Job Recommender Systems

Ass. Prof. Dr. Elisabeth Lex
Graz University of Technology

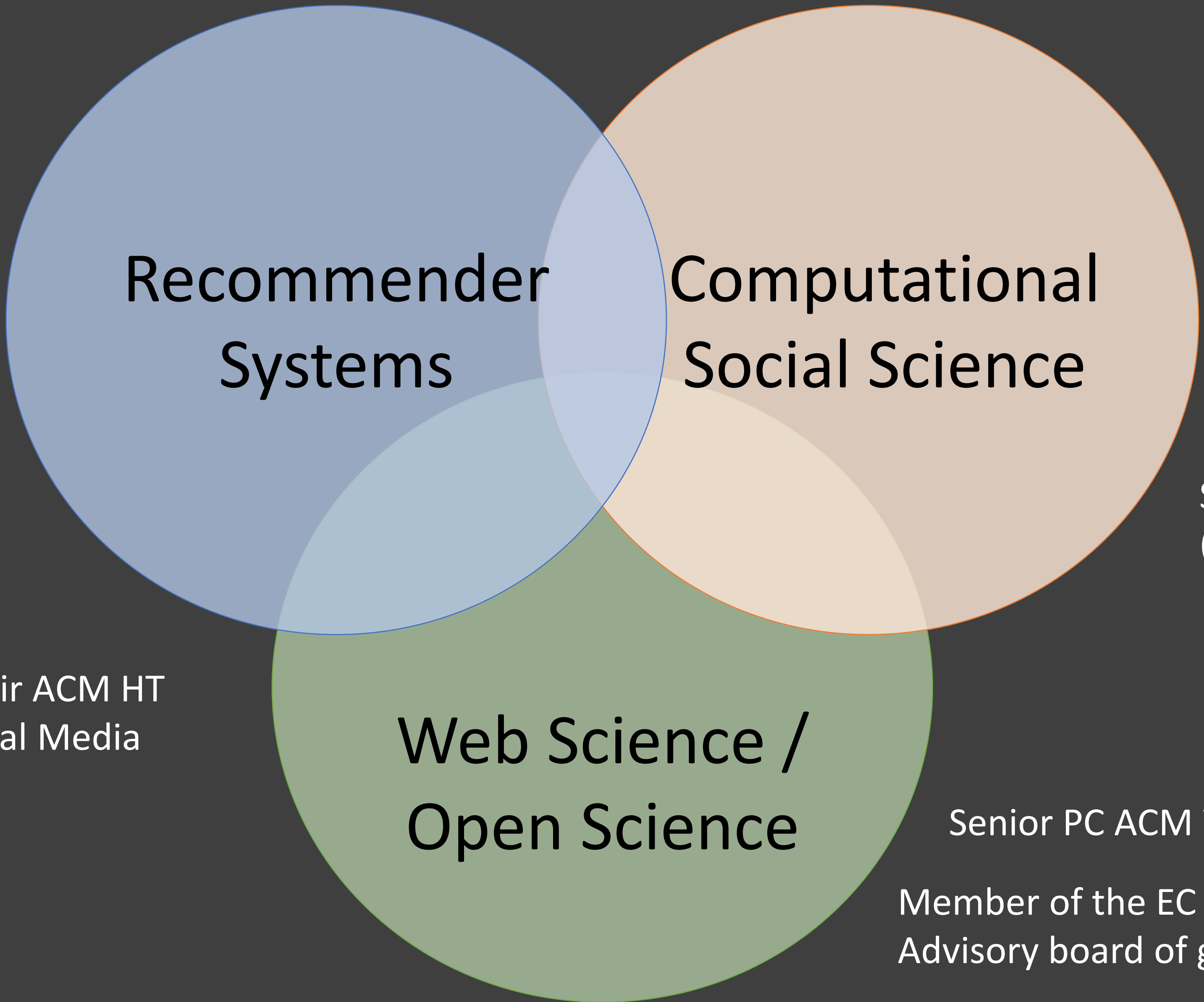
About me

- Assistant Prof @ TU Graz
- Head of Social Computing Lab @ISDS/KC
- PhD + habilitation in Applied Computer Science
- PostDocs at RWTH Aachen, TU Delft, UNSL Argentina
- Organizer of Computer Science Talks (#cstalks) at TU Graz (since ‘18)

I exploit digital behavioral data to create and improve personalized recommender systems and to computationally study (online) social phenomena

Research Fields

Senior PC ACM IUI (since '18)
Reviewer for major RecSys events & journals



Senior PC Social Informatics (since '19)

Track Chair ACM HT '20 – Social Media & RecSys

Senior PC ACM Web Science ('19)

Member of the EC Expert Group on Altmetrics
Advisory board of generationR (https://genr.eu)

This Talk

- Introduction to (Job) Recommender Systems
- Using Deep Learning in job RecSys
  - Creating job embeddings and use them in content-based recsys
  - Providing personalized job recommendations for anonymous users with autoencoders

Introduction

What are Recommender Systems?

- Recommendation systems (RS) help to match users with items
  - Ease information overload
  - Sales assistance (guidance, advisory, persuasion,…)

> *RS are software agents that elicit the interests and preferences of individual consumers […] and make recommendations accordingly.*
> *They have the potential to support and improve the quality of the decisions consumers make while searching for and selecting products online.*
> (Xiao & Benbasat 2007[1])

- Different system designs / paradigms
  - Based on availability of exploitable data
  - Implicit and explicit user feedback
  - Domain characteristics

Xiao and Benbasat, *E-commerce product recommendation agents: Use, characteristics, and impact*, MIS Quarterly 31 (2007), no. 1, 137–209

Purpose & Success of Recommender Systems

- Different perspectives/aspects
  - Depends on domain and purpose
  - No holistic evaluation scenario exists
- Retrieval perspective
  - Reduce search costs
  - Provide "correct" proposals
  - Users know in advance what they want
- Recommendation perspective
  - Serendipity – identify items from the Long Tail
  - Users did not know about existence

When does a RecSys perform well?

