T-Rex: The NASA Technology Taxonomy Recommender System

Ellis Giles
ARES Corporation
NASA Space Technology Mission Directorate
Washington, DC, USA
egiles@arescorporation.com

Ryan Miller
ARES Corporation
NASA Space Technology Mission Directorate
Washington, DC, USA
rmiller@arescorporation.com

Abstract

NASA tracks over 16,000 technology projects across the Agency, from propulsion systems to software. These projects are classified according to NASA's Technology Taxonomy to facilitate data search, extraction, and application. In 2020, the Taxonomy was revised to better align strategic goals with project technical disciplines. Manual re-classification of current and historical projects was estimated to take thousands of technologist labor hours.

Instead of manual classification, our team developed T-Rex, a recommender engine, trained on just a small set of manually classified projects. T-Rex was used to classify the projects and then integrate the data into TechPort to recommend classes to users when updating projects. The system and methodology are used in other NASA projects, and T-Rex has achieved 95% accepted accuracy overall.

CCS Concepts: • **Information systems → Recommender systems**.

Keywords: Space Technology, Portfolio Management, TechPort, Taxonomy, Machine-Learning, Classification, Ontology, Recommender Systems, NASA

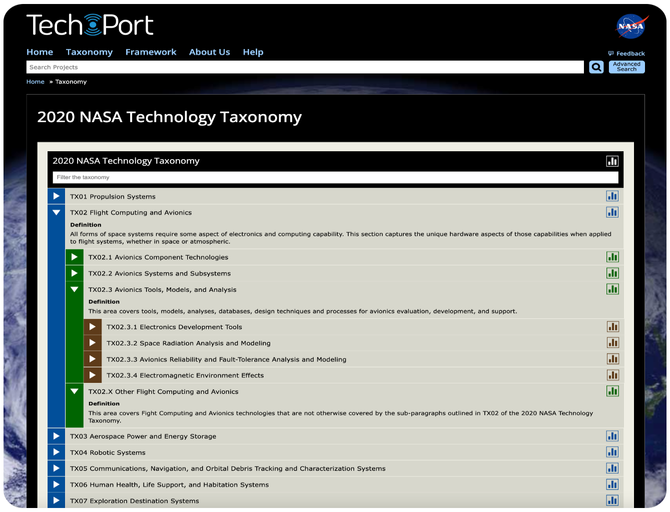
1 Introduction and Overview

In recent years there has been an astronomical increase in the amount of worldwide data. Some studies estimate worldwide data will increase to 175 ZB in 2025 [20], doubling from 2022 levels [22]. With rapid data growth, there is an increased need for ways to determine what data exists, where to find data, how to access the data, and how to determine the value of the data to an organization.

To address these data challenges, many organizations determine common features in collected data and use those features as properties in creating a classification system. Mankind has created classification systems across all domains: biology, with the universal biological Taxonomy introduced by Carl Linnaeus in the 18th century [5], books with the Dewey Decimal Classification System [25], technology in the US patent classification system [4, 13], movies through the Internet Movie Database (IMDB) [16], websites on the Internet in the early Yahoo! website ontology [11], Directory Mozilla (DMOZ), and the Open Directory Project [18], and in specialized technical systems. For instance, the ACM uses the Computing Classification System as a hierarchical ontolgy for classifications in the field of computing [21] which is used to classify this research.

Another example of a technology classification system is the NASA Technology Taxonomy [19]. The taxonomy is comprised of three hierarchical levels with 17 distinct Level 1 Taxonomy Areas encompassing a broad range of technologies from propulsion to information processing. The sub-areas in Level 2 are further refined into 387 Level 3 descriptions encompassing specific types of technologies, as shown in Figure 1. NASA uses this new system to align long-term strategic goals, such as goals in aerospace and environment, with technology projects and investments. Classifying technologies according to a common taxonomy provides a means to efficiently find technologies of interest and correlate investments to avoid duplication of effort and improve existing technology. As new Agency goals and objectives emerge, the Taxonomy is periodically revised, with the most recent edition published in 2020.

