Feature selection for language-independent text forum summarization

Vladislav A. Grozin, Natalia F. Gusarova, Natalia V. Dobrenko

It’s about

- Relevance
- Simple textual features
- Social graph features
- Gradient boosting models (random forest-like)
- Bootstrap
- Normalized cumulative gain metric (precision/recall alternative)

Using search engine

Using search engine

“Emacs vs vim”

What are the pros and cons of Vim and Emacs? - Unix ...
unix.stackexchange.com/.../what-are-the-pros-and-cons-of-**vim**-and-**ema**...
Turning the tables, I have observed **Vim** taking noticeably longer to load than **Emacs** (**vim** -u /dev/null **vs**. **emacs** -q). Admittedly this was on a weird platform ...

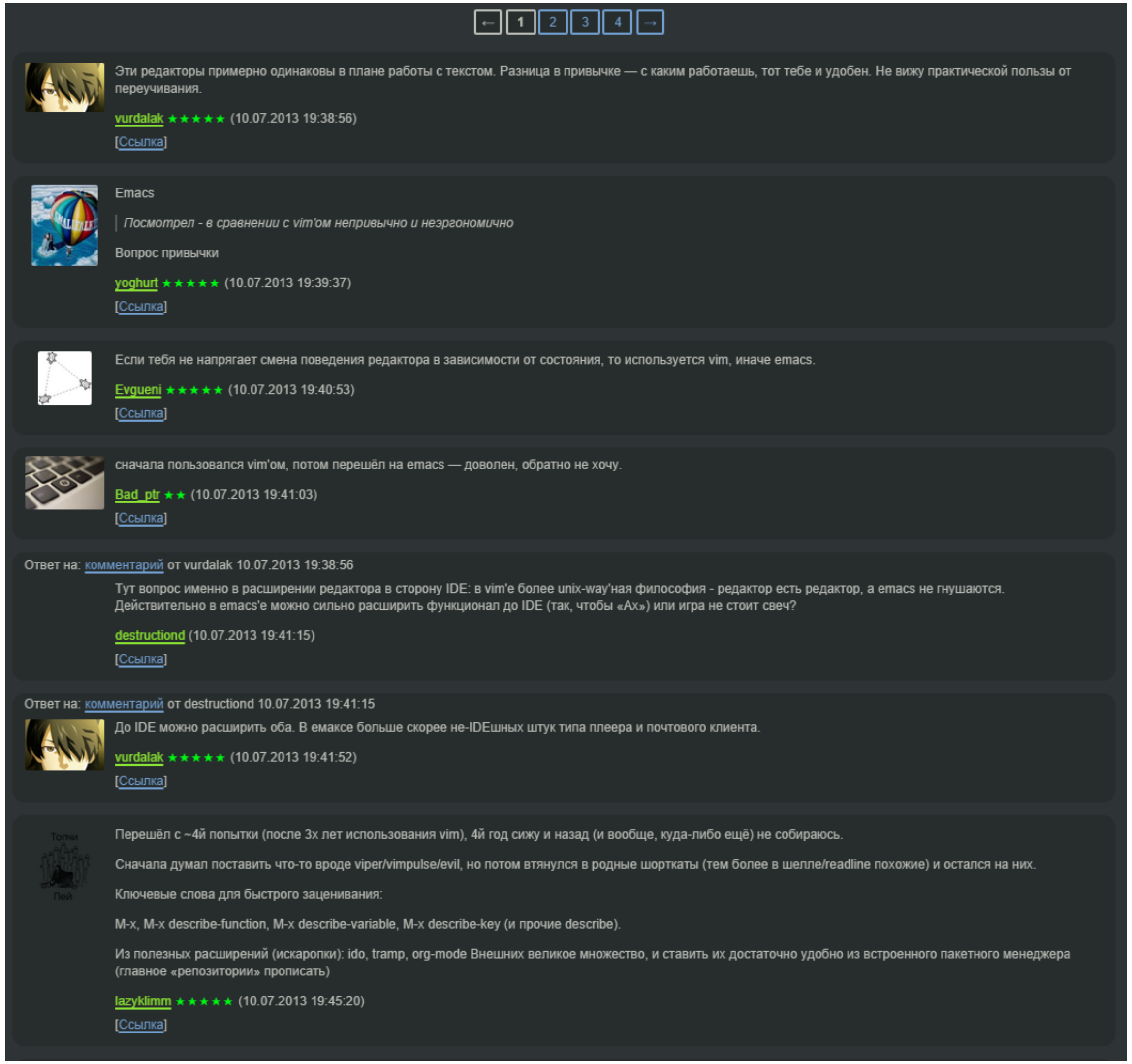
Which is better, Vim or Emacs? Why? - Quora
https://www.quora.com/Text-Editors/Which-is-better-**Vim-or-Emacs**-Why
I feel as if I'm uniquely placed to answer this one because I've been using both for about 25 years now. (15 with almost full time **Vim** and 10 **emacs**). First o...

