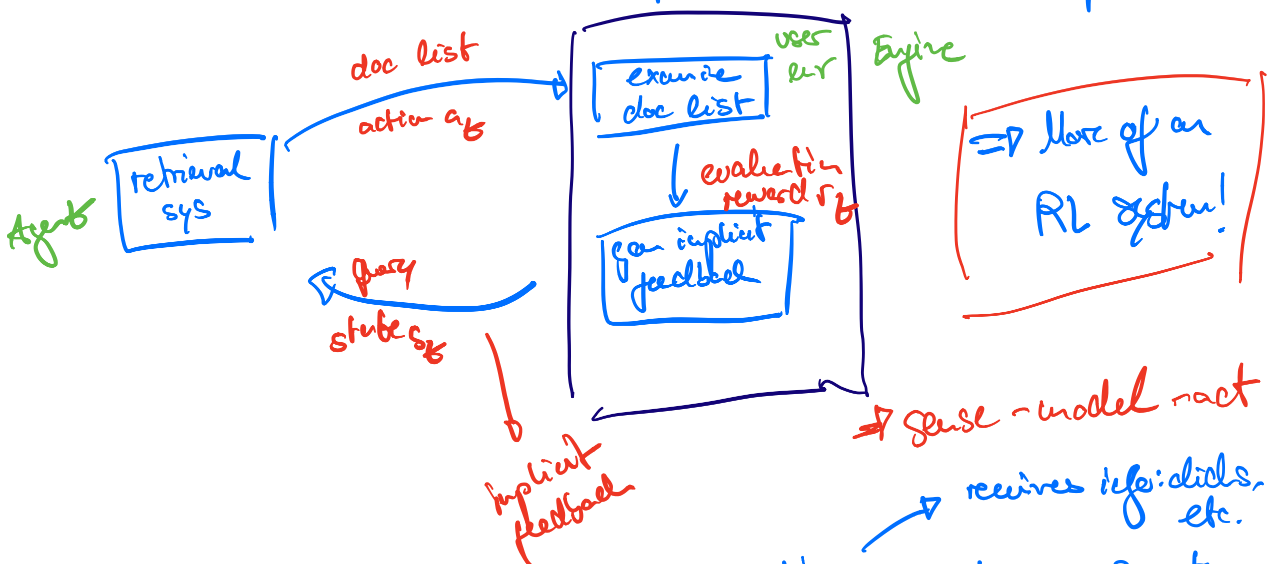
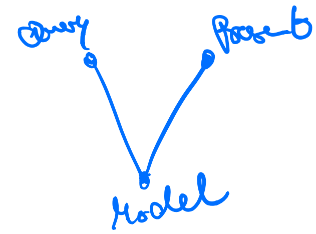
Talk 1 - M. De Rijke - Foundations of IR   04/08

- IR => Ranking engine? -> Development of Agents
  - * right info to right people in the right way
- ↳ individualized -> context dependency => high-dim. problem



- ("Front door") => UI + Query Suggestions



- Evaluation => e.g. A/B Testing
- Modeling => Understanding Query and Max. Reward
- Offline Comp. in IR
  - => No direct user input needed / non-query related
- UX
  - -> efficient/informative feedback -> provide shortcuts
  - -> balance control/automation -> error reduction
  - -> Reduce short-term memory -> rec. importance of details (typos)
  - -> Rec. importance of aesthetics
- AOL log: contains a lot of specific information!

- Query Auto Completion: Drive people away from queries that have no results
- Rank QAC candidates via MLE based on popularity of queries
  → Res.: No dynamics in popularity distr.
- **Evaluation**
  → Metrics
  → Flighting
  → Learning
  → Logging
  → Annotations

① Offline ⇒ Have assessors determine which docs are relevant → metric
↳ Assesses relevance is same for all users
↳ fixed exp. conditions
⇒ reproducible!