NASA currently tracks over 16,000 applied research and experimental development projects developed in the last decade, collecting project data and metadata in NASA's official technology portfolio system known as TechPort [3]. TechPort was tasked with applying the new NASA Technology Taxonomy to project records, requesting that managers



2020 NASA Technology Taxonomy

2020 NASA Technology Taxonomy

- TX01 Propulsion Systems
- TX02 Flight Computing and Avionics

  Definition
  All forms of space systems require some aspect of electronics and computing capability. This section captures the unique hardware aspects of those capabilities when applied to flight systems, whether in space or atmospheric.

  - TX02.1 Avionics Component Technologies
  - TX02.2 Avionics Systems and Subsystems
  - TX02.3 Avionics Tools, Models, and Analysis

    Definition
    This area covers tools, models, analyses, databases, design techniques and processes for avionics evaluation, development, and support.

    - TX02.3.1 Electronics Development Tools
    - TX02.3.2 Space Radiation Analysis and Modeling
    - TX02.3.3 Avionics Reliability and Fault-Tolerance Analysis and Modeling
    - TX02.3.4 Electromagnetic Environment Effects
  - TX02.X Other Flight Computing and Avionics

    Definition
    This area covers Fight Computing and Avionics technologies that are not otherwise covered by the sub-paragraphs outlined in TX02 of the 2020 NASA Technology Taxonomy.
- TX03 Aerospace Power and Energy Storage
- TX04 Robotic Systems
- TX05 Communications, Navigation, and Orbital Debris Tracking and Characterization Systems
- TX06 Human Health, Life Support, and Habitation Systems
- TX07 Exploration Destination Systems

Figure 1. The 2020 NASA Technology Taxonomy

classify their projects accurately and quickly. Requiring over 10,000 project managers to familiarize themselves with 387 classification possibilities would require extensive training and verification, estimated in thousands of hours. Instead of a manual approach, the team recommended the development of a technology that would automatically classify the 16,000 existing projects, and provide a future-proof solution that recommends taxonomy classifications to managers when new records are added to TechPort.

However, development of the classification and recommendation system for the new taxonomy was non-trivial. The prior version of the Technology Taxonomy used to classify records in older NASA systems could not be used reliably. Many projects were missing information and lacked classification altogether. NASA awards short-term projects that span industry, academia, and other agencies, meaning some managers no longer had access or a means to update project data. Even when a classification was available in project metadata, in several cases it was recorded incorrectly or not at a sufficient level of detail.

Additionally, many technology classifications at Level 3 can lead to seemingly ambiguous choices for even a skilled technologist. For instance, class TX4.5.7 is titled "Modeling, Simulation, Analysis, and Test of Rendezvous, Proximity Operations, and Capture" and TX17.3.2 is "Dynamics Analysis, Modeling, and Simulation Tools." Both areas discuss modeling and simulation and responses to forces by aerospace vehicles. However, the first class is in a robotic systems sub-tree, whereas the later is under a generic guidance, navigation, and control tree.

Finally, there were limited data available for training and testing. Only 1,200 projects were classified into the new taxonomy by hand by a group of data experts. These projects mapped to only about 70% of the 387 Level 3 taxonomy classes, leaving more than 100 classes with only a small description of text as one data point. Some categories had 30 projects mapped while others had none. With empty and imbalanced class test and training data, developing any accurate model is difficult. Many existing classification approaches were investigated that resulted in < 50% accuracy.

To solve these challenges, we designed, built, and deployed a system called T-Rex, or the NASA Technology Taxonomy Recommender System. The contributions of T-Rex is notable across multiple areas from model selection to optimization and deployment. Our recommender is integrated into TechPort in a novel, effective manner; and other groups at NASA are now using T-Rex to programmatically classify technology through an exposed API. The system has since been used in other projects to train and build recommenders, such as the NASA Technology Target Destination system. We found our model achieves over 96% accuracy in k-fold training, and over 97.5% accuracy in measured user updates.

2 T-Rex

T-Rex is an amalgamation of classifier models, custom models, and voting optimization and pruning methods for efficient, accurate, deployed models. The system automatically optimizes, selects, and combines multiple individual models. It uses prior classification information only when accurate to suggest a subset of classes as recommendations for other models. Our training algorithm optimizes weighting from individual models from a matrix of individual model outputs to reduce training time.

The T-Rex Machine Learning model is shown in Figure 2. Project metadata for the model includes five textual fields including technology title, description, benefits, and a findings closeout summary. A potential prior technology area reference which may be absent or incorrect is also included in the metadata.

