Query expansion based on linguistic evidence

Alexei Sokirko, Evgeniy Soloviev, Yandex

Overview

- Introduction: search engine linguistics, ‘synonymy’ relation, query terms
- The overall design of query expansion, general features
- Morphological inflection and derivation
- Transliteration and acronyms
- Machine learning in query expansion

Query expansions: the basic idea

Query expansion is the process of reformulating a search engine query to enhance retrieval performance, for example:

[buy cars]: cars -> car
[nato]: nato -> *North Atlantic Treaty Organization*

Why do we need query expansions?

- The larger topic variety in Internet, the more word senses in queries differ.
- The more people use Internet, the less their average educational level and language ability are, the more inaccurate queries are.
- Users do not realize the amount of ambiguity they put into queries, the disambiguation should be done by search engines.

Query or single terms?

- What should be expanded? The whole query or single terms?
- The best solution: expand single terms in local and global contexts.

Search engine linguistics

- User- and query-oriented linguistics
- No need to model real-world objects, informational objects (web-sites, software, reviews, lyrics) can be achieved directly by search engines
- Search engine as an AI agent

Synonymy

- Query term S refers to objects $O=O_1,O_2,\dots O_k$ objects with some distribution A:

  $P(O_k|S)=A_k$.
- If we replace term S with a new term N, then the distribution B ($P(O_k|N)=B_k$) should be as close as possible to A.
- In general, synonymy is the reference distributions similarity.

Я

Query terms

- Query terms could be one word expressions or collocations, for example “Russia” is one-word term, but “The United States of America” is a multiword term.
- Query terms always refer to objects of the same type (the object might be unique), and these objects constitute our naïve taxonomy.

Query term is a fuzzy notion

- “What is Russia?” 70% people could answer;
  “What is France?” 60% people could answer;
  “What is decision tree?” 0.0001% people could answer.
- Terms depend on the language or region.
- Query terms should occur in query logs as stand-alone queries (ad hoc restriction)

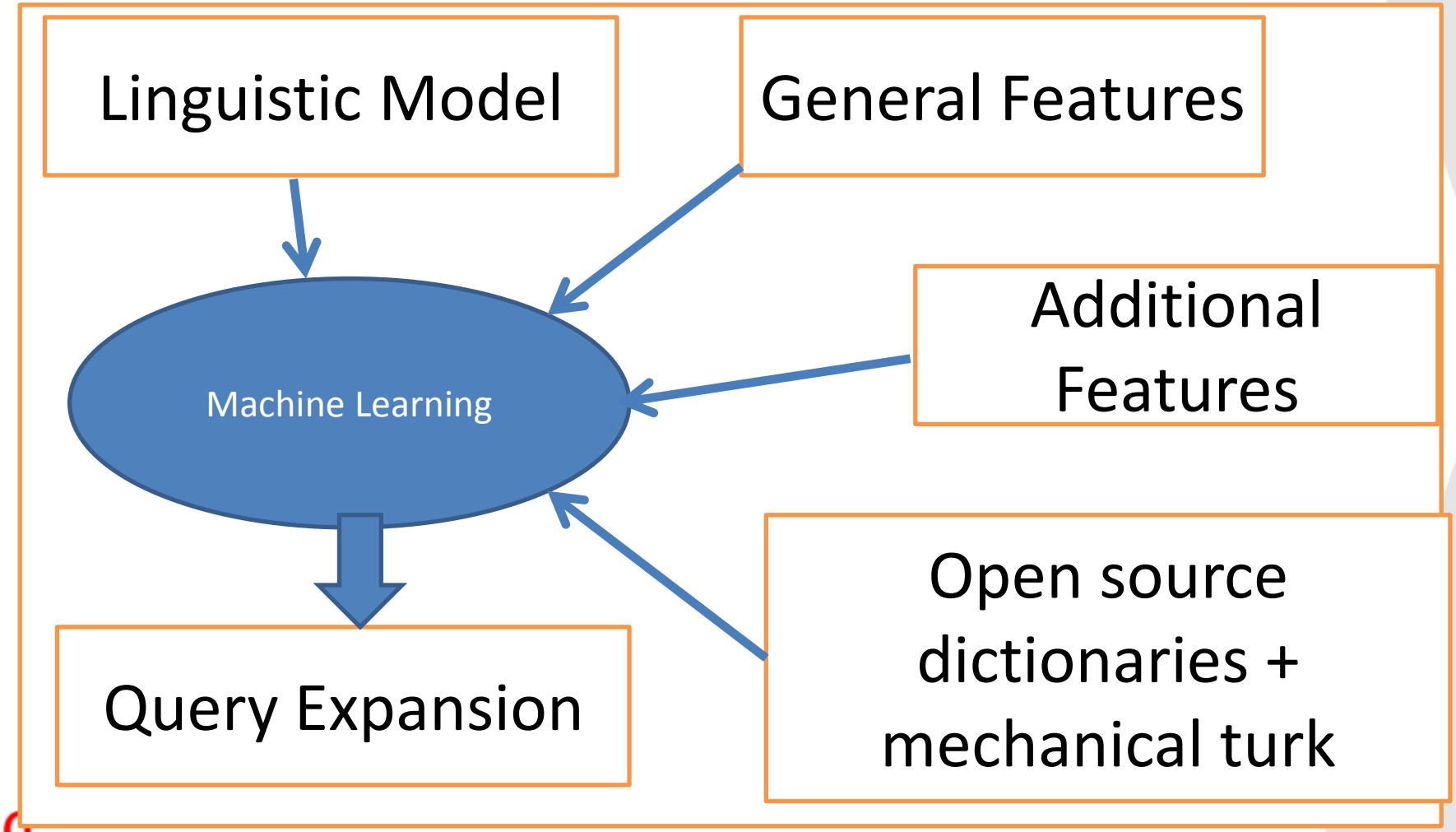
Popular classes of synonymy

- Morphological inflection relation (boy->boys, want->wanting)
- Morphological derivation relation (lemma->lemmatize, lemmatize->lemmatization)
- Transliteration (Bosch->бош, Yandex->Яндекс)
- Acronyms (United States of America -> USA, Russian Federation -> RF)
- Orthographic variants (dogend->dog-end, zeros->zeroes, volcanos->volcanoes)
- Common near-synonyms (error->mistake, mobile phone -> cell phone)

Overall design

- One system for all classes? For each word? For each class?
- Our solution is to supply each class with a separate algorithm of expansion.

