# Matching Experts to Questions: A Comparison of Recommender Systems

Robin Stenzel[1], Max Lübbering[1], Bilge Ulusay[1], Daniel Uedelhoven[1] and Rafet Sifa[1]

[1]Fraunhofer IAIS, Germany

**Abstract**

Community question answering platforms like Stackoverflow are among the most popular interactive environments on the Internet for individuals to share knowledge. Finding experts to answer questions is one of those platforms' major challenges. To this end, we compare SBERT-Rec and LDA-Rec, two recommender system algorithms which are based on the state-of-the-art transformer architecture and well-established probabilistic topic modeling algorithm Latent Dirichlet Allocation, respectively. Our results show that SBERT-Rec significantly outperforms LDA-Rec in terms of average rank score. While SBERT-Rec excels in an open-world scenario with no presumptions about the underlying subjects of the corpus, LDA-Rec carves out distinct and human interpretable topics inside a niche closed-world corpus. Finally, we provide a novel metric for expert matching evaluation that supports partial experts/non-experts annotations.

**Keywords**

Recommender systems, Expert Finding, Community Question Answering, Latent Dirichlet Allocation, BERT

## 1. Introduction

With free online encyclopedias like Wikipedia and search engines, internet users can access any bit of information, regardless of time or location. From the start of the Internet, people used this new freedom to find like-minded people to share knowledge and ideas. Community-driven question and answer websites like Stackoverflow have emerged in recent years, matching questioners with domain experts. Many such niche platforms exist providing a question-and-answer system with an embedded voting mechanism [1].

Since these websites are entirely community-regulated, there is no inherent distinction between experts in specific fields allowing anyone to answer. Thus, users are equally presented with new questions covering a wide range of topics regardless of personal expertise. This raises the central question of this work: Can we match questions to experts solely based on a platform's historical question and answer data without any true expert annotations?

LWDA'22: Lernen, Wissen, Daten, Analysen. October 05–07, 2022, Hildesheim, Germany

robin.stenzel@iais.fraunhofer.de (R. Stenzel); max.luebbering@iais.fraunhofer.de (M. Lübbering); bilge.ulusay@iais.fraunhofer.de (B. Ulusay); daniel.uedelhoven@iais.fraunhofer.de (D. Uedelhoven); rafet.sifa@iais.fraunhofer.de (R. Sifa)

E-commerce websites are faced with a similar problem to increase the conversion rate and leverage recommender systems to suggest products to users matching their interests. Amazon and Netflix are two well-known examples that recommend products and movies based on a user's search or purchase history [2, 3].

This paper examines two different recommender systems (RS) for question-to-expert matching: 1) Topic modeling based RS and 2) transformer-embedding based RS. While the former approach yields interpretable closed-world topics, the latter's embeddings are more expressive due to the incorporation of world knowledge, thus leading to superior recommendation performance. Finally, we propose a new metric for recommender systems that supports partially annotated data.

## 2. Related Work

We look at the existing research from three perspectives: recommender systems, statistical topic models, and expert finding.

### 2.1. Expert Finding and Statistical Topic Models

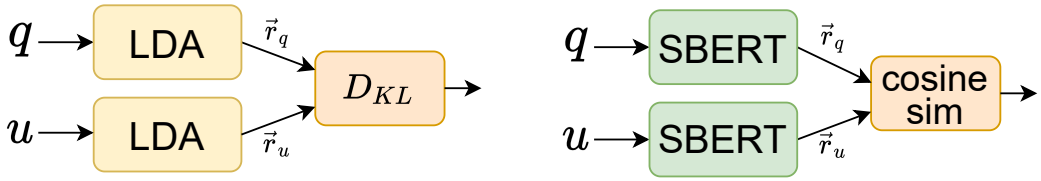
Popular online knowledge-sharing communities like Quora and Stackoverflow have become well-known platforms for finding people with specific knowledge in academic and non-academic fields. The goal of these platforms is to present the most relevant experts to a user searching for a term. LDA topic modeling [4] is one of the methods to achieve this goal. Using LDA, the relevant topics from a corpus are identified and used to establish an association among each user and their field of experts [5]. LDA topic modeling is widely used in expert community question answering [6], [7], [8], expert identification [9], and matching companies to news articles [10].