The textual fields are pruned for stop words and punctuation and word roots are utilized (not shown for space). Once the word roots are determined, metadata on the words is pulled from our model. This word metadata includes word frequencies in each class, overall word frequencies, and word weights for each class. Word weights are determined in our training step, described below, using Neighborhood Component Analysis.

The words and their corresponding metadata are sent to an array of optimized sub-models. As shown in the figure, these models include a K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Hierarchical Decision Tree (Tree), Bag of Words (Bag), and Naïve Bayes model. Tree and Bag did not contribute to the overall accuracy and were removed during optimization. The Map classifier takes a potentially empty prior technology area classification and, if present, produces a set of potential new taxonomy area classifications equally weighted. This map of taxonomy classes for the project may be large depending on the prior areas. For instance one area might have up to 20 similar taxonomy classes, yielding a map vector containing 367 zeros and twenty 0.05 entries.

Each sub-model produces a vector of probabilities, one for each of the 387 output classes. We then multiply the weight vector with a combined matrix of each sub-model output, yielding a vector of 387 probabilities. The weight of the Map

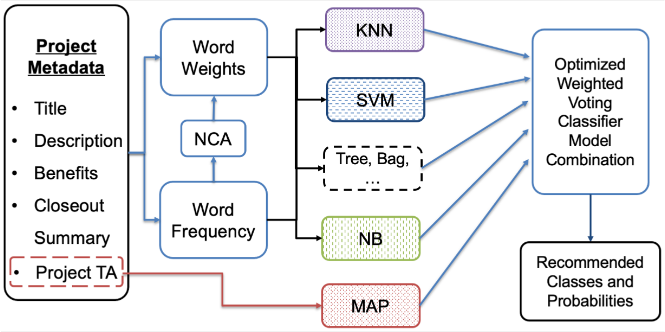


Figure 2. The T-Rex Machine Learning Model Architecture

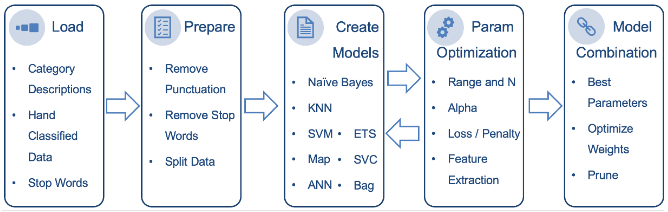


Figure 3. The T-Rex Training and Optimization Process

is low enough such that incorrect priors are overridden by high outputs on other closely matched sub-models. Any sub-model that is weak in one set of class predictions may be overcome by another sub-model.

The training process is shown in Figure 3. We only received approximately 1,200 manually classified projects; that effort alone took several weeks. Some classes did not have any project data, therefore we also utilized the taxonomy class title, description, and taxonomy node parents. We then prepared the data for training by removing punctuation, common or stop words such as *a, at, the, etc.*, and split the projects into train and test data sets. Due to some classes only having one data point, e.g. the class description, we preformed 10-fold cross-validation training, and only pulled final test cases from classes with multiple projects.

In the Create Models step, we produce an array of sub-models. In the Param Optimization step, each model was individually optimized with its features, such as number of neighbors K in KNN. Once the NCA model was optimized, it produced word weights and feature sets which were used in our training steps.

Once individual models were optimized, we combined the sub-models in the final step. For each sub-model M, we recorded the class probability output C for each project data point N, producing an M x C x N matrix U. We then found the weight vector W of size M, such that W times U, had the closest match to the desired output C x N. To find W, we searched the sub-space of combinations of W whose elements summed to 1 for weighted probabilities at 0.01 granularity.

2.1 System Architecture and User Interface

The system architecture is shown in Figure 4. T-Rex was deployed in Amazon SageMaker with staging and production models. We create a REST-based API that is used by the TechPort client application and other NASA users. The client API can be configured to contact the T-Rex instance via the Amazon API Gateway or through the TechPort application server. We found that either method produces fast recommendations for users.

The User Interface for recommendations is shown in Figure 5. When a user creates a project record in TechPort, or edits the classification of an existing record, an asynchronous call is made to T-Rex, returning an array of the top five recommendations. Based on experimental thresholds and the probabilities of the recommended classes, recommendations are shown as High, Medium, or Low, or removed.

