Elements of elements of information retrieval

Czyli jak trudny był doktorski egzamin przedmiotowy?

Dominika Tkaczyk

ICM, Uniwersytet Warszawski

21 June 2016

Introduction

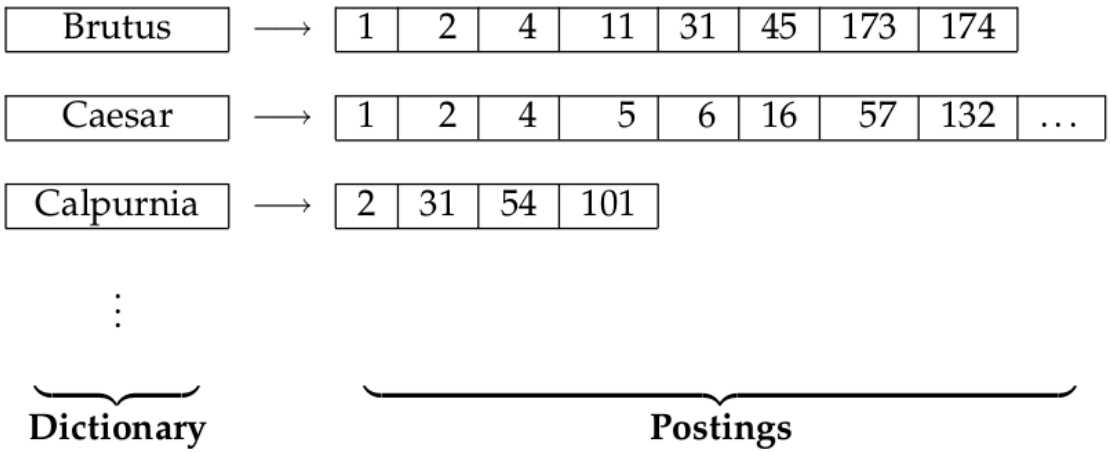
"**Information retrieval** (**IR**) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)."

Basic terms:

- collection, document, term
- information need, query, relevance
- effectiveness, precision, recall

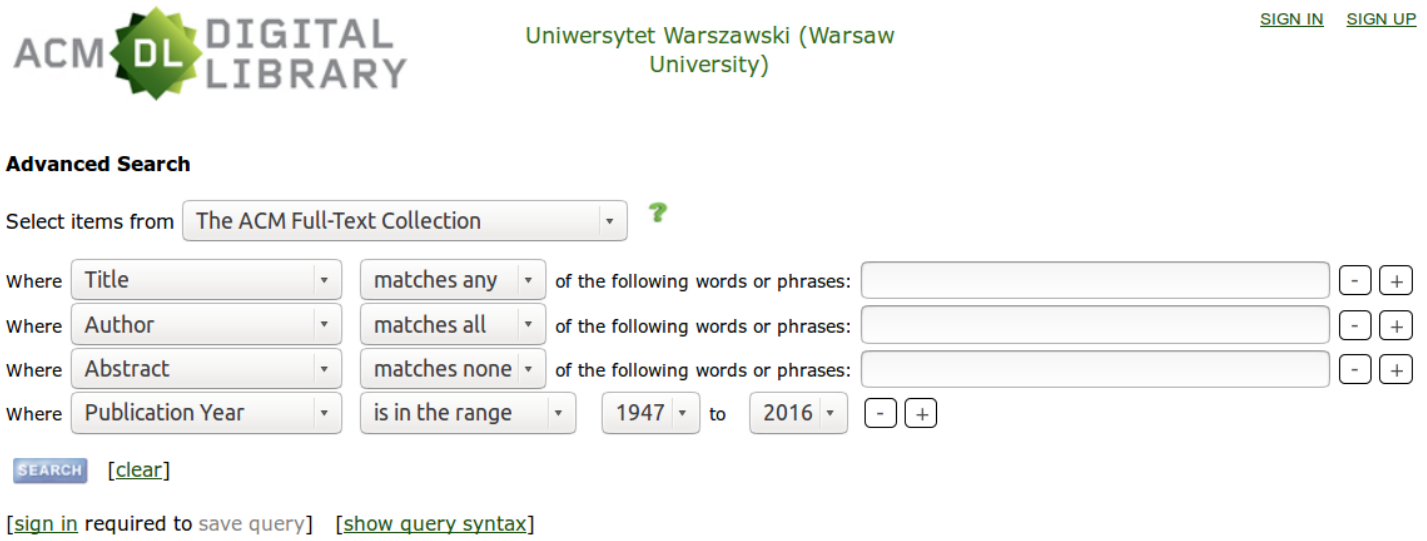
The whole presentation is based on: **Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008**, available at http://nlp.stanford.edu/IR-book/