② User-study ⇒ Exposure to exp. system
↳ Observe behavior, ask questions
↳ Can obtain detailed data
↳ Expensive, difficult to generalize

③ Online ⇒ Observe implicit behavior - clicks, skips, saves, forwards, bookmarks

↳ Infer differences in behavior from different flavors of the live system
↳ A/B testing; $\boxed{\text{Interleaving}}$ ⇒ Expose combination of system versions
↳ based on natural interactions

⇓

Common mistakes:
* Consistent experience ⟷ same person should be in the same group (use cookies)
* Always run A/A test ⟷ observing differences might be indicator of problem

↳ A/B testing vs. Interleaving: Interleaving compares behaviors not populations ⇒ Need for less interactions!

• $\boxed{\text{Online}}$

→ Query Understanding ↓ ↑

* Alterations : translation, search syntax
* Classifiers : query intent, performance prediction, task/device det.
* Annotators : user modeling, localization
* Aggregators : query stats

→ Intent Identification ⇒ Think about it as a multiclass classification problem → mixed allocation possible

* Intent-aware result ranking
* Intent classification
* Intent Discovery

extract structure

S. goal || S. task || Semantic Topics ||

Vertical-oriented intents || Time sensitivity || → news

Location-sensitivity

navigational, informational, transactional → Rose and Levinson (2004)

image, news, video

URL clusters ⇒ express similar needs/intents

vertical might be dynamic! ⇒ Easter e.g. // Elections

→ Ranker parametrization

⇒ type selection || param. selection || direct answers

→ SERP generation

⇒ mix results

* hotfixes ⟷ personalized
* query document features
* ranking model

→ Extrinsic diversity

→ Intrinsic diversity

a lot of criteria!

→ Learning To Rank

- Pointwise ⟺ Score
- Pairwise ⟺ Ordering
- Listwise ⟺ learn a list of rankings

offline initial training ⟷ online updating

GOAL: $C = \sum_{t=0}^{\infty} \gamma^{t-1} r_t(a_t)$

discount, reward, action

↳ needs to generalize across contexts

↳ Interleaved comparison : Infer listwise relative feedback

↳ Dueling bandit GD : Optimizes weight vector for lin. comb.

⟹ Syst. A, Syst. B → Online

→ small step in direction of system that was successful

↳ Probabilistic GM for user interaction

- Random click model
- CTR-based models : Rank-based, Document-based

CTR : click through rate

- Position-based model : Examination ⊕ Attractiveness ⟹ Click
- Cascade model : Dependencies of seq. clicks

↳ Problem with scalability : Personalization vs. Contextualization

tables too large!

- Offline

  → Crawling: URL queue updating
    * problems with scalability
    * sequentially: avoid fetching them at the same time

  → Duplicate detection:
    * Exact vs. Near duplicates

  → Spam detection
    * Lucas' Critique problem ⇒ people want to be ranked high and adapt their behavior to deceive the search algo

  → Aggregation

- Open Research
  → Online ⇒ learn from interaction signals
  → Conversational search ⇒ Chats
    ↳ kind of Bayesian optimization!
  → FAB Web ⇒ Does not get enough attention
    fairness accountability

Talk 2 – M. Lalmas – System-oriented Evaluation in IR 04/09

- DCG: Discounted Cumulated Gain
- System Perf. vs. Retrieval Perf.

  System Perf. → speed of result generation

  Retrieval Perf. → correctness of result

- What to evaluate
  - ⇒ coverage: quality of collection
  - ⇒ time lag // efficiency
  - ⇒ presentation: interface, visualisation
  - ⇒ effort: user satisfaction
  - ⇒ recall // precision: retrieval effectiveness
    - ↳ SYSTEM-ORIENTED EVALUATION ⇒ not users

→ Test collection methodology: Benchmark comparison → important to use it on multiples → test generalization

→ Metrics

→ Info need ⇒ Info query

Info need: translation → description behind query

Info query → $\boxed{\text{Relevance}}$ is assessed relative to the need not the query!

