Car Price Evaluation

Anonymous Author(s)

ABSTRACT

Because of the durability and price of a new car, many users would choose suitable used cars to save their money. However, the value of a used car was much more complicated than a new car. It would be influenced by the production time, mileage, and maintenance. Hence a well-prepared but old car might be more valuable than a newer but poorly maintained car. KBB was a famous car value estimation website. It could estimate a car from every aspect and provide different prices based on the market data.

KBB entirely relied on data mining and machine learning methods. However, we developed a used car value evaluation system similar to KBB but based on the cognitive computing method. This system would use a python crawler to automatically get data from several famous car sale websites and construct its corpus. Then, it would use user input and its corpus to find the price range and generate price hypotheses. Finally, output estimated price of the car. To make the system fit users' needs, we also added a QA phase at the end of the process, which would change all the parameters during the evaluation process. Users could score the result and let the system re-evaluate the car.

KEYWORDS

Cognitive Computing, NLP, Machine Learning, data mining, price evaluation

1 MOTIVATION

The car has become a necessary part of everyone's life. However, choosing a suitable used car is a difficult task for most people, especially we can find countless various used cars in the current market. Hence, we planned to develop a car price evaluation system to help people find a suitable option. Because of the implementation similarity of recommender systems and evaluation systems, we absorbed various excellent ideas from that area.

1.1 Current Related Applications

Recommender systems are used to suggest novel products and personalize the advertisement, are interesting research areas not only because it is a challenging area, but also because of its huge financial potential.[25] There have been many recommender systems that are probably already used such as playlist generation for video and audio content, products recommendation for e-commerce, and social network recommendations. However, not all recommender systems are cognitive and intelligent recommender system. For example, some recommender systems are purely based on simple algorithms such as keyword match, which makes them awkward to use and not providing much value.

A cognitive recommender system should include 3 parts at a high level: datadriven, knowledge-driven, and cognition driven [3]. Data driven means that the recommendation should rely on large set of data to improve the accuracy of recommendation. Knowledge driven means that the system should mimic the knowledge of domain experts to improve desired recommendations while reduce undesirable recommendations. Cognition driven means users' personality and behavior should be take into consideration when recommending a service to customer. Although there are many recommender systems, recommender system for vehicles is seldom available. Most websites only accepts specifications of cars to generate a list to customer.

A semantic search-based recommender system architecture for vehicle sales [22]. The architecture includes mainly 4 layers. An ontology layer where the processed models of the corpus are stored and used to generate the candidates. A context layer to provide user input and related information. A discover layer that is used to understand user input. And a recommendation layer, which combines the discovery layer and the ontology layer to generate the recommendations.

1.2 Our Application

We planed to develop a car price evaluation system. It will be slightly different from a car recommendation system. In most practical scenarios, users need to check each recommendation result whether it fits their needings or not. This process will be challenging, especially for somebody who does not know the car market very well wants to buy a car because there are tons of various technical terms to describe cars.

Moreover, because of the volume of the American car market, it is also hard to decide which car to buy even users know very well about their needings and decided on a specific car model. The price will be various based on the year, the seller and maintenance.

Hence, we want to develop a cognitive system to help car buyers figure out the actual price of a car. We combine cognitive computing with big data and machine learning to build a cognitive system. This system can tell users whether a car is acceptable or not based on the input data, the system corpus and users own needings.

1.2.1 Data Sources. Generally, we have two parts of data sources. The first part is well-prepared datasets from several dataset database websites. The second part will be raw plaintexts, most of them will come from the Internet, about car price evaluation. Furthermore, we plan to implement this part in three candidates ways.

First, We can use the website official API to extract data from The car evaluation cluster like r/whatcarshouldIbuy,r/carbuying. Several sub-topic possibilities will not take too much time, and the previous two subreddits have a high concentration of information on evaluating the car price. Therefore, we will determine it manually. Also, we can build a virtual corpus by searching Reddit with several keywords.