Editor war - Wikipedia, the free encyclopedia
https://en.wikipedia.org/wiki/Editor_war
A **vim**-inspired **Emacs** package (undo-tree) provides a user interface to the tree. Memory usage ... Ability to emulate vi and **vim** (using Evil, Viper **or Vimpulse**).
Differences between Emacs and vi - Humor - Today - See also

Differences between Emacs and Vim - Stack Overflow
stackoverflow.com/questions/.../differences-between-**emacs**-and-**vim**
Without getting into a religious argument about why one is better than the other ... (the text below is my opinion, it should not be taken as fact **or** an insult) ..... I'm a ...

emacs или vim - Talks - Форум - Linux.org.ru
www.linux.org.ru/forum/talks/9349101 Translate this page
Jul 10, 2013 - 51 posts - 24 authors
сначала пользовался **vim**'ом, потом перешёл на **emacs** — доволен, обратно не хочу. .... https://www.youtube.com/watch?v=EQAd41VAXWo.
You visited this page on 9/30/15.

Using search engine



Problems

- Informal forum language
- Posts are not “self-enclosed” (unlike generic web-pages)
  - If web-page has some question, it usually has answer too
  - Forum threads are not the case

  => thread may be relevant, but posts in it are not useful

Solution

System that fetches “good” post from forum threads using user *query*.

It should be:

- Robust to the lexical and grammatical errors
- Language-independent

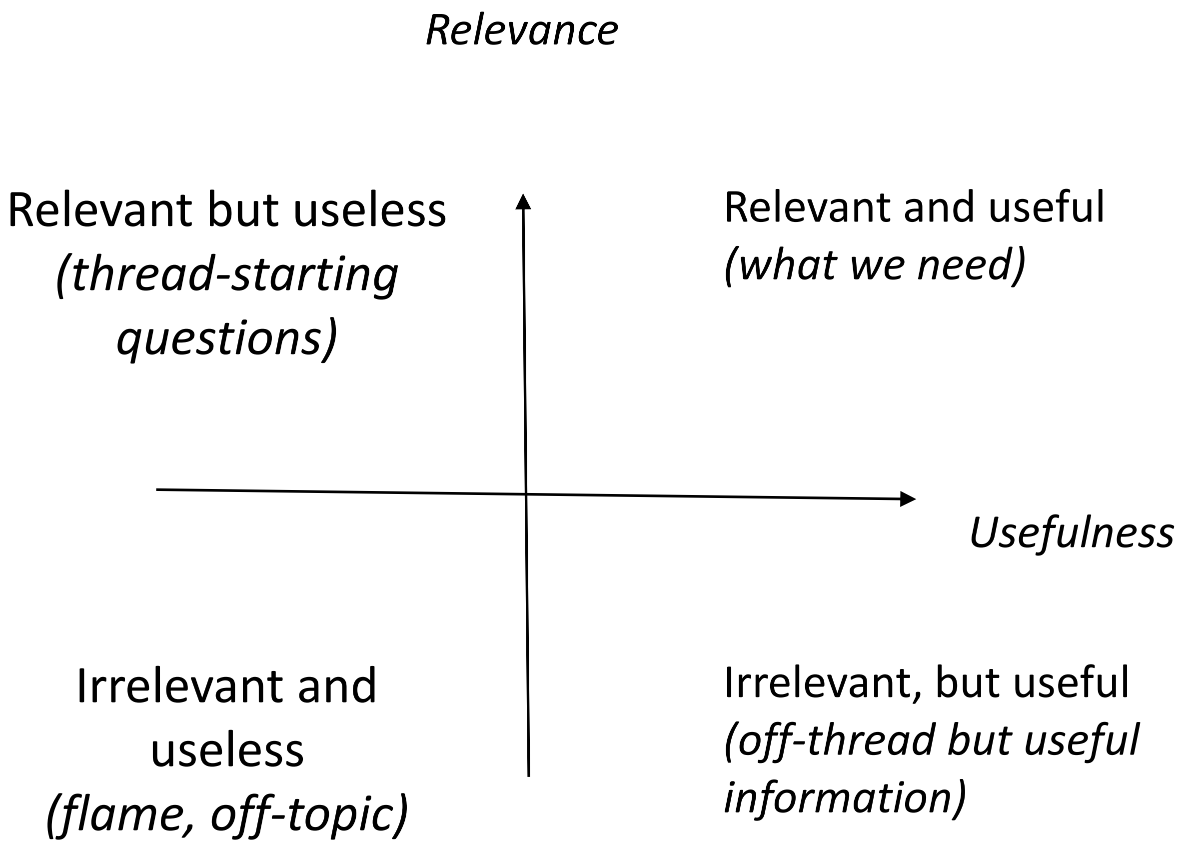
Formal definitions

What is “good” post?

Relevant (post, thread) = matches the query

Useful = has many information within (i.e. how enclosed the post)

(In generic web documents *Relevant* == *Useful*)



Formal definitions

- *Relevance* is well-studied => focus on *usefulness*
- Existing search systems can find *relevant* threads (but not specific posts!) => let’s create system on top of that!

Goal

The goal of our system is to retrieve *useful* posts in context of *query* from the set of *relevant* threads.

Offtopic can appear in *relevant* threads => Utility, our target variable*:*

0. Post does not match the thread or has no useful information *(not useful, or an irrelevant off-topic)*
1. Post contains some information about chosen topic, but lacks arguments or explanations *(relevant, moderate usefulness)*
2. Post contains useful information and arguments / explanation *(relevant and useful)*

Collecting the dataset

1. Take a forum and a narrow topic *(query)* within
2. Select threads which matches the query *(emulating the action of typical retrieval system; irrelevant threads are discarded)*
3. Mark down all posts from these threads:
    1. Thread URL
    2. Author
    3. Text (including quotes)
    4. Sentiment value (-2..+2, manually marked down)
    5. Utility (0..2, manually marked down according to criteria)

Example of a dataset

- Knitting forum
- Narrow topic is “Knitting techniques”

Author	Text	Sentiment	Utility
sgtpam	Wow, Rachel! What a great contribution to the thread!	+2	0
Rachel	Go slow, speed comes with experience. Also talk to yourself i.e.k1, p1, yo, etc. This registers with your brain which sends the message to your fingers, this is scientific fact and helps a lot…	0	2
ArtLady1981	lynn893, no, pin trick does not work that well. I prefer butterfly clips.	-1	1

Collected dataset

- 7 forums
- 94 threads
- 1553 posts

Features

Features must indirectly hint us *Utility*, target variable.

Textual features:

- Length
- Number of *query* keywords
- Sentiment value (marked down by experts)
- Links to external sources

Structural features:

- How many times has this post been quoted
- Position in thread (post ID within the thread)

