection of segments of text T_i having the source language L_i and the target language English.

Experiments and Results

- In a first experiment, we compute the Flesh-Kincaid metric for each language, for all the concatenated files in the English Europarl subcorpus

Language	EN	SV	NL	DA	FI	DE	ET	MT	PL	FR	
Readability	11.45	11.50	11.56	11.95	11.99	12.45	12.71	12.79	12.81	13.29	
Language	LV	SL	HU	CS	BG	SK	LT	ES	RO	IT	PT
Readability	13.34	13.35	13.46	13.75	13.90	13.91	14.69	14.72	15.01	15.54	15.6

Table 1: Flesch-Kincaid values for *Europarl*

Results

- One can notice that the lowest Flesh-Kincaid value belongs to the collection of texts having English as the source language, followed by:
- texts having Germanic source languages,
- texts having Slavic source languages and, finally,
- texts translated from Romance languages.

- Finno-Ugric languages are the only family that doesn't form a cluster with regard to the Flesch-Kincaid metric.
- Among the Romance languages, French is the only one that sets apart from the group, being closer to the Germanic cluster, but this fact is justified by the nature of French

3 EXperiments

- Exp1:
- For each language, we account for the overall readability score computed for all documents of each speaker;
- based on these computed values, we determine outliers and remove them from the dataset;
- then, we rerun the experiments based on Flesch-Kincaid measure for the remaining speakers

Experiments

- Exp2:
- We investigate outliers for each speaker by computing the Flesch-Kincaid metric individually for each document belonging to a speaker.
- We discard documents whose levels of readability are outliers and we compute the Flesch-Kincaid formula again accounting only for the documents having the individual level of readability in $[LF;UF]$ range.

3 Experiments

- Exp. 3:
- In the last experiment we consider, for each language, the readability scores of each document belonging to each speaker.
- We apply the same strategy as before: we detect outliers among documents and remove them from the dataset.
- Then, we compute Flesch-Kincaid measure again, for a text consisting of the concatenation of all remaining documents after outliers removal, for each language.

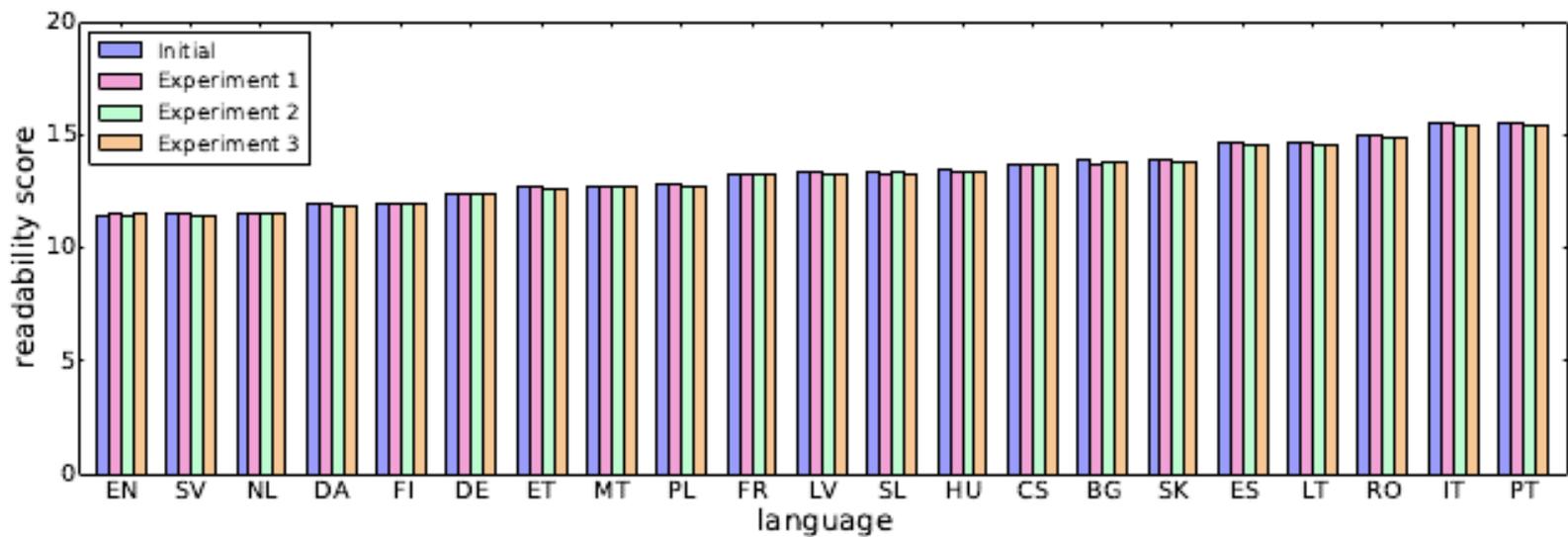


Fig. 2: Flesch-Kincaid values for *Europarl*

Classification

- We investigate the readability of translation as a classification problem.
- Taking as input sentences originally spoken in English and sentences translated from other languages, our goal is to see whether the readability features have enough discriminative power to distinguish original from translated text.
- We train a logistic regression classifier for a binary decision problem: original versus translated text.

Methodology

- We extract randomly 100 sentences originally spoken in English and 100 sentences originally spoken in other languages and translated into English.
- We split this dataset into equal train and test subsets. We choose the optimal value for the regularization parameter performing 3-fold cross-validation over the training set.
- Finally, we evaluate the model on the test set.

Results

- We obtain 58% f-score on the test set in deciding whether a sentence was translated into English or was originally spoken in English.
- The most informative feature is the average number of characters per word (0.69 logistic regression score), followed by the type/token ratio (-0.67 score).
- Adding n-grams of tokens and POS tags as features improves the performance of the model. We obtain 75% f-score.

Conclusions and feature work

- We investigate the behavior of various readability metrics across parallel translations of texts from a source language to target languages.
- We plan to investigate the left-right distinction



Deception Detection

Deception. Definition

- *“To **intentionally** cause another person to have or continue to have a false belief that is truly believed to be false by the person intentionally causing the false belief by bringing about evidence on the basis of which the other person has or continues to have that false belief.”* (Mahon, J.E. (2007). *A Definition of Deceiving. International Journal of Applied Philosophy*, 21, 181-194.
- **No** a general accepted definition

Ingredients

- Intention
- **An act of deceiving is not an act of deceiving unless the result is that another person has a false belief.**
- Deception <> Lies
- Lies definition :
“... to make a believed-false statement with the intention that that statement be believed to be true”. (Mahon, J. E. 2008. Two Definitions of Lying. International Journal of Applied Philosophy, 22(2), 211-230.)

Ingredients (2)

- Deceptive behaviour: **planned** and **unplanned**.
- In planned interactions, people have time to think, reflect and compare situations with past experiences. They know or have time to consider knowing the person who they interact with.
- Planned deceptions are harder to detect.
- Many deceptions types, many medium of communication

Deception exists in various forms...

- Fake (Armstrong) Real (Quintana)



Traditional Approaches:

- *Psychology* and *criminal justice* have studied the behaviors that might be associated, with deception
- Three types of behavior have been examined:
 1. facial expressions and body movements;
 2. vocal behaviors, including prosodic features;
 3. verbal behaviors, including the words and structures that might correlate with deception.

Deception Detection. New trends

- NLP approaches to address the vocal and verbal features that might be associated with deception
- NLP papers on the classification of narratives as truthful or deceptive
- Stylometric techniques, machine learning approaches and models of data collection and processing

EACL-2012 First Workshop on Computational Approaches
to Deception Detection. Avignon, may 2012
<http://aclweb.org/anthology-new/W/W12/W12-04.pdf>



Liviu Dinu, primul (cum altfel?) din stînga, la o conferință despre fraudele prin copiere. Undeva printre participanți este și șeful Interpol Italia, pe care, evident, din motive de securitate, nu vi-l putem indica.

Fake reviews detection: Ott&Tomasso

Ott et al. (2011)

Features

Truthful reviews

- **Tempered** opinions
- More **spatial** details
- More **nouns** and **adjectives**
- More **numbers** and **punctuation**

Fake reviews

- **Exaggerated** opinions
- Greater focus on aspects **external** to the hotel
- More **pronouns, verbs** and **adverbs**
- More **filler** (blah, like)



Cornell University

Which of these reviews is **fake**?

"I have stayed at many hotels traveling for both business and pleasure and I can honestly say that The James is tops. The service at the hotel is first class. The rooms are modern and very comfortable. The location is perfect within walking distance to all of the great sights and restaurants. Highly recommend to both business travellers and couples."