Boolean retrieval model



- documents are **sets of words**
- query is in the form of a **Boolean expression of terms**, eg. "Brutus AND Caesar AND NOT Calpurnia"
- term-document incidence matrix vs. **inverted index**
- what if the query retrieves a lot of documents?

Parametric search



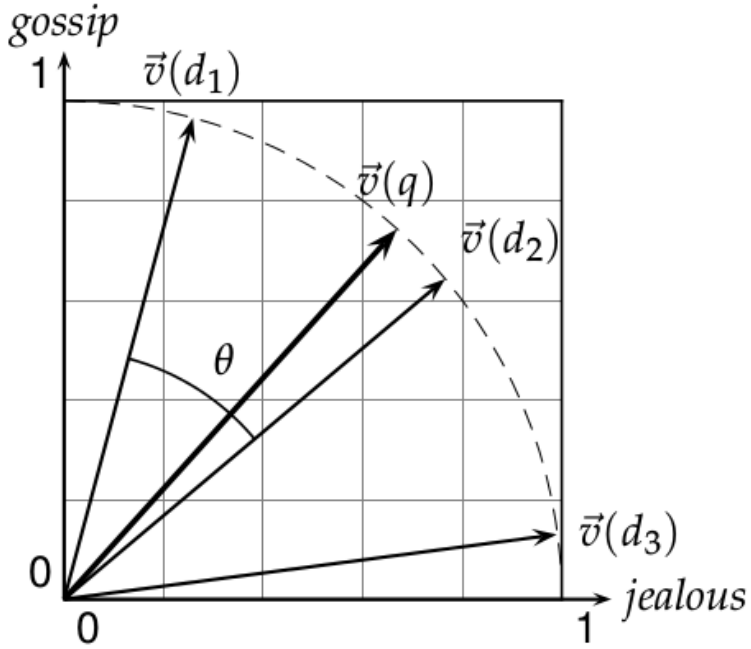
- score $= \sum_{i=1}^{n} g_i s_i$, where $g_i \in [0, 1]$ and $\sum_{i=1}^{n} g_i = 1$
- weights g_i are provided by the user or learned from the training set

Term weighting

- **term frequency**: $\mathrm{tf}_{t,d}$ = how many times the term appears in the document
- **document frequency**: df_t = how many documents in the collections contain the term
- **inverse document frequency**: $\mathrm{idf}_t = \log \frac{N}{\mathrm{df}_t}$
- **term weight**: $\text{tf-idf}_{t,d} = \mathrm{tf}_{t,d} \times \mathrm{idf}_t$

Vector space model

- dimentions $\sim$ (chosen) terms
- documents $\sim$ vectors of tf-idf
- query $\sim$ vector in the same space
- similarity between a document and a query $\sim$ vectors' cosine similarity
- similarity between documents $\sim$ vectors' cosine similarity



Tf-idf variants

Term frequency		Document frequency		Normalization	
n (natural)	$\mathrm{tf}_{t,d}$	n (no)	1	n (none)	1
l (logarithm)	$1+\log(\mathrm{tf}_{t,d})$	t (idf)	$\log \frac{N}{\mathrm{df}_t}$	c (cosine)	$\frac{1}{\sqrt{w_1^2+w_2^2+\ldots+w_M^2}}$
a (augmented)	$0.5+\frac{0.5\times \mathrm{tf}_{t,d}}{\max_t(\mathrm{tf}_{t,d})}$	p (prob idf)	$\max\{0, \log \frac{N-\mathrm{df}_t}{\mathrm{df}_t}\}$	u (pivoted unique)	$1/u$
b (boolean)	$\begin{cases} 1 & \text{if } \mathrm{tf}_{t,d}>0 \\ 0 & \text{otherwise} \end{cases}$			b (byte size)	$1/CharLength^{\alpha}, \alpha<1$
L (log ave)	$\frac{1+\log(\mathrm{tf}_{t,d})}{1+\log(\mathrm{ave}_{t\in d}(\mathrm{tf}_{t,d}))}$				