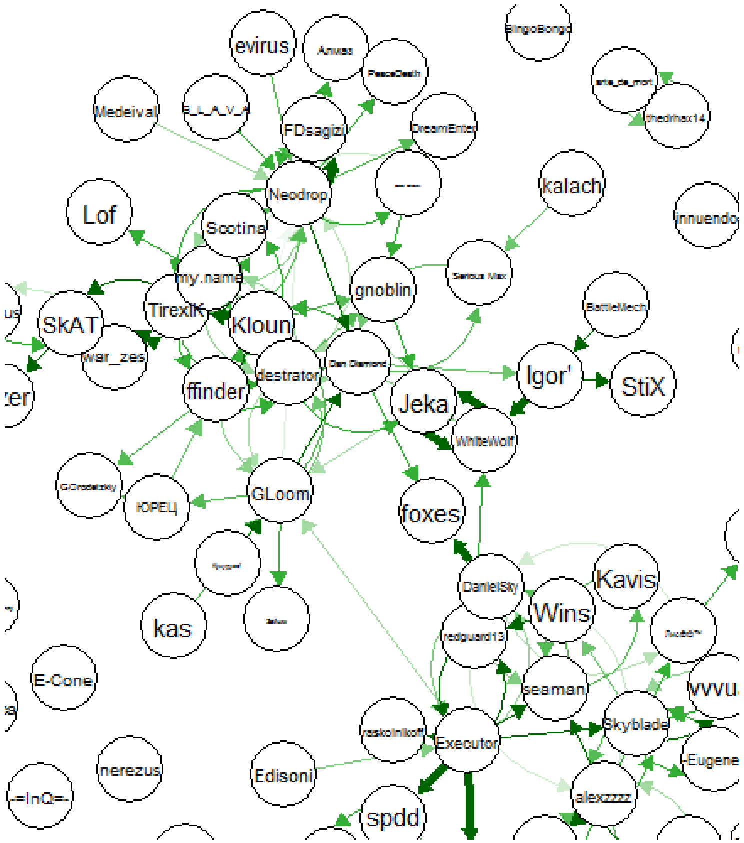
Graph features

Forums users can be represented via social graph *(User = node; directed link = quotation)*. Weights:

- Total number of quotes
- Sentiment value

Features (per user/node):

- Sum of input edges
- Sum of output edges
- Centrality (betweenness; how many shortest path cross the node)



Constructing models

- Linear model
- Gradient boosting model *(iterative random forest)*

These models estimate *Utility*, sort by estimation and choose top *N* (how many posts user wants)

Baselines

- Using keywords from semantic core of the request (imitates complex IR system)

Estimating model quality

Precision/recall? We have multiple classes

Micro/macro? Classes are skewed

Normalized Discounted Cumulative Gain metric – how close the selection of *N* posts from specific dataset if close to the ideal, with logarithmic emphasis on first posts?

rel_i - how good is the i_{th} selected post (Utility)

$IDCG_N$ – the maximum possible DCG_N (for ideal selection)

$$NDCG_N = \frac{DCG_N}{IDCG_N}$$

$$DCG_N = rel_1 + \sum_{i=2}^{N} \frac{rel_i}{\log_2(i)}$$

Normalized cumulative gain metric

Post	Utility
A	0
B	1
C	2
D	1
E	2

N=3
Our model selected: C,B,D
Ideal selection: C, E, B

DCG_3 = $Utility_C$ + $(Utility_B)/(\log_2 2)$ + $(Utility_D)/(\log_2 3)$ = 3,63
$IDCG_3$ = $Utility_C$ + $(Utility_E)/(\log_2 2)$ + $(Utility_B)/(\log_2 3)$ = 4,63