Second,We can use the data from Craigslist and other car shopping sites. The data will be structured and easy to use on those websites. However, there is no existing API on most of these sites. We have to use web scrape or OCR to achieve data. Therefore, maybe we will give up the shopping sites corpus if time is not enough.

Finally, we will directly extract data from existing datasets.We cab use them to build a model based on their possible inputs.

1.2.2 Input Data NLP Process. In order to make the system understand the input plaintexts, we need an input learning process. It will extract certain data from the plaintext and build construct data. Because most data we need is numerical, like miles and date of manufacture, we will extract related numerical data. Then, we will extract numerical values close to the frequency items. Although somebody may write irregularly and input incorrect data to our system, we will still generate hypotheses for each numerical number. We will use evidence provided by other inputs datasets and our corpus to eliminate improper hypotheses finally. Second, to discover the car model and brand, we will use an English dictionary to extract all the words. Then the system will search from corpus to classify them. For this part, we might use Watson API Natural Language Understanding to achieve the function.

1.2.3 Predicting Model. We need a predicting model to generate hypotheses. First, we will use an algorithm to find several close candidate prices to the input data. Then, the system will search for the most similar evidence to the input data. The system also will use distance edges to assign confidence values. We also need machine learning to calculate a correct score for the distance edges and the confidence value in the process. After scoring, we will remove several improper candidates like low confidence or abnormal price and finally calculate our prices by each candidate's proportions based on their score.

1.3 Project Goals

The ultimate goal of this project is to develop a car evaluation system that can automatically get data from the Internet and outputs an evaluation price based on users' input. Moreover, this system should have a feedback system that can dynamically balance each weight of all the parameters in the hypotheses generation and scoring process.

1.4 Roadmap

We will illustrate our project goals in section 2. Then, we will discuss the research area related to our project in section 3. In section 4, we will present our project structure and implementation details. After that, we will discuss the disadvantages of the project and what we can do to improve the project. In section 6 and section 7, we will discuss the legal and ethical considerations about our project. Finally, we will make a brief conclusion in section 8.

2 RELATED WORK

The car price evaluation problem is very similar to question-answering problem that many researchers are working to solve, among which the pioneer work is done by the IBM Watson project [8]. In the QA system, the system typically takes a question in natural language and then provides an answer via information retrivial. Poonam Gupta et al[11] describes the general architecture of Question-Answering system in five components: 1) query preprocessing, 2) query generation, 3) database search, 4) related documents, 5) answer display. Query preprocessing processes the natural language quesiton to analyze the input. Query generation converts the preprocessed query into machine understandable commands. Database search is used to retrive information from the provided documents. Related documents is the process of generating matching results. And answer display is the process to generate the final answer to for the query. Further, QA systems can be divided into two major groups. The first group is purely based on natural language processing and information retrival by matching methods. The START QA system stores web data by tagging the natural language with annotations and store them into a database which can further be used to provide answers to questions [15]. Another type of QA system is more intelliegent and can provide answers based on reasoning. Zhang *et al.* [31] proposed a method using deep learning to perform multip-hop reasoning on knowledge graph to build a QA system with reasoning abilities. Ajitkumar M. Pundge *et al.* [1] and Sanjay K Dwivedi *et al.* [7] discussed various approaches for question answering systems. Linguistic approach tries to understand natural language using NLP techniques such as tokenization, POS tagging and parsing. The linguistic approach is mainly information retrival only and is widely used for closed domain (where the question are from a specific topic) questions [2]. Another approach is pattern approach where the system tries to match the pattern from the question and the corpus data. The pattern approach is useful for questions like definition lookup. Statistical approach uses techniques such as maximum entropy model, n-gram mining *etc* to analyze huge amount of data and is suitable for open domain questions where the internet can be used as the source of corpus. The statistical method is widely used to implement reasoning QA systems, such as CubeQA [12].