→ "has significant and demonstrable bearing on the matter at hand"

→ Assumptions on <u>Relevance</u>:
* Objectivity ↳ Should be incorporated in retrieval model!
* Topicality
* Binary nature: Relevant or Not
* Independence: One documents' relevance doesn't affect relevance of others!

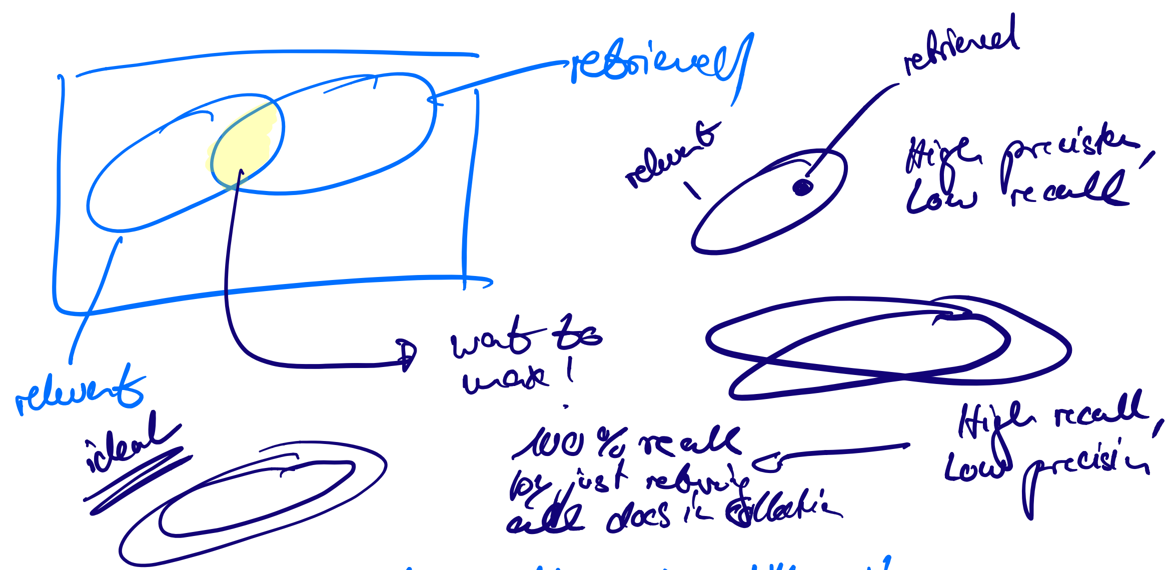
→ Absolute vs. relative comparison of ranking systems!

→ Test collection creation:
* Manual: People have to submit their ranking
  ↳ only possible for small document collections
* Pooling: Top documents are pooled together (proposed by different systems) and then reduced sample is judged by humans

→ TREC: Text Retrieval Conference
* Competition
  - made research systems scalable
  - emphasis on high recall
  - very long queries ⇒ unrealistic
  - focus on batch ranking ⟷ interactions ignored

⇒ PRECISION: Fraction of relevant instances among the retrieved instances.
$$\frac{\text{relevant retrieved}}{\text{total retrieved}}$$
positive predictive value

⇒ RECALL: Fraction of relevant instances that have been retrieved over total amount of relevant instances
$$\frac{\text{relevant retrieved}}{\text{total relevant}}$$
sensitivity



↳ Till now: Set based ⇒ No ranking till now!

↳ Interpolation: Allows for standardized comparisons!
⇒ Let us obtain standardized precision values
⇒ Obtain recall and precision for a lot of queries
⇒ Average results to obtain Average precision

↳ One number: Mean Average Precision (MAP)
⇒ Rank of each relevant doc: ↳ interpolated and non-interpolated
⇒ compute precision at rank $r_1, \dots, r_n$

r_1
r_2
$\vdots$
r_n

$$1/3\left(\frac{1}{1}+\frac{2}{3}+\frac{3}{5}\right) \approx 0.76$$

↳ Average precision for this query

Average over all queries!