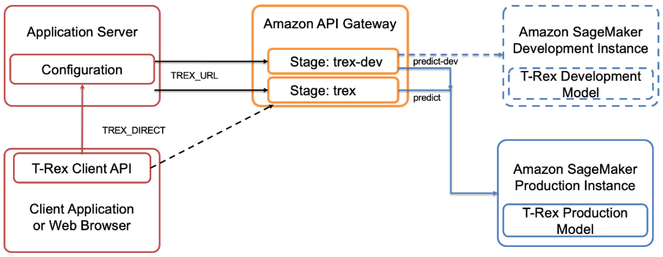
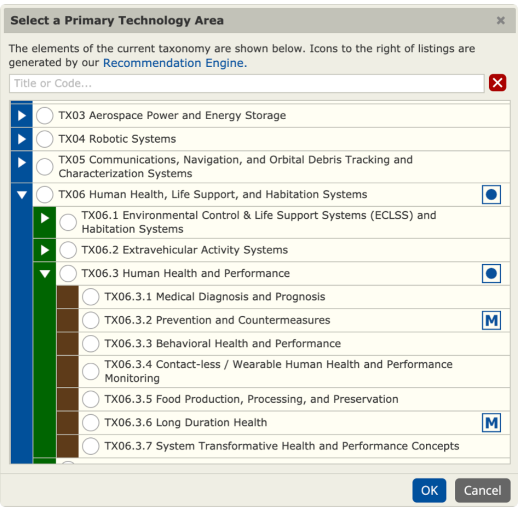


Figure 4. T-Rex System Architecture


Select a Primary Technology Area ×

The elements of the current taxonomy are shown below. Icons to the right of listings are generated by our Recommendation Engine.

Title or Code…

- ○ TX03 Aerospace Power and Energy Storage
- ○ TX04 Robotic Systems
- ○ TX05 Communications, Navigation, and Orbital Debris Tracking and Characterization Systems
- ○ TX06 Human Health, Life Support, and Habitation Systems ●
  - ○ TX06.1 Environmental Control & Life Support Systems (ECLSS) and Habitation Systems
  - ○ TX06.2 Extravehicular Activity Systems
  - ○ TX06.3 Human Health and Performance ●
    - ○ TX06.3.1 Medical Diagnosis and Prognosis
    - ○ TX06.3.2 Prevention and Countermeasures M
    - ○ TX06.3.3 Behavioral Health and Performance
    - ○ TX06.3.4 Contact-less / Wearable Human Health and Performance Monitoring
    - ○ TX06.3.5 Food Production, Processing, and Preservation
    - ○ TX06.3.6 Long Duration Health M
    - ○ TX06.3.7 System Transformative Health and Performance Concepts

Figure 5. Recommendations Highlighted with *H*, *M*, or *L*

3 Evaluation

Individual model performance was increased with the introduction of the NCA word weighting, but was not sufficient. Subsets of the Confusion Matrices are shown in Figure 6 for KNN and Figure 7 for Naïve Bayes. When these models are combined with the optimized weights, the best from KNN can outweigh NB and vice-versa. Figure 8 shows the accuracy versus the combined model with relative model contributions in weights. With just one model, NB performs best with only 47%. However, when combining two models, Map which produces an even distribution favoring no class, is weighted at 0.96 and is boosted by SVM at 0.04 to select amongst the mapped classes. As more models are utilized, a more even distribution of weights is observed; however, there is no accuracy benefit after using 4 sub-models. The overall experimental accuracy is measured over 96%.

