NDBI021, Lecture 4

User preferences, 2/1 ZK+Z,

Wed 12:20 – 13:50 S8

Wed 14:00 – 15:30 SW2 (odd weeks)

https://www.ksi.mff.cuni.cz/~peska/vyuka/ndbi021/2022/

matfyz

siret research group

https://ksi.mff.cuni.cz

How to express user preferences

Feedback variants for users

Non-numeric feedback

How reviews improve personalization

Non-numeric feedback

- ## Textual reviews

Radek99 ★★★★ všechny recenze uživatele

Trochu tendenční film o nádherném a stále ještě mystickém světě vysokohorského Tibetu a jednom mladém Rakušanovi (na Heinricha Harrera ale po uvedení filmu prasklo, že byl v mládí nacista a vše asi bylo trochu jinak, než jak to líčí tenhle životopisně se tvářící film). Každopádně některé pasáže snímku jsou vysloveně povedené - obsazení Tibetu čínskou lidově-demokratickou armádou, Dalajláma nevěřícně osahávající Harrerovy plavé vlasy či Dalajlámovo opojení světem filmu... Není to tak silné a autentické jako některé jiné filmy s obdobnou tématikou týkající se Dalajlámy, Tibetu či jeho anexe, ale i tak se jedná o nadprůměrné filmové dílo... (svůj možný potenciál ale využil jen tak na 75 %, Jean - Jacques Annaud natočil již mnohem lepší a hlubší filmy...viz Jméno růže či Boj o oheň...) (27.09.2007)

- ## Semi-textual reviews

dejv

Hodnoceno 25.09.2019, varianta 32" ViewSonic VX3276-2K-MHD

★★★★★

Dlouho jsem poohlížel po levném a velkém monitoru. Tento monitor je v nízké cenové relaci a jak se říká za málo peněz hodně muziky

- Cena
- Velikost
- Kvalita
- IPS

- zatím žádné

Textual reviews

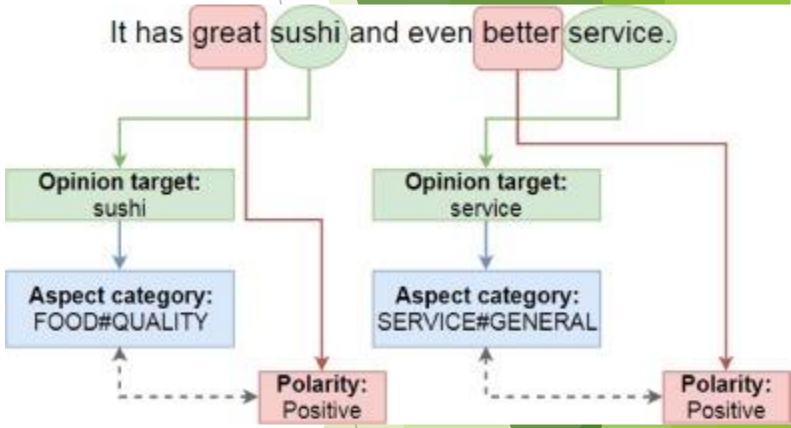
- **Main usage:**
  - Rating prediction from reviews
    - Multi-criteria rating prediction => recommendation
  - Explanations
- How:
  - (explicit) Sentiment analysis
    - https://dl.acm.org/doi/abs/10.1145/3109859.3109905 (restaurants)
    - https://dl.acm.org/doi/10.1007/s11257-015-9157-3 (hotels, fixed aspects)
  - Latent Dirichlet Allocation (and related approaches)
    - https://ieeexplore.ieee.org/abstract/document/8813018

Textual reviews

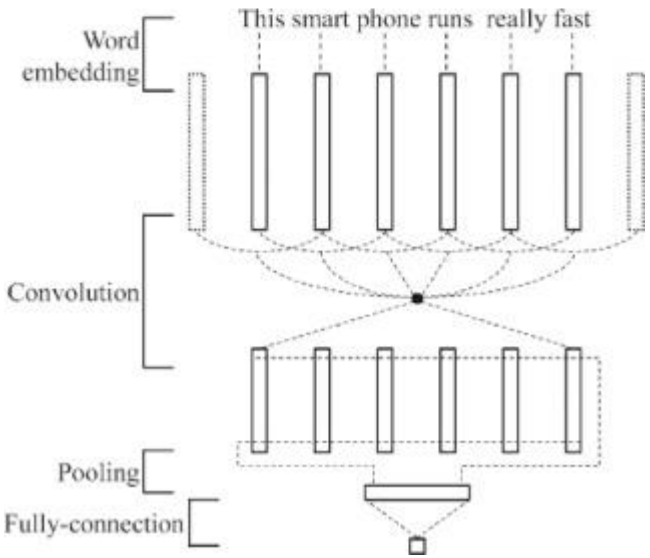
- Sentiment analysis https://dl.acm.org/doi/abs/10.1145/3109859.3109905
  - A **Multi-criteria** Recommender System Exploiting **Aspect-based Sentiment Analysis** of Users' Reviews
    - „SABRE“ framework, Output: aspect, sub-aspect, its relevance for reviewer & its sentiment https://link.springer.com/chapter/10.1007/978-3-319-46135-9_4
      - Aspect modeling as relatively simple frequency analysis – most common nouns [room for improvement]
      - Afinn wordlist for sentiment (annotated words)
    - Neighborhood-based recommendation model
      - Treat each aspect as independent rating, use multi-dimensional euclidean distance (serialize pairs of item-aspect into a single vector)