Look at queries which don't work!
↳ learn about model

→ DCG: Discounted Cumulative Gain

* Highly relevant docs are more useful than marginally relevant ones
* Graded relevance ⟺ gain: accumulated starting at the top of the ranking ⟷ reduced (discounted) at lower ranks
  ↳ typical discount: $1/\log(\text{rank})$

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

need to define a relevance scale!
⇒ proxy by CTR (click-through-rate)

* Normalized version: $NDCG := \frac{DCG_i}{\text{ideal } DCG}$ $i = 1, \dots, S$ — order of experiments/systems

→ user-evaluation: experience, etc.! ⇒ in the end it is all about the user!

Talk 3 - Joemon M Jose - Neural IR 05/09

User Query → Similarity Matching ⟷ Doc. Collections

Retrieved Results ← Similarity Matching

• Problems:
  * Intention of query intention
  * Representation of documents // content] How to match?

Real Info Need (RIN) → Perceived Info Need (PIN) → Request → Query

Perception, Expression, Formalise ⟺ Types

PIN ⟷ implicit in mind

Request ⟷ Language representation

• Y. Zhang et al ⇒ 'Neural Information Retrieval: A Lit. Review'

• One-hot vector: 0s in all cells with exception of single 1 in a cell used to uniquely identify the word
  ↳ no concept of similarity

• Dist. Representation:
  → Embedding for words in continuous space
    ↳ captures hidden structure
  → Hyperspace Analogue to Language (HAL)
    LSA ⇒ matrix factorization of co-occurrence matrix

• Bengio et al (2003): Express joint prob of word seq. in terms of feature vectors of word sequences

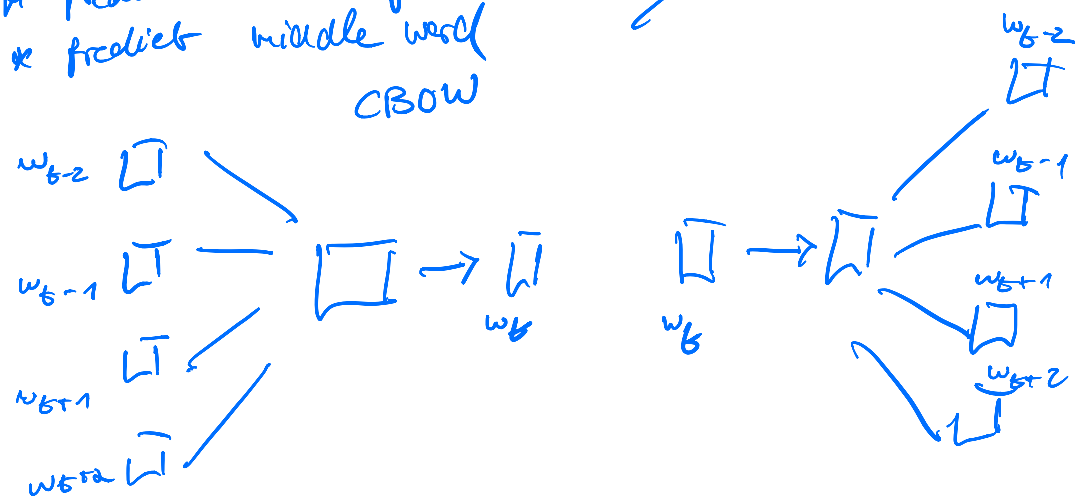
• Mikolov et al (2013):
  ⇒ Xin Rong: word2vec Param. Learning Explained