Evaluation

We need:

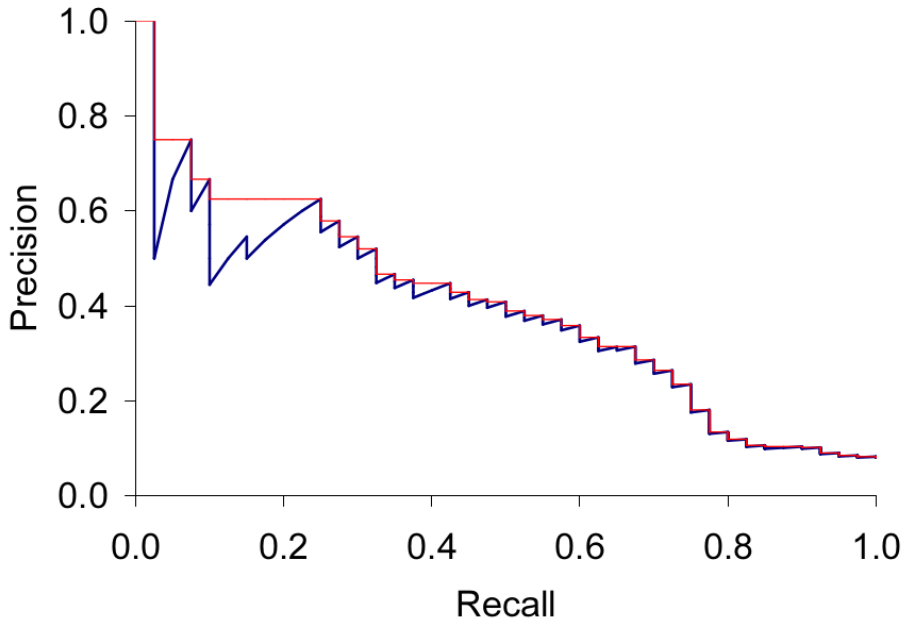
- a document collection
- a test suite of information needs, expressible as queries
- a set of relevance judgments, usually a binary assessment of either *relevant* or *nonrelevant* for each query-document pair

Classification-related measures

- $\text{precision} = \frac{\#(\text{relevant items retrieved})}{\#(\text{retrieved items})} = P(\text{relevant} \mid \text{retrieved})$
- $\text{recall} = \frac{\#(\text{relevant items retrieved})}{\#(\text{relevant items})} = P(\text{retrieved} \mid \text{relevant})$
- $F_1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$
- $\text{accuracy} = \frac{\#(\text{relevant items retrieved}) + \#(\text{nonrelevant items not retrieved})}{\#(\text{all items})}$

Ranked retrieval measures

- 11-point interpolated average precision
- Mean Average Precision (MAP)
- precision at k
- R-precision