One algorithm



Evaluation (3 metrics)

Estimate the dictionaries:

- No context, therefore one could almost always invent a context where the particular pair could be synonymous;
- Estimation of the similarity measure demands high expertise in various domains;
- Useful only for coarse-grained estimation:
  <ericsson, эриссон> is bad
  <ericsson, эриксон> is good

Я

Metric 2: Estimate a synonym pair for each query

- This assessment could be done almost definitive, it is more simple and precise;
- Assessor evaluation data show reference distribution
- Example:

[AAUP Frankfurt Book Fair] (AAUP -> Association of American University Presses)

[AAUP censure List] (AAUP -> American Association of University Professors)

Я

Metric 3: search engine results

- This metric measures the ultimate impact of synonym pairs on ranking of relevant documents.
- Industrial search engines use synonym pairs implicitly, therefore the impact is very hard to estimate
- The second metric (judge expansion in query contexts) is the most important.

General Features

- **DocFeature**: how often S1 and S2 occur on the same web-page or on the same web-site;
- **LinkFeature**: how often S1 and S2 occur in anchor texts of the links that point to the same web-site;
- **DocLinkFeature**: how often an anchor text contains S1 while the target website contains S2;
- **UserSessionFeature**: how often a user replaces S1 to S2 in a search query during one search session;
- **ClicksFeature**: how often a user clicks on a web-page that contains S1 while the search query contains S2;
- **ContextFeature**: how representative are the common contexts (of web-pages or queries) of S1 and S2.

DocFeature

- How often S1 and S2 occur on the same web-page or on the same web-site;
- Distance between S1 and S2 is not relevant;
- Document weight or site weight could be judged;
- Spam filtering is absolutely necessary in order to avoid deviations.

LinkFeature

- How often S1 and S2 occur in anchor texts of the links that point to the same web-site;
- The length of anchor text is relevant;
- The weight of the source host could be estimated

UserSessionFeature

- How often a user replaces S1 to S2 in a search query during one search session;
- Search sessions are not simple to determine, that’s why the distance (in seconds) between queries could help a lot;
- The order of word replacement is important.

ClicksFeature

- How often a user clicks on a web-page that contains S1 while the search query contains S2;
- The position of the clicked link is relevant: the further, the more important click is. The search result pagination should be taken into consideration.
- User makes choice considering only document snippets.

ContextFeature

- How representative the common contexts (of web-pages or queries) of S1 and S2 are;
- The quality and the frequency of common contexts should be taken into consideration.
- The number negative contexts (for S1, but not for S2 or contrariwise)

Morphological inflection

Flexia Models:

- *monitor* -> *monitor*(N,sg), *monitor-s*(N,pl)
  FlexiaModel1 = -, -s
  Freq(FlexiaModel2) = 72500

- *use* -> *us-e*(V,inf), *us-es*(V,3), *us-ing*(V,ger), *us-ed*(V,pp)
  FlexiaModel2 = -e, -es, -ing, -ed
  Freq(FlexiaModel2) = 745

Productive flexia models

- The kernel lexicon is not productive, the kernel flexia models are obsolete and therefore should be hand-made.
- There are obsolete flexia models, that still can be found in Internet (the language of the 19th century), or there are new flexia models, that are yet not enough popular (padonkaff’z language).

Я

Additional Features for inflection

- SuffixFeature: measures the similarity between word endings (*memorize* is a verb, *memorization* is a noun)
- TaggerFeature: uses a part of speech tagger trained on some corpora, estimates all contexts of the input word, deduces the most probable tag for the input word
- ProperFeature: measures the number of times the input word was uppercased

Evaluation (new word inflection, Metric 2)

Precision ≈ 92%

Recall ≈ 96%

F-Measure ≈ 93,5%

Promising directions: detecting language adoptions, new suffix models, new ML methods

Morphological derivation

- The linguistic model consists of the same suffix transformation(=flexia models), like:
  memorize -> memorization: -e,-ation
- There are enough false positives, like sense -> sensibility.
- Generalize models in order to unify the following transformations:
  memorize-> memorization : e->ation
  induce -> induction: e -> tion
  publish -> publication sh->cation

Я

Sense deviation, term boundaries

- F-measure for the dictionary is around 87% (Metric 1).
- F-measure for query expanding by derivation pairs is 65% (Metric 2).

[Australian population] (*Australian*=>*Australia* +)

[Australian gold] (*Australian* => *Australia* -)

[milk diet] (diet => dietary +)

[The Diet of the German Empire] (diet=>dietary -)
(a kind of Parliament)

Я

Transliteration

Transliteration

What's it about?

- To have a high quality search we should take into account

photo фото фωто

- Russian language is not an exception – it uses Cyrillics while Latin is prevalent
- Transliteration is a systematic way of transforming words from one writing system to another and it is very important synonymous type

Transliteration

What are the main transliteration cases?

- Proper names:
  Albert Einstein ↔ Альберт Эйнштейн
  Яндекс ↔ Yandex
- Loanwords:
  computer ↔ компьютер
  перестройка ↔ perestroyka
- URLs, logins and other ids that are in Latin due to system restrictions

Я

Transliteration

How is the transliteration being performed?

- *Transliteration by dictionary* (**offline**) – uses pre-generated dictionary, the correspondences are refined in a very precise way
- *“On-the-fly” transliteration* (**online**) – usually has dubious impact on search results due to lack of required statistics at runtime

Transliteration

What are the sources for transliteration synonyms?

- Sources of data containing every Yandex query and all the possible answers

Transliteration

But how to use such an enormous and unstructured data?