One key thing in the car price evaluation problem is keywords extraction from natural language. Keyword extraction is useful both to build the corpus and to analyze the user query. There are mainly four categories of keyword extraction methods: statistics, linguistic, machine learning, and hybrid methods [4, 26]. Statistical methods usually uses techniques such as TF-IDF, term position, term co-occurrence to identify the key terms from documents [19]. The linguistic approach detects and extracts keywords based on the lexical, syntactic, or discourse analsis [21]. The machine learning approch usually uses machine learning models such as support vector machine [30] to learn from training samples where keywords are known and then use the learned models to get keywords from other documents. The hybrid method simply combines more than one methods to try to get better result.

Another challenge in the research is answer ranking and validation, which is to confirm how relevant the provided answer is to the original question. There are mainly two techniques for answer validation, statistical method and content based method **??**. The statistical method is mainly based on similarity measure. Fangtao Li *et al.* [17] proposed an information distance calculation approach to solve the answer validation problem where the answer should have the smallest distance from the question among all the candidate. Miyanishi *et al.* [20] studied hyphothesis generation by similarities between different events. Content based answer validation mainly focuses on if the answer can be confirmed by the supporting materials. Masatsugu Tonoike *et al.* [28] discussed answer validation by keyword association where the keyword from the answer should match the keyword from the question. Anselmo Peñas *et al.* [24]

describe the revelation confirmation as a textual entailment problem by verifying that if the given result supports the input, which is based on reasoning of the logical relationship.

3 DESIGN AND IMPLEMENTATION

In the following, we will first introduce the framework of our cognitive system. And then, we will introduce each step concretely from corpus construction through hypothesis generation and scoring to the output.

3.1 System Architecture

The system first implemented several browser-simulation-based web scrapers to build corpus through the python library request and beautifulsoup. Then, we choose python library nltk to perform NLP to preprocess phase. After the data is structured, the project implements several classification algorithms and feature selection to hypothesis generation and evaluation through the python library scikit learn. In order to evaluate our result, we use python matplotlib to visualize our result and evaluate the performance. Finally, we code a user interface by python library PySimpleGUI to enable our cognitive system to interact with humans. The process are spefied in the fig.

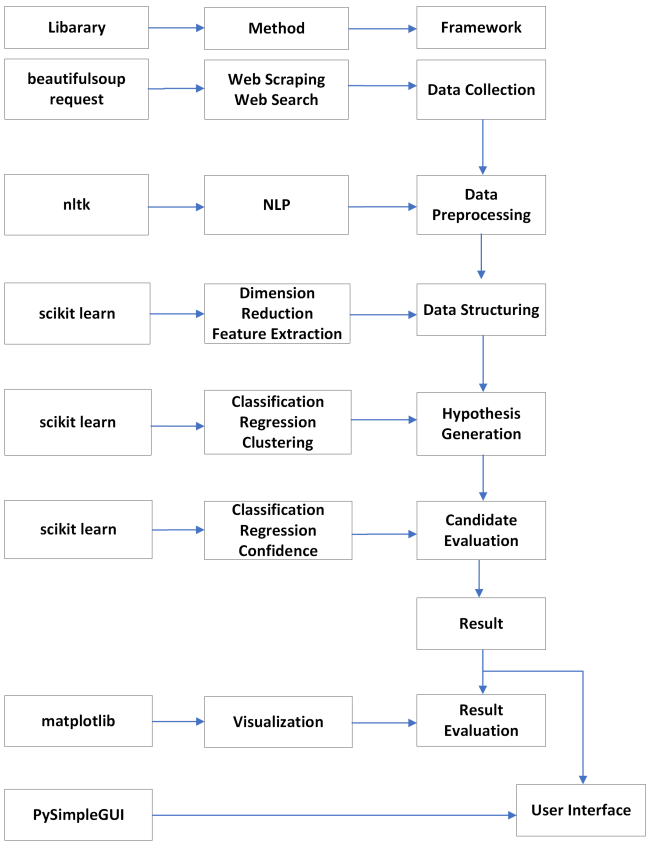

Figure 1: Structure

3.2 Input and Output

Input will be natural language sentences, for example: "Looking at buying a 2010 Subaru Outback 2.5i Premium for my son. I know that without an inspection, you can't give great advice, but what is the consensus in these cars? Typically a good buy? It has 159k miles. I didn't notice anything negative in the test drive. No odd noises or any concerns. I'm somewhat handy with car, but by no means an expert. Brakes seemed good."

