Analyzing Unstructured Data: Dating App Algorithms

Nicole Ziola
Rady School of Management
University of California, San Diego

Jerry Wu
Rady School of Management
University of California, San Diego

Bixiao (Chloe) Feng
Rady School of Management
University of California, San Diego

Wei Zhou
Rady School of Management
University of California, San Diego

GitHub Link: https://github.com/rsm-b3feng/MGTA415-Final-Project.git

Abstract

The increasing popularity of dating applications has generated a wave of user data, presenting the opportunity to leverage machine learning techniques to enhance matchmaking. This study explores the development of an algorithm that conducts matchmaking based on similarity and compatibility (in this context, “compatibility” is referring to sexual orientation matching, as we do not have ground truth on long-term outcomes). Using a dataset containing various data points on OkCupid application users, we conduct multiple techniques to optimize pairing. We will undergo feature engineering to identify key factors in matchmaking, and experiment with different algorithms, including K-means clustering and similarity-based matching, to demonstrate how data-driven matchmaking can effectively identify potential partners. This research contributes to the growing applications of real-world implementations of text-based algorithmic pairing.

Dataset Exploratory Analysis

Statistics and Properties

The dataset, sourced from Kaggle, comes from the dating app OkCupid. The dataset consists of 59, 946 entries with 31 columns. Key information includes:

- User Demographics: age, sex, orientation
- Lifestyle Variables: drinking, smoking, drugs, diet
- Education Level, Occupation, and Location
- Essays: user-written responses to prompts

One potential problem with the data is the vast number of missing values. Since responder input is not mandatory, there are many columns that have a high number of missing information, including body type, diet, drugs, income, offspring, pets, and religion. Many essays have null values, likely due to users responding to 2-3 prompts and not all 10. This can potentially disrupt our methodology because the similarity and cluster analysis are not reliable when we are missing information for the majority of datapoints.

Demographic Representation

In other contexts, similarity-based matching does not have to take compatibility in the context of orientation and gender into account. In this case, we need to be aware of the orientation distribution because this is going to be an integral piece of how we match users.

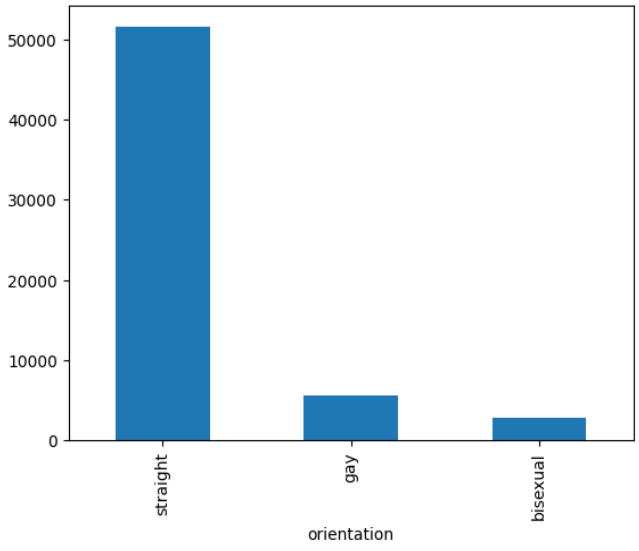

Figure 1: Orientation distribution.

Word Cloud Analysis

For each of the ten essay prompts in the data, we conducted word analysis using the WordCloud package to create word clouds of the most popular words in each essay. The dataset does not have the accompanying questions, so the popular words are not extraordinarily meaningful, but since we are doing similarity matching based on the text data, we are interested in common words.