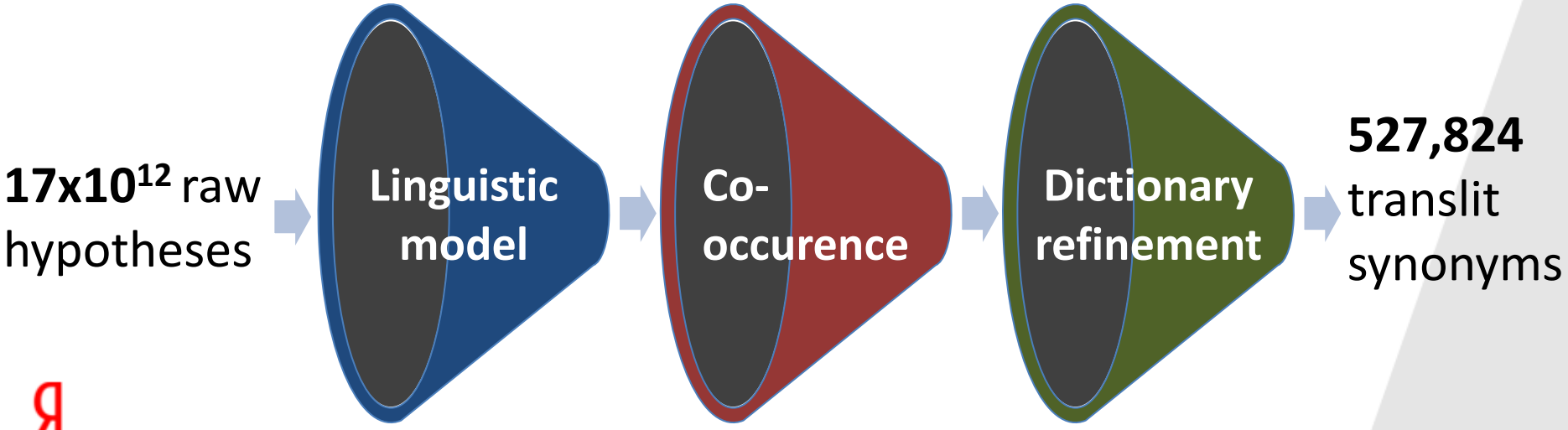
- There’re about 12 millions of known Russian and English words
- About 72×10^{12} possible one-word synonym hypotheses
- About 17×10^{12} pairs from different writing systems

Я

Transliteration

But how to mine the transliteration synonyms?

The main idea: iteratively reduce the number of hypotheses by keeping only those that have any chance to prove their utility



Transliteration

“Linguistic model”

- Our aim here is to mine transliteration type synonyms only
- Linguistic transliteration model is a formal description of what transliteration is
- Using the model we could greatly decrease the number of hypotheses

Transliteration

Rule-based transcription model (M1)

- uses known rules and standards for cross-lingual transcription
- represented as several transition tables, one for each of the most popular languages (English, French, etc.)
- finds syllable-by-syllable transition, penalyzing for letters remaining after transition

Transliteration

Rule-based transcription model - example

a	↔	a
ai	↔	e
ai	↔	э
eau	↔	o
eu	↔	э
eu	↔	ë
es	↔	_
ville	↔	виль

Transliteration

Fuzzy language transcription model *by Yuri Zelenkov* (M2)

- learned on a big corpus of good transcriptions
- model is a probability distribution of possible transliterations given the original syllable pattern
- for each hypothesis pair calculates the probability of its “transliteness”

Transliteration

Fuzzy language transcription model – example

a.ch.aue	→	а (1.000)
a.ch.ay	→	а (0.833) ачае (0.167)
a.ch.aye	→	а (1.000)
a.ch.e	→	а (0.957) е (0.016) ей (0.014) э (0.006) о (0.005) эй (0.003) я (0.003)
a.ch.ea	→	а (0.778) о (0.111) ей (0.111)
a.ch.ee	→	а (1.000)
a.ch.ei	→	а (1.000)
a.ch.ey	→	а (0.500) ей (0.250) э (0.250)

Я

Transliteration

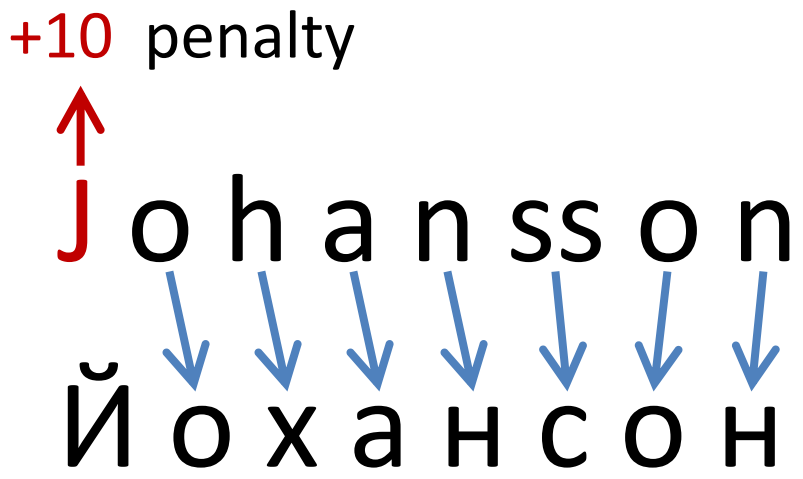
Rule-based transcription – application example

Is the pair “**Johansson → Йохансон**” a proper translit? Let’s look at the table:

J	↔	дж
o	↔	о
h	↔	г
h	↔	х
a	↔	а
a	↔	э
n	↔	н
ss	↔	с

Transliteration

Rule-based transcription – application example



Transliteration

Fuzzy transcription model - results

йохансон (6.446)	йогансон (5.745)	йоханссон (4.919)	иохансон (1.422)	джохансон (1.311)
иогансон (1.269)	иоханссон (1.085)	джоханссон (1.000)	ёхансон (0.427)	юхансон (0.387)
йохонсон (0.342)	югансон (0.341)	хансон (0.333)	гансон (0.298)	юханссон (0.292)
ханссон (0.255)	янсон (0.192)	джохэнсон (0.142)	йонсон (0.103)	йонссон (0.079)
хогенсон (0.068)	джансон (0.067)	жансон (0.066)	хэнсон (0.036)	йоханссен (0.027)

Я

Transliteration

Linguistic model - results

- number of hypotheses reduced to **59** millions transliterations
- **>90%** recall from each of models, but precision is still very low
- “transliteness” ratings from both models for further precision improvement