### 2.2. Recommender Systems

A recommender system is a crucial component of e-commerce, marketing, and social media platforms that predicts what consumers find interesting. Moreover, recommender systems are increasingly used in expert finding [11] and, more specifically, in finding experts in software development for design decision making [12].

A typical recommender system runs on one of the three fundamental engines: content-based systems, hybrid filtering, and collaborative filtering-based systems. Content-based filtering [13] recommends products or services based on item similarity and previous online activity. This filter avoids a cold start for new items by not relying on other users' comments [14],[15]. Unlike content-based filtering, collaborative filtering [16] suggests people based on shared interests. Lastly, hybrid filtering methods combine these two approaches [17].

By using natural language processing techniques such as BERT [18], recommender systems can provide more relevant suggestions to users. Several studies have used BERT in collaborative filtering [19] and other recommender systems, significantly improving recommendations [20],[21]. Furthermore, SBERT [22], a more computational efficient




(a) LDA-Rec (b) SBERT-Rec

**Figure 1:** Proposed recommender system architectures LDA-Rec and SBERT-Rec: A user/question pair is vectorized in terms of LDA's topic space or SBERT's embeddings space. The resulting question and user representation are then compared via Kullback-Leibler divergence $D_{KL}$ within LDA-Rec and cosine similarity within SBERT-Rec.

variant of BERT optimized for sentence similarity estimation, has been used to match jobs and job seekers [23].

## 3. Approach

We propose two different recommender system approaches, namely LDA-Rec and SBERT-Rec, as shown in Fig. 1. LDA-Rec utilizes LDA's probabilistic topic modeling to represent questions and users within the topic space. The dissimilarity of question and user pairs is estimated via Kullback-Leibler divergence. The SBERT-Rec recommender system comprises a SBERT model for question/user representation and a subsequent cosine similarity estimation module. LDA-Rec differs fundamentally from SBERT-Rec in the way the representations are estimated. The LDA model makes a closed-world assumption learning granular topics within a corpus, whereas the SBERT model incorporates world knowledge, making it less focused on a particular subject. Further, LDA provides interpretable corpus-inherent topics, providing valuable insights into the subjects discussed on these platforms.

### 3.1. Latent Dirichlet Allocation

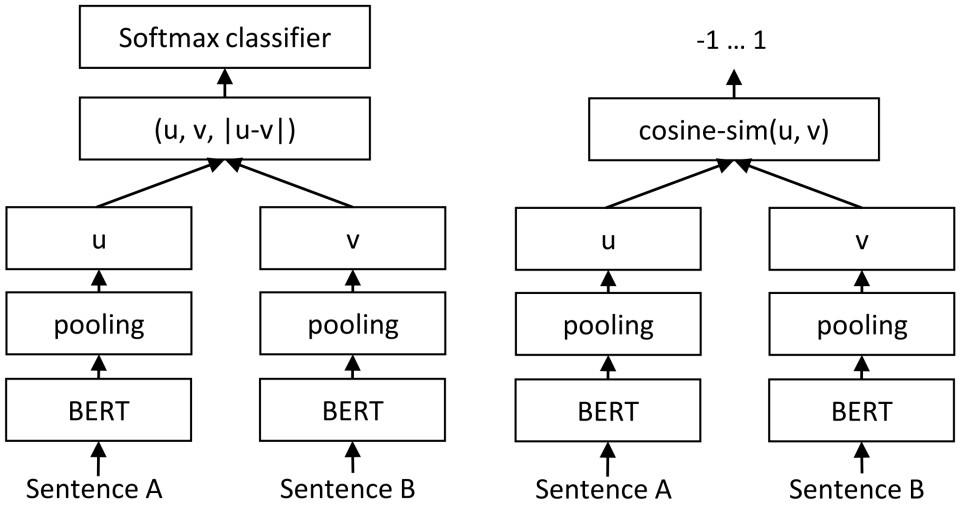
Latent Dirichlet Allocation (LDA) is a generative probabilistic model for topic modeling [4]. It is a hierarchical Bayesian model that estimates the probability distributions of topics appearing in a document and of words associated with those topics. For each topic the model iteratively ranks the vocabulary to maximize their likelihood using variational methods and the EM algorithm [4].