Textual reviews

- Deep Learning for Aspect-Based Sentiment Analysis: A Comparative Review (2018) https://www.sciencedirect.com/science/article/pii/S0957417418306456
  - CNN, RNN, Recursive NN

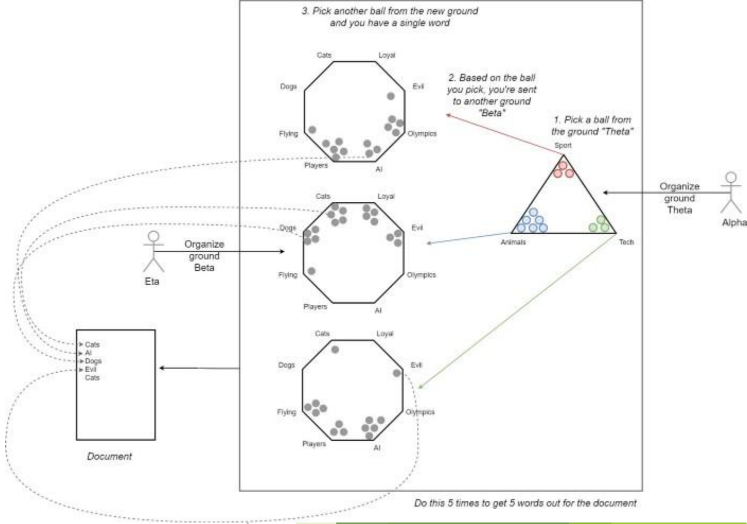


Three subtasks of aspect-based sentiment analysis:
(i) Opinion target extraction (OTE),
(ii) Aspect category detection (ACD
(iii) Sentiment Polarity (SP),



An example of CNN architecture for aspect category and sentiment polarity. Adapted from Gu, Gu, & Wu (2017) https://doi.org/10.1007/s11063-017-9605-7.

Textual reviews



- https://ieeexplore.ieee.org/abstract/document/8813018
- Customer Reviews Analysis With Deep Neural Networks for E-Commerce Recommender Systems
  - Latent Dirichlet Allocation (LDA, https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation , https://towardsdatascience.com/light-on-math-machine-learning-intuitive-guide-to-latent-dirichlet-allocation-437c81220158)
    - Documents; fixed set of latent topics; each document is a mixture of topics, each topic is characterized as a Dirichlet distribution over words
    - Assume generative model for documents and then try to reverse-ingeneer it
    - Several ways to learn, e.g. Variational inference / EM alg.
    - https://en.wikipedia.org/wiki/Expectation%E2%80%93maximization_algorithm
    - https://en.wikipedia.org/wiki/Variational_Bayesian_methods

The way to do this is to minimise the KL divergence between the approximation and true posterior as an optimisation problem. Again I'm not going to swim through the details as this is out of scope.

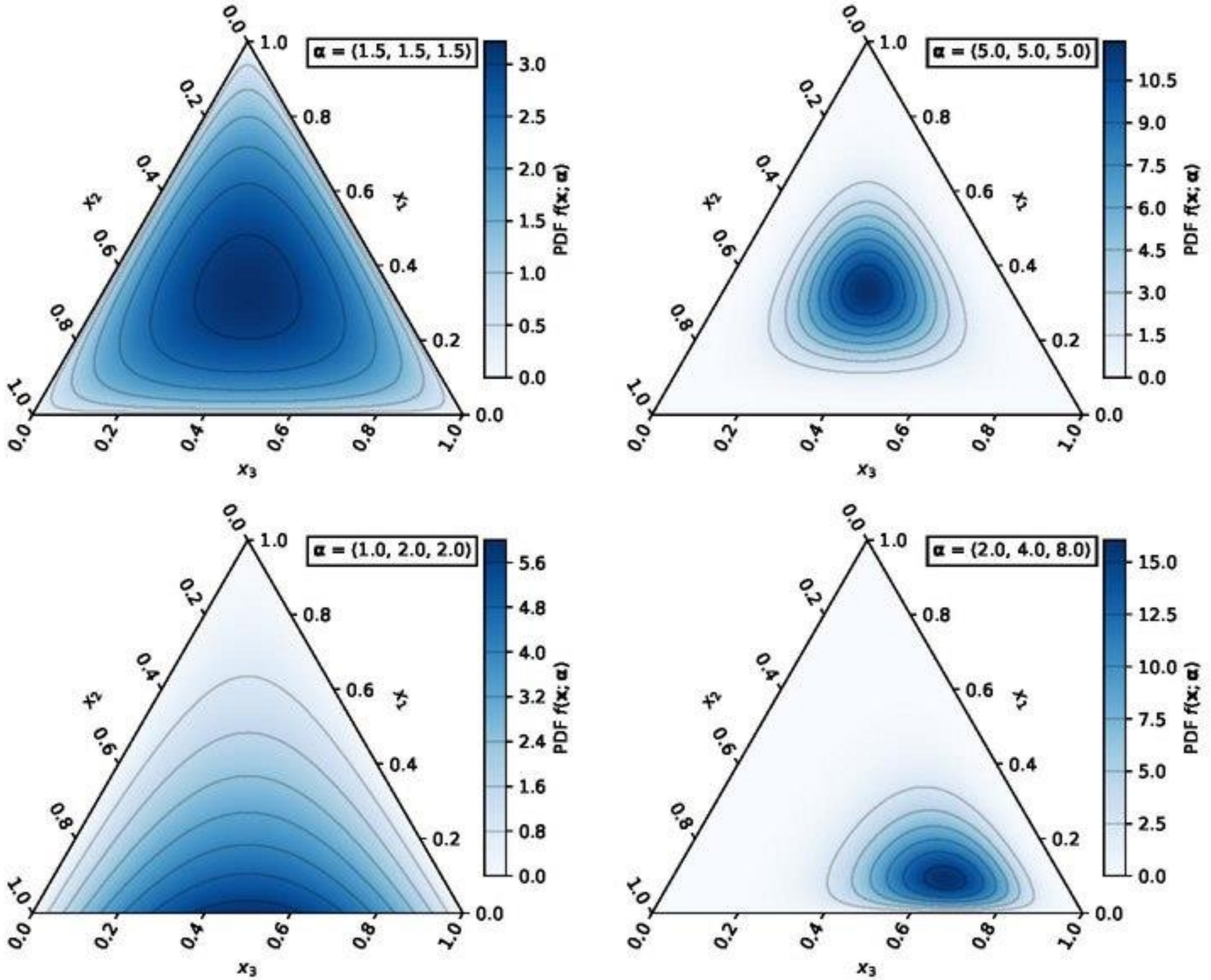
But we'll take a quick look at the optimization problem