Transliteration

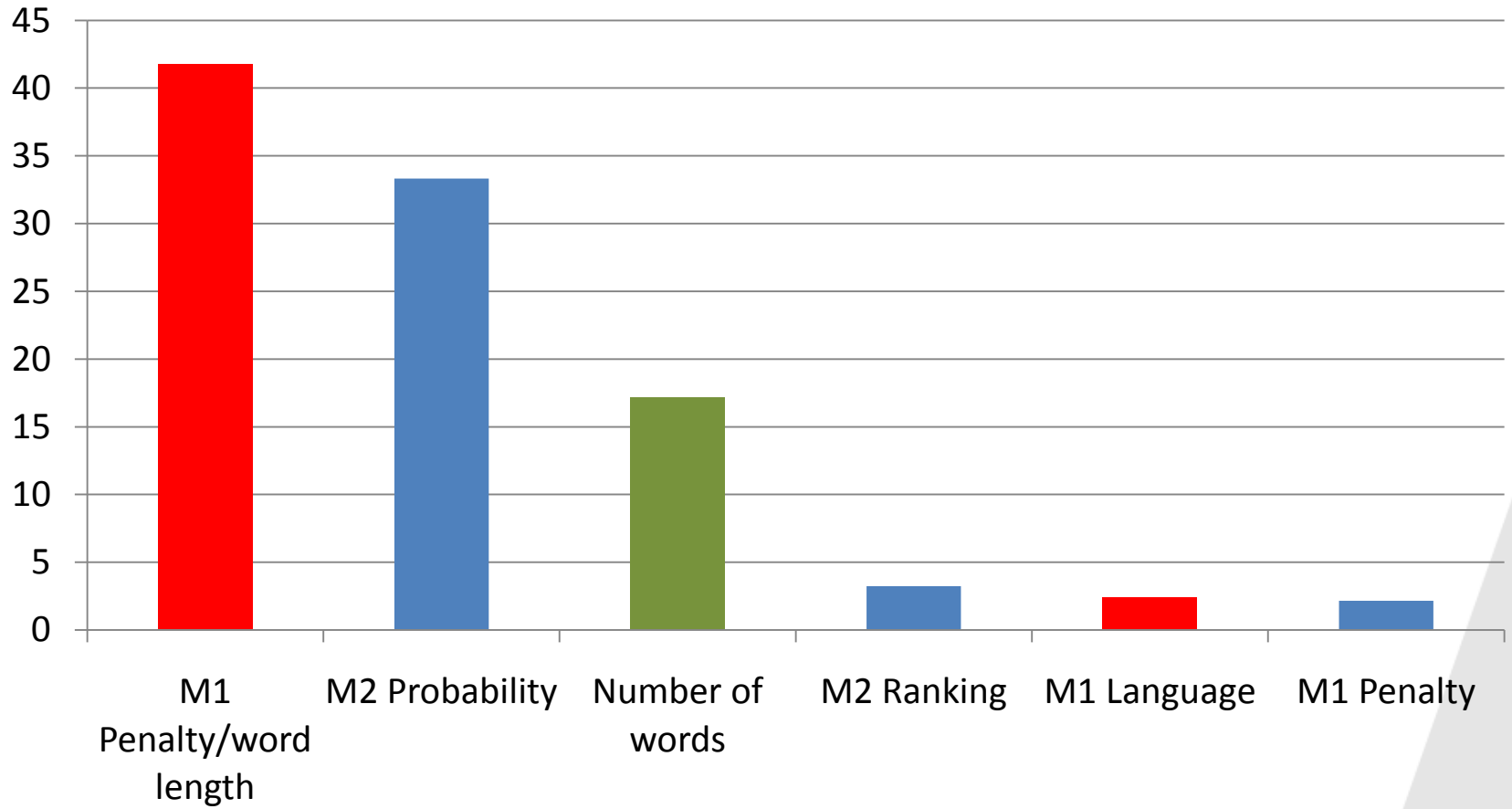
Linguistic model – ML refinement

- combine the ratings from 2 models using ML to reveal their full power
- refined 95,2% recall and 91% precision of “translitness”
- hypotheses count reduced to 1,8 millions
- rating of features’ importance:

Transliteration

Linguistic model – ML refinement

Filtering by language model – features importance



Transliteration

Good transliterations –
not necessary good synonyms

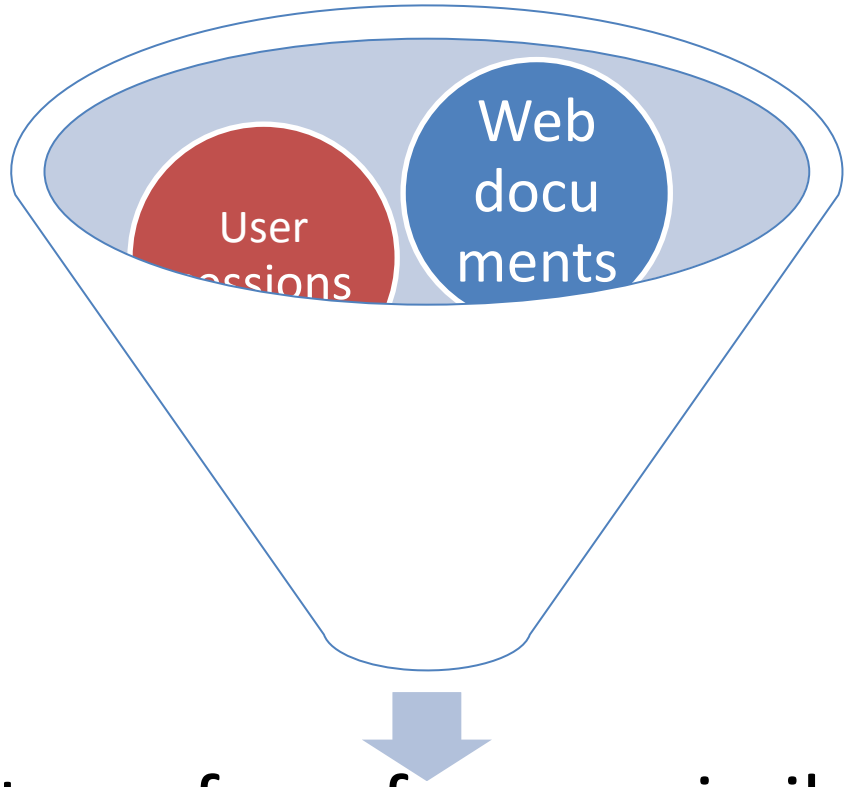
- possible change of lexical meaning:
  magazine → **магазин** (meaning “**shop**”)
- change of reference object (difficult to catch):
  respublica → **республика**
- just trashy transliterations