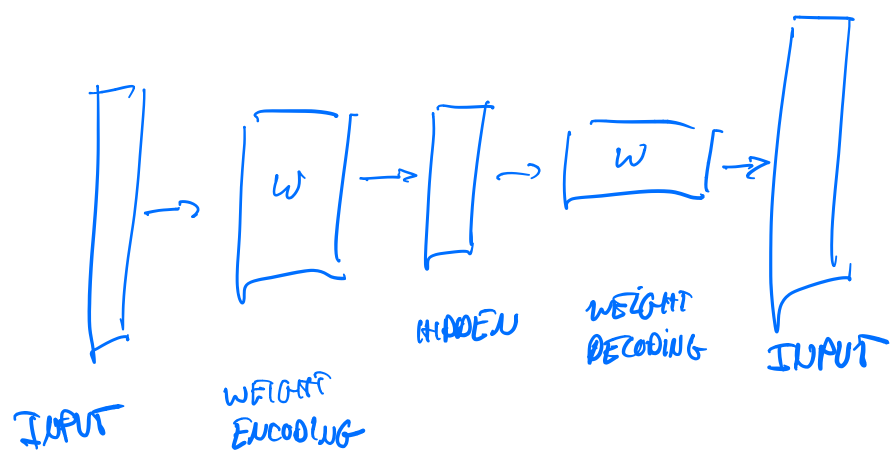
CBOW
Skip-Gram

> Efficient and simple way to learn word-vector representations

↳ Distr. Hypothesis: Same Context ≈ Same Meaning
* Word similarity ⟺ vector similarity
* Predict surrounding words
* Predict middle word

> CBOW





GENERAL STRUCTURE

- Negative Sampling: Each training sample only modifies a small percentage of the weights, rather than all of them
- Zuccon et al (2015): Integrating and Evaluating Neural Word Embeddings...

- Zheng, Callan (2015)
  - ⇒ Reweighting: subtract mean of models from query vector
  - ↳ Run regression to obtain weights
- Query Embeddings!
- Also word embeddings can be used to complete queries
- Dual Embedding Space Model (DESM)
  - → neighbors of word represented in the $\boxed{IN}$ embedding vector into $\boxed{OUT}$ space are typically similar words
  - → In same embedding space functionally similar words are closer

  ↳ different types of embeddings induce different forms of relations!
- Learning to Match: Matching distr. representations complements matching with traditional local reps.

↳ Q: Is there any way how to build IR version of embeddings ⇔ IR task is different from NLP task!

Talk 4 - E. Alfonseca - Conversational Search 06/09

- Critical shift: Voice Search! || Knowledge Graph → structural representations → tags to DB query!
  - 50% of google search comes from mobile!
  - → Conversational search more common
  - → Also: Queryless search → Anticipate needs
- Anticipation || Answer || Converse
  - → Speech input ↔ Text ↔ Speech output
  - Text ↔ NLP understanding
  - CONTEXT — Pisa vs. Pizza
  - Speech input → 20% of all mobile queries are audio → differs by market → lower the "modern" markets; higher if text-entry complex
  - is → Local queries are very common ↔ Only look at small # users ↔ longer queries length
  - → Conversation means that search is not stateless
- How to answer?
  - → no textual sense
  - → answer needs to be kept short
- [•] How to set confidence threshold for when to use knowledge graph? ↔ optimized systems?
  - ⇒ Query is send to different backends
  - ↳ Answer is provided when there is a 88% precision

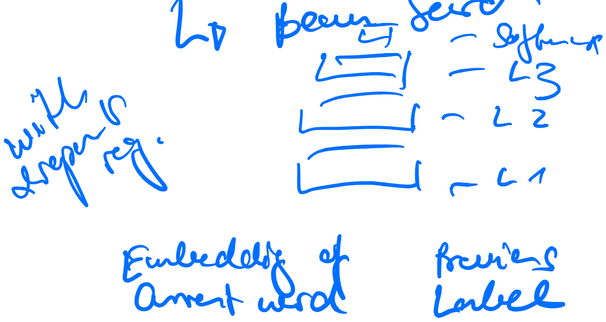
- Mobile queries ⇒ different intents → news / local info
- Answer generation has to account for different language specifics!
- Knowledge Graph covers around ≈1000 relations!
  → Construction of Lexicons by linguists

Hard coded answers by computer linguists! ⇒ few thousand relations covers 95% of all queries

↳ No DL // RNNs // LSTMs used ↔ Only obtain 60-70% accuracy → 10 attributes make up 90%