$$\gamma^\star, \phi^\star, \lambda^\star = \text{argmin}_{(\gamma,\phi,\lambda)} D(q(\theta, \mathbf{z}, \beta|\gamma, \phi, \lambda)||p(\theta, \mathbf{z}, \beta|\mathcal{D}; \alpha, \eta)$$

γ , ϕ and λ represent the free variational parameters we approximate θ,z and β with, respectively. Here D(q||p) represents the KL divergence between q and p. And by changing γ,ϕ and λ, we get different q distributions having different distances from the true posterior p. Our goal is to find the γ^* , ϕ^* and λ^* that minimise the KL divergence between the approximation q and the true posterior p.

Textual reviews

- Dirichlet distribution
  - Multi-variate generalization of Beta distribution
  - https://en.wikipedia.org/wiki/Dirichlet_distribution



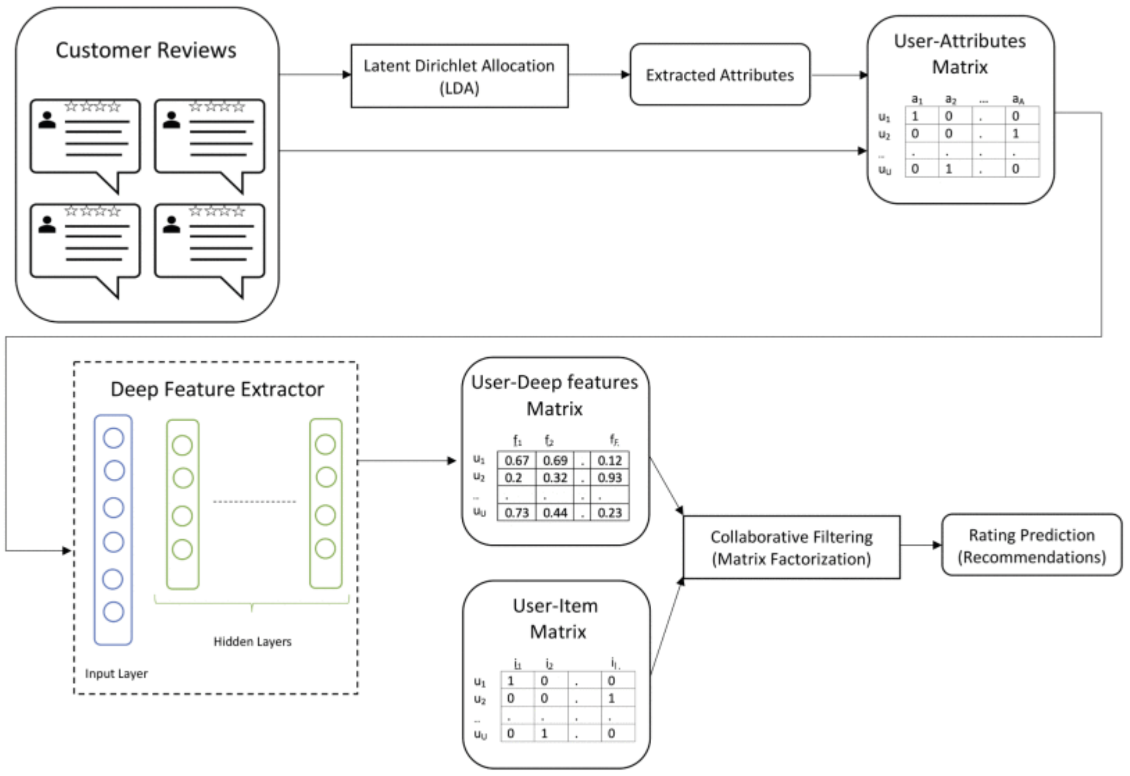
Textual reviews

- LDA is a variant of topic modeling algorithms, there are other options, see e.g.:
  - https://arxiv.org/pdf/2103.00498.pdf (Topic Modelling Meets Deep Neural Networks: A Survey)
  - https://medium.com/data-folks-indonesia/recent-works-in-topic-modeling-56c38da8dfc4