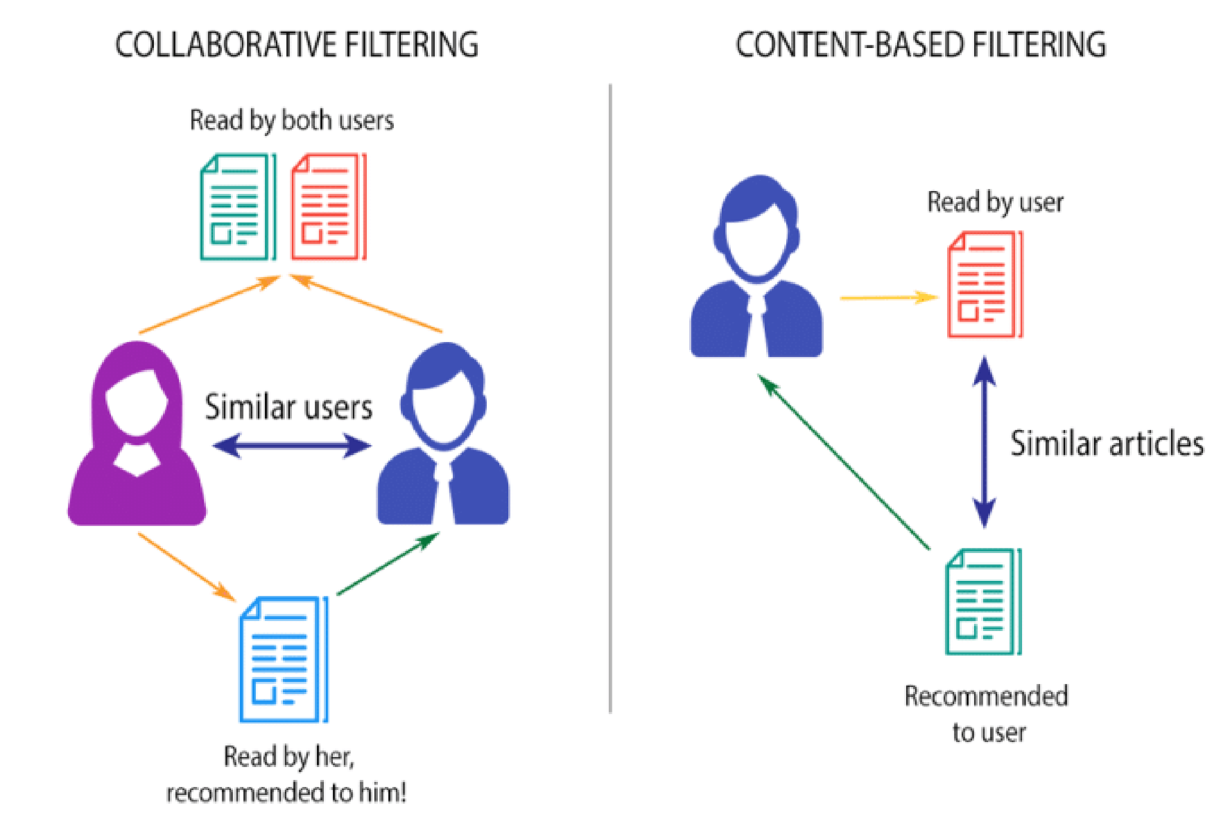
- Prediction perspective
  - Predict to what degree users like an item
  - Most popular evaluation scenario in research
- Interaction perspective
  - Give users a "good feeling"
  - Educate users about the product domain
  - Convince/persuade users - explain
- Conversion perspective
  - Commercial situations
  - Increase "hit", "clickthrough", "lookers to bookers" rates
  - Optimize sales margins and profit

How does it work?

- Given:
  - User model (e.g. ratings, preferences, demographics, situational context)
  - Items (with or without description of item characteristics)
- Find:
  - Relevance score. Used for ranking.

Basic Paradigms in RecSys

- Collaborative: "Tell me what's popular among my peers"
- Content-based: "Show me more of the same what I've liked“
- Hybrid: combinations of various inputs and/or composition of different mechanism



Job Recommender Systems

Why Job Recommender Systems?

- Abundant overload of job vacancies
- Dynamic labor market: need to support job mobility
- Emergence of business-oriented social networks (e.g. LinkedIn, Xing, CareerBuilder, Talto)

What makes Job RecSys challenging?

- Ephemeral nature of jobs
- User experience: need for
  - increased diversity
  - Explanations
  - exploration
- Multi-stakeholder problem
- Reciprocity: getting a job requires reciprocated interest
- “technical” challenges
  - Data sparsity, cold-start problem

Some Job RecSys Approaches

- Collaborative filtering (CF)
- Content-based filtering (CBF)
- Hybrids
- Searching for a suitable candidate for a job and recommending them that job
- Inference based on job transition patterns
- Sequence modeling
- …

Using Deep Learning to recommend jobs

Lacic, E., Kowald, D., Reiter-Haas, M., Slawicek, V., & Lex, E. (2017). Beyond accuracy optimization: On the value of item embeddings for student job recommendations. FUP Workshop @ ACM WSDM ’18, February 6, 2018, Los Angeles, USA