### 3.2. Sentence BERT

Sentence BERT (SBERT) is a fine-tuned BERT model optimized for text similarity estimation. As shown by [24, 22], the original BERT architecture provides state-of-the-art results on semantic textual similarity (STL). This setup, however, requires both sentences as input for similarity estimation, leading to considerable computational

| | Mathematics | Computer Science |
|---|---|---|
| Users | 84,896 | 245,701 |
| Questions | 1,371,686 | 1,806,605 |
| Answers | 1,770,606 | 3,891,942 |

**Table 1**
Number of questions, answers, and active users in the mathematics and computer science datasets.




(a) Training (b) Inference

**Figure 2:** Fine-tuning of BERT via siamese network setup [26]: Two sentences are forwarded through the BERT model with tied weights yielding BERT embeddings *u* and *v*. The network is fine-tuned via cross-entropy loss using the softmax outputs. At inference time, sentence similarity is computed via cosine similarity of embeddings *u* and *v*. Illustrations taken from [25].

inefficiencies. [25] points out that determining the most similar sentence pairs out of 10,000 sentences requires 50,000,000 inferences. This computational overhead also renders our recommender system infeasible since our objective is to match questions to potential experts from a set of more than 10,000 users.

SBERT provides a solution to this problem in two ways as illustrated in Fig. 2: a) The embeddings are fine-tuned in a Siamese network setup, which yields embeddings better capturing a sentence's semantics. b) The cosine similarity measure is decoupled from the network itself. The two adaptations allow computing the embeddings only once and then calculating the sentence similarity offline. The authors of [25] has shown that this more efficient setup provides state-of-the-art results, making it a superior approach.

| Approach | Dataset | average rank |
|---|---|---|
| LDA-Rec | Math | 76.1 |
| SBERT-Rec | Math | 83.8 |
| LDA-Rec | CS | 77.5 |
| SBERT-Rec | CS | 90.5 |

**Table 2**
Average rank scores of the LDA and SBERT recommender models on the math and computer science datasets

# 4. Experiments

## 4.1. Evaluation Approach

While the two datasets represent a typical recommender system setting for expert matching, they impose complicated challenges on recommender system evaluation. This is because only a tiny fraction of experts respond to a question and the remaining unknown experts are indistinguishable from non-experts.

Unknown experts can be expected to achieve high similarity scores, rendering recommender metrics such as precision@k and recall@k with their constant $k$ infeasible. Similarly, the order-aware mean reciprocal rank metric only considers the top-ranked true expert, which greatly depends on the unknown ratio of true experts vs. unknown experts, and disregards the remaining true experts.