Textual reviews

- https://ieeexplore.ieee.org/abstract/document/8813018
- Customer Reviews Analysis With Deep Neural Networks for E-Commerce Recommender Systems
  - Latent Dirichlet Allocation (LDA) on all user reviews
  - Get binary user-attribute matrix (sparse)

-> DL [maybe redundant] for dense vector
-> Nearest neighbor model for rating prediction
(user-user similarities)



Textual reviews

- https://dl.acm.org/doi/pdf/10.1145/3412841.3442065
- Utilizing Textual Reviews in Latent Factor Models for Recommender Systems
- Latent Dirichlet Allocation (LDA); document = all reviews for item
  - Use LDA to get item-attributes, use them in matrix factorization
  - Joint optimization model for MF based and LDA based parts
  - EM procedure for optimization

$$f(\mathcal{T} \mid \Theta, \Phi, \kappa, z) = \sum_{u,i \in \mathcal{T}} (r_{u,i} - \hat{r}_{u,i})^2 + \lambda(\|p_u\|_2^2 + \|b_i\|_2^2 + \|b_u\|_2^2) - \mu l(\mathcal{T} \mid \theta, \phi, z) \tag{14}$$

where $\Theta = \{\alpha, \mathrm{b}_u, \mathrm{b}_i, \mathrm{p}_u, \mathrm{q}_i\}$ and $\Phi = \{\theta, \phi\}$ represent the set of parameters of the LFM and LDA model, respectively. The first term of Equation 14 represents the prediction error corresponding to LFM, the second term represents the regularization of model parameters $\mathrm{b}_u, \mathrm{b}_i, \mathrm{p}_u$ and the third term represents the log-likelihood of the corpus of ratings and users from Equation 11. The parameter $\mu \in R^+$ trades-off the importance of these two effects. We observe that in

Textual reviews

- http://ceur-ws.org/Vol-2068/exss8.pdf [vision paper]
- Explaining Recommendations by Means of User Reviews
  - Extract & summarize arguments about products from reviews
  - Use them in Personalized explanations

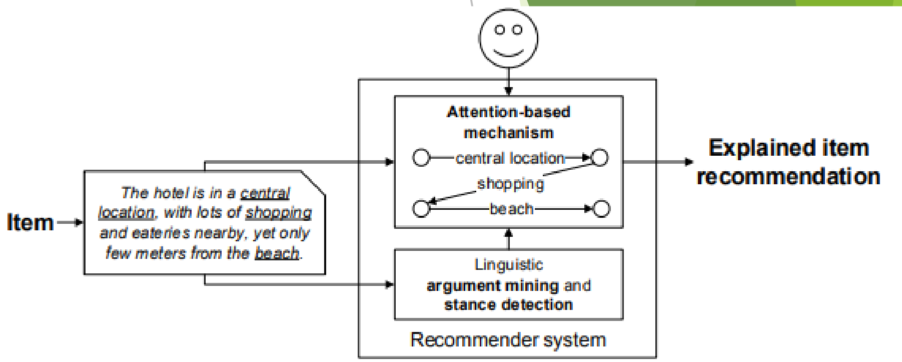


Figure 1. In our framework, a review is analyzed linguistically and via an attention-based mechanism. This allows to implement an argumentation flow based on information provided in the review while deeply integrating the user. Eventually, a personalized recommendation is presented together with individual arguments for or against this product.

Challenges:
- Linguistically analyzing review texts via argument mining and stance detection.
- Identifying important concepts for a target user via an attention-based mechanism.
- Deriving an argumentation flow via multiple applications of the attention-based mechanism.
- Unifying the linguistic analyses and the attention-based mechanism.

Textual reviews

- https://dl.acm.org/doi/pdf/10.1145/3320435.3320457 (2019) Justifying Recommendations through Aspect-based Sentiment Analysis of Users’ Reviews
- Aspoect extraction:
  - Part-of-Speech (POS) tagging algorithm (nouns = possible aspects)
  - Aspect ranking: relevant+positive & distinguishing
    - For each aspect number of sentences + average sentiment + IDF

$$score(a, R_i) = \left(\alpha \frac{n_{a,R_i}}{|S_{R_i}|} + \beta \frac{pos(a, R_i)}{pos(a, R_i) + neg(a, R_i)}\right) * IAF(a, R_i) \quad (1)$$

Sentences containing aspect *a*

Positive vs negative sentiment

IDF-like weighting

"I recommend you 300 because people who liked the movie think that the war scenes are really well done. Moreover, people liked 300 since the soundtrack is very appropriate."

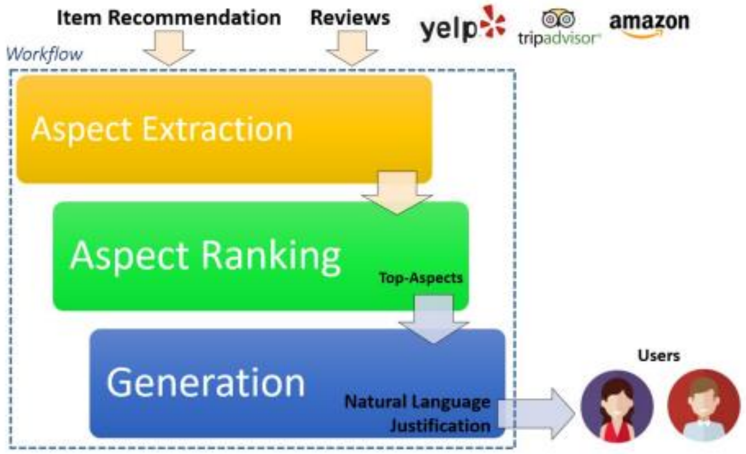


Figure 2: Workflow carried out by our framework.