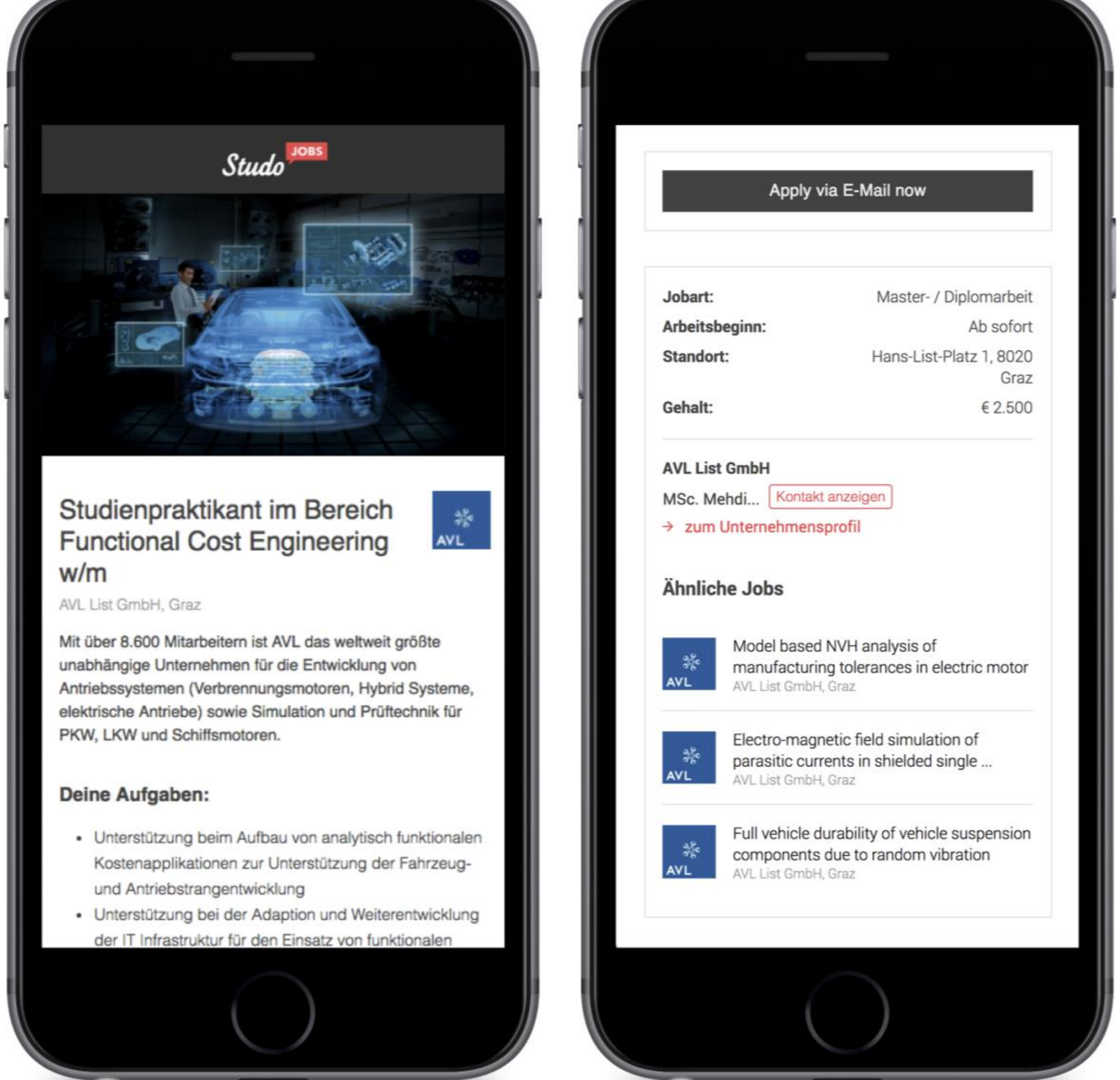
Setting the scene

Task: recommend jobs to university students

Aim: explore the impact of using item embeddings in a content-based job recsys

Scope

- Studo Jobs platform
  - Focus on student jobs
- Aim: content-based job recommender systems to recommend jobs similar to the currently viewed one
- Plus: improve recommendation diversity



Dataset

- Data types
  - Job postings: text & metadata
  - Users: demographics, also anonymous
  - Interactions: timestamp, user-item relation