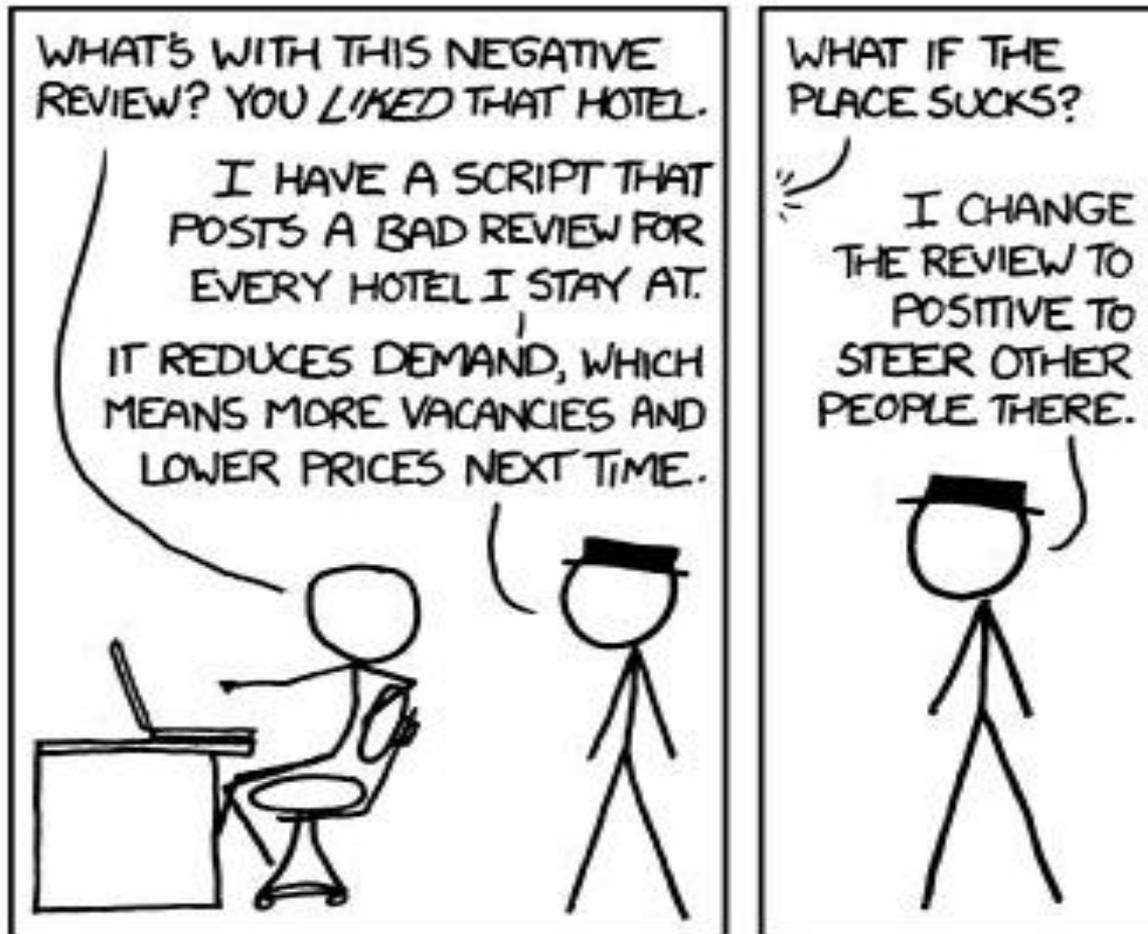
"My husband and I stayed at the James Chicago Hotel for our anniversary. This place is fantastic! We knew as soon as we arrived we made the right choice! The rooms are BEAUTIFUL and the staff very attentive and wonderful!! The area of the hotel is great, since I love to shop I couldn't ask for more!! We will definatly be back to Chicago and we will for sure be back to the James Chicago."



Cornell University

In Search of a Gold Standard in Studies of Deception

Stephanie Gokhman, Jeff Hancock, Poornima Prabhu, Myle Ott and Claire Cardie (Deception detection ws)





MULTUMESC!

Lingvistica Matematica si Computationala

Liviu P. Dinu,

ldinu@fmi.unibuc.ro

University of Bucharest

Center for Computational Linguistics,

Faculty of Mathematics and Computer Science

nlp.unibuc.ro



Authorship identification

Authorship identification

- „ Then there is the letter he said I wrote him. In his hopeless ignorance of civilized conduct and the usages of society, he read it aloud. . . . but I ask you, how would you reply if I were **to deny** ever having sent you that letter? Where is your witness to contradict me? Would you **prove it** by the handwriting? . . . but how could you when the letter is in the hand of a secretary?” (Cicero, Philippics II, Bailey 1986:37)

*“De-acum i se va putea atribui oricui orice în incontrolabilul (sau greu controlabilului) mediu electronic. **Dacă nu se vor pune la punct tehnici care să permită mergerea la sursa inițială și identificarea autorului** în cazuri de acest fel, potențialul de calomnie, fals și minciună devine copleșitor.”* (Mircea Cartarescu, 2009)

Motivation

- The problem of authorship identification is based on the assumption that there are stylistic features that help distinguish the real author from any other possibility.
- Literary-linguistic research is limited by the human capacity to analyze and combine a small number of text parameters, to help solve the authorship problem.

Motivation

- We can surpass limitation problems using computational and discrete methods, which allow us to explore various text parameters and characteristics and their combinations.
- The text characteristics and parameters used to determine text paternity need not have aesthetic relevance. They must be objective, unambiguously identifiable, and quantifiable, such that they can be easily differentiated for different authors.

Human stylom (van Halteren et al, 2005)

- Stilistical Fingerprint.
- Human stylom (van Halteren et al, 2005): The set of language use characteristics - stylistic, lexical, syntactic - form the human stylom

Standard problems (cf. S. Marcus)

1. A text attributed to one author seems nonhomogeneous, lacking unity, which raises the suspicion that there may be more than one author.
2. If based on certain circumstances, arising from literature history, the paternity is disputed between two possibilities, A and B, we have to decide if A is preferred to B, or the other way around.

Problems

1. A text is anonymous. If the author of a text is unknown, then based on the location, time frame and cultural context, we can conjecture who the author may be and test this hypothesis
2. Based on literary history information, a text seems to be the result of the collaboration of two authors, an ulterior analysis should establish, for each of the two authors, their corresponding text fragments.

Solutions

- Two strategies:
- The first strategy is based on Support Vector Machines (SVM) with a string kernel
- The second one is a new strategy based on the similarity of rankings of function words.

Rank distance and authorship

- We propose Rank distance as a new distance measure designed to reflect stylistic similarity between texts.
- As style markers we used the function word frequencies.
- Function words are generally considered good indicators of style because their use is very unlikely to be under the conscious control of the author and because of their psychological and cognitive role (Chung and Pennebaker, 2007).

Function word

- Also function words prove to be very effective in many author attribution studies
- Given a fixed set of function words (usually the most frequent ones), a ranking of these function words according to their frequencies is built for each text; the obtained ranked lists are subsequently used to compute the distance between two texts.
- To calculate the distance between two rankings we used Rank distance

Function word (2)

- In all our english experiments we used the set of 70 function words identified by Mosteller and Wallace (Mosteller and Wallace, 1964) as good candidates for authorattribution studies
- In all our Romanian experiments we used the set of function words identified by (Dinu and Popescu)

a	been	had	its	one	that	was
all	but	has	may	only	the	were
also	by	have	more	or	their	what
an	can	her	must	our	then	when
and	do	his	my	shall	there	which
any	down	if	no	should	things	who
are	even	in	not	so	this	will
as	every	into	now	some	to	with
at	for	is	of	such	up	would
be	from	it	on	than	upon	your

Table 1: Function words used in computing the distance

Method

- Once the set of function words is established, for each text a ranking of these function words is computed.
- The ranking is done according to the function word frequencies in the text.
 - Rank 1 will be assigned to the most frequent function word, rank 2 will be assigned to the second most frequent function word, and so on

Distance

- The distance between two texts will be the Rank distance between the two rankings of the function words corresponding to the respective texts.
- We use it as a base for a hierarchical clustering algorithm.
- The family trees (dendrogram) thus obtained can reveal a lot about the distance measure behavior

Experiments

- We cluster a collection of 21 nineteenth century English books written by 10 different authors and spanning a variety of genres (Table 2).
- The books were used by Koppel et al. (Koppel et al., 2007) in their authorship verification experiments.
- the family tree produced is a very good one, accurately reflecting the stylistic relations between books.

Analyse

- The books were grouped in three big clusters (the first three branches of the tree) corresponding to the three genre:
- dramas (lower branch),
- essays (middle branch)
- and novels (upper branch).
- Inside each branch the works were first clustered according to their author.

Analyse

- The only exceptions are the two essays of Emerson which instead of being first cluster together and after that merged in the cluster of essays, they were added one by one to this cluster.
- Even more, in the cluster of novels one may distinguished two branches clearly separated that can correspond to the gender or nationality of the authors: female English (lower part) and male American (upper part).

Group	Author	Book
American Novelists	Hawthorne	Dr. Grimshawe's Secret
		House of Seven Gables
		Redburn
	Melville	Moby Dick
		The Last of the Mohicans
	Cooper	The Spy
Water Witch		
American Essayists	Thoreau	Walden
		A Week on Concord
	Emerson	Conduct Of Life
		English Traits
British Playwrights	Shaw	Pygmalion
		Misalliance
		Getting Married
	Wilde	An Ideal Husband
		Woman of No Importance
Bronte Sisters	Anne	Agnes Grey
		Tenant Of Wildfell Hall
	Charlotte	The Professor
		Jane Eyre
	Emily	Wuthering Heights