To this end, we propose the average rank metric

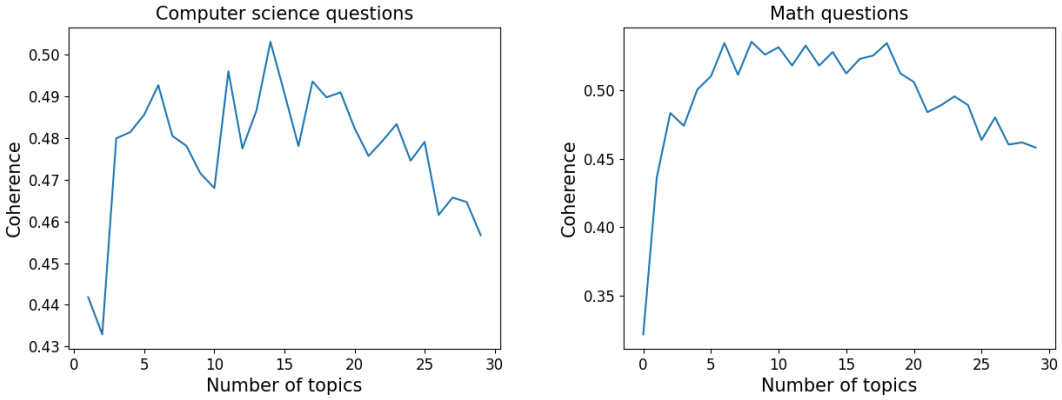
$$s(q,e,u) = \begin{cases} 1, & \text{if } \mathrm{sim}(q,e) > \mathrm{sim}(q,u) \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

$$\text{rel rank}(q,e) = \frac{1}{|U \backslash \{e\}|} \sum_{u \in U \backslash \{e\}} s(q,e,u) \tag{2}$$

$$\text{average rank}(q) = \frac{1}{|E_q|} \sum_{e \in E_q} \text{rel rank}(q,e), \tag{3}$$

which averages the relative rank of every true expert $e \in E_q$ of question $q$ among all remaining users $u \in U \backslash \{e\}$ based on similarity function $\mathrm{sim}(q,u)$, irrespective of true experts, non-experts, and unknown experts. Analogous to AUROC in outlier detection setting [27, 28], this metric can be interpreted as the probability of a random true expert being ranked higher than a randomly sampled user. As a result, the average rank metric is independent of dataset size and considers all true experts within the evaluation.

LDA-Rec and SBERT-Rec are evaluated with respect to their relative rank over all questions in a dataset. In case of KBL divergence, we compute its inverse so a higher value corresponds to a higher similarity.