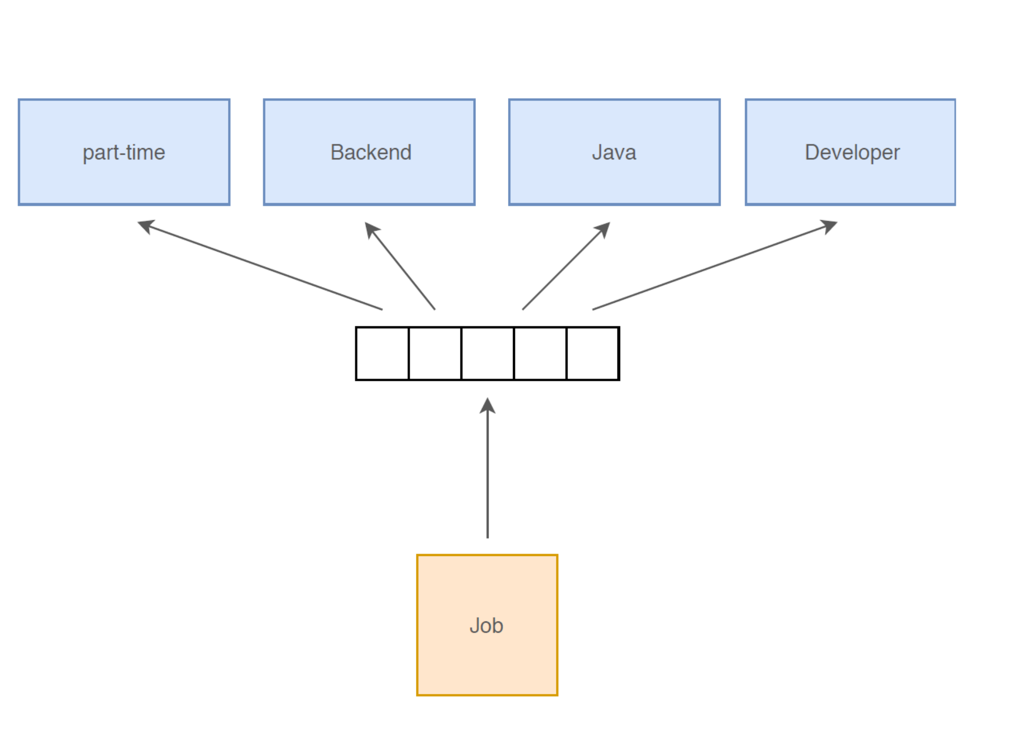
Approach

- Input: job posting that the user currently views
- Doc2Vec algorithm to obtain job vectors representing the key features of that job posting
  - Vector representations → “embeddings”: job description text as dense vector
  - Accounts for semantic & syntactic structure
- Use these job embeddings in a CBF recommendation scenario

Doc2Vec

- Extension of Word2vec
- Word2vec: word embedding
  - Type of representation that stores contextual information in low-dimensional vector
- Doc2vec: encodes entire documents
  - Doc2vec vectors represent theme / overall meaning of a document

→ Job embeddings



Quoc Le and Tomas Mikolov. "Distributed representations of sentences and documents". In: Proc. of ICML'14. 2014

Constructing the embeddings

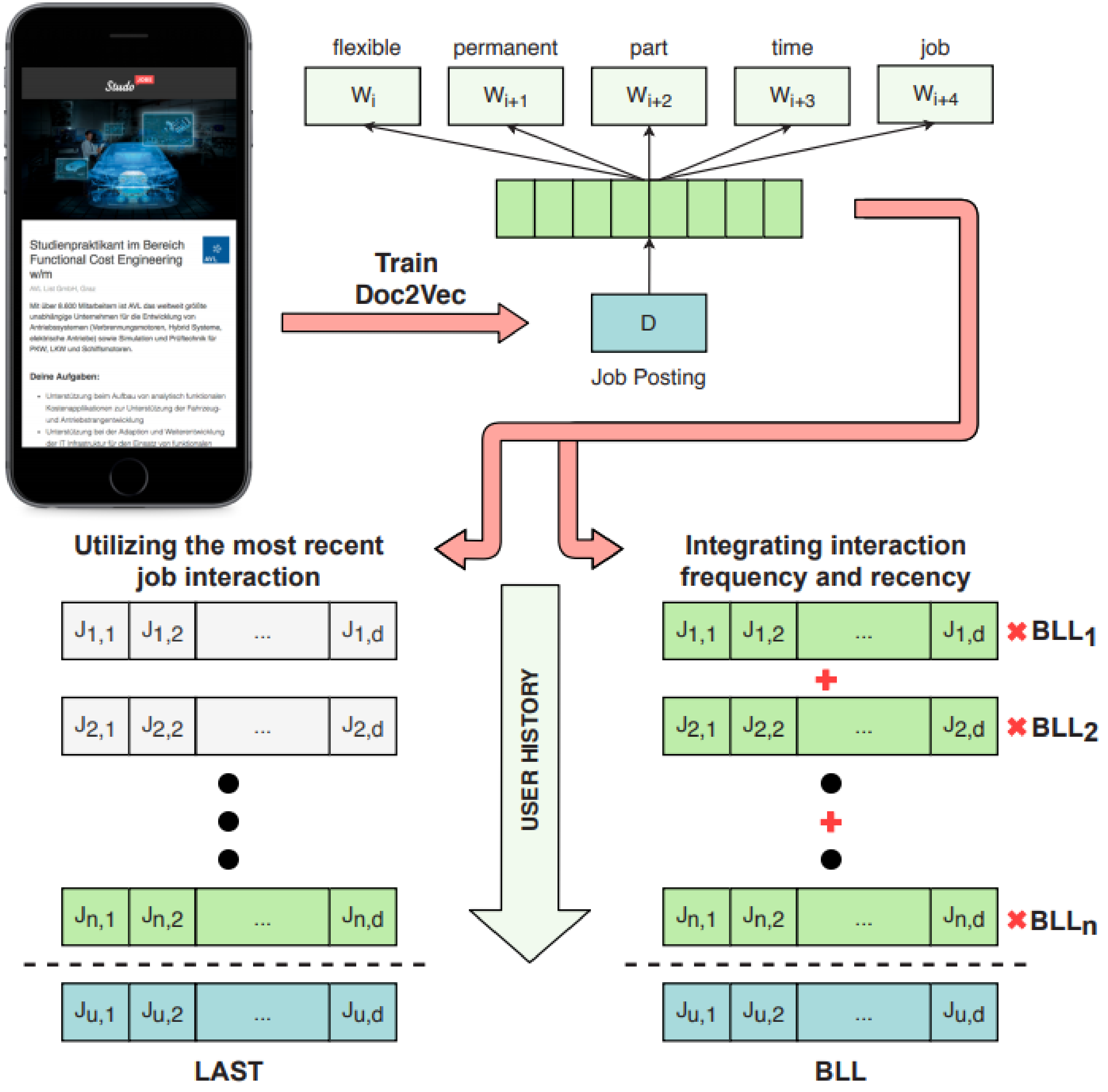
- 3 strategies:
  1. LAST: Vector representation of most recently interacted job posting of user
     → recency
  2. AVG: Vector representation of all jobs a user has interacted with – averaging column vectors
     → frequency
  3. BLL: Multiply job vectors from a user's history with BLL values given by

$$BLL_{u,j} = \ln\left(\sum_{i=1}^{n} (TS_{ref} - TS_{j,i})^{-d}\right)$$

     → Both recency & frequency

We exploit multiple information sources!