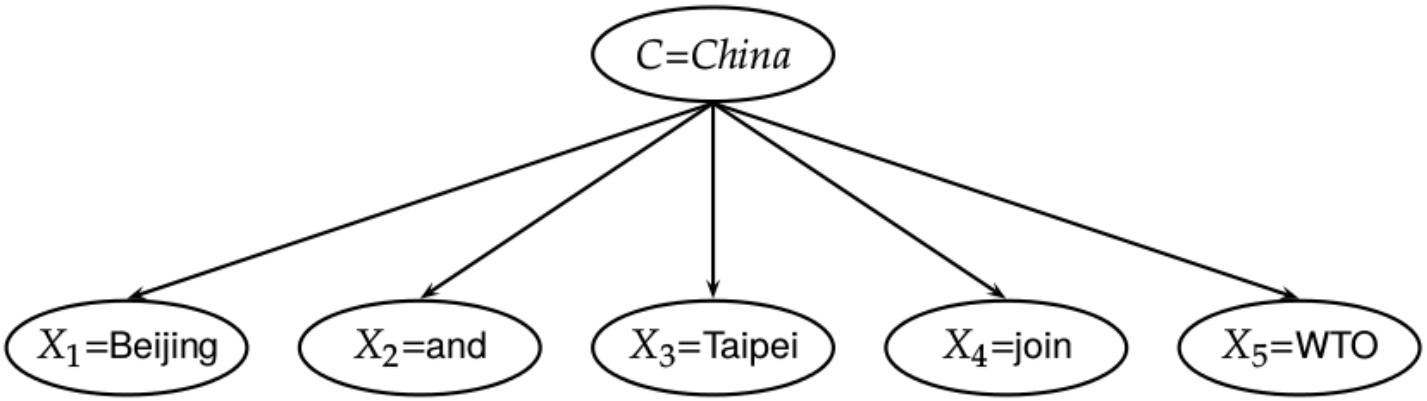
Evaluation: other issues

- How reliable are judges' assessments?
- How to know the set of relevant documents in a large collection?
- Marginal relevance: is a document still useful after the user has looked at certain other documents?
- A/B testing, clickthrough log analysis

Text classification

- assigning topics/categories to documents
- vertical search engines restrict searches to a particular topic
- rule-based vs ML-based
- the goal in text classification is high accuracy on new data

Multinomial Naïve Bayes



- $c_d = \arg\max_{c \in C} \; \mathrm{P}(c|d)$
- $\mathrm{P}(c|d) \propto \mathrm{P}(c) \prod_{1 \leqslant k \leqslant n_d} \mathrm{P}(t_k|c)$
- parameters estimation: MLE + add-one smoothing

Why is Bayes naïve?

- conditional independence assumption

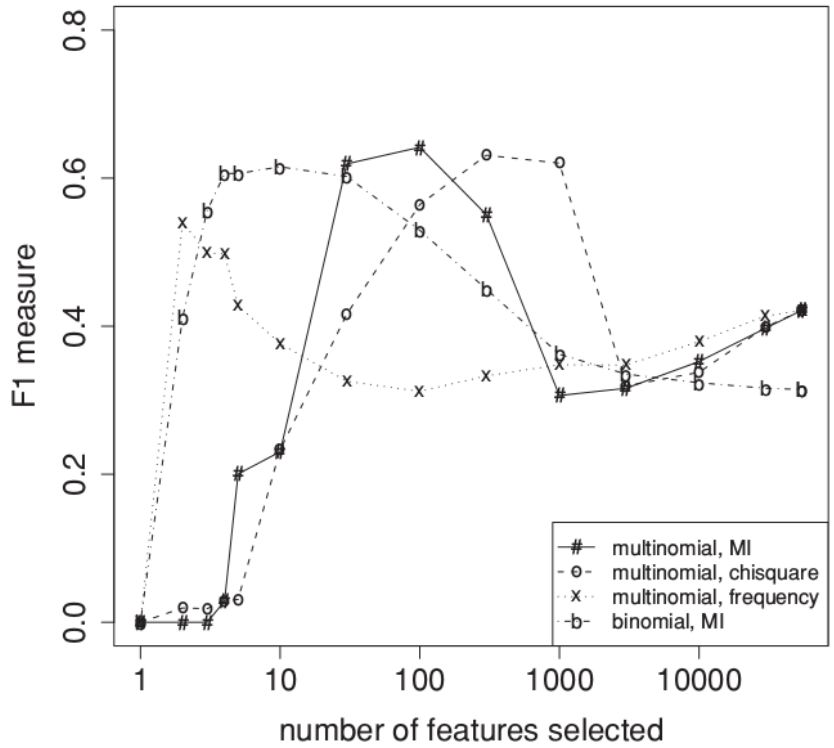
$$P(t_1, ..., t_{n_d}|c) = \prod_{1 \leqslant k \leqslant n_d} P(X_k = t_k|c)$$