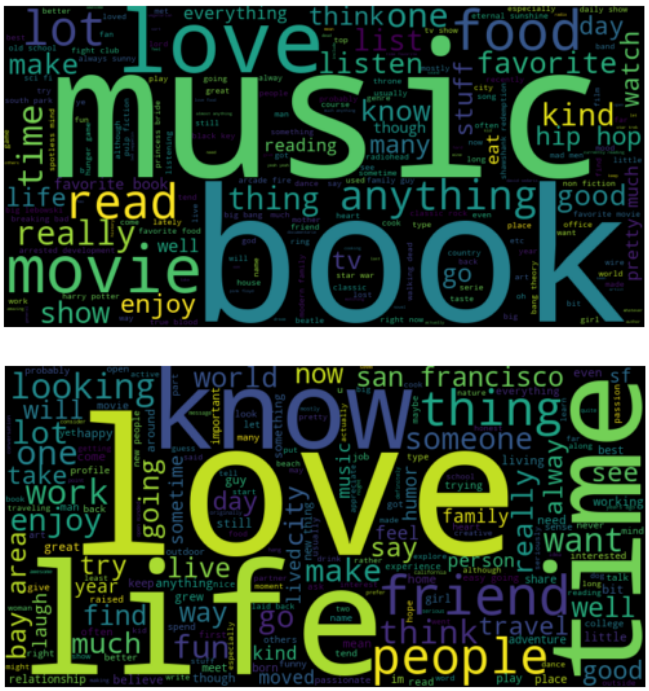

Figure 2: Example Prompt Word Cloud.

Location Analysis

The location analysis shows us that most users are in the Bay Area. Specifically, over 50% of the users in the data set are in San Francisco. This streamlines the matchmaking process because cross-referencing for location compatibility will not cause issues since the majority of people are in the same central location.

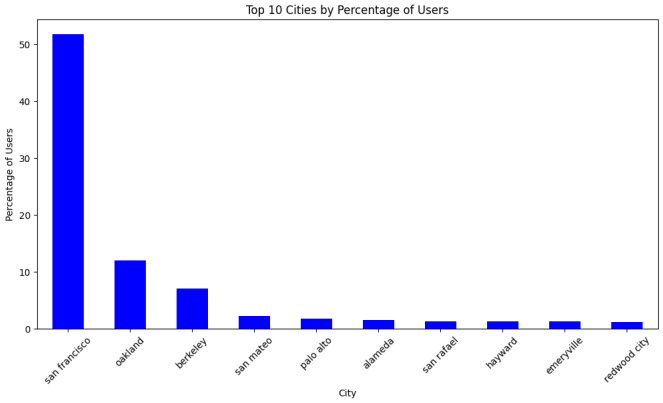

Figure 3: Percentage of users per city.

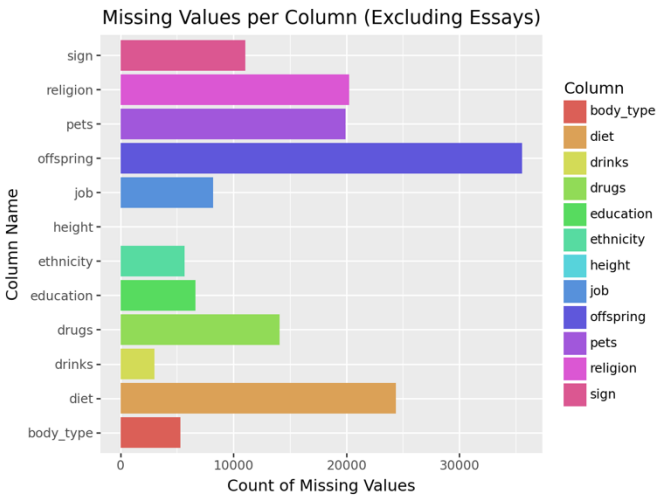

Figure 4: Missing Values per Variable

A notable concern in the dataset is the frequency of missing values. Since this is real data from a dating app, users are not required to disclose any information. Therefore, we must be aware of this and cautious as we move forward with our analysis.

Literature Review

Dataset Background

The explosion of dating applications has resulted in an abundance of user-generated data, offering the opportunity for data-driven matchmaking. Our dataset comes from OkCupid, with prior Kaggle users utilizing the data for matchmaking via K-means clustering. However, their clustering primarily focuses on using the numerical data available. We aim to build on this by incorporating the text data.

Similar Datasets

In addition to OkCupid data, other datasets have been utilized for similar experiments regarding matchmaking research, such as data generated by Tinder and eHarmony. Additionally, non-dating related datasets from apps such as Good Reads and Yelp have been used to develop similarity-based recommendation methods like what we aim to use on the OkCupid data. For example, data from MovieLens has been used to experiment with preference-based matchmaking since it contains extensive user-generated data (Harper & Konstan, 2016). Similarly, social media datasets from platforms like Facebook and Twitter have been analyzed for compatibility predictions based on shared interests and engagement patterns (Kosinski et al., 2013).

Current State-of-the-Art

With the proliferation of dating apps, applying machine learning to online dating has gained increasing attention by researchers. K-means clustering has been widely used to segment users based on shared characteristics, enabling more personalization of recommendations (Xia et al., 2020).