- Content: embeddings
- Time: BLL
- Similarity: hybrid CF



Evaluation:

- Dataset:
  - Extracted from Studo
  - ~3k users getting recommendations from 2.4k job postings; ~140k job views
- Baseline algorithms
  - Content-based filtering
  - Collaborative filtering
  - Most popular

Evaluation metrics

- Accuracy metrics
  - Precision, recall, nDCG

- Beyond-accuracy metrics
  - novelty, diversity, serendipity

$$Novelty@k = 1 - \frac{1}{k} \sum_{i \in k} \frac{\log_2(pop_i + 1)}{\log_2(pop_{MAX} + 1)}$$

$$Diversity@k = \frac{1}{k \cdot (k-1)} \sum_{i \in R} \sum_{j \in u^k, j \neq i} d(i,j)$$

$$Serendipity@k = \frac{1}{|R_k| * |H_s|} \sum_{i \in R_k} \sum_{j \in H_s} d(i,j)$$

Results

- MP, CF good accuracy, diversity
- CBF, LAST: good novelty
- LAST: ok diversity

Approach			k = 3				k = 6			
			Novelty*	Novelty	Diversity	nDCG	Novelty*	Novelty	Diversity	nDCG
MP			.5849	.1649	.7261	.0395	.6057	.1857	.7156	.0722
CBF			.8124	.7676	.4536	.0122	.7965	.7835	.4854	.0156
CF			.7718	.3518	.6736	.0889	.7860	.3660	.6814	.1292
Doc2Vec	LAST	d=100	.8331	.7469	.4845	.0170	.8161	.7639	.5486	.0217
		d=200	.8411	.7389	.5091	.0182	.8212	.7588	.5854	.0219
		d=300	.8448	.7352	.5163	.0177	.8206	.7594	.5953	.0220
	AVG	d=100	.8275	.7525	.6929	.0107	.8144	.7656	.7239	.0154
		d=200	.7930	.7870	.7455	.0099	.8050	.7750	.7830	.0135
		d=300	.7715	.8085	.7439	.0091	.8003	.7797	.7796	.0133
	BLL	d=100	.8500	.7300	.5974	.0156	.8322	.7478	.6486	.0198
		d=200	.8284	.7516	.6408	.0146	.8275	.7525	.7006	.0188
		d=300	.8191	.7609	.6388	.0144	.8222	.7578	.7015	.0186
CF + BLL			.8731	.4531	.6820	.0721	.9578	.5378	.6890	.0900

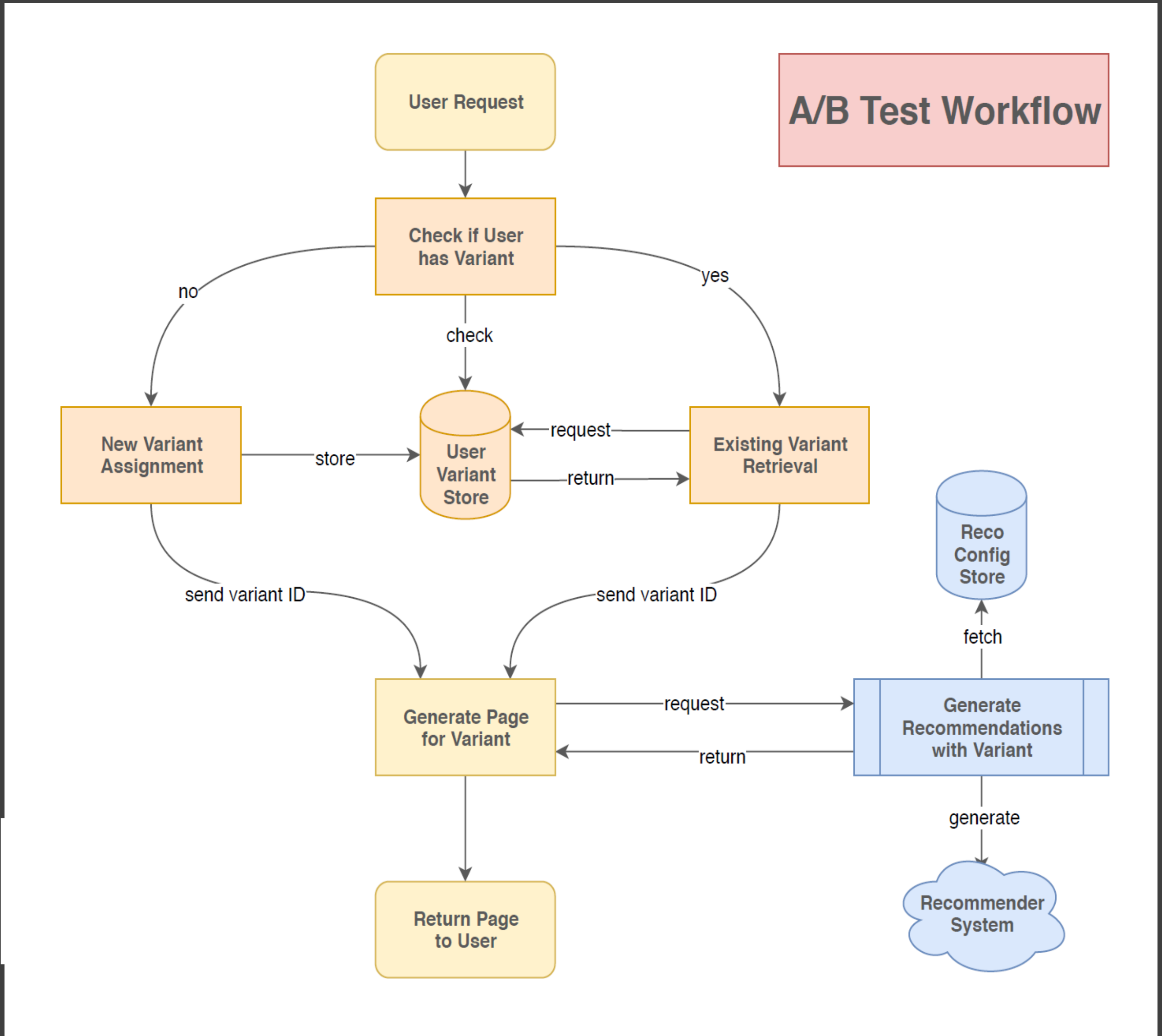
Are the recommendations useful?