- positional independence

$$P(X_{k_1} = t|c) = P(X_{k_2} = t|c)$$

Feature selection

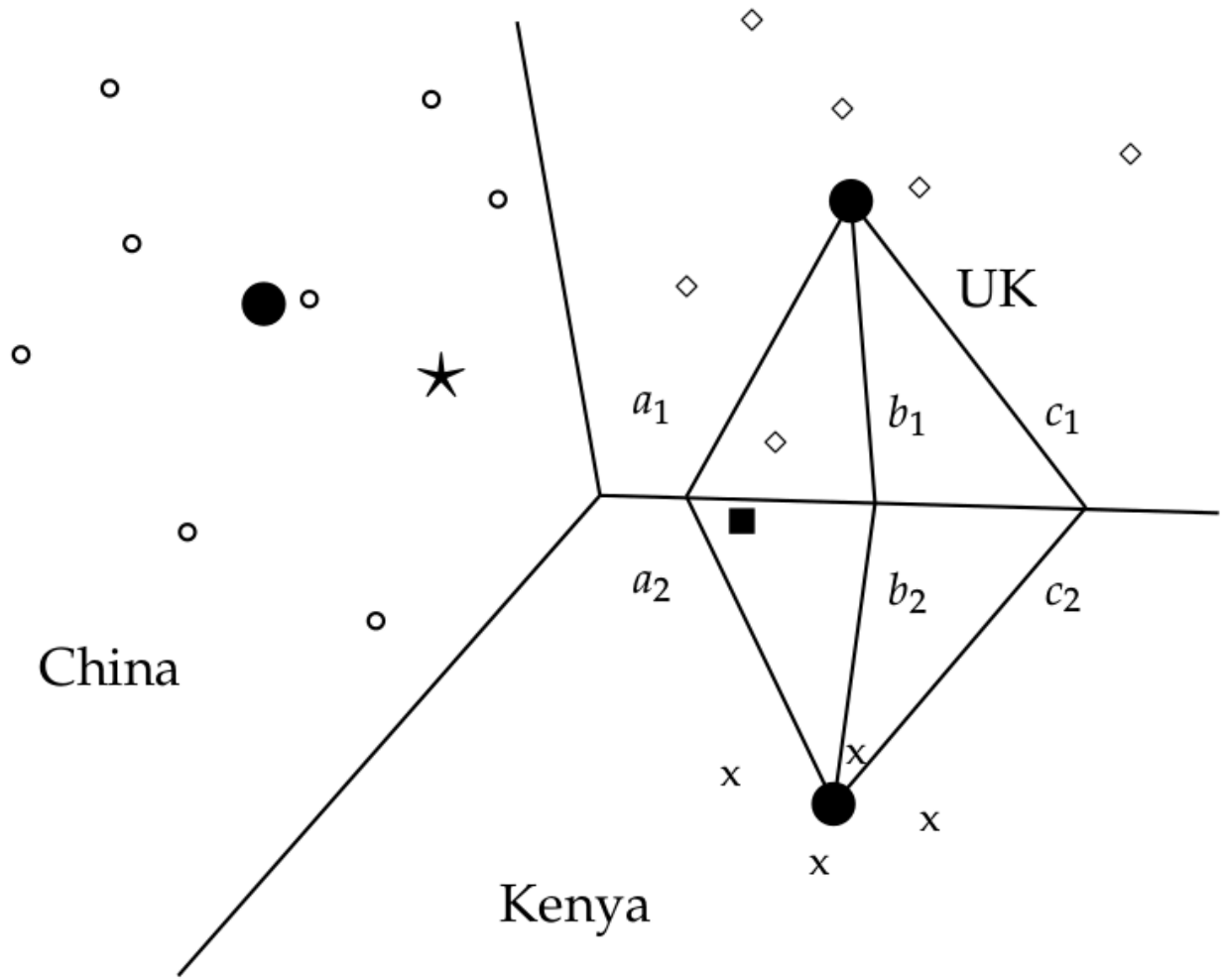
- decreases the vocabulary size
- eliminates noise features
- select k terms with the highest **utility measure**
  - mutual information
  - χ^2 test
  - frequency-based