Table 2: The list of books used in the experiment

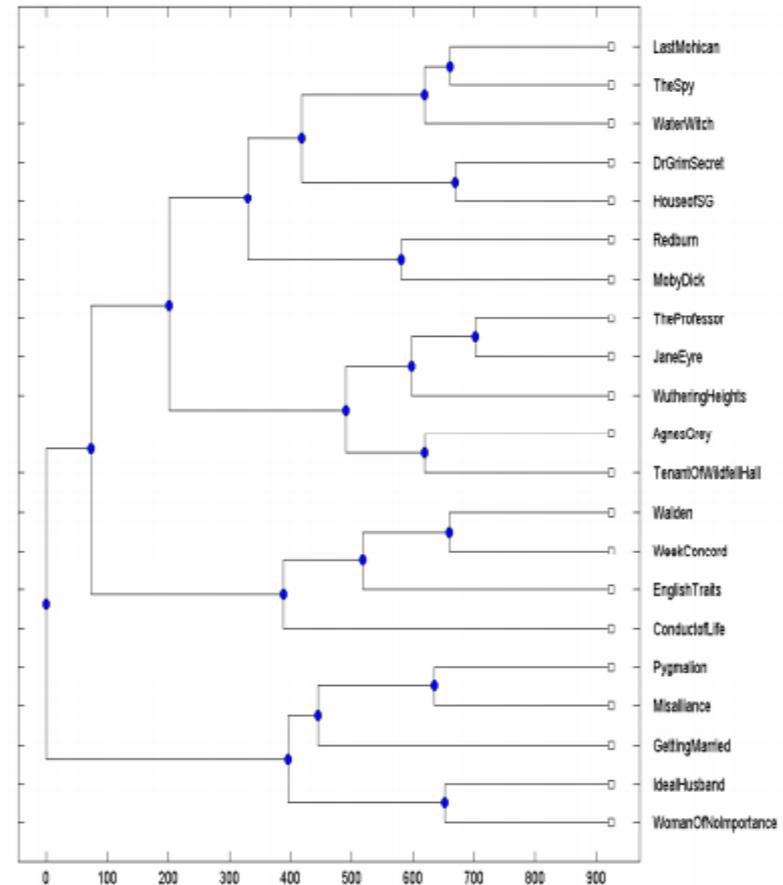


Figure 1: Dendrogram of 21 nineteenth century En-

Binary classification experiment

- We tested the nearest neighbor classification algorithm combined with both rank distance and euclidean distance on the case of the 12 disputed federalist papers (Mosteller and Wallace, 1964).
- We followed the Mosteller and Wallace setting, treating the problem as a binary classification problem.
- Each one of the 12 disputed papers has to be classified as being written by Hamilton or Madison. For training are used the 51 papers written by Hamilton and the 14 papers written by Madison

Comparision

- Tested on disputed papers, the nearest neighbor classification algorithm combined with rank distance attributed all the 12 papers to Madison.
- This matches the results obtained by Mosteller and Wallace and is in agreement with today accepted thesis that the disputed papers belong to Madison.
- When the nearest neighbor classification algorithm was combined with euclidean distance only 11 papers were attributed to Madison, the paper 56 was attributed to Hamilton.



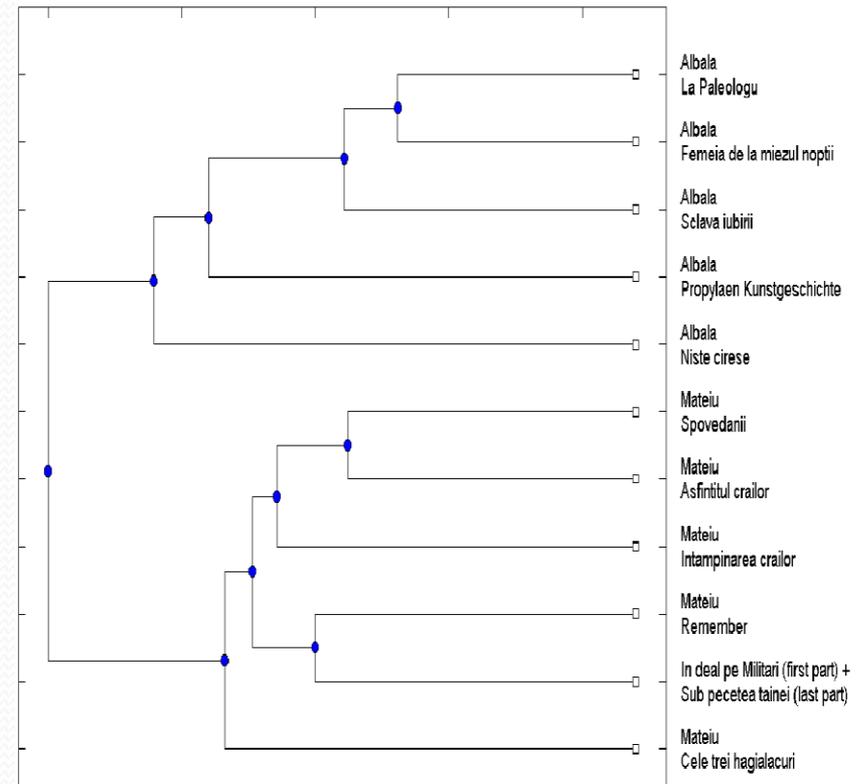
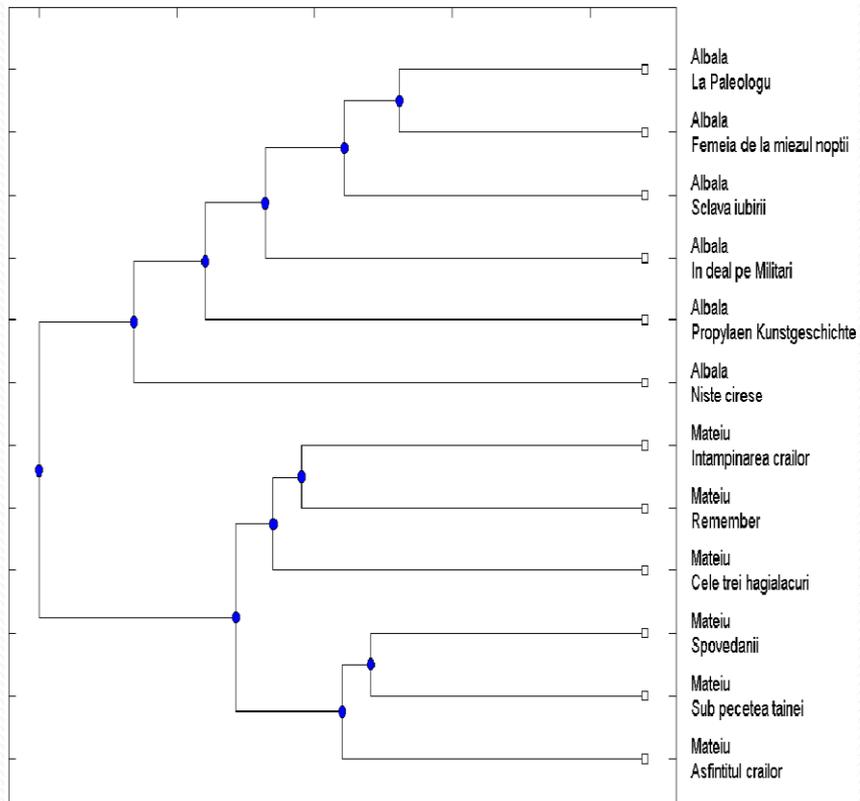
MORE EXPERIMENTS