  tekst pesni

Я

Transliteration

Refining synonyms by co-occurrence statistics



Features for reference similarity measurement

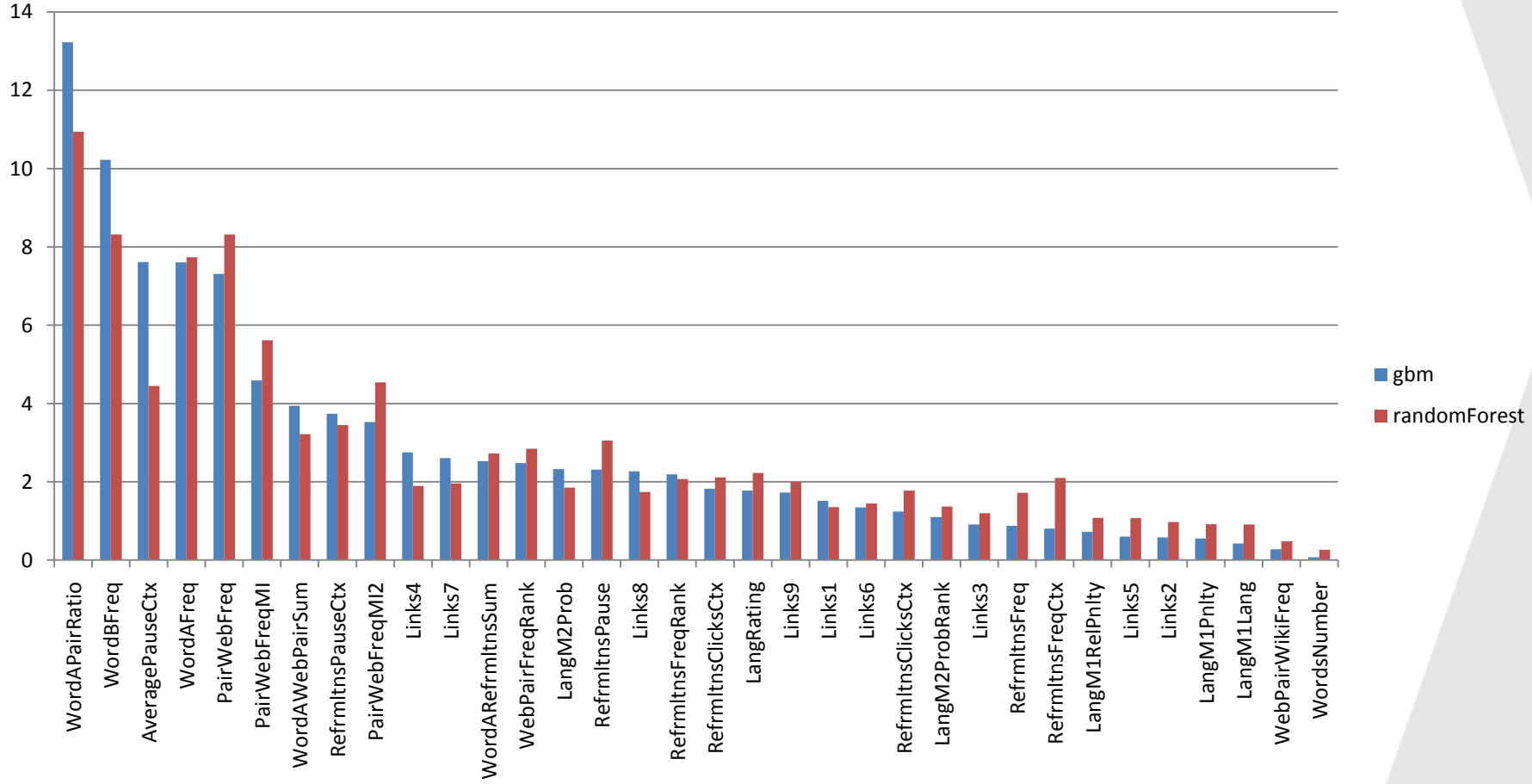
Transliteration

ML methods used for synonyms refinement

Model type	Train error	Test error	Annotations
gbm	0,22%	11,81%	distribution="adaboost" interaction.depth=4
randomForest	0,00%	13,38%	ntree=100
Logistic regression	25,42%	23,62%	
SVM	3,71%	13,38%	nu = 0.5,gamma = 1 radial nu-classification
Decision trees	17,21%	30,70%	