For deployed model accuracy, we measure changed project classifications on the deployed system. On May 22, 2020, the remaining 14,800 non-manually classified projects were analyzed by T-Rex. The top recommendation for each of these projects was then uploaded into the TechPort database as the primary class for each project. Almost 2.5 years later, in October 2022, only 810 projects had been changed to a different taxonomy class. It is important to note that NASA policy requires project data in TechPort to be validated and

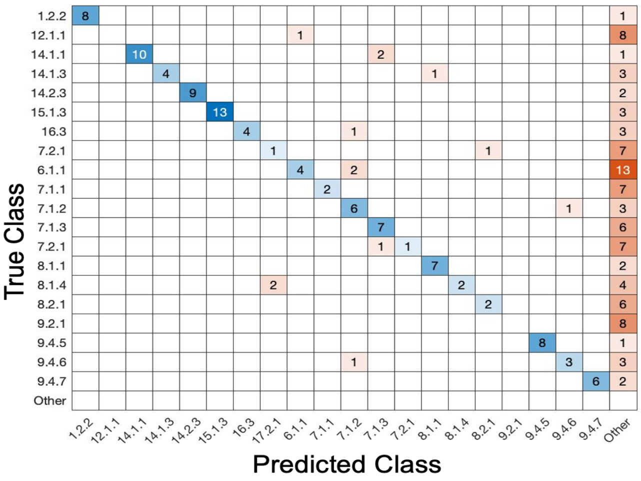


Figure 6. Subset of Confusion Matrix for KNN Sub-Model

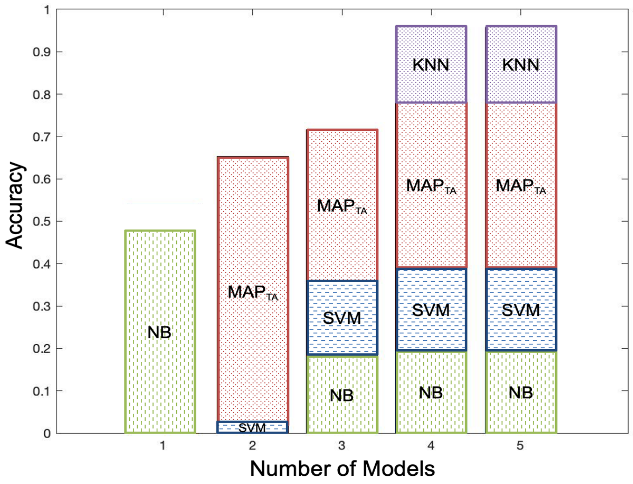


Figure 8. Combined Model Accuracy with Weights

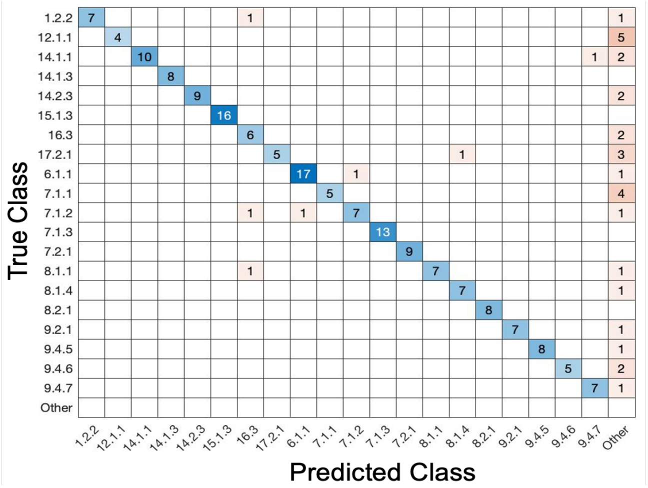


Figure 7. Subset of Confusion Matrix for Naïve Bayes

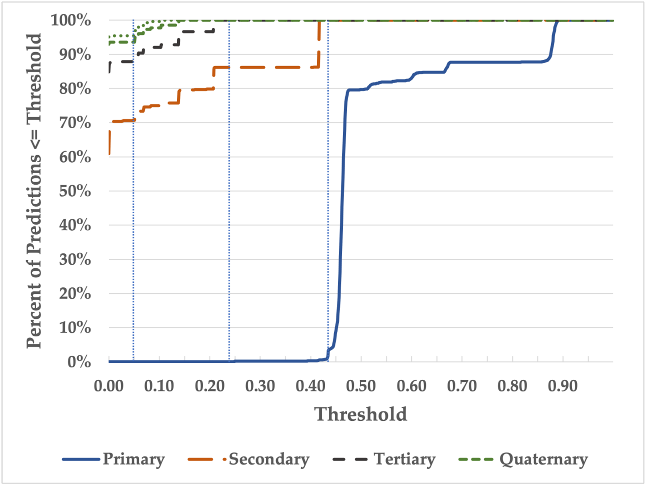


Figure 9. Cumulative Dist of Recommendation Probabilities

updated bi-annually, and the automatic classification by T-Rex was communicated and reviewed by managers. The small number of class changes for projects correlates to an observed 94.5% acceptance accuracy.