Stilistic Deception. Mateiu and followers

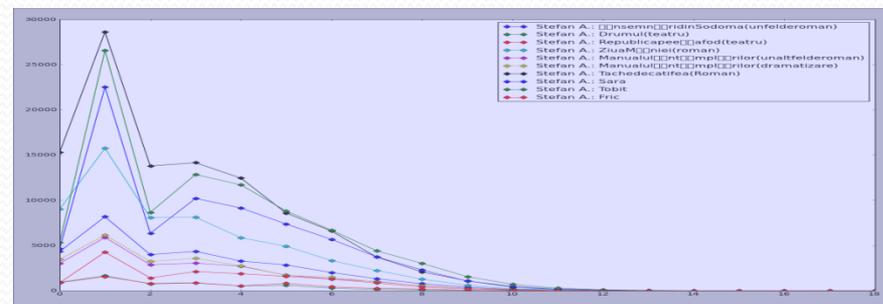
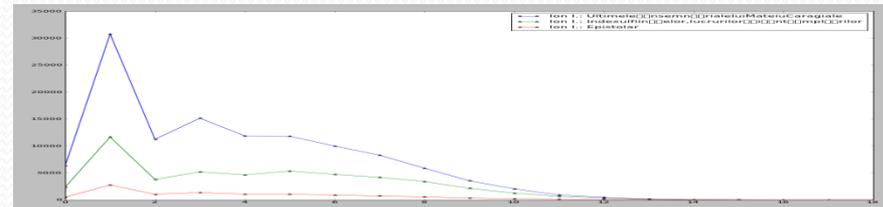
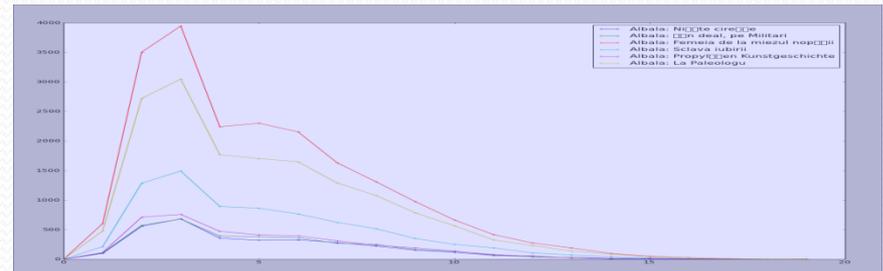
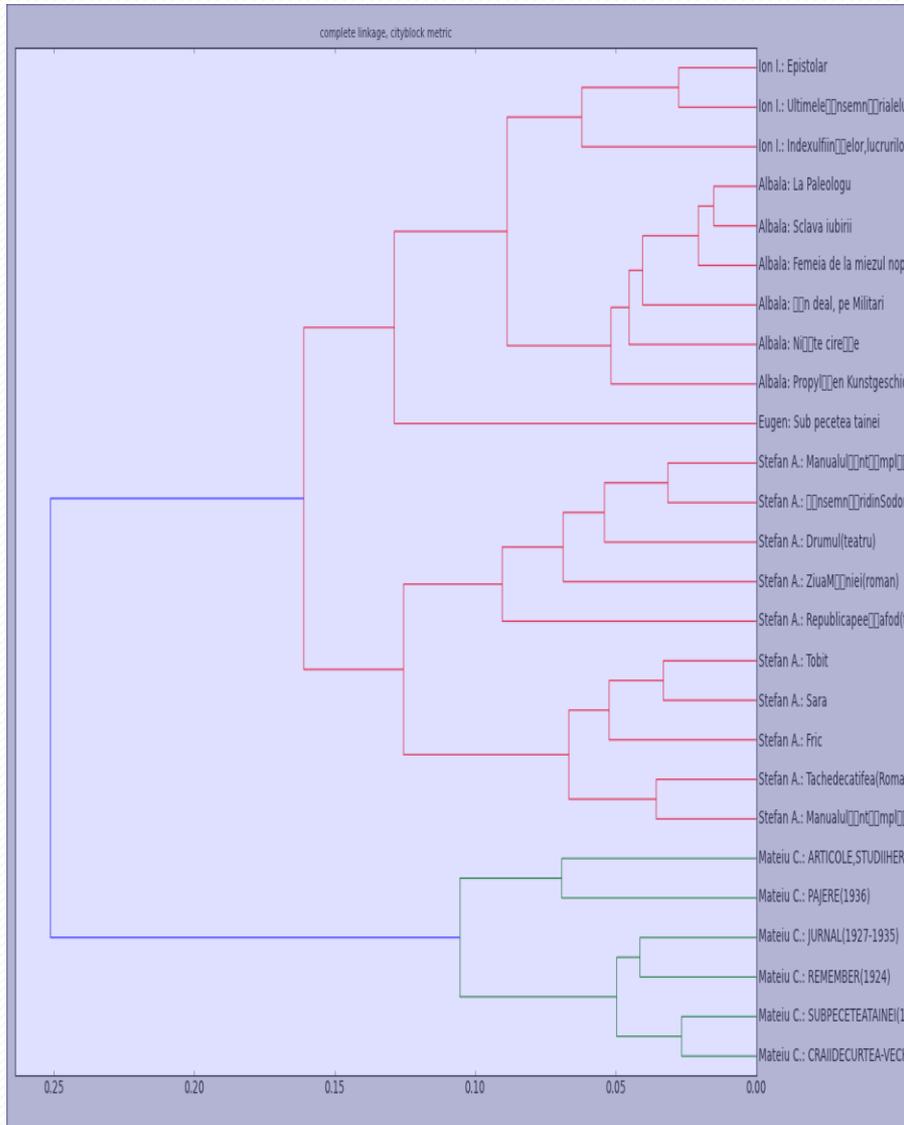
- *Mateiu Caragiale* died on 1936, at age of 51. In 1929 he begun to works to the novel "*Sub pecetea tainei*", but *unfortunately* he died before finishing this novel.
- Many authors attempted to write different endings to the novel: Radu Albala, Al. George, George Balan

- In 2008, Ion Iovan published the so-called *Last Notes of Mateiu Caragiale*, composed of sections written from Iovan's voice, and another section in the style of a personal diary describing the life of Mateiu Caragiale, suggesting that this is really Caragiale's diary

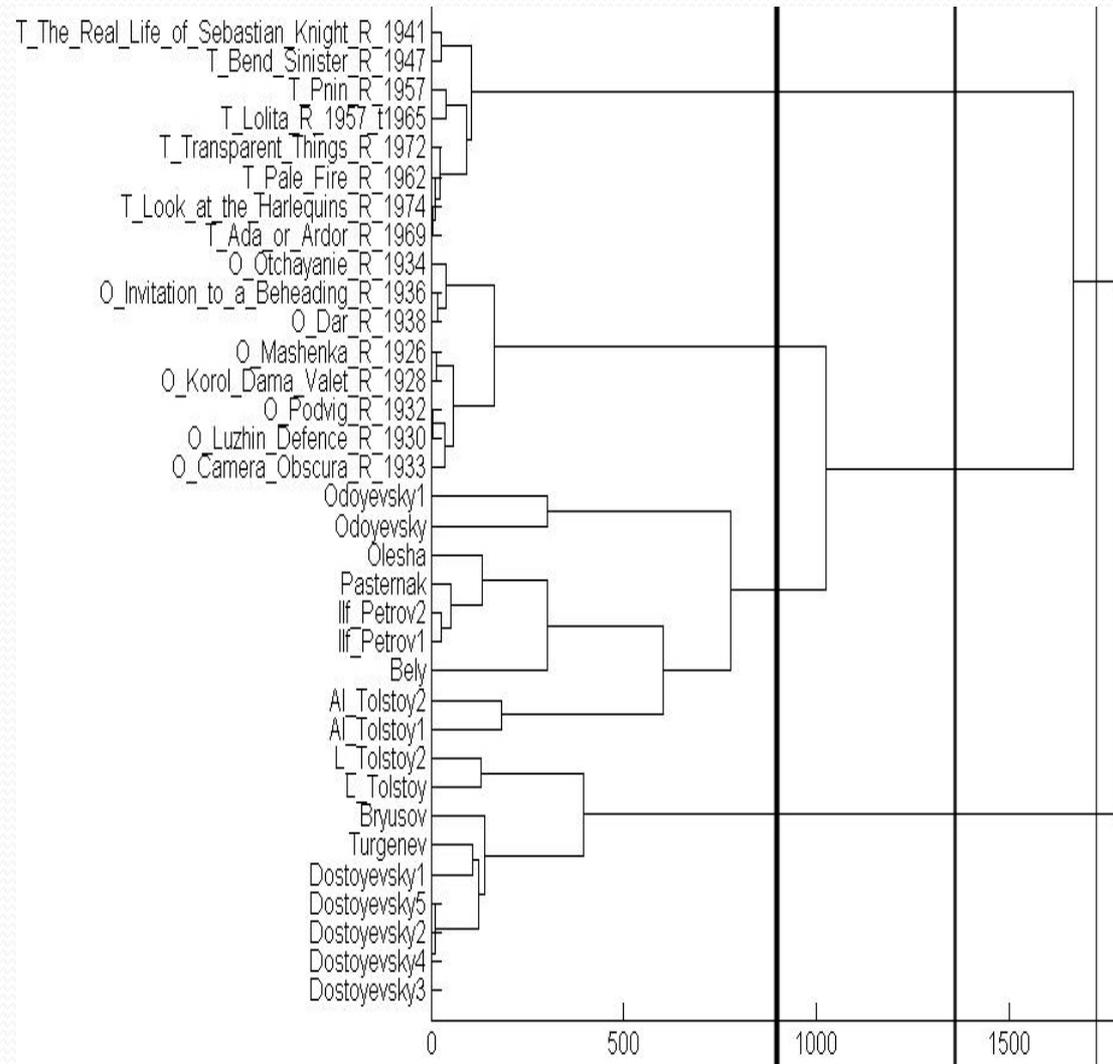
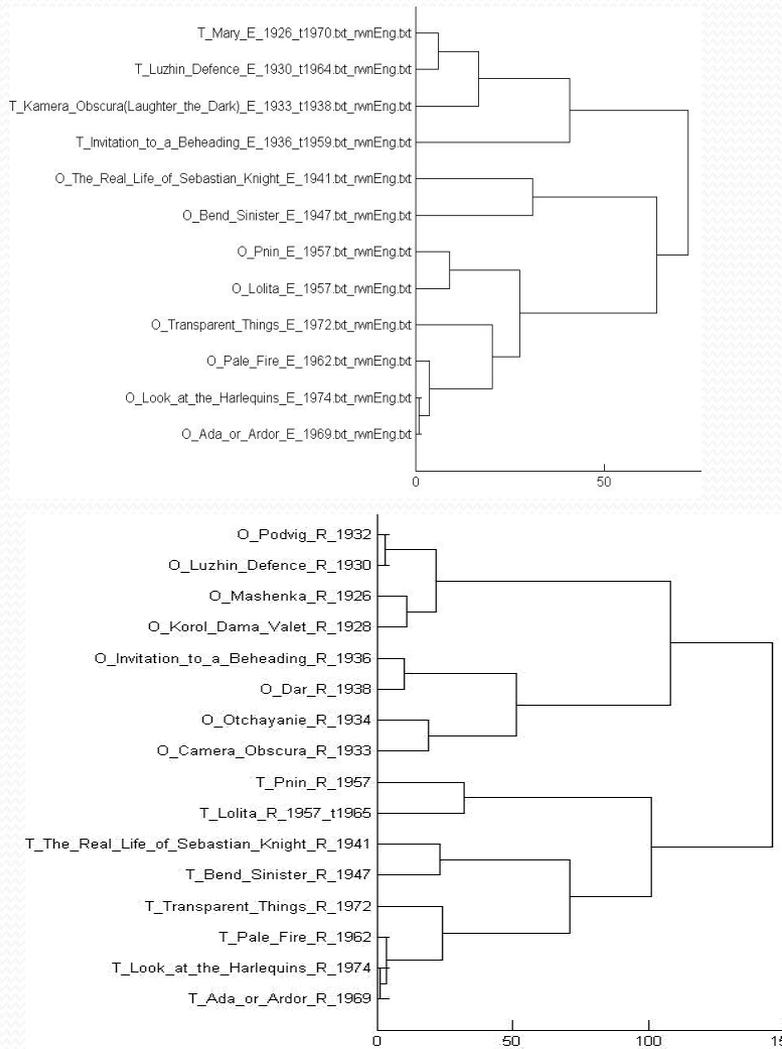
Albala vs Mateiu (Dinu, Popescu & Dinu, LREC08)