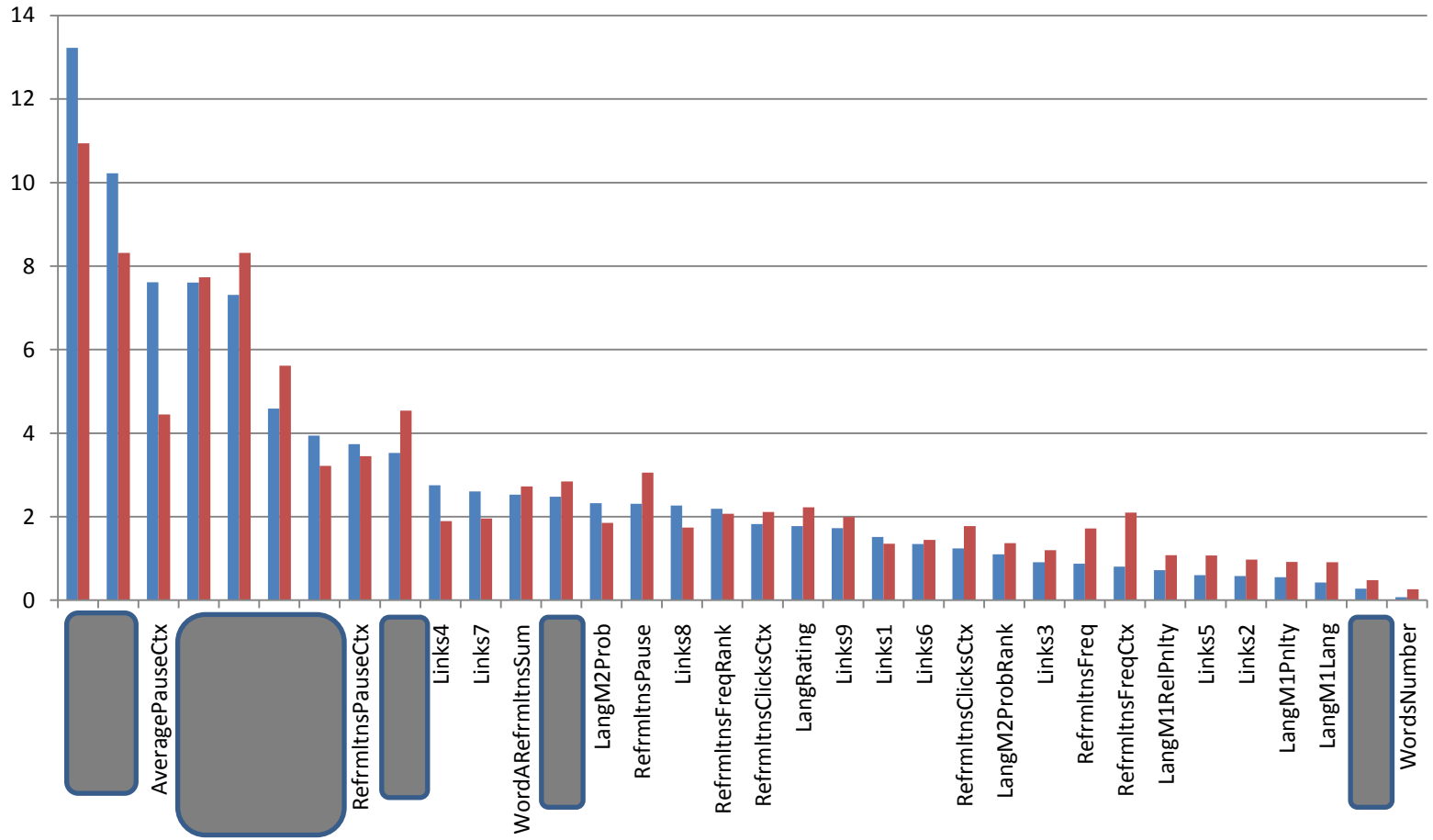
Transliteration

Features importance for synonyms refinement



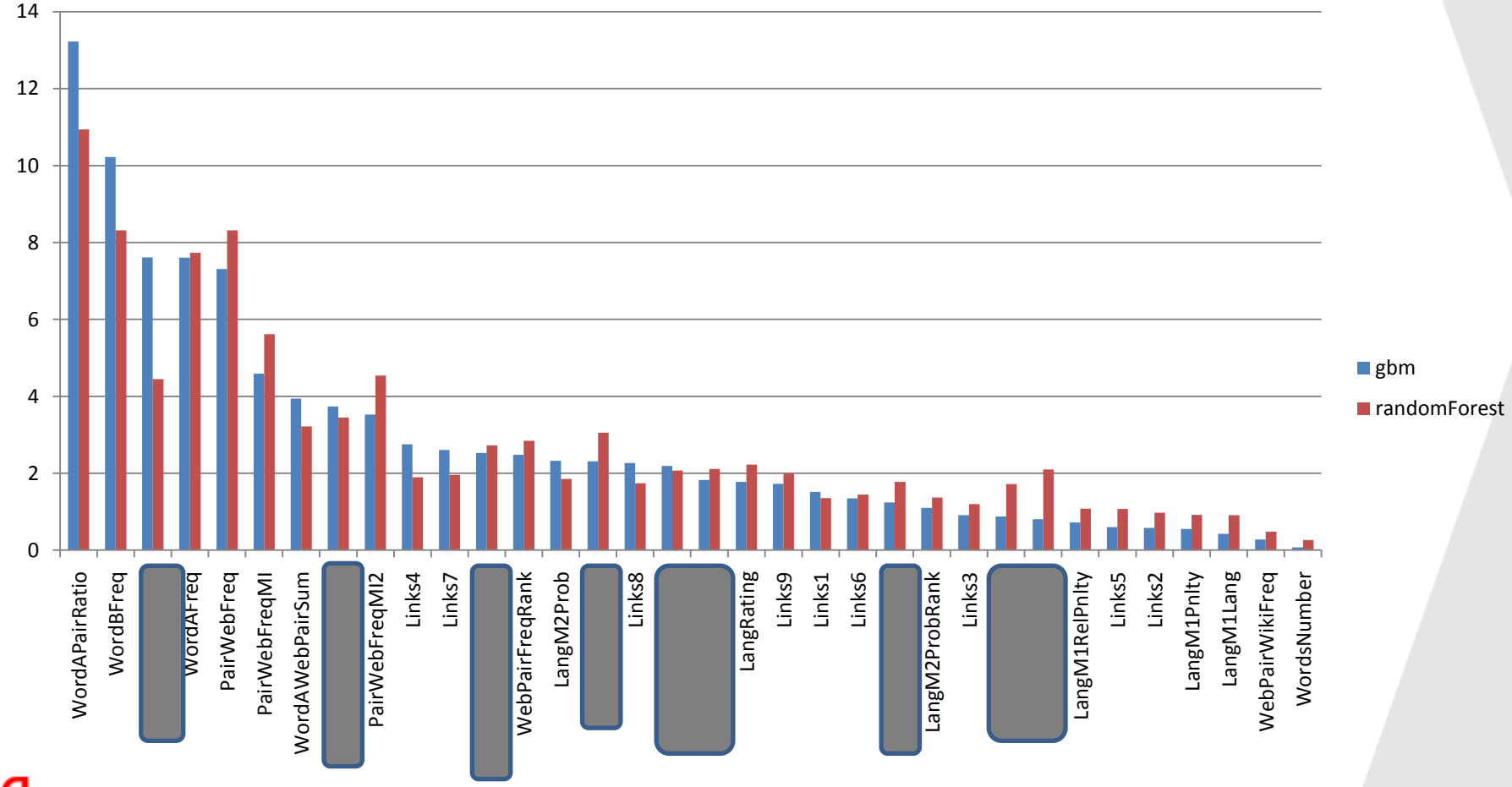
Transliteration

Features importance for synonyms refinement – Web statistics



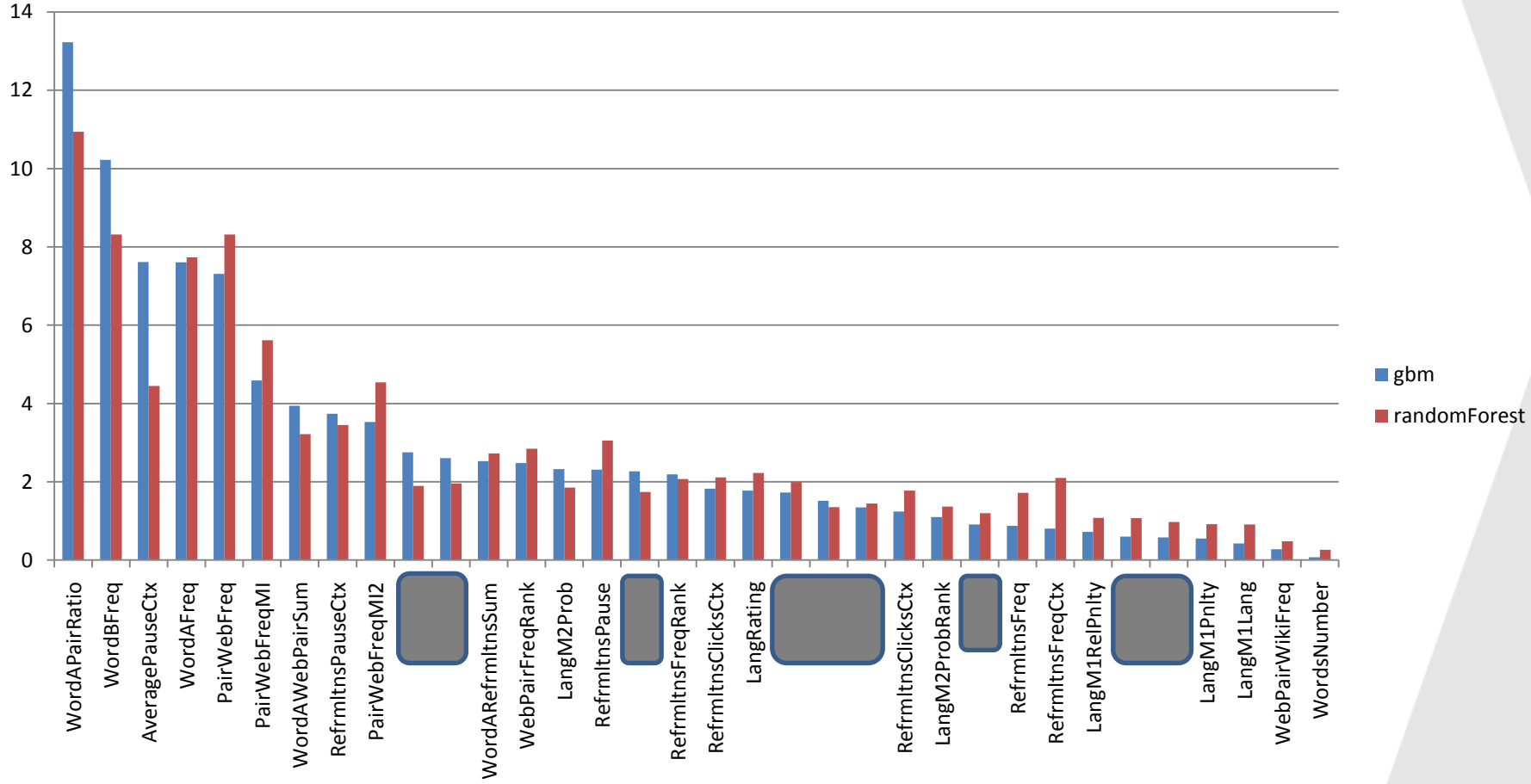
Transliteration

Features importance for synonyms refinement – query reformulations



Transliteration

Features importance for synonyms refinement – links statistics



Abbreviations

Abbreviations

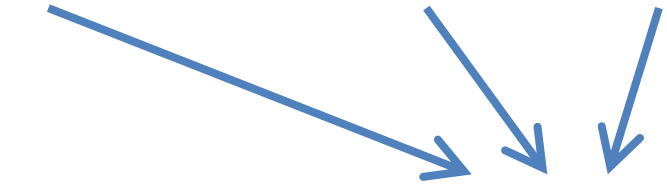
What is it?

- used to shorten well-established phrases and terms
- linguistic model looks quite simple – take the fist letter(s) from each word
- but that is not as easy…