⟨*justification*⟩ ::= ⟨*intro*⟩ because ⟨*excerpts*⟩

⟨*intro*⟩ ::= I suggest you ⟨*item_name*⟩ | I propose you ⟨*item_name*⟩ | I recommend you ⟨*item_name*⟩

⟨*excerpts*⟩ ::= ⟨*first_static_phrase*⟩ ⟨*review_excerpt*⟩ . ⟨*adverb*⟩ , ⟨*second_static_phrase*⟩ ⟨*review_excerpt*⟩ .

⟨*first_static_phrase*⟩ ::= people who liked this movie think that | people who watched the movie think that

⟨*second_static_phrase*⟩ ::= other people think that | people liked <item_name> since

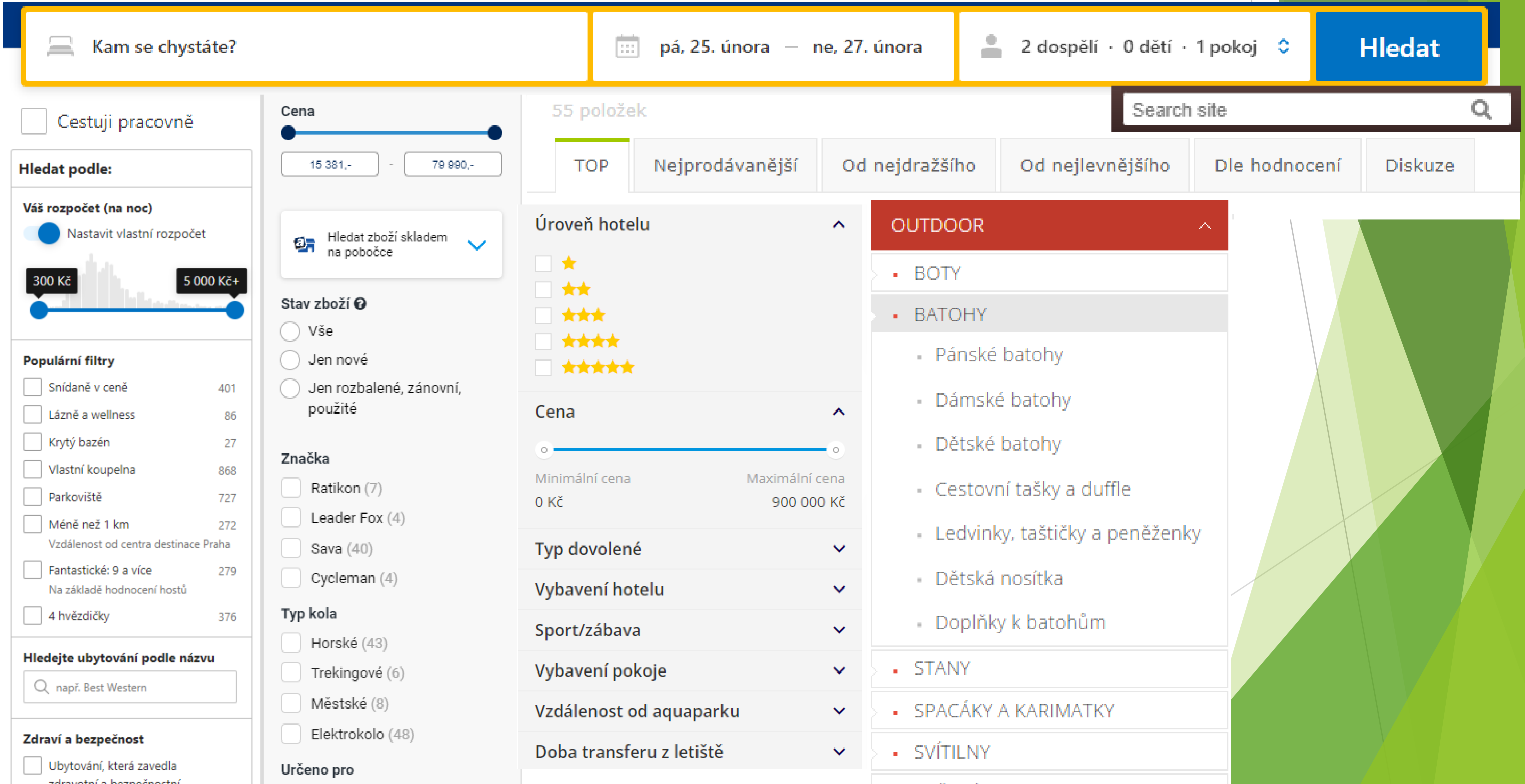
⟨*adverb*⟩ ::= Furthermore | Moreover | Besides

⟨*item_name*⟩ ::= name of the recommended item

⟨*review_excerpt*⟩ ::= a compliant sentence

Searching and filtering as feedback

What would the user be willing to do?



What would the user be willing to do?

Most users do:

- **Filter content manually**
  - Browse categories
  - Apply facet search
    - Mostly direct mapping to object's attributes
  - Use fulltext search
  - Can be utilized in the construction of attribute-level preferences
    - Beware of long-term preferences vs. short-term goals

All users do:

- Evaluate & consume content:
  - Browse items, open details, read content, play, purchase,…
  - Preferences based on implicit feedback

How to utilize searching / querying feedback?