(a) CS (b) Math

**Figure 3:** LDA topic coherence with varying number of topics.

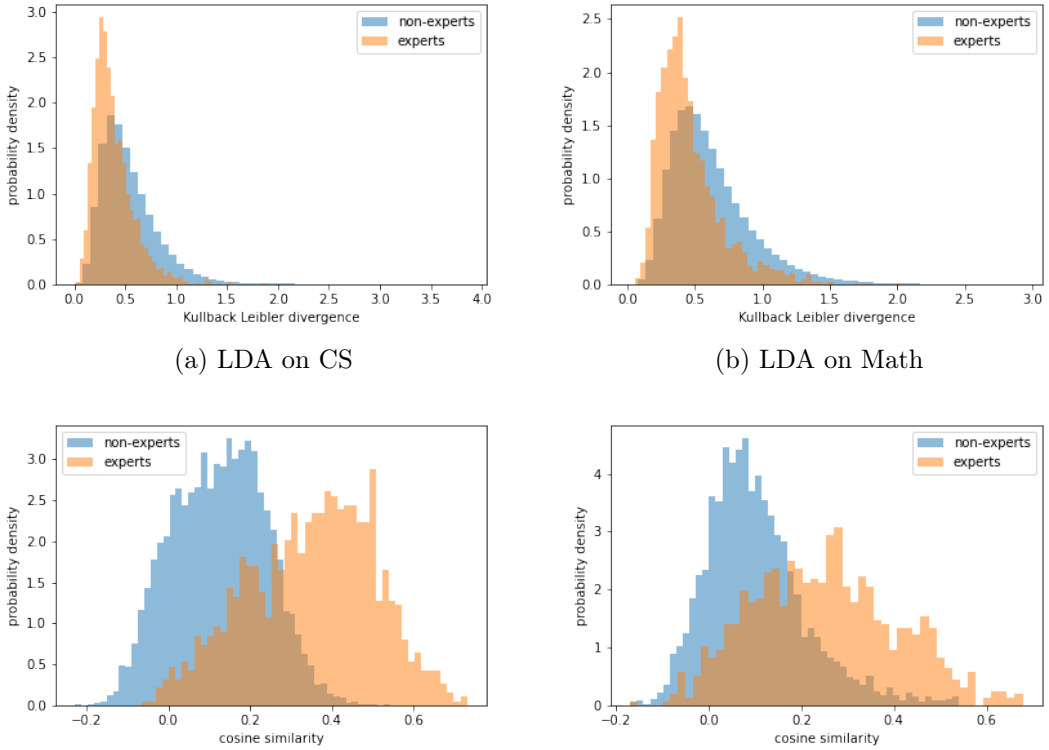
### 4.2. Datasets

We trained and evaluated our models on two community question answering datasets. The first one covers mathematics and contains questions and answers from Math Stack Exchange[1], while the second covers computer science and software engineering from Stack Overflow[2].The Internet Archive[3] has a data dump from both sites' posts. The datasets are large-scale, well-known, and contain a wide range of topics, questions, and users. They are an excellent fit for our research question since many learning platforms are designed similarly.

We applied multiple preprocessing steps to the datasets. First, we created questions by concatenating the titles with the corresponding main texts and removed users with less than 50 answers to assure expert representations of high quality. Then we split each question's text into its constituent words while converting each character into lower case and applying lemmatization. Moreover, we filtered out special characters, stop words, and one-letter words.

The key idea is to represent a user by all the questions he answered and measure its similarity to the representation of the question at hand. We assume a user to be a true expert for all questions that he answered. Likewise, we consider a user that answered similar questions before but not this one as an unknown expert and everyone else as non-experts. Note that unknown experts are indistinguishable from non-experts within the dataset.

Table 1 shows the final number of questions, answers, and users in each dataset.




(a) LDA on CS

(b) LDA on Math

(c) SBERT on CS

(d) SBERT on Math

**Figure 4:** Density histograms of question/expert and question/non-expert similarity measured by $D_{KL}$ for SBERT and cosine similarity for LDA. Note that $D_{KL}$ measures dissimilarities causing the histograms to be flipped.

### 4.3. Results

The LDA model has been trained for 500 iterations and 100 passes. For the Math and CS datasets, we determined the number of topics to be 12 and 14, respectively, as supported by the topic coherence development, shown in Fig. 3. For the SBERT-Rec, we used the pre-trained model without further fine-tuning.

As shown in Table 2, SBERT-Rec significantly outperforms LDA-Rec on both Math and CS datasets by 7 and 13 percentage points, respectively. Further, LDA only improves slightly on the CS dataset compared to the Math dataset, whereas SBERT's enhances 7 percentage points. As a topic in CS can be arbitrary, e.g., due to the advent of new technologies every year, it can be assumed that the CS dataset comprises more distinct topics than the Math dataset. This is in line with the increase in average rank scores on the CS dataset for both models.

[1]https://math.stackexchange.com, Math Stack Exchange
[2]https://stackoverflow.com, Stack Overflow
[3]https://archive.org/details/stackexchange, The Internet Archive

| Id | Name | Top four words |
|---|---|---|
| 1 | General request | use, like, would, want |
| 2 | Code | new, class, string, public |
| 3 | Backend | application, server, run, error |
| 4 | File handling | file, c, x, b |
| 5 | User input | name, value, type, form |
| 6 | Java script | function, page, text, html |
| 7 | Databases | table, query, database, id |
| 8 | Websites | http, com, xml, url |
| 9 | Authentification | user, view, self, model |
| 10 | Java webservice | java, org, service, web |
| 11 | HTML | div, td, px, width |
| 12 | Android development | android, layout, parent, id |
| 13 | .net web service | event, asp, control, net |
| 14 | Images | list, image, item, li |

**Table 3**
Learned computer science topics represented by the four words having highest topic importance

Additionally to the average rank scores, we plotted the density histograms of expert/question similarities and non-expert/question similarities, as displayed in Fig. 4. On both datasets, the similarity histograms are better separated, compliant with SBERT-Rec's higher average rank scores.