↳ Hard coding is worth it right now

- Sentence compression ⇒ No reordering! → Extraction!
  → Integer Linear Program: decide whether or not to keep an edge dependency!
  → Minimum Spanning Subtree that covers all headlines
  → Syntactic and preprocessing errors propagate
    ↳ Sentence read token-by-token is translated into zero-one sequence
    ↳ LSTM encoder-decoder ⇒ seq2seq → binary: keep or drop word
    ↳ Beam Search



→ Headline only used to filter out

→ before bidirectional LSTMs

not for compression

- LSTMs very very slow! ⇒ Have to be run at query time!

- Parsey McParseface: directed NN ~~moves~~ Feedforward from l-to-r
  ↳ moves through the tree and constructs subtree!
  ↳ make dependencies explicit! ⇒ previous decisions are feed into next network ⇒ feature engineering became more important!
  ↳ 100 times faster than 3 layer LSTM
  ↳ beam size (word window) matters!
  ⇒ global beam solution: cost fct. encodes penaliz. ⇒ any local decision that might destroy subtree

⇒ no metric

☑ SpeechToText:
→ Only work with top recommendation
Speech ⟷ Text

- Problem: Self-selection of answers by queries!
→ Look at feedback

Talk 5 – A. Karatzoglou – DL for RecSys 06/09

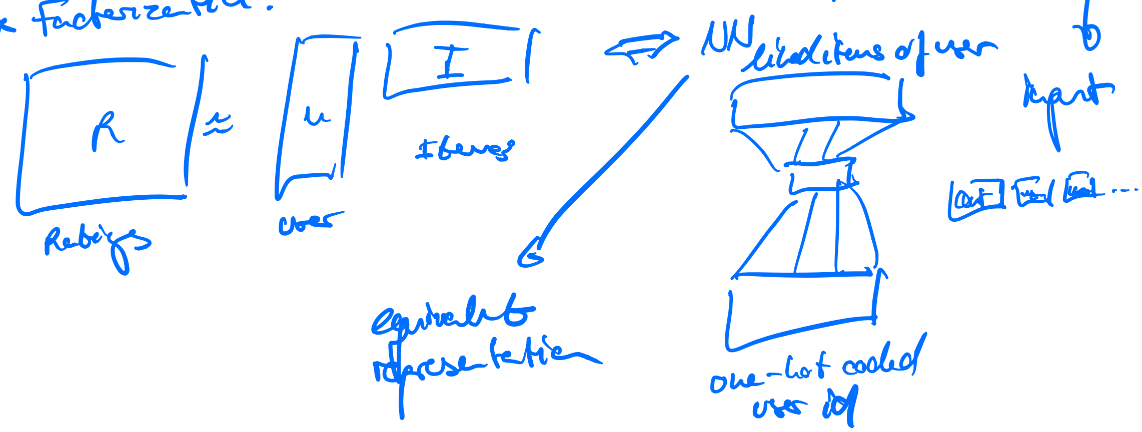
- Switchboard || Hub 5000 data ⇒ Speech to Text
- Backprop problem of vanishing gradients
  - ↳ sigmoid: vanishing gradients ⇒ slow learning
  - ↳ ReLU as solution
- Dropout: Ensembling ⇒ train different models and average
  - ↳ leads to robustness!
- DL RecSys:
  - * Learning Item Embedding ⇒ 2 Vec Models
  - * Deep CF
  - * Feature extraction from complex datasets / content
  - * Sequential

  content / talk
- CPU/GPU optimization is different!
  - ↳ moving data to memory is very expensive on GPUs
- Embeddings: learned real value vector representing an entity
  - ⇒ good for contrasting synonyms!
  - ⇒ Item 2 item Recommendations
    - ↳ input for more complicated methods

  skip gram
  use to work better!