Query refinement

- User gives some (textual) query, we recommend him/her query extensions/modifications
- Traditional approach: https://link.springer.com/chapter/10.1007/978-3-540-30192-9_58
  - Query Recommendation Using Query Logs in Search Engines (2004)

1. Queries along with the text of their clicked URLs extracted from the Web log are clustered. This is a preprocessing phase of the algorithm that can be conducted at periodical and regular intervals.
2. Given an input query (i.e., a query submitted to the search engine) we first find the cluster to which the input query belongs. Then we compute a rank score for each query in the cluster. The method for computing the rank score is presented next in this section.
3. Finally, the related queries are returned ordered according to their rank score. The rank score of a related query measures its interest and is obtained by combining the following notions:
   1. Similarity of the Query. The similarity of the query to the input query. It is measured using the notion of similarity introduced in Section 3.1.
   2. Support of the Query. This is a measure of how relevant is the query in the cluster. We measure the support of the query as the fraction of the documents returned by the query that captured the attention of users (clicked documents). It is estimated from the query log as well.

How to utilize searching / querying feedback?

Query refinement

- Not just similarity, but rather expansion of the query
- Diversity of the recommended expansions
- Beyond bag-of-words models (NLP, deep learning)
- Sequential models (bandits, RNN)

Further readning:

- https://link.springer.com/chapter/10.1007/978-3-030-72240-1_54
- https://proceedings.mlr.press/v157/puthiya-parambath21a.html
- https://dl.acm.org/doi/10.1145/3269206.3271808

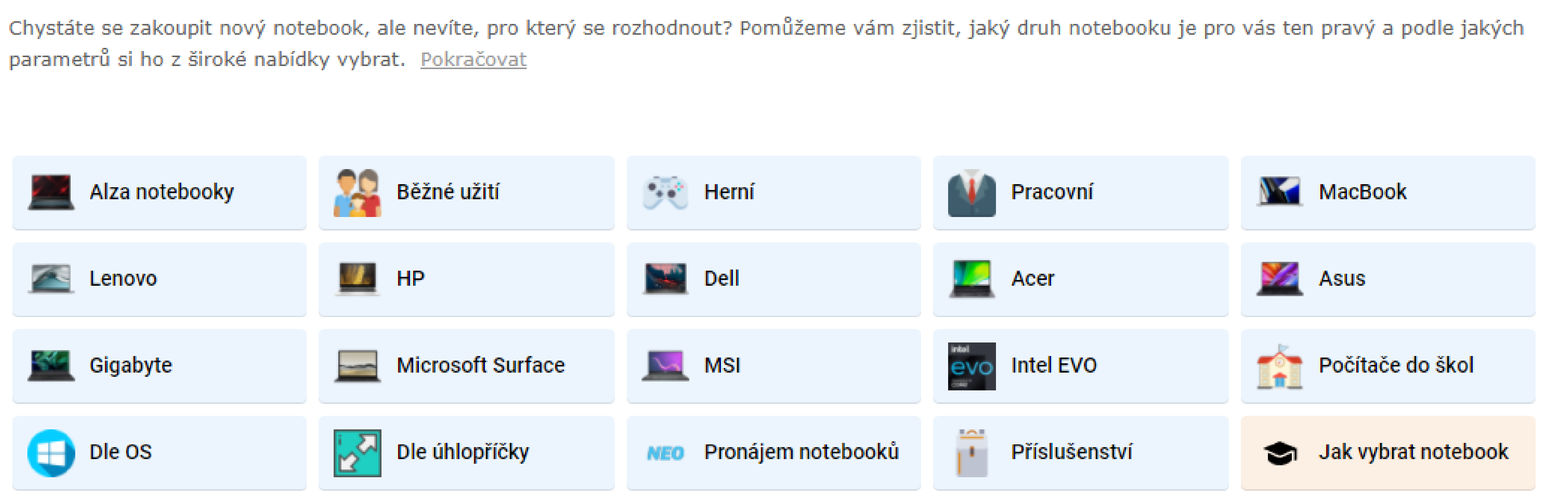
In general:

- Usable for current information need of the user
- Limited applicability for estimating long-term preferences

How to utilize searching / querying feedback?

Query refinement

- In theory, applicable also for faceted search logs / category browsing (transformation to key-value pairs)
- Not very clear how to present it to the user
  - Customized banners such as Alza have?
  - Needs additional description generation model => but then, why not to search simply by keywords?



How to utilize faceted search logs / browsed categories?

???

How to utilize faceted search logs / browsed categories?

???

- Almost no available literature
  - No (to the best of my knowledge) available datasets combining recommendations and facet search logs
  - => Largely ignored by academic researchers
  - No confirmed info from the industry

- So, why should we bother?

How to utilize faceted search logs / browsed categories?

- So, why should we bother?
  - Depending on the domain (based on the data I have available)

    Visits of objects vs. visits of search / browsing pages are approx. 50:50

    - Recommendation-first designs are less informative (users did not filter anything manually), but e.g. E-commerce websites may be highly relevant
    - User’s intent can be inferred from the searched / filtered terms & it can be done faster than if only feedback from visited objects is collected

- How to distinguish short-term needs vs. long-term preferences?
- How to detect interest / preference drift?