Another common approach is collaborative filtering, which analyzes similarities in user interactions and preferences (Guan & Li, 2018). Deep learning approaches, including neural networks and reinforcement learning, have been integrated into matchmaking systems to optimize recommendations dynamically based on user feedback (Lee et al., 2023).

Similarity to Existing Works

This research aligns with previous studies exploring similarity-based techniques matchmaking. Nguyen et al. (2019) explored using cosine similarity and Jaccard similarity measures to match individuals based on shared attributes, an approach that is integral in our methodology. Additionally, past research on feature engineering has identified key matchmaking factors, such as demographic data, shared interests, and textual self-descriptions, which are key in our model development (Rudder, 2014). Our study builds upon prior research approaches by taking a hybrid method that combines both clustering and similarity-based matching to enhance data-driven matchmaking.

Methods

1. Baseline

The baseline model for matchmaking is K-means clustering based solely on numerical features. Due to the unlabeled nature of the data, we do not have a baseline metric on performance; however, we consider this the simplest approach of clustering and matching. We aimed to improve on the baseline model by incorporating additional features and methods, beginning with K-means clustering on both numerical features and text data.

2. K-Means with Text Data

Our first approach was to add in text data for the K-means clustering, building on the baseline model by incorporating text similarity into the clustering method. Prior to running the K-means algorithm, we performed the following feature engineering and preprocessing:

1) Use mean imputation on missing values for numerical columns.
2) Change sex into a categorical variable.

3) Simplify body type, education, drinking, drugs, smoking, and pets into a smaller number of categories.
4) Perform standard text preprocessing and stack all essay responses into one document for analysis.
5) Use TF-IDF on the essay document to vectorize the text.

Next, we ran K-means analysis with ten clusters and included all features, including numerical, categorical, and text data.

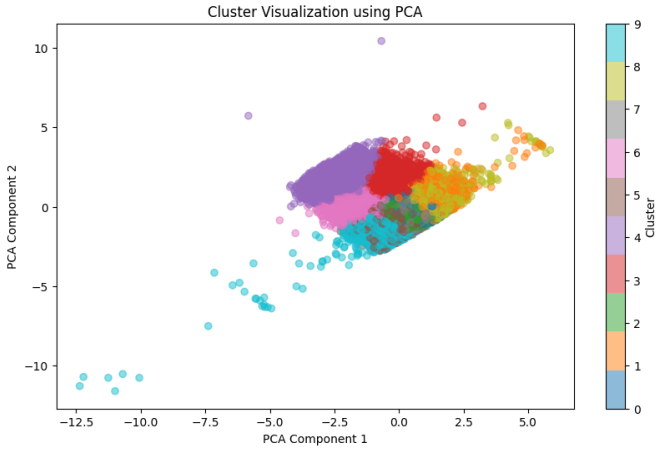

Figure 5: K-means Cluster Visualization

To examine the effectiveness of our clusters, we conducted a case study by randomly selecting one user and generating compatible (based on orientation and gender) counterparts ranked by affinity. Then, we could manually evaluate how successful the matching process was.

We discovered that the matching does well with clustering and the counterparts matched very well in similarity; however, this is not technically a measure of how truly compatible the users are as this a relatively simple measure that builds slightly on the baseline.

3. Contextualized Weak Supervision

The next approach we took was experimenting with three different algorithms that fall under contextualized weak supervision. This helps us combat the problem of having a lack of labeled data as we leverage a minimally supervised approach by manually defining categories for matching.

Firstly, we divide user descriptions (where users describe themselves) and expectations (where users describe their expectations for a romantic partner) into ten groups and manually define a category label name for each one. Then, we match labels accordingly. Specifically, we manually define ten categories that are each associated with 3 or 10 seed words. State-of-the-art methods such as WeSTClass and ConWear require significant computational resources, so we adopt a structural approximation instead. We determine category assignment by comparing frequency or similarity between seed words and text within each category (figure 6). To ensure the categories are both mutually independent and representative, as well as that seed words within the categories are meaningful, we generate them using structured dialogue with ChatGPT.