  Paragraph2vec
  Doc2vec
  - ↳ keep context vector

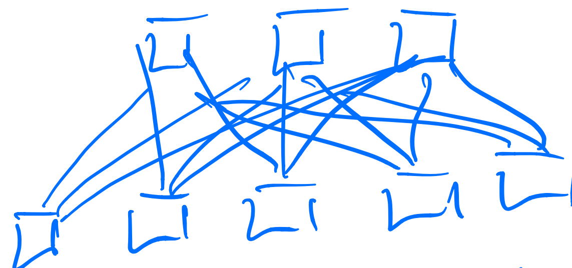
- Matrix Factorization:



- 2vec Models in RecSys
  → words become items in session/user profile
  → Grbovic et al (2015): prod2vec // Bagged prod2vec
  → Meta-Prod2Vec ⇒ recovery of meta data
  → User-Prod2Vec
- Don't do rating prediction: spend a lot of computing power on what the user is going to rate badly!
- Unclear why shallow nets perform better than deep ones!

- Deep Collaborative Filtering
  → Neural SVD

→ Restrictive Boltzmann Machines



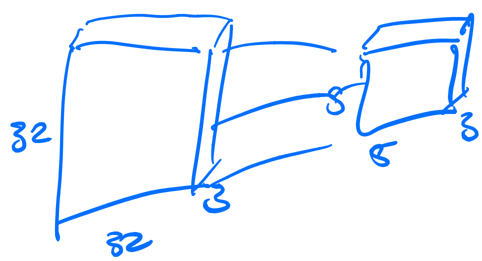
↳ forward and backward activation
⇔ sigmoid activation: prob. of firing an ⟷ gets recommendations by observing which items are also activated that user has not activated

Autoencoder
* different encoding and decoding weights
* denoising ⇒ corrupt input!
* Sparse Version ⇒ Combat that Autoencoder learns the identity function!

→ Bayesian Stacked denoising Autoencoder

- TDSSM => Temporal Deep Semantic Structured Model
  - -> similar to MV-DNN but dynamic over time!
    - ↳ time dependent part is modeled by an RNN!
- Coevolving features => Items and users develop dynamically
- Wide & DL (Cheng et al 2016)
  - -> Shallow model: simple recommendations
  - -> Deep model: more creative recommendations
- YouTube
  - -> Candidate Generation => Recs as binary classification
  - -> Ranking => Use more complicated/hard to compute features
- Feature Extraction from Content
  - -> Hybrid systems
  - -> CNNs

can solve coldstart problem



- filter moves over image with filter to obtain conv.
  - ↳ summing up after multiplication

- -> Oord et al (2013)
  - ↳ Music representations -> Treat image // spectrogram as input to obtain features / embed item in product space!

- Session-based Recs
→ RNNs: horizontally deep and not necessarily stacked
  ↳ Encoder-Decoder: Two RNNs for Translation
  ↳ $s_{t-1} = \sigma(W s_{t-1} + \dots)$
    ↳ $\lambda(W) < 1$: $\nabla \to 0$
      $\lambda(W) > 1$: $\nabla \to \infty$
  ↳ LSTM combats this problem automatically!
→ Vanishing gradient ⟺ solved by Deep Belief Nets

→ GRU4Rec

↳ GANs? ⇒ No applications for RecSys right now ⇒ Idea: similar to VAE - Generate samples and choose item that user has not chosen and is most similar to existing one
↳ multi-modal ⇒ fast eval? ↳ Yes, easily possible! 2-5 ms on GPU
↳ Deep Belief Nets
  ⇒ Made clever training with logistic sigmoids possible!
  ↳ Use Boltzmann Machines to initialize weights!

- Convs on Text ⇒ 1 dim. window actor

↳ CNNs work the best right now!
  ⇒ Transform almost everything into some form of image representation

Talk 6 - E. Gómez - Music IR

07/09

- Extract query description $\xleftrightarrow{\text{match}}$ music description

  → Low, mid, high specificity of query

  → small, large granularity ⇒ complete vs. time fragments

- Tasks:

  * Identification, fingerprinting
  * Alignment : audio - to - score (notes), audio - to - lyrics
  * Cover song identification
  * Recommendation
  * Play list generation
  * Visualization
  * query by label, humming, etc.