Abbreviations

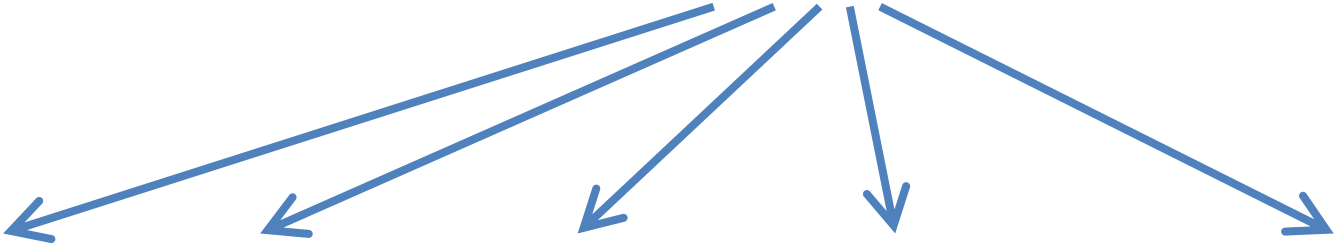
Abbreviations are formed quite simply…

Moscow **S**tate **U**niversity



MSU

RuSSIR



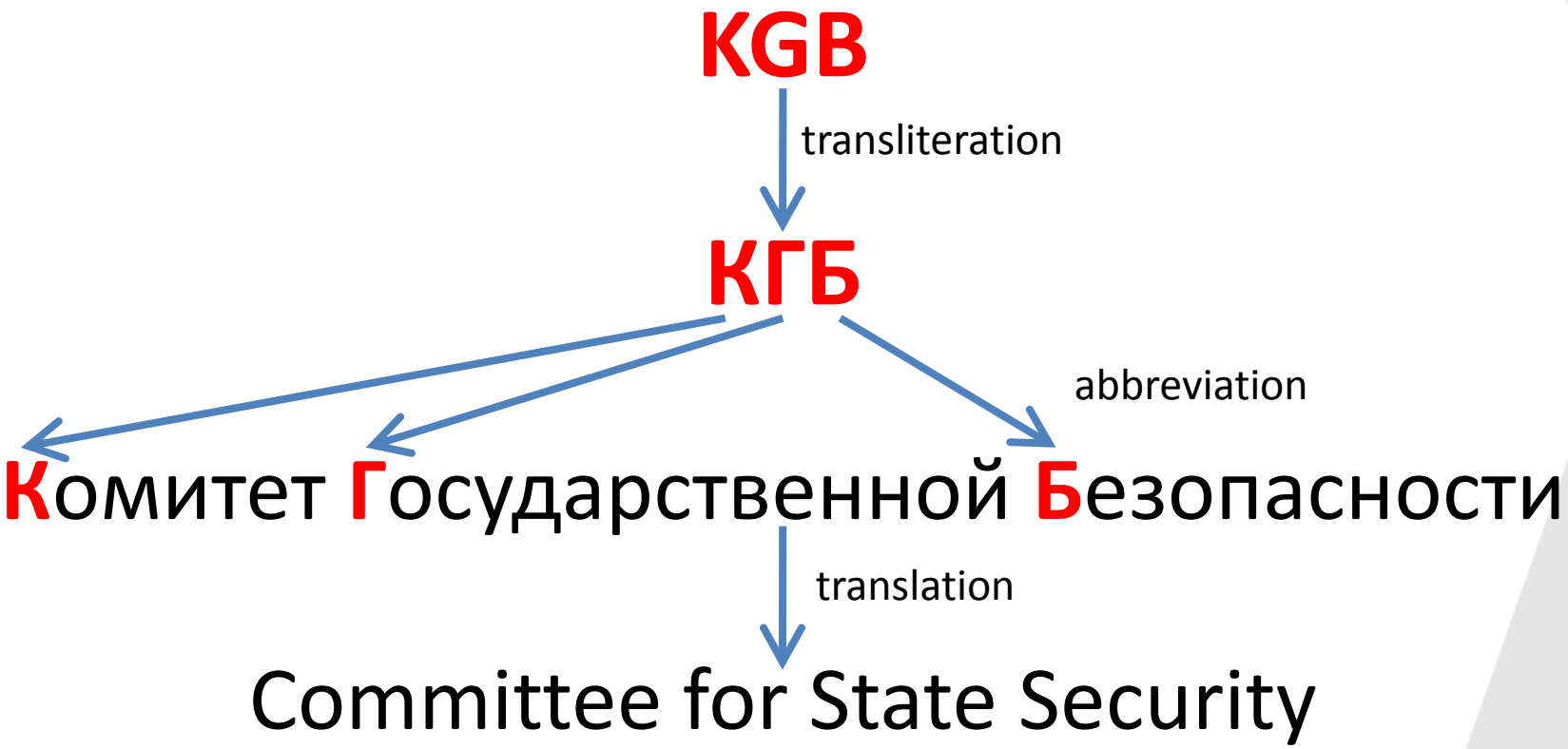
Russian **S**ummer **S**chool ~~**in**~~ **I**nformation **R**etrieval

Abbreviations

…or could be more complex

Abbreviations

…or could be more complex



Abbreviations

Not every phrase forms an abbreviation

- only small portion of all possible phrases forms abbreviations
- for example, let’s look at most frequent phrases forming “RuSSIR”:

Abbreviations

What may word “RuSSIR” stand for?

Russian Summer School in Information Retrieval	ruption of the serotonin system in immature rats
ru siteuri scrise in romana	rupa se sparge i radu
rue statement showing in respect	ruj si sklepy i restauracje
ru sodo sklypas irvint rajone	rung setzt sich im rahmen
rujce stan systemu i raportujce	run the shell scripts in the rc

Я

Abbreviations

What may the word “RuSSIR” stand for?

Russian Summer School in Information Retrieval	run the shell scripts in the rc
ru siteuri scrise in romana	ructuri sanitare situate in regiunile
rue statement showing in respect	running sun’s implementation of rmid
ru sodo sklypas irvint rajone	ructor's signature is required
rujce stan systemu i raportujce	ruffle straight style is reversible
ruption of the serotonin system in immature rats	rural support service is responsible
rupa se sparge i radu	run the same services in runlevel
ruj si sklepy i restauracje	ruzione secondaria superiore in relazione
rust s score is represented	rudman says she is really
rung setzt sich im rahmen	runescape special service include runescape

Abbreviations

Abbreviations homonymy

- easy case – non-homonymous (virtually) abbreviations:

IEEE (*I triple E*) - **I**nstitute of **E**lectrical and **E**lectronics **E**ngineers

MSU – **M**oscow **S**tate **U**niversity

Я

Abbreviations

Abbreviations homonymy

- tough case – abbreviations are ambiguous:

- to other words:

мэг -> Мэг Райан vs **м**оно**э**тилен**г**ликоль

- to other abbreviations:

CSS^(**c**ascading **s**tyle **s**heets) styles vs.

CSS^(**c**ontent **s**crambling **s**ystem) license

…and even **MSU** could be “**M**ordovian **S**tate **U**niversity” in Mordovia! ☺

Я

Abbreviations

How to resolve ambiguity?

- pre-collect context statistics of expansion:
  - context words frequencies – bigrams or bags of words
  - query semantics
  - any other context information showing significant correlation with expansion (i.e. user region etc.)

Questions