3.3 Corpus Construction

There are two major approaches to build the corpus. The first part is well-prepared datasets from several dataset database websites. The advantage of the approach is that the data set will be well structured and have minimal effort. However, the approach suffers from several following shortcomings; first, the datasets might be outdated due to most datasets updating very slow; second, the datasets might not be huge and suitable enough for training because the most dataset will only collect a specific part of data to fulfil their own demand; finally, the most datasets will not indicate where the datasets from so that the confidant of the datasets can not be guarantee. This fault will have a big negative impact on the cognitive system. Therefore, To build a more huge and update in the real-time corpus, the project will use existing datasets downloaded from online to build a basic car evaluation training set, two scrapers for Reddit posts, and information from www.cars.com.

There are several ways to build a web scraping, the first is the web API, which might be the best way because the data will be organized in a suitable manner and high efficient, but most websites will not provide API. The second way is to simulate the browser and analyze on HTML structure to get the needed information. Thus, the program will use API as the first choice.

The corpus first needs data to extract keywords that are the most important words to evaluate a car, like cars' model, body style, mileage, etc. So To get the keywords, we build a web scraper on "www.cars.com", which have a sound filter system for extracting keywords. Because the websites don't have API, we explored the python library requests to request HTML and Beautifulsoup to analyze the HTML and finally successfully got all the keywords from the filter. The lack of data is the cars' information to generate hypotheses and especially the latest car selling data based on the keywords. We use the same library to build a search scaper to search and retrieve car information depending on a specific query. Thus, once the internet web updates, our search will achieve the latest data from the Internet. The last data is the input training data that is for our NLP inputs. Fortunately, Reddit provided API, so we built a scraper based on Reddit API, which extracted every post in the car evaluation subreddits.

3.4 NLP Process

To make the system understand the input plaintexts, we need an input learning process. It will extract specific data from the plaintext and build construct data. Because most data we need is numerical, like miles and date of manufacture, we will extract related numerical data. Then, we will extract numerical values close to the frequency items. Although somebody may write irregularly and input incorrect data to our system, we will still generate hypotheses

for each numerical number. We will use evidence provided by other inputs datasets and our corpus to eliminate improper hypotheses finally. Second, to discover the car model and brand, we will use an English dictionary to extract all the words. Then the system will search from corpus to classify them. For this part, we might use Watson API Natural Language Understanding to achieve the function.

Also, there are many edge cases in the process. For instance, many prices will be separated by "," . and the previous NLP process will separate them improperly. Another problem is that our NLP process will be in trouble if the substring is meaningful in the corpus. For instance, the Audi a3 e-tron can be separated into Audi a3, also a model or Audi a3 e-tron. In order to overcome the two-issue, we first perform the associate rule discovery to have a better relationship between ambiguous words and other context words. We build a parser to read the sentence and separate data by probability based on this probability relationship.

3.5 Candidate Machine Learning Algorithm

Classification is a process of structuring or labeling data into small clusters that the data in each small collection have similar properties. In contrast, a regression algorithm can be deemed as classifying data based on continuous input. Classification and regression have been widely used in big data, data mining, and cognitive computing. The most famous fundamental classification and regression algorithms are k-NN, Decision Trees(DT), Support Vector Machine(SVM), Gradient Descent(GD), Decision, regression tree(Cart), Linear Regression, Artificial Neural Network and improvement work on them like dimensionality reduction, random forest. Decision trees commonly con is divided into two categories; one is the Classification Tree, which predicts discrete values like binary; another is the Regression Tree, which can predict continuous values. Later on, Breiman combines the two trees as the classification and regression tree(CART). [18]