Mateiu Caragiale. Pastiche (Dinu et al., ws at EACL12)



Nabokov (Dinu&Nisioi, RANLP13)



The stilistics unity of Pauline Epistles

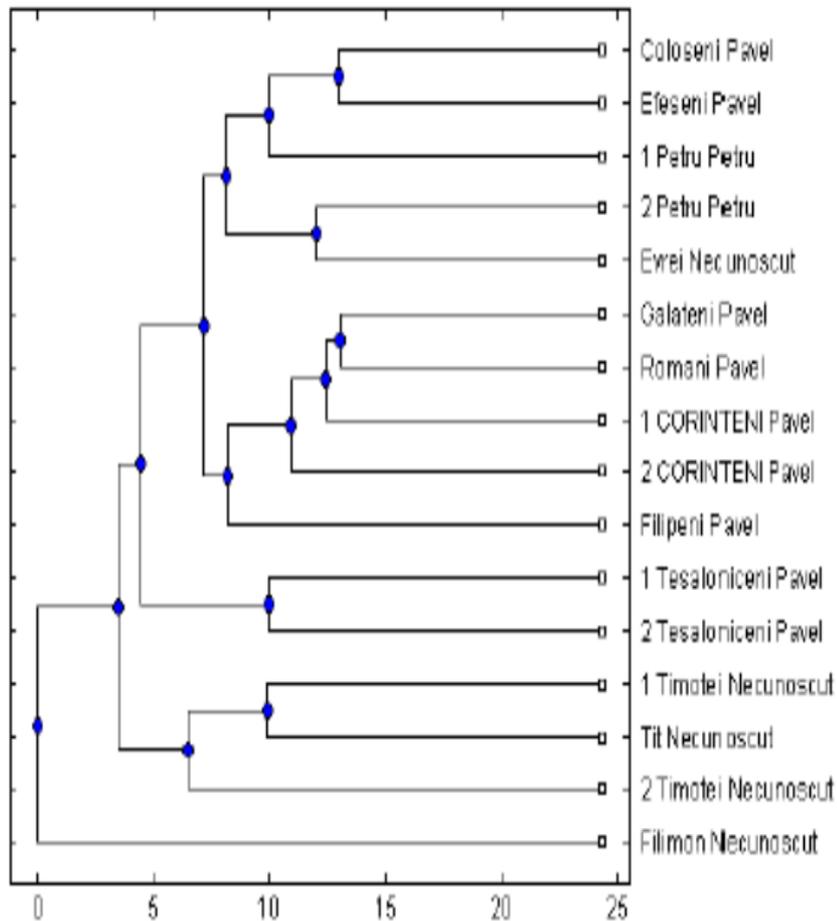
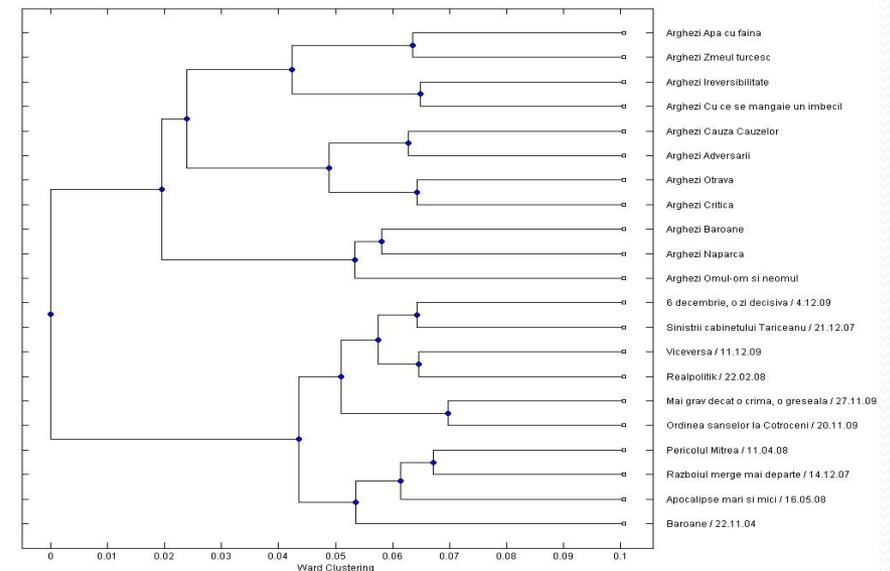
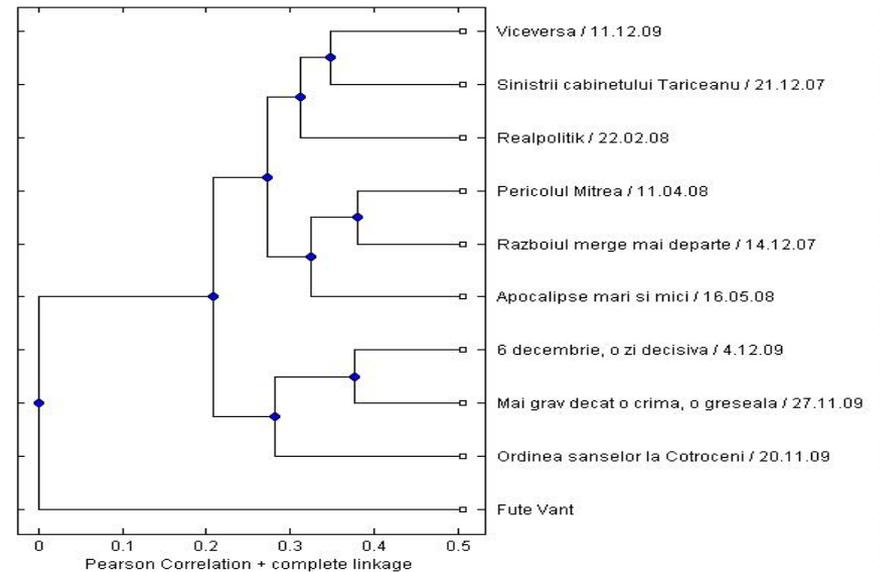


Figure 3: Dendrogram of Paul and Peter epistles

- St. Paul seems to dictate his letters to his disciples Timothy, Silvanus (= Silas)
- Philemon is a single cluster (was written during the jail period)

Other results

- The paternity of Eminescu publicistics
- Mircea Cartarescu
- Federalist Papers
- ...



More applications...

- Temporal text classifications (EACL 2014)
- Opinion mining and sentiment analysis
- Text categorization
- Political ideology detection

(more at <http://www.kenbenoit.net/new-directions-in-analyzing-text-as-data-workshop-2013/>)

- ...

A receipt for happiness

(<http://www.cse.unt.edu/~rada/>)

- Ingredients

1. - Something new
2. - Lots of food that you enjoy
3. - Your favorite drink
4. - An interesting social place

- Directions : *“go shop for something new ... Then have lots of food, for dinner preferably, as the times of breakfast and lunch are to be avoided. Consider also including .. your favorite drinks. Then go to an interesting place, it could be a movie, a concert, a party, or any other social place. Having fun, and optionally getting drunk... Note that you should avoid any unnecessary actions, as they can occasionally trigger feelings of unhappiness. Ideally the recipe should be served on a Saturday, for maximum happiness effect.*

Bon appétit!

MULTUMESC!

Temporal Text Ranking and Automatic Dating of Texts

EACL 2014, Göteborg

Vlad Niculae (Max Planck Institute for Software Systems)

Marcos Zampieri (Saarland University)

Liviu P. Dinu (University of Bucharest)

Alina Maria Ciobanu (University of Bucharest)

1. Text Dating

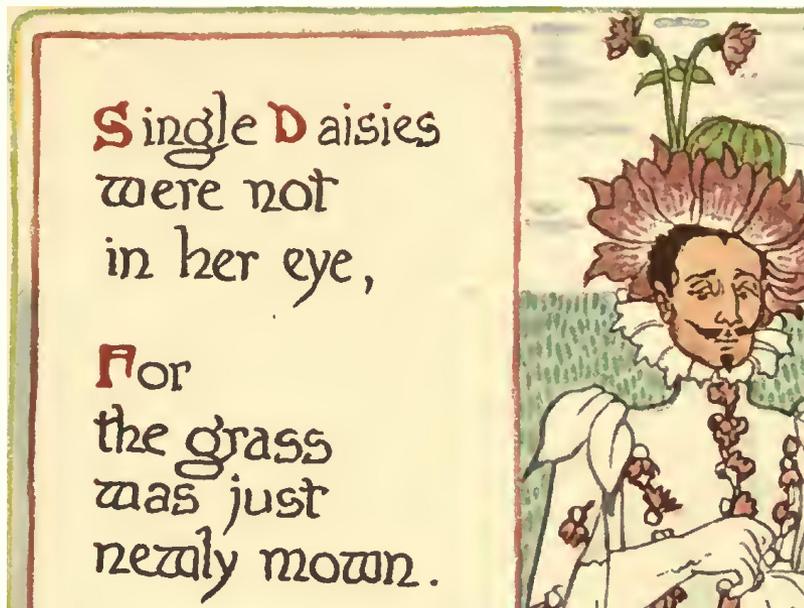
Estimate the writing date of a text.

(Linguistic complement to *material dating*.)

1. Text Dating

Estimate the writing date of a text.

(Linguistic complement to *material dating*.)