Lacic, E., Reiter-Haas, M., Duricic, T., Slawicek, V., & Lex, E. (2019). Should we embed?: a study on the online performance of utilizing embeddings for real-time job recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems* (pp. 496-500). ACM.

Online Evaluation

- A/B tests in Studo
- Click-through-rate (CTR)

$$CTR = \frac{\#actions}{\#impressions}$$



Study setup

- Online study in Studo Jobs platform (now Talto.com)

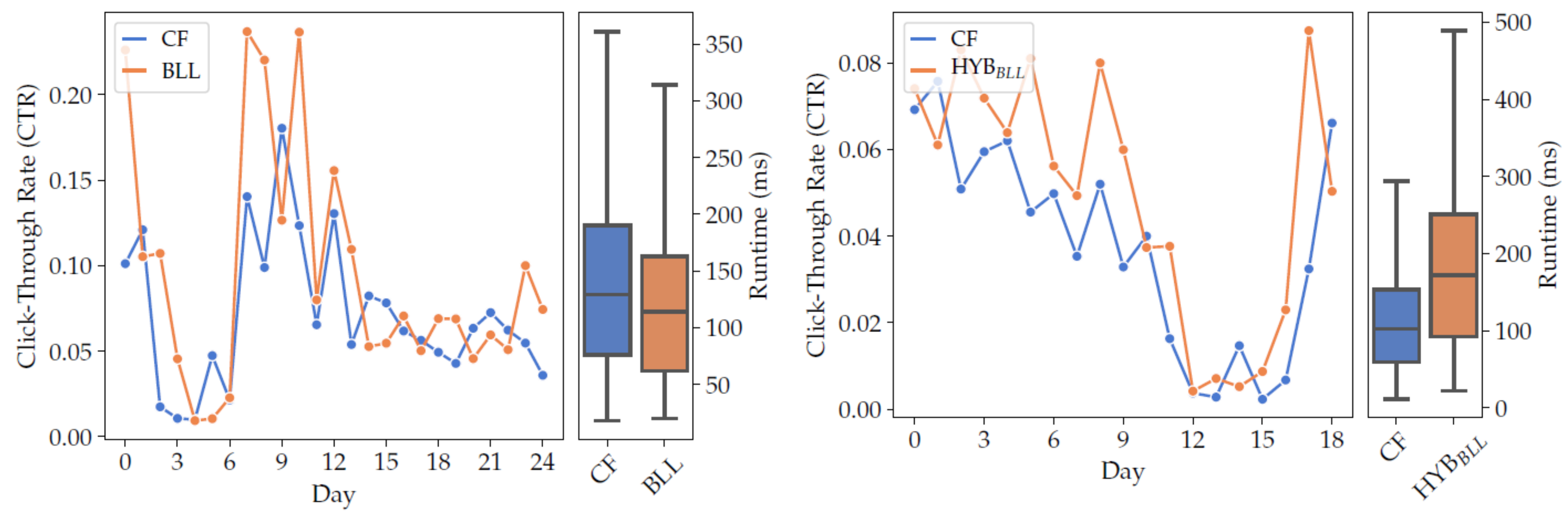
- CTR: percentage of recommended job postings a user has clicked
- Split user base in two equally sized groups

Results

	Test	Distinct Users	Reco Requests	Days	Approach	CTR	↗	Runtime (ms)	↘
Similar Jobs	Impact of embeddings	8,576	31,968	32	CBF	0.0194	18.04%	51	23.53%
					LAST	**0.0229***		**39****	
	Influence of frequency and recency	4,715	18,464	15	LAST	**0.0249****	75.35%	**67****	28.72%
					BLL	0.0142		94	
	Merit of recency	3,375	11,992	15	$BLL_{d=0.6}$	**0.0174***	35.94%	97	2.06%
					$BLL_{d=0.4}$	0.0128		**95**	
Homepage	Influence of frequency and recency	9,620	26,334	25	BLL	**0.0671***	15.69%	**114****	13.64%
					CF	0.0580		132	
	Combining frequency and recency	9,313	24,907	19	HYB_{BLL}	**0.0471****	33.05%	172	38.37%
					CF	0.0354		**106****	

Embeddings based on most recent interaction (LAST) have best online performance when recommending similar jobs

Providing recommendations on a user's homepage in Studo



(a) Influence of frequency and recency **(b) Combining frequency and recency**

BLL Leads to **higher user engagement**

Findings

- Embeddings → effective representation of job postings

- Recency of interaction is a crucial factor
- Same as recency & frequency of interaction

→ Also verified in online study

How can we provide personalization if we do not have any information about the user?