Biggest Conf.: ISMIR

→ Trends: * DL
* Real-world music
* Adaptation
  ↳ generalize MIR systems

- How to describe / compare music ?

  → semantic gap: difference between data extractions and human description

  → spectrograms

- History

  → early 2000s: Audio and Score ⇒ CONTENT

  → Mid 2000s: Web pages, images, tags ⇒ CONTEXT

  → Recently: Feature design via ML

  ↳ system-centric ⇒ user-centric

- Content Description
  - → Feature extraction
  - → Global vs. Local
    - ↳ Temporal scope
  - → High vs. Low
    - ↳ Abstraction level

Timbre — Melody

Rhythm — Structure

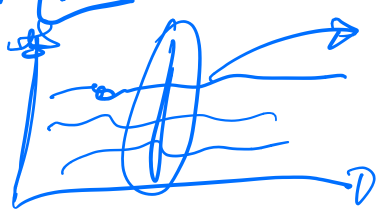
↳ all interrelated

[2d] – High Freq.

[20] – Low freq.

Anything outside can't really be processed → high freq. might be different

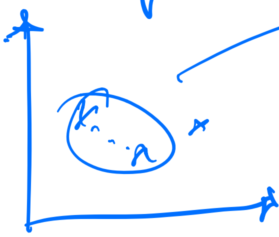
- Deep Learning ⇒ Use spectrogram as input to CNN
  - ↳ don't even need to preprocess by fourier transform / choice
  - ↳ Yes ⇒ But: a lot of room to handcraft filters



→ very rapid change: □ - square filter can only partially group this!

- Shazam Algorithm:

freq.



f.

→ peaks of energy in terms of spectrum

↳ very robust!

→ Match with database

↳ does not work for versions/remixes

- Pitch: Fundamental freq. ⇒ gives melody to the music
  - ↳ higher level
  - ↳ allows for retrieval via humming!

f_0

Only one! ⇒ allows identification of style!

- Onset detection => rhythm
- Tempo <=> Pitch <=> Rhythm
  - → descriptors work better for vocal and simple music → complexity of music is important
  - → music similarity depends on task!
    - ↳ low-level spectral descriptors
    - mid-level musical descriptors
    - semantic descriptors

- Context description
  - → web data, ratings, etc.
  - → similarity of artists => co-occurence on web page, etc.

- $\boxed{\text{Content}}$ vs. $\boxed{\text{Context}}$

Content:
- extraction from audio file
- no cultural bias
- requires audio
- hard to share data
- semantic gap => $\boxed{\text{HYBRID}}$ approaches
  - ↳ self-organizing maps => spectral ± acoustic distance

Context:
- captures special aspect
- mostly contextual

- Evaluation
  - → MIREX => some very hard tasks - e.g. emotion perception
  - → problem of copyrights!
  - → user-centered perspective → new measure:
    - emotional state change
    - novelty
    - aesthetics
  - → List vs. Visual search
    - List: more effective
    - Visual: more enjoyment

Talk 7 – S. Rueger – Visual Retrieval and Mining 07/09

- Web-based image searching
  → retrieving images // querying w/ images
  → multimedia IR

- Video Querying
  → Compression : iframe ⊕ dynamics model ⇒ slots → allows efficient storage
    ↳ Cut leads to new iframe!
  → Can decompress video info using iframes
    ↳ Way you input query is up to you!

content dependent & user informs of relevant features

- Multimedia IR // $\boxed{\text{near}}$-duplicate-detection
  → works well for 2d and near 2d (buildings, object)
  → does not work well for 3d (faces)

- Word spotting – OCR – find near-duplicates in hand-written documents

- Fingerprinting
  → salient / invariant points ⇒ rotation and scaling
    ↳ SIFT – Lowe (2015) – key point detector
  → Object Segmentation ⇒ fast RCNN → should work better!