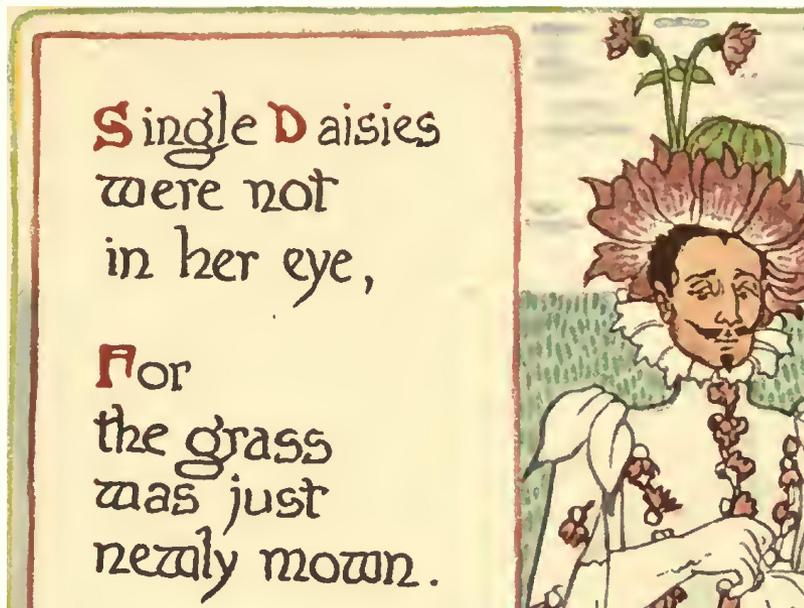


- 1930? 1899? 1823?
(Regression)
(Preoțiu-Pietro and Cohn, 2013)
- 18th / 19th century?
(Classification)
(de Jong et al, 2005)
and our previous work

1. Text Dating

Estimate the writing date of a text.

(Linguistic complement to *material dating*.)



- Which is newer?

A Relation

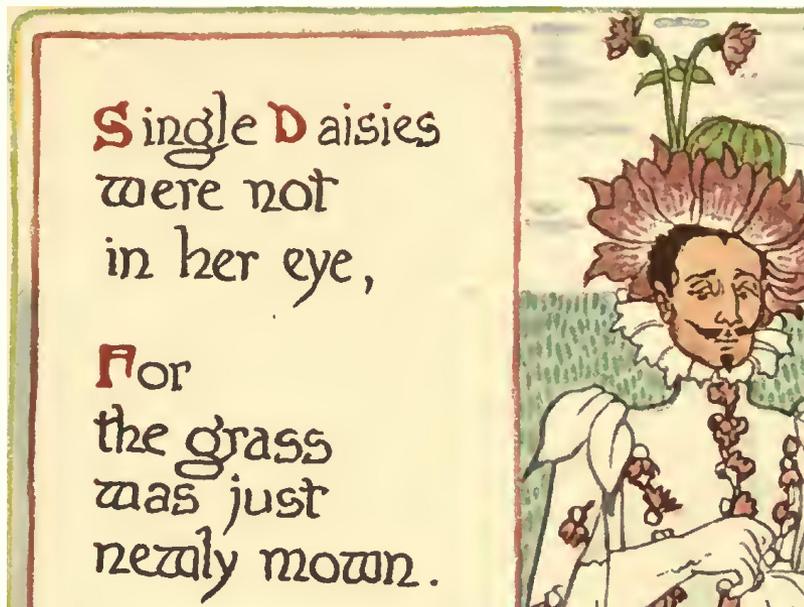
Of some Trials of the same Operation, lately made in France.

1. *M. Denys*, Professor of the *Mathematicks* and *Natural Philosophy* at *Paris*, in a Letter of his to the *Publisher* relateth, That they had lately transmitted the Blood of four *Weathers* into a *Horse* of 26 years old, and that this *Horse* had thence received much strength, and more than an ordinary stomach.

1. Text Dating

Estimate the writing date of a text.

(Linguistic complement to *material dating*.)



1899. W. Crane, A Floral Fantasy in an Old English Garden

- Which is newer?

A Relation

Of some Trials of the same Operation, lately made in France.

1. *M. Denys*, Professor of the *Mathematicks* and *Natural Philosophy* at *Paris*, in a Letter of his to the *Publisher* relateth, That they had lately transmitted the Blood of four *Weathers* into a *Horse* of 26 years old, and that this *Horse* had thence received much strength, and more than an ordinary stomach.

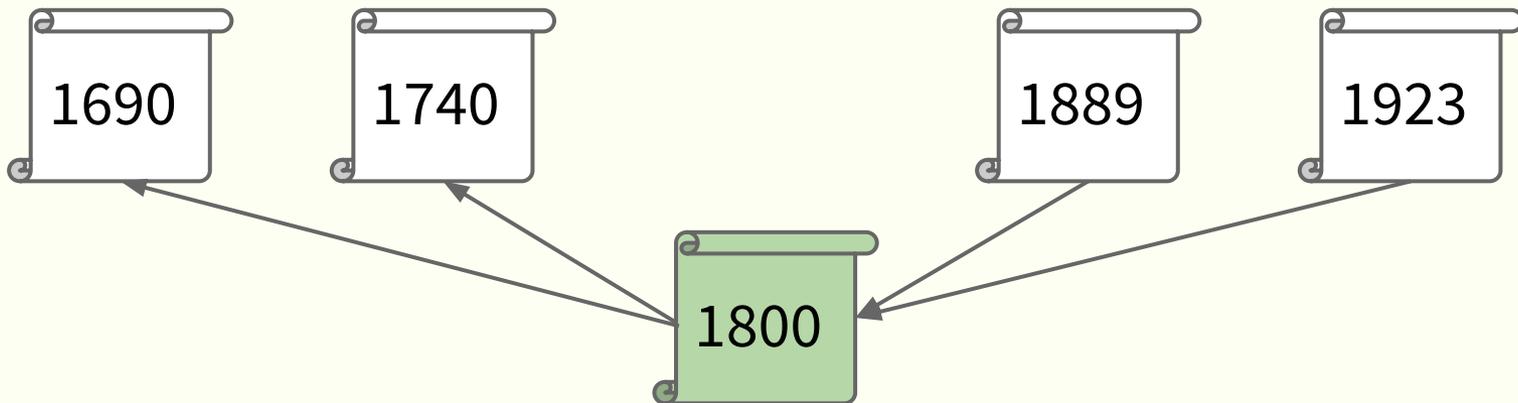
1667. An Account Of The Experiment Of Transfusion Practiced Upon A Man In London

2. This Work: Pairwise Ranking

Input: **pairs** of documents

Output: “<”, “>”

Not all input samples need to be comparable.

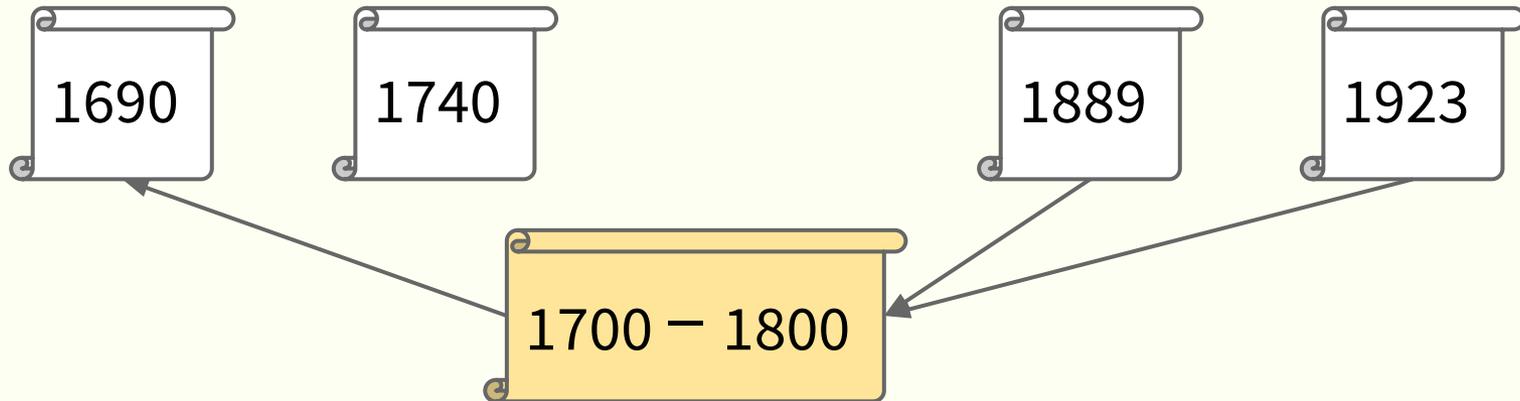


2. This Work: Pairwise Ranking

Input: **pairs** of documents

Output: “<”, “>”

Not all input samples need to be comparable.



3. Behind the Scenes

Binary classification of pairs.

$$g(d_1, d_2) > 0$$

But we want the dates, not a ranking!