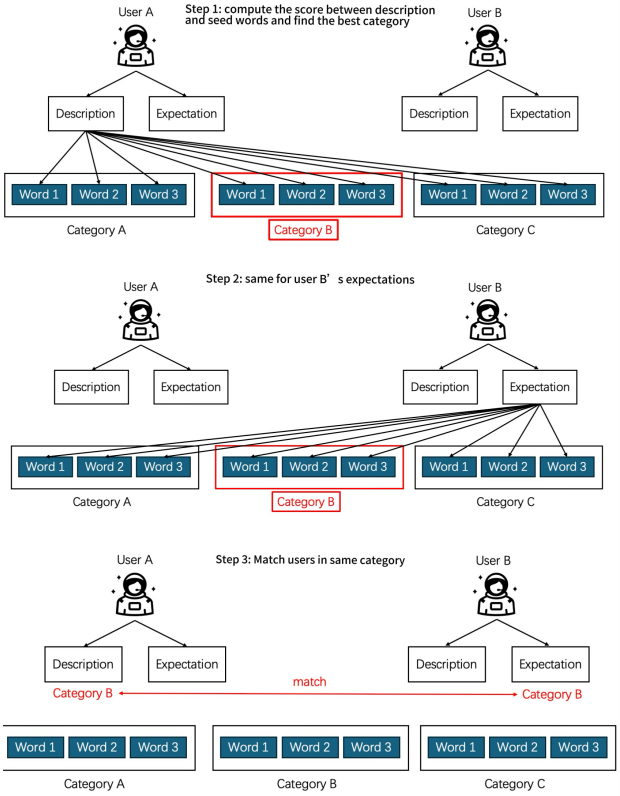

Figure 6: General Flow Chart

3.1 Approach 1: String Matching

We assume that if certain keywords appear frequently in a document, the document is likely to belong to the corresponding category. Based on this assumption, we calculate the frequency of seed words from different categories appearing in each user's describe section and assign the label of the category with the highest frequency.

Initially, we determined labels solely based on the total frequency of all seed words. However, this approach presented a limitation: If a specific seed word (e.g., love) is widely used across different contexts, many users may be incorrectly classified into that category.

To mitigate this issue, we refined our method by first assessing the coverage of seed words in the text before considering their frequency. Specifically, we first count the number of unique seed words from each category that appear in each text. If multiple categories contain the same number of unique seed words, we then compare the total occurrences of all three seed words within the text. This refined approach effectively prevents overgeneralization into a single dominant category and provides a more accurate representation of users' actual preferences.

This approach has a limitation: the set of seed words is finite. For instance, the seed words for the Adventurous Outdoorsy category are travel, hike, and adventure. However, if a text describes outdoor activities such as camping, mountain climbing, or skiing, it may not be classified into this category due to the absence of these specific keywords.

As a result, when the number of seed words is set to three, one-third of the describe entries and two-thirds of the expect entries cannot be classified. Expanding the seed word list to ten reduces the number of unclassified instances, but still, 28% of the describe entries and two-thirds of the expect entries remain unclassified.

The higher proportion of missing classifications in expect than describe is primarily because most users provide only a short sentence for their expectations, whereas their self-descriptions tend to be more informative.

3.2 Approach 2: Similarity Comparison on Entire Document

We refined approach 1 by incorporating similarity comparisons. If all seed words are missing, we leverage similar words in the text to minimize the number of unclassified instances. There're two ways to measure similarity, cosine and Euclidean distance. We chose cosine based on our observation that the seed words from different categories have significant differences in vector magnitudes, which could result in most users being classified into a single dominant category. Cosine similarity, by ignoring vector magnitude and focusing on angular difference, provides a more balanced classification approach.

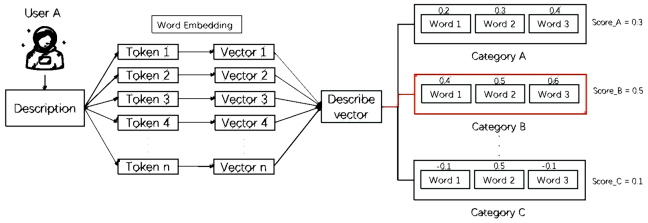

Figure 7: Approach Two Flow Chart

We first vectorize the describe text using the word2vec-google-news-300 model, obtaining a matrix of size (number of words in the document × 300). We then compute the column-wise mean of this matrix to derive a single 1 × 300 vector representation for the entire document.

Next, we apply the same word embedding method to the seed words of each category. For a category with three seed words, we obtain a 3 × 300 matrix.