$NDCG_3$ = 0,78

$$NDCG_N = \frac{DCG_N}{IDCG_N}$$

$$DCG_N = rel_1 + \sum_{i=2}^{N} \frac{rel_i}{\log_2(i)}$$

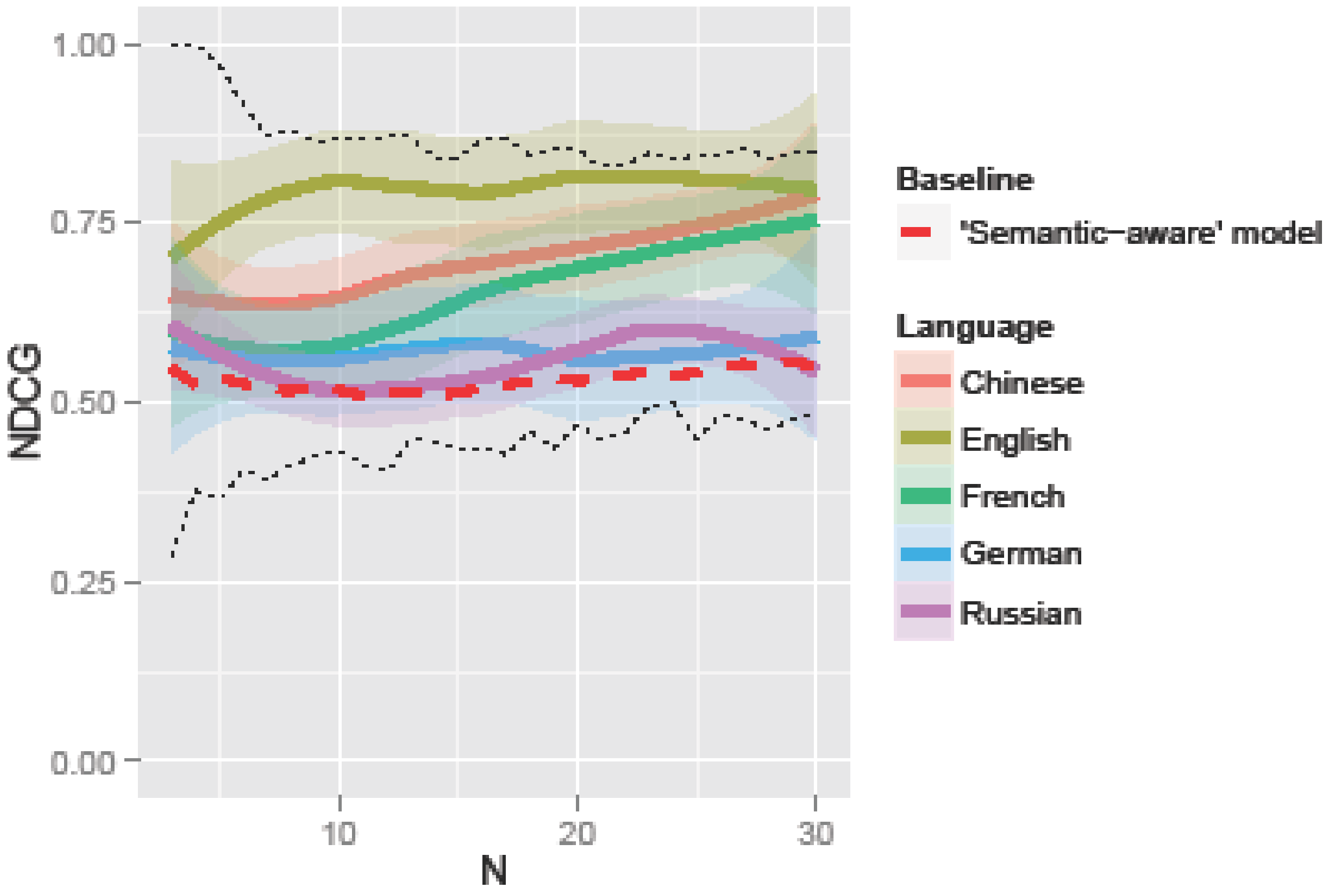
Stability of results

Bootstrap:

1. Resample dataset with replacement
2. Split the data into train/test sets
3. Train our model (for each language)
4. Evaluate model quality

Repeat ~200 times, calculate averages and variances

Results



1.00
0.75
0.50
0.25
0.00
NDCG
10
20
30
N
Baseline
'Semantic–aware' model
Language
Chinese
English
French
German
Russian

Getting best features

- p-value for linear regression, relative variable influence for GBM.
- Best features = recurring features from top-5 best features for each language

Best features

The best features are: sentiment value, text length, position in thread; number of keywords is fine too

Chinese	Russian	German	French	English
Sentiment value	Sentiment value	Text length	Sentiment value	Text length
Text length	Text length	Position in thread	Text length	Sentiment value
Position in thread	Author betweenness, non-sentiment graph	Sentiment value	Number of threads author is participating in	Author betweenness, non-sentiment graph
Number of keywords	Number of keywords	inDegree, non-sentiment graph	Number of links	outDegree, sentiment graph
Number of links	Position in thread	outDegree, sentiment graph	Number of keywords	Number of keywords

Conclusion

- The problem was defined
- Dataset was collected
- Textual and non-textual features were extracted
- Linear and GBM were constructed
- Model quality was estimated and checked for stability using bootstrap
- Best features were selected

TODO

- Apply methods to social media
- Develop experimental forum search

Q&A