While SBERT-Rec can be regarded as a black-box model, each topic learned by LDA is the vocabulary ranked by topic relevancy and thus interpretable. As shown in Table 3 and Table 4, the set of learned topics within CS is more distinct than the topics within the Math dataset, explaining the superior scores on the CS dataset. Nevertheless, for LDA-Rec, the average rank score difference between the two datasets is less significant in comparison to SBERT-Rec. This pinpoints LDA's advantage as a closed-set topic modeling algorithm to learn niche corpora.

In conclusion, we have empirically shown that both methods are practical recommender systems tailored for orthogonal settings. While SBERT-Rec performs best in an open-world scenario without presumptions on the corpus-inherent topics, LDA-Rec can carve out clear topics within a niche closed-world corpus.

## 5. Conclusion

This study presents two recommender systems for expert finding in community question answering platforms like Stack Overflow, using LDA topic modeling and the transformer-baed SBERT model. We test our approach on large-scale datasets from Stack Overflow and Math Stack Exchange, demonstrating its effectiveness and delivering high-quality expert matching results. Our results reveal that SBERT-Rec outscored LDA-Rec on both datasets based on the average rank score. While SBERT-Rec performs better in an open-world scenario with no presumptions about the corpus's underlying subjects, LDA-Rec finds unique topics inside a particular closed-world corpus. In terms of future

| Id | Name | Top four words |
|---|---|---|
| 1 | Formulas | alpha, cdot, sigma, beta |
| 2 | Probability theory | theta, probability, variable, random |
| 3 | Graphs | point, line, graph, circle |
| 4 | Group theory | mathcal, omega, group, element |
| 5 | Set theory | set, subset, space, open |
| 6 | Equations | frac, right, left, int |
| 7 | Logarithms | text, mu, gamma, log |
| 8 | Matrices | end, begin, matrix, lambda |
| 9 | Functions | mathbb, function, delta, epsilon |
| 10 | Number theory | number, integer, prime, polynomial |
| 11 | Proofs | prove, show, let, proof |
| 12 | General request | find, would, question, use |

**Table 4**
Learned math topics represented by the four words having highest topic importance

research, it would be interesting to employ the auto-regressive language model Generative Pre-trained Transformer 3 (GPT-3), which incorporates even more world knowledge. We will also investigate the effect of tags on the representation of the questions.

## Acknowledgments

In parts, the authors of this work were funded by the Federal Ministry of Education and Research of Germany. The authors would also like to thank the Daniel Jung Media GmbH for their insightful input.

## References

[1] C. Treude, O. Barzilay, M.-A. Storey, How do programmers ask and answer questions on the web?(nier track), in: Proceedings of the 33rd international conference on software engineering, 2011, pp. 804–807.
[2] B. Smith, G. Linden, Two decades of recommender systems at amazon. com, Ieee internet computing 21 (2017) 12–18.
[3] C. A. Gomez-Uribe, N. Hunt, The netflix recommender system: Algorithms, business value, and innovation, ACM Transactions on Management Information Systems (TMIS) 6 (2015) 1–19.
[4] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, the Journal of machine Learning research 3 (2003) 993–1022.
[5] S. Momtazi, F. Naumann, Topic modeling for expert finding using latent dirichlet allocation, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 3 (2013). doi:10.1002/widm.1102.
[6] F. Riahi, Z. Zolaktaf, M. Shafiei, E. Milios, Finding expert users in community question answering, WWW'12 - Proceedings of the 21st Annual Conference on World Wide Web Companion (2012). doi:10.1145/2187980.2188202.
[7] H. Dong, J. Wang, H. Lin, B. Xu, Z. Yang, Predicting best answerers for new questions: An approach leveraging distributed representations of words in community question answering, in: 2015 Ninth International Conference on Frontier of Computer Science and Technology, 2015, pp. 13–18. doi:10.1109/FCST.2015.56.
[8] H. Li, S. Jin, S. LI, A hybrid model for experts finding in community question answering, in: 2015

International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, 2015, pp. 176–185. doi:10.1109/CyberC.2015.87.