Furthermore, many optimizations work like a random forest tree to improve the classification by random building many trees to get more diverse usage of data [6]. Also, there are many other improvements like Chi-square automatic interaction detection (CHAID) [14], Conditional Inference Trees [13]. The K-NN is a distance-based algorithm that utilizes the nature of a similar item that will have less distance in its attributes. However, K-NN will suffer from computational expenses and the curse of dimension. So, Vladimir N. Vapnik and Alexey Ya. Chervonenkis introduced the SVM that projects the high dimension object to the lower dimension. [5], And other dimensions reduction works like Principal Component Analysis(PCA) [27], Linear discriminant analysis (LDA) [29], Generalized discriminant analysis (GDA) [32].

3.6 Hypothesis Generating and Scoring

We need a predicting model to generate hypotheses. First, we will use an algorithm to find several close candidate prices to the input data. Then, the system will search for the most similar evidence to the input data. The system also will use distance edges to assign confidence values. We also need machine learning to calculate a correct score for the distance edges and the confidence value in the process. After scoring, we will remove several improper candidates like low confidence or abnormal price and finally calculate our prices by each candidate's proportions based on their score.

3.7 Output

Output estimates price with the highest confidence or a graph where the x is price, and y is confidence level.

4 ANALYSIS

In the following, we will first introduce the shortcoming of our cognitive system and so that the future work. The issue will include the corpus build, machine learning algorithm, user interface. And then, we will introduce what we learned from the project. Which will include the technique learning and general learning

4.1 Shortcoming and Future Work

First, our training data is based on the current selling data but not on sold data, which means that the selling item might not accurately feedback the actual value. Also, we do not have the data for the price that change on time. So, the system only reflects the current price but cannot be more meaningful to predict future prices. Therefore future work needs to scrape data to perform Time Series Forecasting continuously. Second, our domain is relatively small and well structured. We might improve our system to suit a more enormous and "noise" corpus in the future. Finally, our web scraping needs many human efforts and only works well on several websites. In the future, we might build a more intelligent web scraper to suit huge corpus requirements, reducing the human effort to analyze each HTML structure.

Moreover, In the machine learning phase, we trained our hyperparameters by hand. The machine learning process might not suit new data since the dimension, and other attributes have changed dramatically. The solution can be to improve the NLP process, and the program can have more knowledge of the data. With a better understanding of the data, we can build an automated hyperparameters program.

Furthermore, our system cannot deal with user feedback for our evaluation system. That means our system cannot adapt when the demand and world change and cannot evolve during the human-machine interaction. Moreover, the user feedback can be somehow a label for each instance to get more valuable data in the interaction. Therefore, we can build a better user interface and collect the user's evaluation on each output in the future. In this way, The system can be more adaptively by automated learning this user feedback to improve the system.

Finally, our system can be more meaningful by providing more functions. First, the feedback and discussion for a car are also crucial for consumers. So, we can build a more complicated recommend system analyzing this feedback and any relevant data. Second, our user interface is too simple just to include the input and output. Therefore, we can improve the interface by providing more buttons or several highly recommended cars suitable for specific customers.

4.2 Learned from the Project

Generally speaking, we learned a lot about the cognitive structure and how to build cognitive systems empirically. Also, we learned how to work together and communicate with each other to achieve

our goals. For example, we need to separate a big project into small pieces and collaborate well on the small pieces. Hence, good documentation and communication frequently are incredibly critical. Because we do not just need to make the program work, the group members also need to connect well on each other's parts.

Finally, we learned many machine algorithms about their underlying concepts and their library. We learned how to write a web scraper in the corpus build process, and the corresponding library includes python beautifulsoup and request.Also, we learned the HTML structure and grammar to build the web scraper. In the preprocessing process, we learned how to write a simple NLP program to deal with row data and the corresponding libraries like nltk. In the feature extraction phase, we learned several dimension reduction algorithms like Kernel PCA. Moreover, we learned many machine learning algorithms in the hypothesis-generating and scoring phase, including regression, classification, and clustering. Also, we learned many model selection techniques like grid search, cross-validation, metrics. Finally, we learned how to implement visualization through mapplotlib to evaluate our result and user interface.