Lacic, E., Reiter-Haas, M., Kowald, D., Dareddy, M. R., Cho, J., & Lex, E. (2020). Using autoencoders for session-based job recommendations. *User Modeling and User-Adapted Interaction*, *30*(4), 617-658.

Session-based Recommendation

- Input: sequentially ordered log of user interactions
  - E.g item views, listening events,…
- In many cases:
  - user cannot be identified (first-time users, users not logged in)
  - no longer-term preference information is available
  - user interest/intention/preferences must be assessed from a small set of interactions

Very common in real-world scenarios!

Operationalizing the Problem

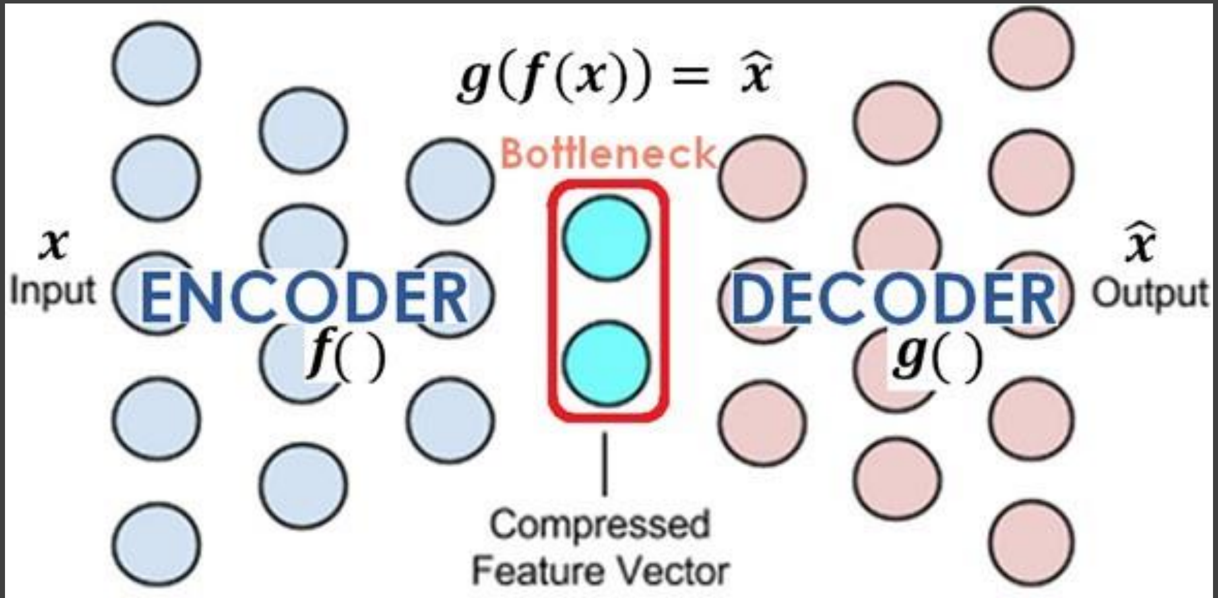
- Background
  - User intention is not known – e.g. shall we recommend more similar items?
- Computational task
  - Predict subsequent user actions, given
    - The last N actions the user has taken (e.g. in the current session)
    - Other types of information (e.g. metadata, community behavior)
- Evaluation
  - Standard IP metrics (precision, recall, …)
  - Use publicly available log datasets (ideally)

Quick Overview of Selection of Approaches

- Item co-occurrences in individual sessions
  - “users who bought”
- Simple Markov Chains
  - Count how often item appeared after others in the training data
- Session-based nearest neighbors
  - Find similar previous sessions
  - Weighted prediction scheme
  - Choose similarity function
- Deep Learning
  - Recurrent Neural Networks – a natural choice for the problem
    - SOTA: GRU4Rec (Hidasi et al.), based on gated recurrent units