How to utilize faceted search logs / browsed categories?

- Option #0 just filter the recommendations
  - Implicit assumption: User preferences are binary & exactly as stated in the search
    - Post-process any recommendations to fulfill searched conditions (or their slightly relaxed versions)
    - Use e.g. the last search record to filter recommendations given on particular object (a.k.a. similar objects)

How to utilize faceted search logs / browsed categories?

- **Option #1 content-based representation**
  - Model search pages / browsed pages in the same way as visited objects
    - Vector representation of object's attributes
    - The same representation for searched terms (leave blank if unknown)
    - Alternatively, page is represented as a (weighted) sum of items it displays
  - Apply any suitable sequence-based recommender system on such data
  - Diploma thesis of Kaan Yos: „Deep Learning For Implicit Feedback-based Recommender Systems", https://dspace.cuni.cz/handle/20.500.11956/121242
    - Limited search data on a travel agency (dates, tour type, accomodation type)
    - LSTM, several encoding variants
    - Next item recommendations
  - Suitable for short-term user needs (sessions)
  - Possible extensions: aggregated information from past sessions => latent model for long-term pref. (similar as https://dl.acm.org/doi/10.5555/3367471.3367627)

How to utilize faceted search logs / browsed categories?

- **Option #1 content-based representation - extension**
  - Adaptive user modeling with long and short-term preferences for personalized recommendation
    - https://dl.acm.org/doi/10.5555/3367471.3367627
    - Latent model based on two components: long-term and short-term user preferences
  - Short-term: based on LSTM trained on the sequence of user behavior (tweaks with time distance)
  - Long-term: assymetric SVD
    - users are represented through weighted sum of items they interacted with
    - This representation can be modified e.g. to cover searched terms

  - Adaptive fusion of long and short term preferences to derive final latent vector for user

How to utilize faceted search logs / browsed categories?

- **Option #1 content-based representation - extension**
  - Beware on how to represent search terms
    - Different ranges for the same attributes throughout various categories (e.g. Fridge vs. Keyboard)
    - Different set of attributes for various categories
    - The same value may have a different meaning throughout the time
      - „500GB HDD“ now vs. 5 years ago
      - „Movies from 2018“ now vs. 3 years ago

  - Try to compensate for these biases

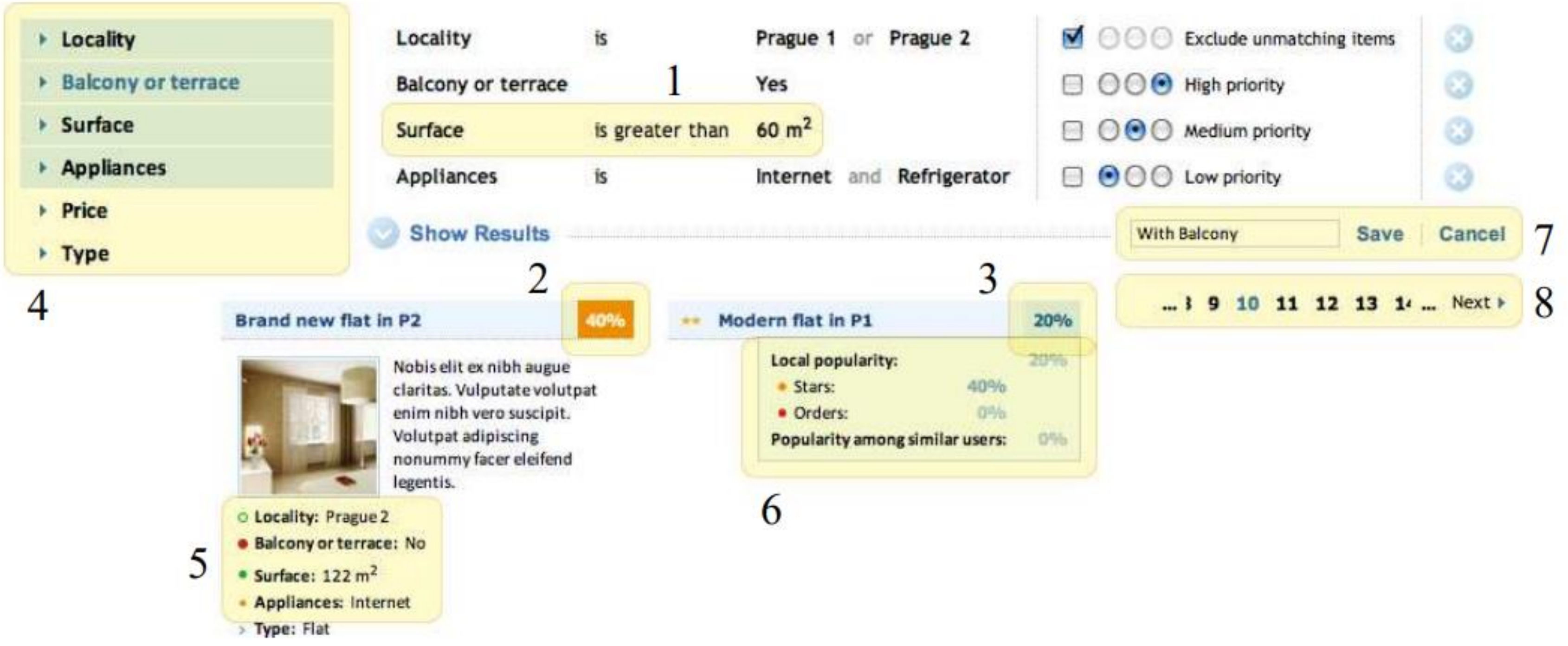
How to utilize faceted search logs / browsed categories?