[9] R. Chi, L. Wang, Expert identification based on dynamic lda topic model, 2018, pp. 881–888. doi:10.1109/DSC.2018.00141.

[10] M. Lübbering, J. Kunkel, P. Farrell, What company does my news article refer to? tackling multiclass problems with topic modeling (2019).

[11] H. Chen, A. G. O. II, C. L. Giles, Expertseer: a keyphrase based expert recommender for digital libraries, CoRR abs/1511.02058 (2015). URL: http://arxiv.org/abs/1511.02058. arXiv:1511.02058.

[12] M. Bhat, K. Shumaiev, K. Koch, U. Hohenstein, A. Biesdorf, F. Matthes, An expert recommendation system for design decision making: Who should be involved in making a design decision?, in: 2018 IEEE International Conference on Software Architecture (ICSA), 2018, pp. 85–8509. doi:10.1109/ICSA.2018.00018.

[13] M. J. Pazzani, D. Billsus, Content-based recommendation systems, in: The Adaptive Web, 2007.

[14] S. Zahoor, Addressing cold start problem in recommendation systems with collaborative filtering and reverse collaborative filtering, International Journal of Computer Sciences and Engineering 6 (2018) 211–214. doi:10.26438/ijcse/v6i4.211214.

[15] J. Bobadilla, F. Ortega, A. Hernando, J. Bernal, A collaborative filtering approach to mitigate the new user cold start problem, Knowl. Based Syst. 26 (2012) 225–238.

[16] J. B. Schafer, D. Frankowski, J. Herlocker, S. Sen, Collaborative filtering recommender systems, 2007.

[17] R. Burke, Hybrid recommender systems: Survey and experiments, User Modeling and User-Adapted Interaction 12 (2002). doi:10.1023/A:1021240730564.

[18] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, CoRR abs/1810.04805 (2018). URL: http://arxiv.org/abs/1810.04805. arXiv:1810.04805.

[19] T. Wang, Y. Fu, Item-based collaborative filtering with bert, 2020, pp. 54–58. doi:10.18653/v1/2020.ecnlp-1.8.

[20] G. Cenikj, S. Gievska, Boosting recommender systems with advanced embedding models, 2020, pp. 385–389. doi:10.1145/3366424.3383300.

[21] Z. Qiu, X. Wu, J. Gao, W. Fan, U-bert: Pre-training user representations for improved recommendation, Proceedings of the AAAI Conference on Artificial Intelligence 35 (2021) 4320–4327. URL: https://ojs.aaai.org/index.php/AAAI/article/view/16557.

[22] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).

[23] D. Lavi, V. Medentsiy, D. Graus, consultantbert: Fine-tuned siamese sentence-bert for matching jobs and job seekers, CoRR abs/2109.06501 (2021). URL: https://arxiv.org/abs/2109.06501. arXiv:2109.06501.

[24] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[25] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, arXiv preprint arXiv:1908.10084 (2019).

[26] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015). URL: http://dx.doi.org/10.1109/CVPR.2015.7298682. doi:10.1109/cvpr.2015.7298682.

[27] M. Lübbering, M. Gebauer, R. Ramamurthy, C. Bauckhage, R. Sifa, Decoupling autoencoders for robust one-vs-rest classification, in: 2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA), IEEE, 2021, pp. 1–10.

[28] M. Lübbering, M. Gebauer, R. Ramamurthy, C. Bauckhage, R. Sifa, Bounding open space risk with decoupling autoencoders in open set recognition, International Journal of Data Science and Analytics (2022) 1–23.