5 LEGAL CONSIDERATIONS

In the development of car price evaluatin system, the price recommendation and evaluation should be clear on theuse of data. The legal considerations should include how data are collected, stored, shared, and inferred from. For example, these issues in the European context are already regulated with the Genernal Data Protection Regulations in May 2018 [23]. Specifically, the following general guidelines should be followed. We should only collect data that are required to perform the functionality of the car price evaluation system. The data collected should be only used and processed for the car price evaluation purpose. The data should be processed in a way that avoids potential data leaking risks.

Although the car price evaluation system may take some personal data into consideration. The price evaluation for the same car under the same market should be the same regardless of the user. Otherwise, the system is acting price discrimination against users which is prohibited by the Sherman Antitrust Act [9]. This legal considerstaion should be evaluated against the various machine learning algorithms to make sure that they don't take unrelated information into the model to avoid price discrimination against different users.

Another legal considerstaion is that the car price evaluation system should provide a method to allow users to remove their person information. The system should also notify the customer and ask for the customer's consent if the data needs to be transferred to another company for other analysis. Failing to comply with these methods may result in violation of the California Consumer Privacy Act (CCPA) if the system is used in California, USA [16].

6 ETHICAL CONSIDERATIONS

Car prices evaluation and recommend system can collect, reorganize, and evaluate a large amount of car price data, potential customer and previous customer's data and output a tailored result to users. In recent years, the unprecedented popularity of recommender applications has raised several issues related to the ethical and legal implications of the car price evaluation system. To assess the relevant ethical issues, we rely on the emerging principles across the ethics fields.

According to ACM ethnical code 1.6 [10].Regarding privacy concerns to our system, we will establish transparent policies and procedures that allow individuals to understand how the system collects the data and how data is used. The system result will keep consistent with different users. However, the privacy issue regarding the car price evaluation system is focused on how data are collected, stored, and shared. To give informed consent for automatic data collection, review, obtain, correct inaccuracies, and delete their data, balance a reasonable trade-off between accuracy and privacy. Moreover, the evaluation system should allow individual information for legitimate ends and avoid ignoring the rights of personalities and the community.

According to ACM ethnical code 1.1 [10]. The car price evaluation system should consider whether the results of their efforts will respect diversity should use in a socially responsible way. The car price evaluation system's natural language processing runs the risk of exploiting and reinforcing the societal biases present in the underlying data. For example, the "," is the customary decimal notation in North American, but not familiar with the person who uses the other expression in decimal notation. Another example is that the number "13" is unlucky in western culture, and some of the car colours mean luck, but some of them are not. The evaluation system should have a worldwide perspective.

Fairness in cognitive computing is an important topic. The fairness may primarily come from data analytics and machine learning models, as they are widely used in cognitive computing. The underlying reason for the fairness issue is from two sides: 1) unfairness from data. 2) unfairness amplification from a model. The unfairness is from data. If a cognitive system is missing the data, it may cause fairness issues. The unfairness amplification from models thus, the car's price evaluation should foster fair participation of all people. That means all the person's requests and should be displayed in the car price evaluation system. According to ACM ethnical code 1.4 [10]. The car price evaluation system should make the car product data about the criteria for evaluating and recommending their product. The car evaluation system should aim for a robust, updated, and verified data set to guarantee procedural fairness. The designer should complete and update the database they are working from.

7 CONCLUSIONS

In this report, we presented a car evaluation system based on cognitive computing. This system can automatically search data from the Internet based on the user's input and output an estimated price. Moreover, the system will always focus on the latest real car data because of the data searching method. Our current system only searches data from one website. In the future, the system should get data from more data sources. We believe this system can help car buyers evaluate the value of a used car, especially for those who do not know the car market very well and do not have much time to do detailed research.