Threshold selection for High, Medium, and Low is shown in Figure 9. The cumulative distributions for each of primary, secondary, tertiary, and quaternary selections were plotted. To recommend approximately 4 classes with high probability, maximum non-interfering thresholds were determined to be 0.44, 0.24, and 0.5. The application uses the probabilities returned from the API for highlighting in the UI.

4 Related Work

A survey of classification systems across various domains was performed in [23]. For websites, a multi-agent recommender system was developed [15, 17], and a categorization engine with boosting was developed with 76% accuracy [10]. In video content, the Netflix recommender system is described [6], IMDB recommendation using graphs [7], and custom models using a variety of methods [2] has been explored. Deep learning has been used in patent classification [12, 14], other technical document classification [9], music features [8], and other recommender systems [28]. Patents have also been classified using a combination of SVM and LDA [27] and NLP with NN [24]. Other recommender systems have used Reinforcement Learning [1, 26, 29].

5 Summary

We presented and described T-Rex, the NASA Technology Taxonomy Recommender System. The system is comprised of a novel model selection, mapping, and optimization process, maximizing performance and pruning unused models.

The system has been deployed for over two years and recommended almost 40,000 classifications. It is used by multiple groups, and similar methods have been used in another recommender system for technology target destinations. T-Rex achieves high accuracy of almost 95% on both small sets of training data and observations in practice.

Acknowledgements

We would like to thank our sponsors, data providers, and many technical assistants. Our sponsors at NASA include T-Rex's original sponsor, Patrick Murphy, and current sponsor Alesyn Lowry. Training data were provided and analyzed by Carie Mullins and Madeline Shaughnessy. Extensive testing and evaluation was performed by Nathaniel Booth and Stephanie Booth. User interface design was developed by Paul Herzing, complex database queries and data gathering by Christine Lehman, preliminary investigations Hristo Iankov, and integration by Dillon Gresham.

References

[1] M. Mehdi Afsar, Trafford Crump, and Behrouz Far. 2022. Reinforcement Learning Based Recommender Systems: A Survey. https://doi.org/10.1145/3543846 Just Accepted.

[2] Warda Ruheen Bristi, Zakia Zaman, and Nishat Sultana. 2019. Predicting IMDb Rating of Movies by Machine Learning Techniques. , 5 pages. https://doi.org/10.1109/ICCCNT45670.2019.8944604

[3] NASA Space Technology Mission Directorate. 2022. TechPort. https://techport.nasa.gov

[4] Louis Falasco. 2002. United States patent classification: system organization. *World Patent Information* 24, 2 (2002), 111–117. https://doi.org/10.1016/S0172-2190(02)00007-8

[5] H Charles J Godfray. 2007. Linnaeus in the information age. *Nature* 446, 7133 (2007), 259–260.

[6] Carlos A. Gomez-Uribe and Neil Hunt. 2016. The Netflix Recommender System: Algorithms, Business Value, and Innovation. *ACM Trans. Manage. Inf. Syst.* 6, 4, Article 13 (dec 2016), 19 pages. https://doi.org/10.1145/2843948

[7] Jelena Grujić. 2008. Movies recommendation networks as bipartite graphs. , 576–583 pages.

[8] Aniket Jha, Sagar Gupta, Priyanshu Dubey, and Aditi Chhabria. 2022. Music Feature Extraction And Recommendation Using CNN Algorithm. , 03026 pages.

[9] Shuo Jiang, Jie Hu, Christopher L. Magee, and Jianxi Luo. 2022. Deep Learning for Technical Document Classification. , 17 pages. https://doi.org/10.1109/TEM.2022.3152216

[10] Aldin Kovačević, Zerina Mašetić, and Dino Kečo. 2021. *Naive Website Categorization Based on Text Coverage.* Springer International Publishing, Cham, 435–448. https://doi.org/10.1007/978-3-030-54765-3{_}30

[11] Yannis Labrou and Tim Finin. 1999. Yahoo! As an Ontology: Using Yahoo! Categories to Describe Documents. In *Proceedings of the Eighth International Conference on Information and Knowledge Management* (Kansas City, Missouri, USA) *(CIKM '99)*. Association for Computing Machinery, New York, NY, USA, 180–187. https://doi.org/10.1145/319950.319976

[12] Jieh-Sheng Lee and Jieh Hsiang. 2019. Patentbert: Patent classification with fine-tuning a pre-trained bert model.