Our Approach: Neural Autoencoders

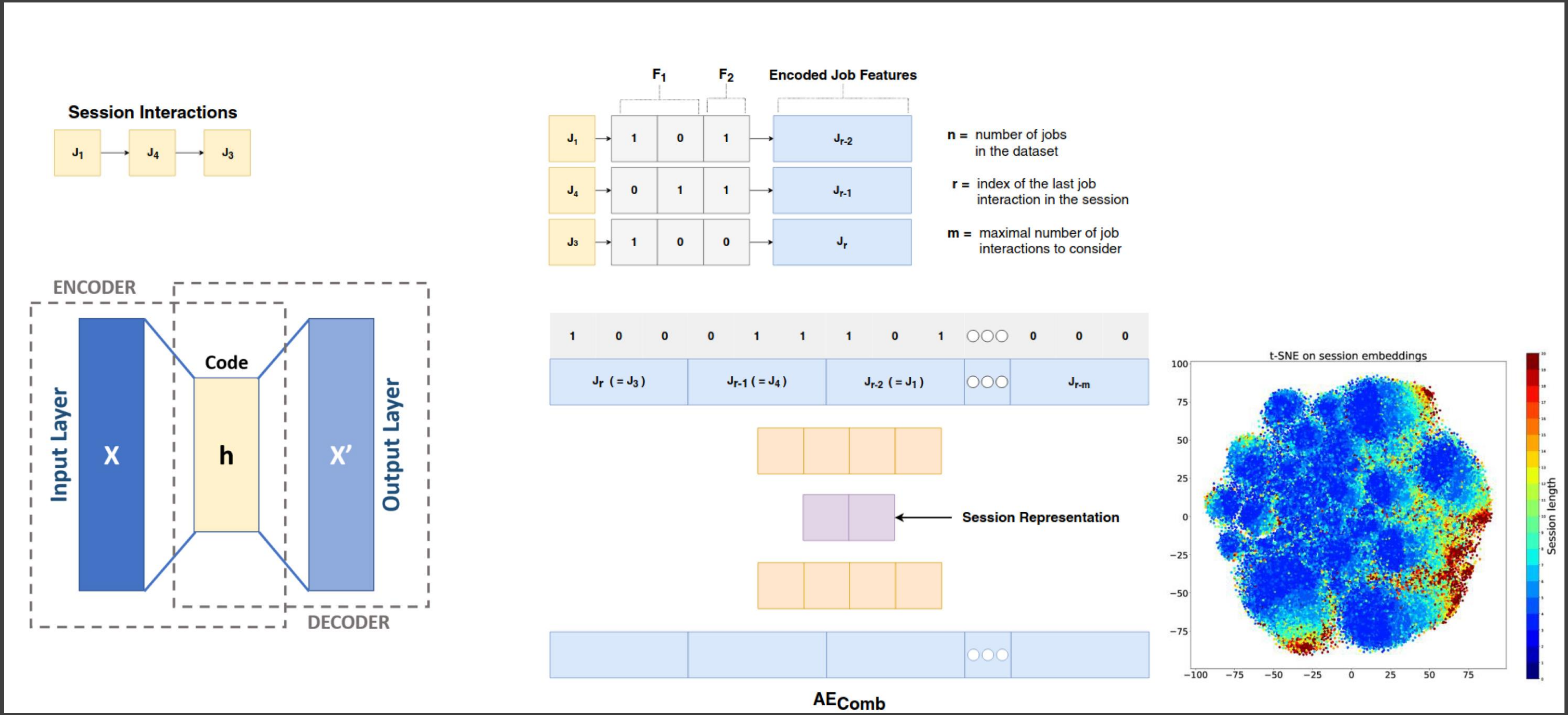
- Dimensionality reduction technique
- Encoder f(): learns transformations that compress input to shrink via bottleneck
- Decoder g(): reconstructs input



Approach

- Different autoencoder architectures (AE, VAE, DAE)
  - Use them to infer latent session representations
  - Representations used in k-nn manner to recommend jobs within a session
- 2 types of input
  - Interactions with job postings within a session
  - Interactions in combination with content features extracted from job posts
- Evaluation with beyond-accuracy metrics
  - Diversity, serendipity, novelty

Latent session representations w/ autoencoders



Session-based Job Recommender System

- Predict next clicked item based on similar sessions
- Optimize for beyond accuracy metrics
  - Diversity
  - Serendipity
  - Coverage

	Accuracy	Beyond Accuracy	Coverage
VAE_{Int}	++	++	++
VAE_{Comb}	+	++	++
sKNN	+	o	+
V-sKNN	++	+	++
S-sKNN	++	+	+
GRU4Rec	++	+	+
pRNN	- -	- -	- -
Bayes	- -	- -	o
iKNN	o	-	+
BPR-MF	-	- -	++
POP	- -	- -	- -

Lacic, E., Reiter-Haas, M., Kowald, D., Dareddy, M. R., Cho, J., & Lex, E. (2020). **Using autoencoders for session-based job recommendations**. User Modeling and User-Adapted Interaction, 30(4), 617-658.

Implemented in Live System talto.com

Modeberater (m/w/d)

Kastner & Öhler Mode GmbH, Graz

Wir zählen zu den führenden Modeanbietern in Österreich und suchen für unser Haupthaus in Graz Modeberater/innen Vollzeit, Teilzeit, Geringfügig samstags.

Jetzt online bewerben

Treffen folgende Punkte auf Sie zu?

- Modeaffinität
- Berufserfahrung im Verkauf sowie Kundenkontakt
- Flexibilität
- Höchste Kundenorientierung und Freundlichkeit
- Freude an der Arbeit mit Menschen
- Teamfähigkeit
- Ausgezeichnete kommunikative Fähigkeiten

Ja? Dann sind Sie bei uns richtig!

Unser Angebot

- Interne Verkaufsschulungen unterstützen Sie in Ihrer beruflichen und persönlichen Weiterentwicklung
- Einen zukunftsorientierten Arbeitsplatz in einem führenden Modeunternehmen
- Verantwortungsvolle Tätigkeit in einem spannenden und herausfordernden Aufgabengebiet
- Für diese Position gilt ein Jahresbruttogehalt ab € 22.900,- (Kollektivvertrag für

Jobart:	Nebenjob, Teilzeit
Arbeitsbeginn:	Ab sofort
Standort:	Graz
Gehalt:	ab 22.900 € Jahresbruttogehalt auf Vollzeitbasis
Berufsfelder:	Verkauf, Kundenbetreuung

Kastner & Öhler Mode GmbH

Anita Juri... Kontakt anzeigen

→ zum Unternehmensprofil

Weitere Jobs

B2B-Community Manager (m/w/d)
Studo GmbH, Graz

DekorateurIn
Kastner & Öhler Mode GmbH, Graz

Verkäuferinnen und Verkäufer für unsere Grazer Filialen
MARTIN AUER GMBH, Graz

https://talto.com/jobs/

Summary

- Job RecSys challenging problem
- Embeddings are effective tools to represent jobs
- Recency and frequency of interactions with job postings important
- Personalization for anonymous browsers possible
- Autoencoders well suited to improve beyond accuracy performance
- “Simple” neighborhood-based methods also highly effective

Thank you for your attention

Happy to answer your questions!

elisabeth.lex@tugraz.at
http://www.elisabethlex.info
follow me: @elisab79
follow my lab: @socialcomplab