- **Option #1 content-based representation - explicit model**
  - Latent vs. Explicit model (previously described is latent)
  - Explicit model:
    - Distribution on searched values vs. all possible values
    - Probably relevant only for a subset of attributes
    - What about context (of other searched criteria)
    - Be especially aware of biases – category agnostic predictor (use CDF or similar rather than raw data)

    - Given other searched terms, try to predict what values would be searched by the user in not-yet-filled facets => use this to rank items / recommend particularly good ones

How to utilize faceted search logs / browsed categories?

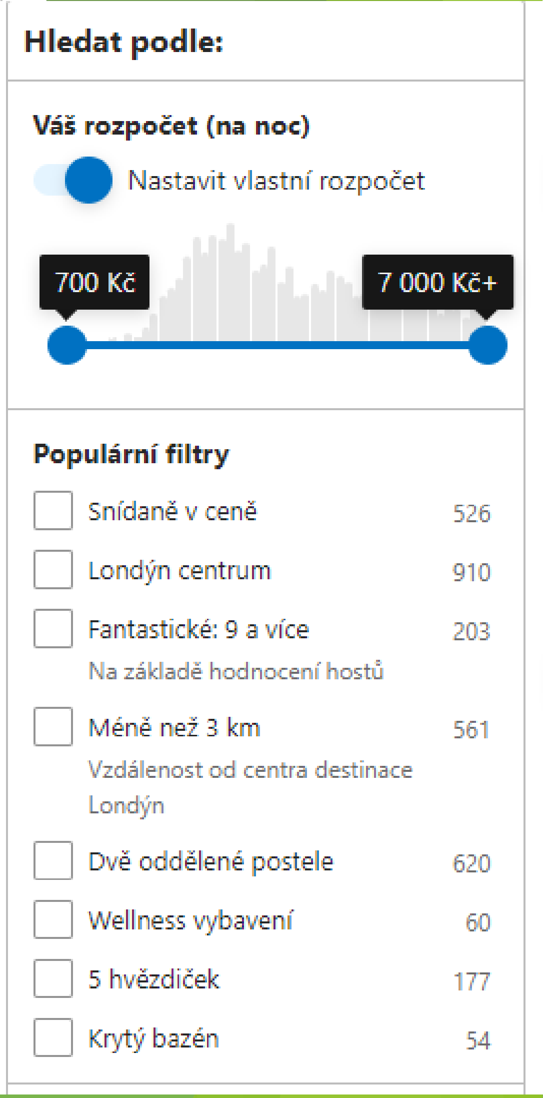
- **Option #2 extend facet search with automated ranking**
  - Soft & hard constraints / importance of individual constraints
    - https://dl.acm.org/doi/pdf/10.1145/3425603
    - Diploma thesis of Bronislav Vaclav „Models of user preferences in e-shop environment“ https://dspace.cuni.cz/handle/20.500.11956/30703



!! If all constraints are met, items are undistinguishable !!

How to utilize faceted search logs / browsed categories?

- **Option #3 recommend/re-order filtering options**
  - If there are too many filtering options, the relevant ones might be difficult to find
    - Recommend best options for the user
      - Nowadays, this is usually done in a non-personalized fashion
    - Personalization based on
      - Utilization statistics (the more used the higher position – multiarmed bandits, beware of feedback loops – discoverability models)
        - Collaborative/contextual model possible in case of insufficient data per user
      - Background user preference model & ability to distinguish preferred vs. unpreferred (e.g. Information gain, https://en.wikipedia.org/wiki/Information_gain_in_decision_trees)



How to model UP

Tenative solutions for show-cases

How to model UP

Simple movies recommendation:

- **Task: discover what to watch tonight**
- How to use UP: Collaborative recommendation of movies

Basic model of UP:

- Preferences on movies (rating, watching)
  - If insufficient data: openning movie details, top search results

Enhancements:

- Learned (confirmed) preferences towards genres (multiple confirmation, enough data)
- Learned (confirmed) preferences towards other named entities (actor, director)
- List-wise preferences (Y was selected from results of XYWZ)
  - Remember impressions, not just usage

How to model UP

(Food) Recipes recommendation:

- **Task: help to decide what to cook**
  - How to use UP: **personalized searching, front-page recommendation**

Basic model of UP:

- Preferences on recipes (likes, add to list, reading sufficiently long)
- Preferences on ingredients (search count, contained in prefered recipes, confirmation?)
  - Ingredients granularity?

Enhancements:

- Learned preferences towards tags & attributes
  - Verify on a well-known subset of users (RecSys OPS)
- Best out of similar choices
  - Which goulash does the user prefer? Would that say something more generic about him/her?
- Should we allow users to further refine recommendations?
  - Faceted recommendations (https://www.researchgate.net/publication/301321425_FeRoSA_A_Faceted_Recommendation_System_for_Scientific_Articles)

How to model UP

Group music recommendation:

- **Task: create a background music playlist for an evening with friends**
- How to utilize it: fairness-aware playlist construction

- Individual preference
  - Track -> Album -> Artist (playcount, play from search, likes)
  - Maybe, preferred sequences (low-level audio analysis, but probably not for individual users)

- Group preferences
  - Playlist modifications