[13] Loet Leydesdorff. 2008. Patent classifications as indicators of intellectual organization. *Journal of the American Society for Information Science and Technology* 59, 10 (2008), 1582–1597.

[14] Shaobo Li, Jie Hu, Yuxin Cui, and Jianjun Hu. 2018. DeepPatent: patent classification with convolutional neural networks and word embedding. *Scientometrics* 117, 2 (2018), 721–744. https://doi.org/10.1007/s11192-018-2905-5

[15] A. Jorge Morais, Eugénio Oliveira, and Alípio Mário Jorge. 2012. A Multi-Agent Recommender System. In *Distributed Computing and Artificial Intelligence*, Sigeru Omatu, Juan F. De Paz Santana, Sara Rodríguez González, Jose M. Molina, Ana M. Bernardos, and Juan M. Corchado Rodríguez (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 281–288.

[16] Col Needham. 1998. Internet movie database.

[17] Joaquim Neto and A. Jorge Morais. 2014. Multi-Agent Web Recommendations. In *Distributed Computing and Artificial Intelligence, 11th International Conference*, Sigeru Omatu, Hugues Bersini, Juan M. Corchado, Sara Rodríguez, Paweł Pawlewski, and Edgardo Bucciarelli (Eds.). Springer International Publishing, Cham, 235–242.

[18] ODP. Last accessed on 2022-10-15. Web Directory of High-Quality Resources: The Open Directory Project. https://odp.org

[19] Office of the Chief Technologist. 2020. NASA Technology Taxonomy. https://www.nasa.gov/offices/oct/taxonomy/index.html

[20] David Reinsel, John F. Gantz, and John Rydning. 2020. *The Digitization of the World: From Edge to Core.* Technical Report. Whitepaper, International Data Corporation (IDC).

[21] Bernard Rous. 2012. Major update to ACM's computing classification system. *Commun. ACM* 55, 11 (2012), 12–12.

[22] John Rydning. 2022. *Worldwide IDC Global DataSphere Forecast, 2022–2026: Enterprise Organizations Driving Most of the Data Growth.* Technical Report. Whitepaper, International Data Corporation (IDC).

[23] Carlos N. Silla and Alex A. Freitas. 2011. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery* 22, 1 (2011), 31–72. https://doi.org/10.1007/s10618-010-0175-9

[24] Amy Trappey, Charles V Trappey, and Alex Hsieh. 2021. An intelligent patent recommender adopting machine learning approach for natural language processing: A case study for smart machinery technology mining. *Technological Forecasting and Social Change* 164 (2021), 120511.

[25] Jun Wang. 2009. An extensive study on automated Dewey Decimal Classification. *Journal of the American Society for Information Science and Technology* 60, 11 (2009), 2269–2286.

[26] Xin Xin, Alexandros Karatzoglou, Ioannis Arapakis, and Joemon M. Jose. 2020. Self-Supervised Reinforcement Learning for Recommender Systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 931–940. https://doi.org/10.1145/3397271.3401147

[27] Junghwan Yun and Youngjung Geum. 2020. Automated classification of patents: A topic modeling approach. *Computers & Industrial Engineering* 147 (2020), 106636. https://doi.org/10.1016/j.cie.2020.106636

[28] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep Learning Based Recommender System: A Survey and New Perspectives. *ACM Comput. Surv.* 52, 1, Article 5 (feb 2019), 38 pages. https://doi.org/10.1145/3285029

[29] Lixin Zou, Long Xia, Zhuoye Ding, Jiaxing Song, Weidong Liu, and Dawei Yin. 2019. Reinforcement Learning to Optimize Long-Term User Engagement in Recommender Systems. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Anchorage, AK, USA) *(KDD '19)*. Association for Computing Machinery, New York, NY, USA, 2810–2818. https://doi.org/10.1145/3292500.3330668