We then compute the cosine similarity between the document vector and each of the three seed word vectors within a category. The category is selected based on the following rules: 1. Compute the average cosine similarity of the three seed words for each category and select the category with the highest value. 2. If the highest average similarity among all categories is ≤ 0, then select the category with the highest individual similarity score. 3. If the highest individual similarity score among all categories is still ≤ 0, return None (figure 7).

The same classification procedure is applied to the expect text.

This approach effectively reduces the issue of unclassified instances present in approach 1. However, a small number of descriptions still cannot be classified due to insufficient information. (e.g. someone wrote the expectation as "at least 5'4"") With the three-seed-word setting, the proportion of unclassified entries is 1% for describe and 2% for expect.

Additionally, this approach has a limitation: averaging the vectors of all words compresses a significant amount of information. This approach assumes that all words contribute equally to the final representation, which is not always realistic.

3.3 Approach 3: Similarity Comparison on Key Words

To emphasize the contribution of important words, we use TF-IDF to select the top 10 most important words from each describe entry. We set min_df = 2 to ignore words that appear only once, reducing noise caused by typographical errors. For sentences containing fewer than ten words, we select all words with a TF-IDF score > 0.

Next, we vectorize all selected keywords using word2vec-google-news-300, resulting in a (number of keywords × 300) matrix. We then compute the cosine similarity between each keyword and each category's seed words. For each category, we take the highest similarity score among all keyword–seed word pairs as the category's final score. The category with the highest score is assigned as the label (figure 8).

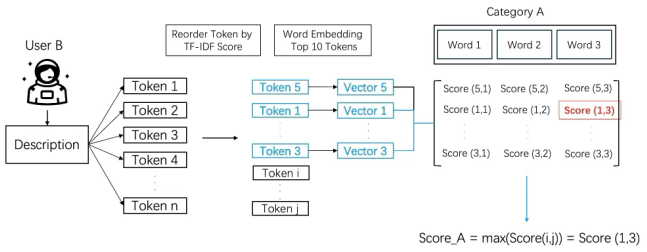

Figure 8: Approach Three Flow Chart

3.4 Combination of Three Approaches

Due to the lack of labeled data, we could only randomly sample instances for evaluation. Upon manual inspection, we found that each of the three approaches had its own advantages. Therefore, we selected the most frequently assigned category across all six approaches as the final label.

4. LDA Topic Analysis

In Topic Analysis Approach, we are using only the text essay data and do not have access to other information. This means when we do match making with it, it is possible we mismatch people, as we are not able to identify user's sex and the orientation. We assume with further information like sex and orientation, the company can filter the people to recommend in the real application.

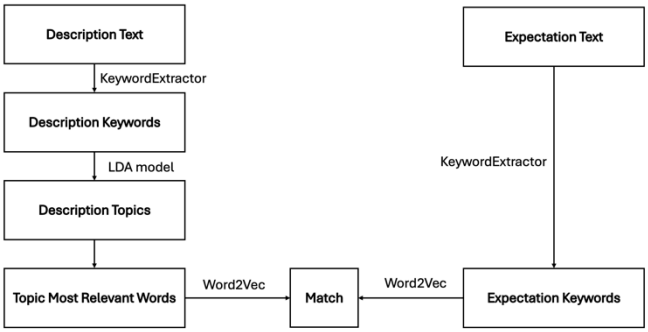

Figure 9: Flow Chart for LDA Analysis

We will do the same text processing on description and expectation texts to make sure the match is made based on the same vocabulary. To implement our topic analysis approach, we will:

1) Process description and expectation texts and extract five keywords to have five representative keywords for description and expectation for each person in the dataset.

2) Conduct topic analysis on description text with LDA model, producing several topics summarizing different groups of people's self-description. Each topic now has a group of members that can be ranked by their probability of belonging to this topic.

3) Use expectation keywords to match with the topics' top five relevant words. To do this, we use google-news-300 Word2Vec to vectorize the expectation keywords and topic top five relevant words. Then, we can calculate the distance between the expectation keywords and each of the topic relevant words. We then select the top three closest topics, and we consider the users in these topics to be matches with the user who listed the same topic as their expectation.

Next, we used Coherence Score and visualizations to evaluate the model. This measures the semantic

similarity between high-probability words within a topic, helping determine whether the topics are meaningful and interpretable. Here we specifically use CV Coherence, which is based on word embedding similarity and cosine similarity between top words in a topic. It is usually better aligned with human judgments of topic quality.