3. Behind the Scenes

Binary classification of pairs.

$$g(d_1, d_2) > 0$$

But we want the dates, not a ranking!

$$w \cdot (d_1 - d_2) > 0$$

$$w \cdot d_1 > w \cdot d_2$$

3. Behind the Scenes

Binary classification of pairs.

$$g(d_1, d_2) > 0$$

But we want the dates, not a ranking!

$$w \cdot (d_1 - d_2) > 0$$

$$w \cdot d_1 > w \cdot d_2$$

Use a moment in time instead of a document:

$$w \cdot d_1 > \theta(1850)$$

Evaluation

4. Historical Corpora

Three languages:

- Colonia Corpus of Historical **Portuguese**
(Zampieri and Becker, 2013)
- Corpus of Late Modern **English** Texts (CLMET)
(de Smet, 2005)
- **Romanian** Historical Corpus
(Ciobanu et al. 2013)

5. Simple Features

A. lexical (word counts)

B. naive morphological

(character n-grams at the end of words)

+ feature transformation and selection

6. Results

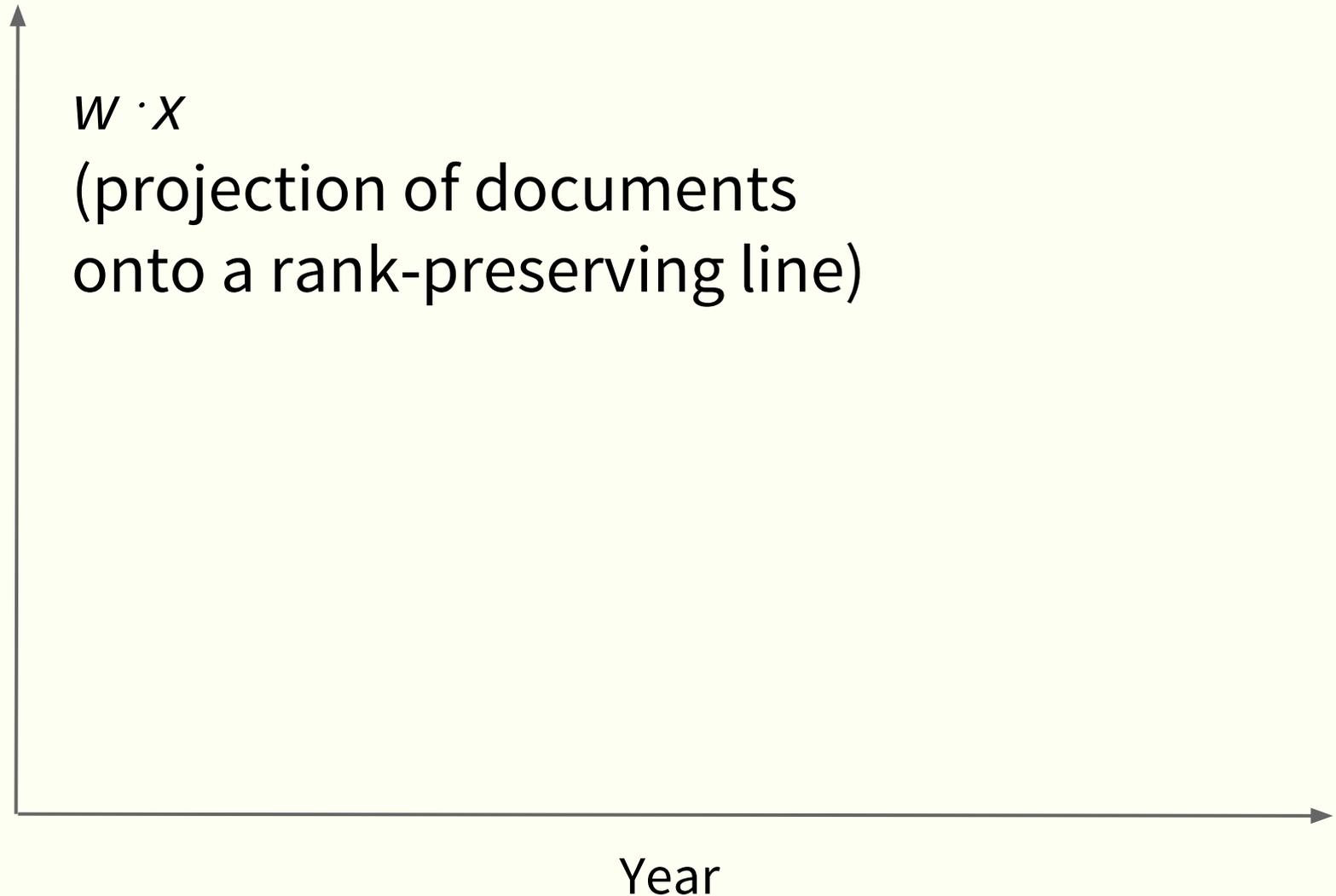
Comparable to the regression approach

	size	pairwise score	Ridge pairwise score
en	293	83.8%	83.7%
pt	87	82.9%	81.9%
ro	42	92.9%	92.4%