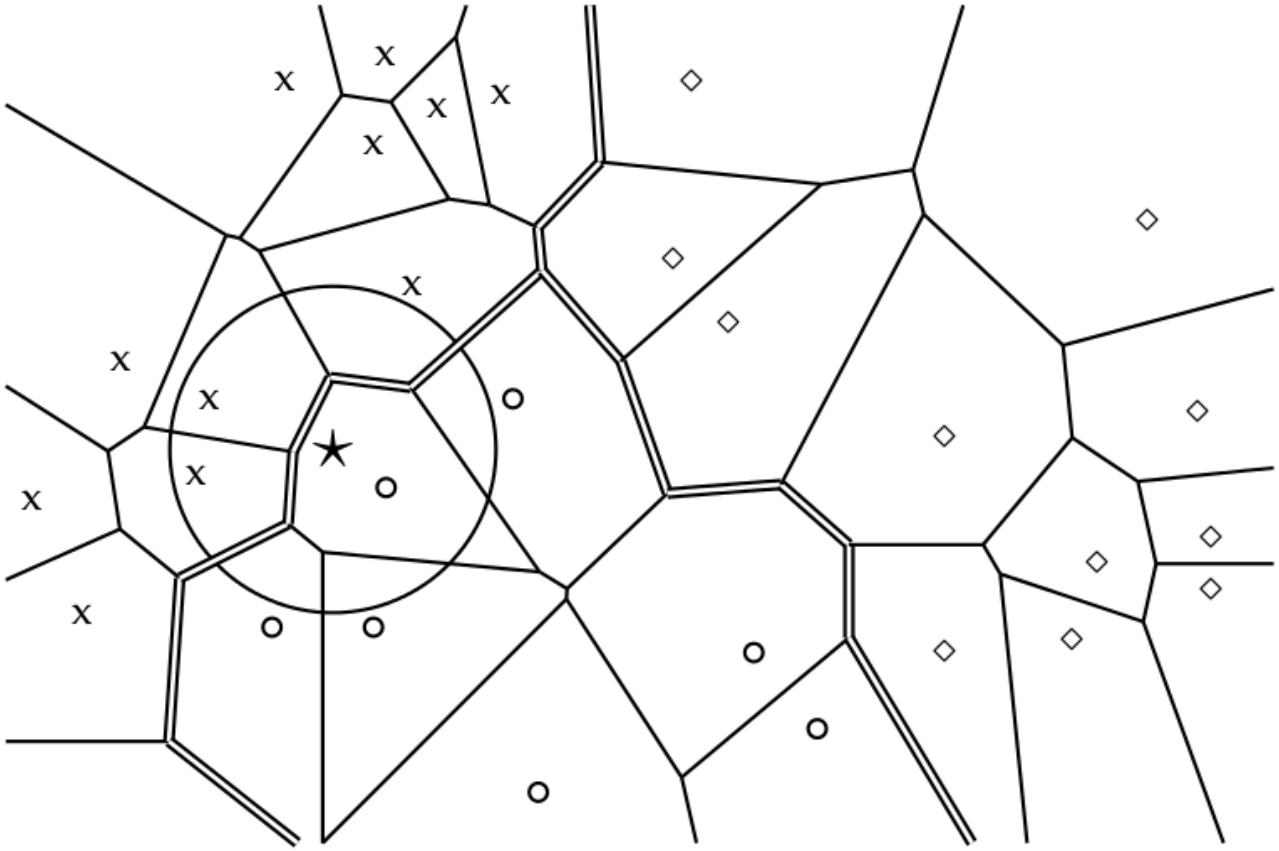
Vector space classification

- represent documents in **vector space model** as before
- now we can apply **various classifiers** to **numerical vectors of features**

Rocchio classification



kNN classification



Text classification evaluation

The usual stuff:

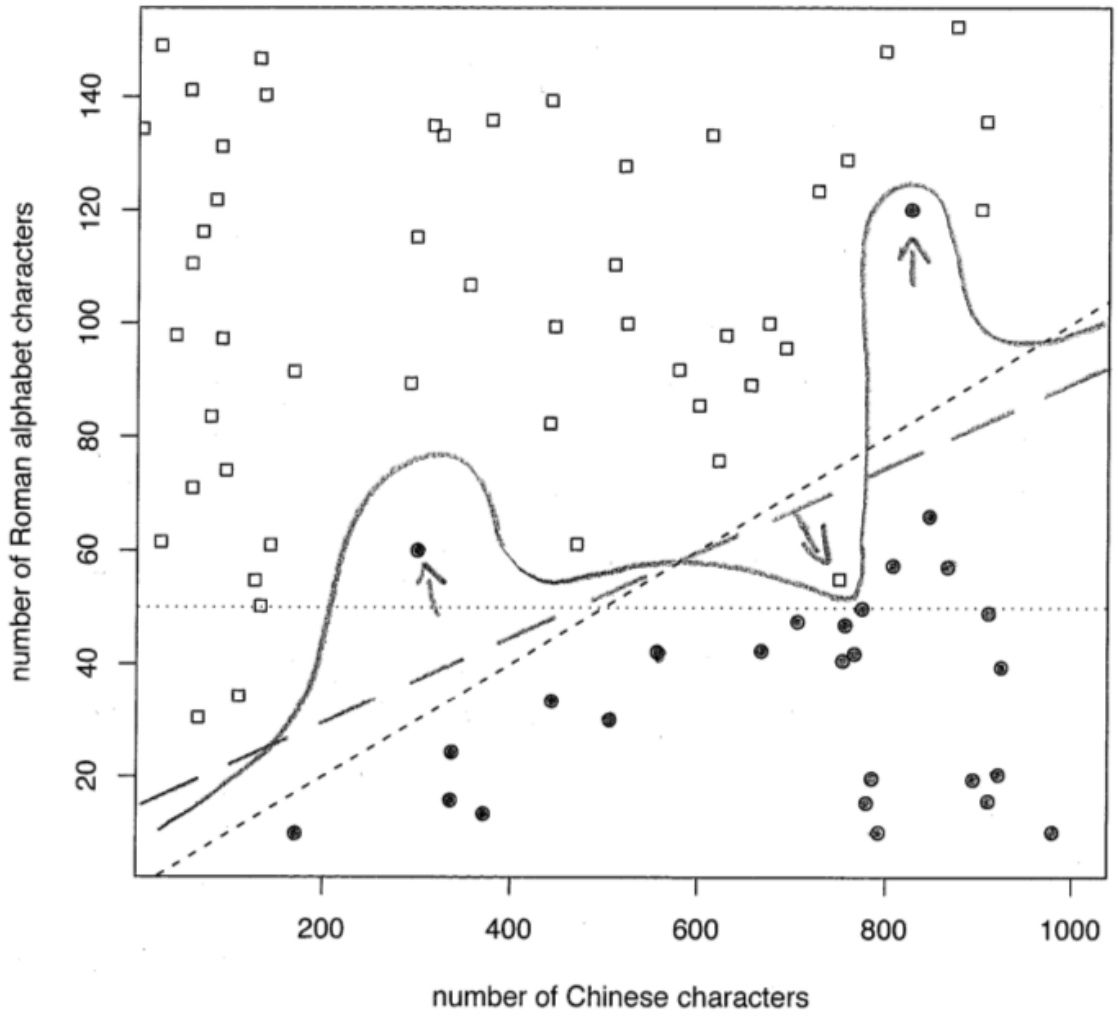
- precision, recall, F1
- microaveraging vs. macroaveraging

Bias-variance tradeoff

$$\begin{aligned}
\text{learning-error}(\Gamma) &= E_{\mathbb{D}}[\text{MSE}(\Gamma_{\mathbb{D}})] \\
&= E_{\mathbb{D}} E_d[\Gamma_{\mathbb{D}}(d) - P(c|d)]^2 \\
&= E_d[\text{bias}(\Gamma, d) + \text{variance}(\Gamma, d)]
\end{aligned}$$

$$\begin{aligned}
\text{bias}(\Gamma, d) &= [P(c|d) - E_{\mathbb{D}}\Gamma_{\mathbb{D}}(d)]^2 \\
\text{variance}(\Gamma, d) &= E_{\mathbb{D}}[\Gamma_{\mathbb{D}}(d) - E_{\mathbb{D}}\Gamma_{\mathbb{D}}(d)]^2
\end{aligned}$$

Bias-variance tradeoff



Thank you!

Dominika Tkaczyk
d.tkaczyk@icm.edu.pl