Visualization can help us evaluate the LDA model qualitatively. We can observe topic overlap, topic size and topic distribution in the plot. With a good LDA model, we can observe well-separated, non-overlapping, and evenly distributed topics.

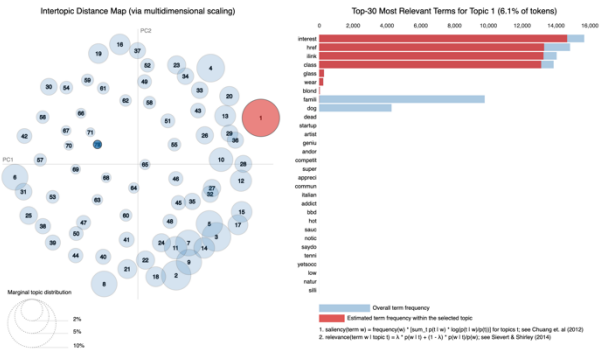

Figure 10: Topic Visualization

The Coherence Score of our model was 0.44. In addition, we also randomly selected some users to manually evaluate their matches. We found that users with long and informative descriptions had better matching outcomes. Overall, this method was strong when descriptions were offered sufficient information, despite limitations.

Results

To evaluate and compare different approaches, we randomly select some expectation texts, and we provide 3 matches from each approach for the same expectation text. We manually evaluate them, but we ignore the sex and orientation during evaluation due to the limitation of the current dataset. Here we present two examples:

Expectation 1:

you have hair...and teeth. you like to have fun, and you know how to. you were born a female.

Match:

Method	Description for the Match	Good Match
K-Means	i am a faithful guy who enjoys being a pc consultant and cyber junkie. i am honest, truthful, and sincere	✗
	one might find me relaxing at home on a Friday night, recovering from a busy work week or hosting a casual game night with friends. sometimes it might be checking out a new restaurant or singing my heart out with friends at karaoke.	✓
	i love games (video more than board), and silly, smart people. i read the new york times in morning, and spend way too much time talking to people on gchat	✓
Contextualized Weak Supervision	I'm not someone who needs to be in a relationship to feel complete. I like myself (though I suppose I could be more ambitious) and don't mind being alone. in fact, I enjoy it. that way, I always get to do what I want...	✗
	chill guy here, masculine, smart, fun, honest, and loyal. I'm looking to get to know other guys to hang out with and go from there. good people and good conversation are important...	✓
	I'm pretty much a big goofy free spirit, I'm honest but can be very sarcastic. I love to joke around, make funny faces/sounds and make people laugh. I graduated from a small private college outside of boston and then moved out to the bay about three years ago...	✓
Topic Analysis	I'm just a girl in the city trying to have it all.	✓
	fun I'm like. born Thailand. the parents they bring me China. i in us. my aunt adopt me here now my home here. look for nice boyfriend. i like the asian guy nice or if cute then i go the other one too hehe. or try the american one i try	✓
	hi, i'm david. i'm from virginia. i moved to san francisco less than a year ago and am doing reasonably well here. i make music.	✗

Expectation 2:

if you like having fun, like adventures! dont judge. and dont care about deep conversations or politics (thank you in advance!)

Match:

Method	Description for the Match	Good Match
K-Means	i am a single mom....3 live at home with me and my oldest is in the marines. recently went back to school and i am working towards a degree in social services. i was lucky enough to land a job i love and that i find very fulfilling.	✗
	one might find me relaxing at home on a Friday night, recovering from a busy work week or hosting a casual game night with friends. sometimes it might be checking out a new restaurant or singing my heart out with friends at karaoke.	✗
	i am a faithful guy who enjoys being a pc consultant and cyber junkies. i am honest, truthful, and sincere	✗
Contextualized Weak Supervision	i am looking for a fun, intelligent, laid back and most importantly drama free girl who I can explore the city with and with whom i can at least be friends with if nothing else ensues. i like to live life to its fullest and do what makes me happy and hope tha you do too.	✓
	on a lighter note, and there is a lighter note. i like the accidents of life. the chance meeting of an old friend, the happening of a new band, or new discovery. i love the sweet smell of weeds and flowers on a morning hike, but i also like the with whiffs and riffs of music.	✓