REFERENCES

[1] Khillare S.A. Ajitkumar M. Pundge and C. Namrata Mahender. 2016. Question Answering System, Approaches and Techniques: A Review. *International Journal*

of Computer Applications 141, 3 (May 2016), 34–39. https://doi.org/10.5120/ijca2016909587

[2] Frank Anette, Krieger Hans-Ulrich, Xu Feiyu, Uszkoreit Hans, Crysmann Berthold, Jörg Brigitte, and Schäfer Ulrich. 2007. Question answering from structured knowledge sources. *Journal of Applied Logic* 5, 1 (2007), 20–48. https://doi.org/10.1016/j.jal.2005.12.006 Questions and Answers: Theoretical and Applied Perspectives.

[3] Amin Beheshti, Shahpar Yakhchi, Salman Mousaeirad, Seyed Mohssen Ghafari, Srinivasa Reddy Goluguri, and Mohammad Amin Edrisi. 2020. Towards Cognitive Recommender Systems. *Algorithms* 13, 8 (2020). https://doi.org/10.3390/a13080176

[4] Drsantosh Bharti and Korra Babu. 2017. Automatic Keyword Extraction for Text Summarization: A Survey. *Computation and Language* (04 2017), 1–12. arXiv:1704.03242

[5] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. 1992. A Training Algorithm for Optimal Margin Classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory* (Pittsburgh, Pennsylvania, USA) *(COLT '92)*. Association for Computing Machinery, New York, NY, USA, 144–152. https://doi.org/10.1145/130385.130401

[6] Leo Breiman. 2001. Random Forests. *Machine Learning* 45, 1 (2001), 5–32. https://doi.org/10.1023/A:1010933404324

[7] Sanjay K. Dwivedi and Vaishali Singh. 2013. Research and Reviews in Question Answering System. *Procedia Technology* 10 (2013), 417–424. https://doi.org/10.1016/j.protcy.2013.12.378

[8] David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer, and Chris Welty. 2010. Building Watson: An Overview of the DeepQA Project. *AI Magazine* 3 (2010), 59–79. Issue 31.

[9] Eric Goldman. 2020. An Introduction to the California Consumer Privacy Act (CCPA). *Santa Clara University Legal Studies Research paper* (2020), 1–7. http://dx.doi.org/10.2139/ssrn.3211013

[10] D. Gotterbarn, B. Brinkman, C. Flick, M. S. Kirkpatrick, K. Miller, K. Vazansky, and M. J. Wolf. 2018. ACM Code of Ethics and Professional Conduct. https://www.acm.org/code-of-ethics

[11] Poonam Gupta and Vishal Gupta. 2012. A Survey of Text Question Answering Techniques. *International Journal of Computer Applications* 53, 4 (September 2012), 1–8. Full text available.

[12] Konrad Höffner, Jens Lehmann, and Ricardo Usbeck. 2016. CubeQA—Question Answering on RDF Data Cubes. In *International Semantic Web Conference*. Kobe, Japan, 325–340. https://doi.org/10.1007/978-3-319-46523-4_20

[13] Torsten Hothorn, Kurt Hornik, and Achim Zeileis. 2006. Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical statistics* 15, 3 (2006), 651–674. https://doi.org/10.1198/106186006X133933

[14] G. V. Kass. 1980. An exploratory technique for investigating large quantities of categorical data. *Applied Statistics* 29, 2 (1980), 119–127. https://doi.org/10.2307/2986296

[15] Boris Katz. 1997. Annotating the World Wide Web Using Natural Language. In *Computer-Assisted Information Searching on Internet* (Montreal, Quebec, Canada) *(RIAO '97)*. 136–155.

[16] Samuel F Kava. 2019. The Extraterritorial Application of the Sherman Anti-Trust Act in the Age of Globalization: The Need to Amend the Foreign Trade Antitrust Improvements Act (FTAIA) & Vigorously Apply International Comity. *J. Bus. & Tech. L.* 15 (2019), 135.