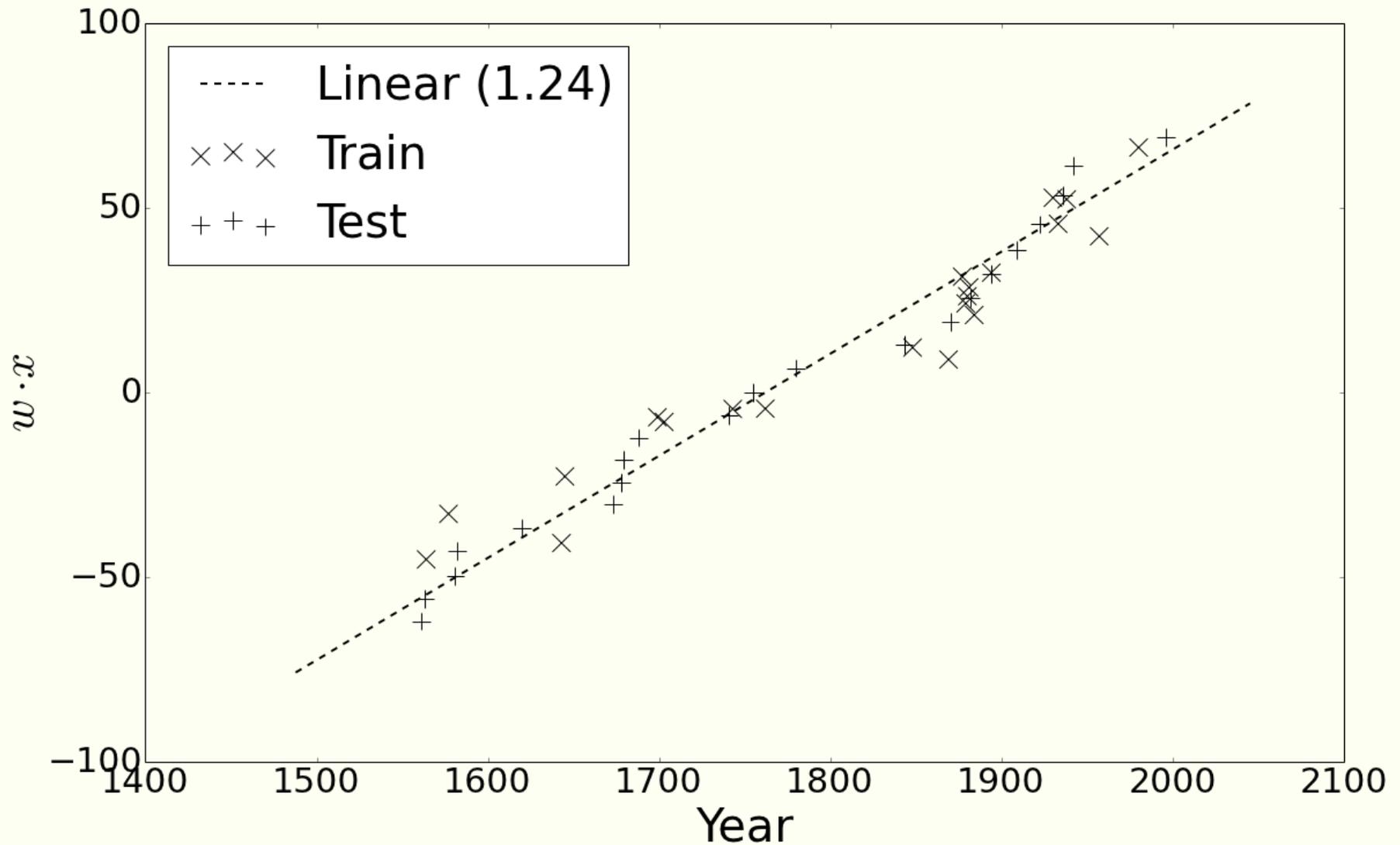


our system

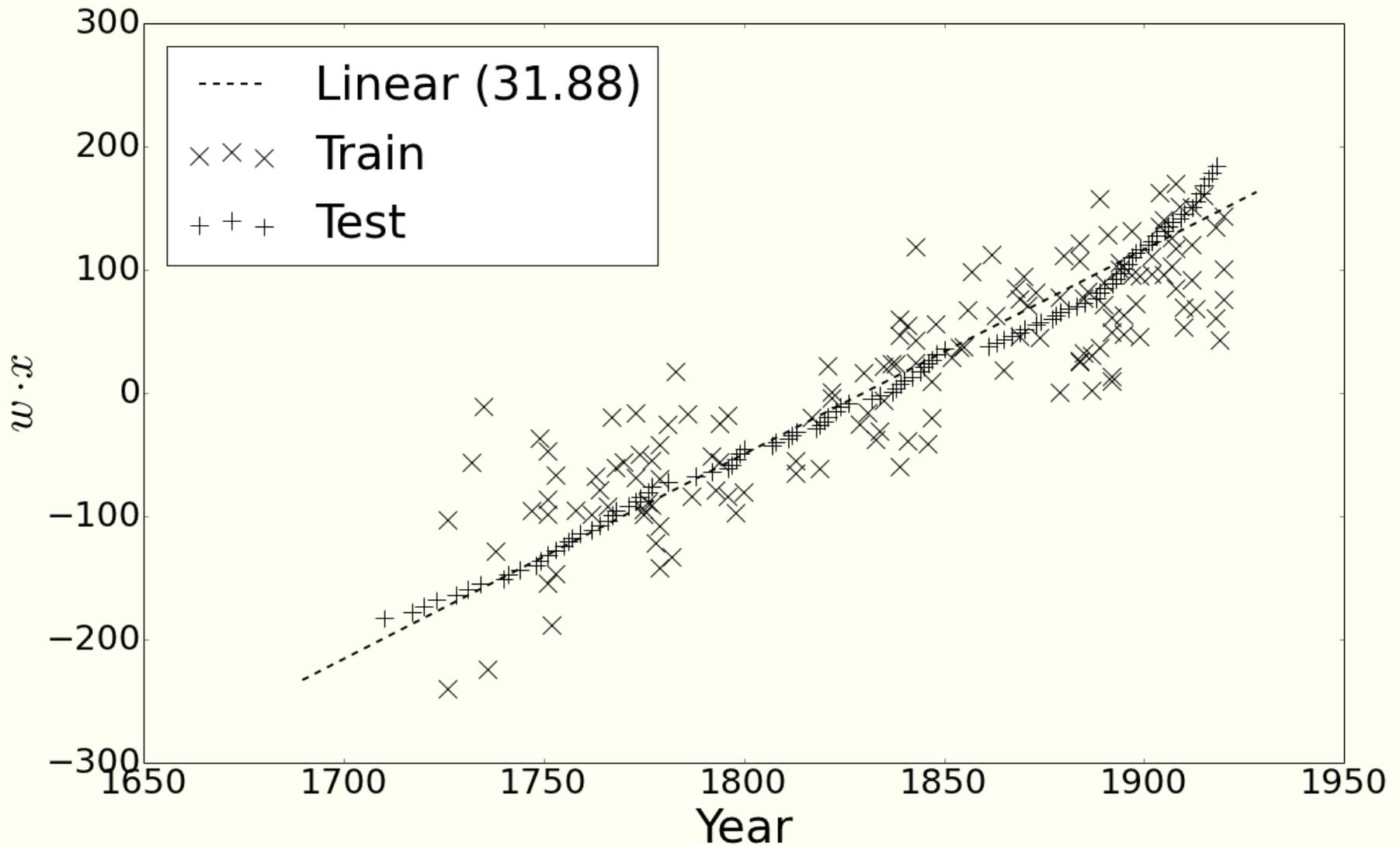
7. Function estimation (θ)



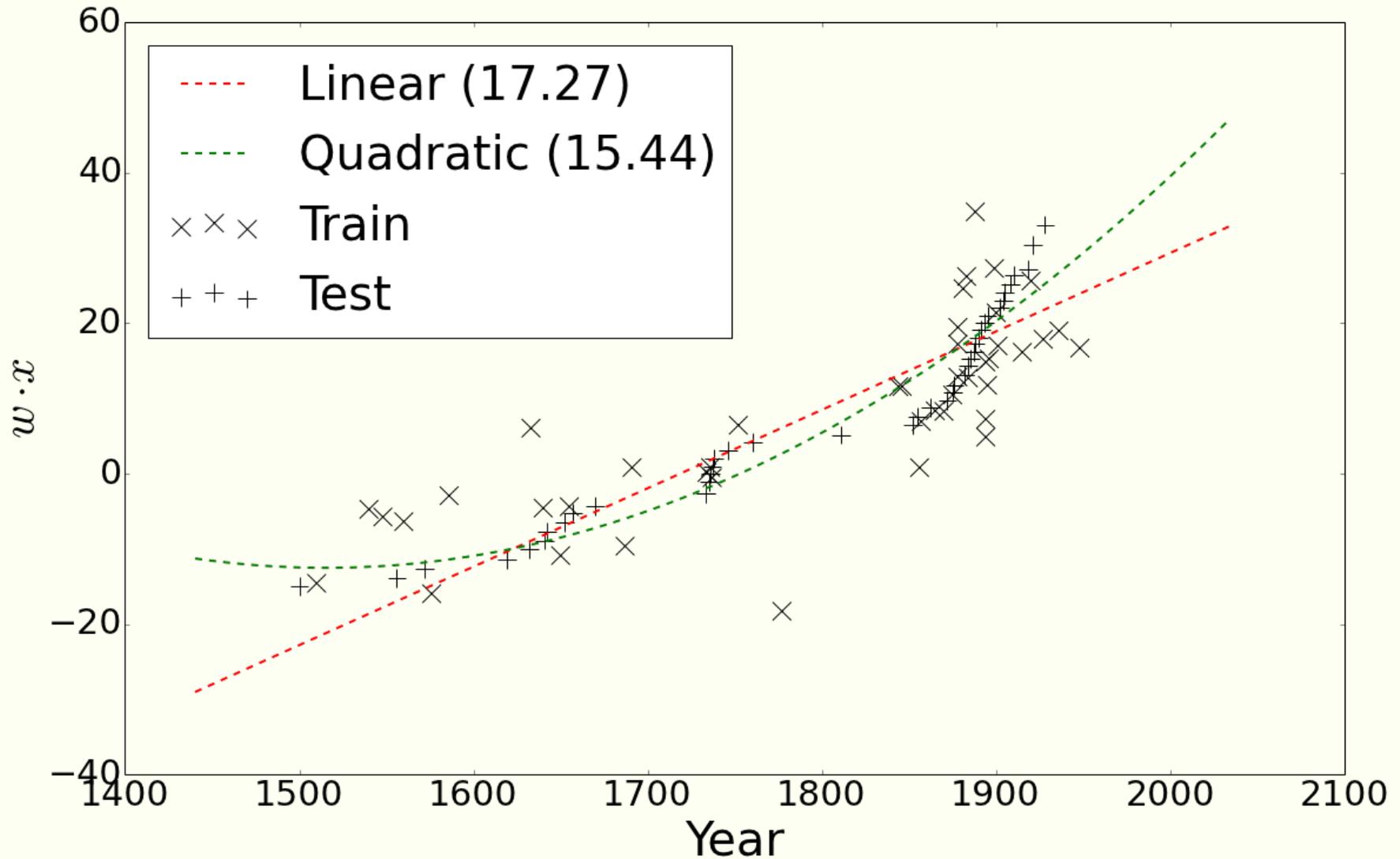
8. Function estimation (Romanian)



9. Function estimation (English)



10. Function estimation (Portuguese)



11. Dating uncertain texts

C. Cantacuzino (1650 – 1716), *Istoria Țării Rumânești*

Important historical work, contested writing time.

Published: 19th century.

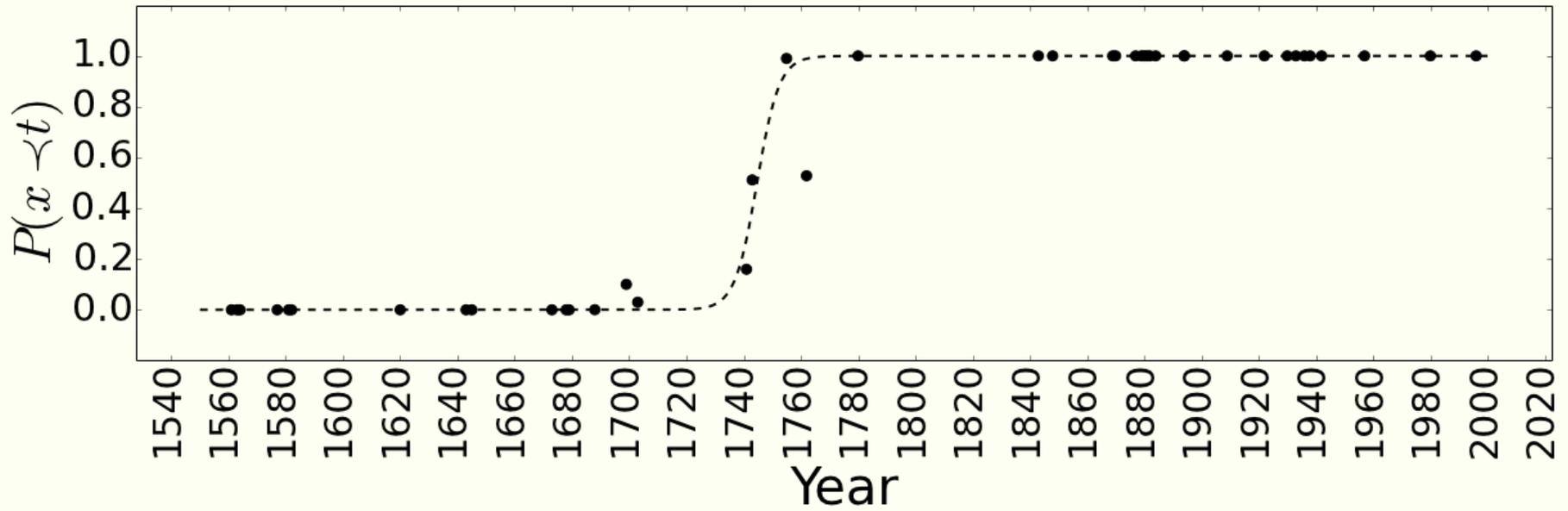
11. Dating uncertain texts

C. Cantacuzino (1650 – 1716), *Istoria Țării Rumânești*

Important historical work, contested writing time.

Published: 19th century.

We predict 1736.2 – 1753.2:



12. Conclusion & Future Work

- ranking approach to temporal modelling
- important gain on flexibility
- acceptable performance with simple features

12. Conclusion & Future Work

- ranking approach to temporal modelling
- important gain on flexibility
- acceptable performance with simple features

- application-specific feature engineering
- other historical corpora wanted!



MULTUMESC!