	i love the city life and everything it has to offer, bu t i really need nature. i love hiking, camping, gettin g dirty, biking, and staying active. even if it's just a bike ride over the gg bridge.	✓
Topic Analysis	i am one of the nicest, funniest, committed, most a thletic people you'll ever meet ... i love sports, (esp ecially soccer) i have a dog. i am trying to speak a handful of different languages (spanish, german, fr ench, dutch, and italian) i also take martial arts. an d i will always stay committed in a relationship.	✓
	i love the outdoors, but i'm not an adrenaline junk y type...i'd rather go for a hike than mountain clim bing, surf than waterski, yoga than krav maga. i love to travel...my favorite part is to learn more a bout the people and their food and culture. i'd rat her stay in a small out of the way town than the m ain tourist cities...	✓
	if i could do anything i would sell sno-cones on the beach. Not that i'm lazy it just sounds really peace ful and stress free.	✖

Limitations

Each of our matchmaking algorithms build off the baseline simplified model by adding additional features, incorporating text data, and utilizing algorithms for text-similarity pairing. However, each approach carries some form of limitation.

The K-means approach does not distinguish between expectation text and description text. For example, a user may self-describe as introverted and a homebody, but still seek an extraverted, adventurous partner. Since the clusters group all text data together in one document, it is unable to distinguish between these in cluster development, a significant limitation in this approach.

The Contextualized Weak Learning algorithm also carries limitations.

First, all approaches assume that each user belongs to a single category, making it difficult to accurately classify individuals with diverse interests. In dating apps, most users tend to present multiple facets of themselves, leading to high variance in our classification results.

Second, none of the approaches account for negative exclusions. For example:

1. If a user's expect states, "You should message me if you're not a nerdy guy," the model would delete the stop word "not" and likely classify them under Intellectual/Bookish based on "nerdy", which contradicts their actual preference.
2. Similarly, if someone's expect states, "You should message me if you accept an unconventional family and want to have a child," the presence of keywords like child and family may lead to classification under Family-Oriented. However, this expectation does not align with the typical preferences of users labeled as Family-Oriented.

Finally, the LDA method is reliant on topics from descriptions. Therefore, users with short, incomplete, or inaccurate descriptions will not yield strong results. Additionally, all approaches do not take data on orientation or gender, meaning this filter would have to be applied prior to applying those on the data.

Conclusion

Considering our limitations, case studies, and methodologies, we conclude that a combination of all three methods will sufficiently counteract the limitations of alternative methods.

Our evaluations were done through sampling and manual examination due to the lack of ground truth labeling. We found that all three were sufficient in matching, but due to limitations, could not be used independently with confidence.

In addition to limitations, each method also has strengths and best use cases. Thus, combining the three methods can help combat limitations of others and improve matching. In application, we send more than one match to users, so dating app algorithms have the option to use several forms of matchmaking.

For example, K-means clustering incorporates more features than other algorithms. This allows us to include features such as age, height, social preferences, etc., and conduct feature engineering to improve results. Our Contextualized Weak Learning Algorithm prioritizes hobby matching, encouraging compatibility based on interests and hobby interests. Finally, the LDA approach considers multiple aspects and takes expectation and description matching into account.

References

Binns, R. (2020). *On the apparent conflict between individual and group fairness.* In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (pp. 514-524). ACM.

Guan, W., & Li, L. (2018). *A survey on collaborative filtering algorithms for recommender systems.* Journal of Artificial Intelligence Research, 62, 217-253.

Harper, F. M., & Konstan, J. A. (2016). *The MovieLens datasets: History and context.* ACM Transactions on Interactive Intelligent Systems (TiiS), 5(4), 1-19.

Kosinski, M., Stillwell, D., & Graepel, T. (2013). *Private traits and attributes are predictable from digital records of human behavior.* Proceedings of the National Academy of Sciences, 110(15), 5802-5805.

Lee, J., Kim, H., & Cho, J. (2023). *Deep learning-based matchmaking: An approach to dynamic compatibility prediction in online dating.* Expert Systems with Applications, 220, 119671.

Nguyen, T., Pham, H., & Tran, D. (2019). *Measuring user similarity in online dating applications using cosine and Jaccard similarity.* In *Proceedings of the 2019 International Conference on Machine Learning and Data Science* (pp. 89-97).

Rudder, C. (2014). *Dataclysm: Who we are when we think no one's looking.* Crown.

Xia, S., Wang, J., & Zhang, L. (2020). *User clustering and preference modeling in recommender systems: A K-means*

clustering approach. Journal of Information Science, 46(5), 702-716.