[17] Fangtao Li, Xian Zhang, and Xiaoyan Zhu. 2008. Answer validation by information distance calculation. In *Proceedings of the 2nd workshop on Information Retrieval for Question Answering (IR4QA)*. Manchester, UK, 42–49. https://doi.org/10.3115/1641451.1641457

[18] Wei-Yin Loh. 2011. Classification and regression trees. *WIREs Data Mining and Knowledge Discovery* 1, 1 (2011), 14–23. https://doi.org/10.1002/widm.8 arXiv:https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/widm.8

[19] Y Matsuo and Mitsuru Ishizuka. 2004. Keyword Extraction From A Single Document Using Word Co-Occurrence Statistical Information. *International Journal on Artificial Intelligence Tools* 13 (05 2004), 157–169.

[20] Taiki Miyanishi, Kazuhiro Seki, and Kuniaki Uehara. 2010. Hypothesis Generation and Ranking Based on Event Similarities. In *Proceedings of the 2010 ACM Symposium on Applied Computing* (Sierre, Switzerland) *(SAC '10)*. Association for Computing Machinery, New York, NY, USA, 1552–1558. https://doi.org/10.1145/1774088.1774421

[21] Paolo Nesi, Gianni Pantaleo, and Gianmarco Sanesi. 2015. A Distributed Framework for NLP-Based Keyword and Keyphrase Extraction From Web Pages and Documents. 21st International Conference on Distributed Multimedia Systems, Hyatt Regency, Vancouver, Canada, 1–7. https://doi.org/10.18293/DMS2015-024

[22] Fabio Paiva, José Costa, and Claudio Silva. 2014. An Ontology-Based Recommender System Architecture for Semantic Searches in Vehicles Sales Portals. In *International Conference on Hybrid Artificial Intelligence Systems* (Salamanca, Spain). Springer, 537–548.

[23] Voigt Paul and Von dem Bussche Axel. 2017. *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer.

[24] Anselmo Peñas, Álvaro Rodrigo, Valentín Sama, and M. Verdejo. 2008. Testing the Reasoning for Question Answering Validation. *Journal of Logic and Computation* 18 (06 2008). https://doi.org/10.1093/logcom/exm072

[25] Baptiste Rocca. 2019. Introduction to recommender systems. https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada

[26] Sifatullah Siddiqi and Aditi Sharan. 2015. Keyword and Keyphrase Extraction Techniques: A Literature Review. *International Journal of Computer Applications* 109 (01 2015), 18–23. https://doi.org/10.5120/19161-0607

[27] Wold Svante, Esbensen Kim, and Geladi Paul. 1987. Principal component analysis, In Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists. *Chemometrics and Intelligent Laboratory Systems* 2, 1, 37–52. https://doi.org/10.1016/0169-7439(87)80084-9

[28] Masatsugu Tonoike, Takehito Utsuro, and Satoshi Sato. 2004. Answer Validation by Keyword Association. In *Proceedings of the 3rd Workshop on Robust Methods in Analysis of Natural Language Data* (Geneva) *(ROMAND '04)*. Association for Computational Linguistics, USA, 95–103.

[29] Petros Xanthopoulos, Panos M. Pardalos, and Theodore B. Trafalis. 2013. *Linear Discriminant Analysis*. Springer New York, New York, NY, 27–33. https://doi.org/10.1007/978-1-4419-9878-1_4

[30] Kuo Zhang, Hui Xu, Jie Tang, and Juanzi Li. 2006. Keyword Extraction Using Support Vector Machine. In *Advances in Web-Age Information Management*, Jeffrey Xu Yu, Masaru Kitsuregawa, and Hong Va Leong (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 85–96.

[31] Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alex Smola, and Le Song. 2018. Variational Reasoning for Question Answering with Knowledge Graph. In *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*. New Orleans, Louisiana, USA, 6069–6076.

[32] Yu Zhang and Dit-Yan Yeung. 2011. Semisupervised Generalized Discriminant Analysis. *IEEE Transactions on Neural Networks* 22, 8 (2011), 1207–1217. https://doi.org/10.1109/TNN.2011.2156808