

AI-Powered Member Matchmaking Agent

Author

Vedansh Rotale

Research Scholar

vedanshrotale9@gmail.com

Soham Bobade

Research Scholar

sohambobade381@gmail.com

Aryan Tekade

Research Scholar

aryantekade9@gmail.com

Shashank Dudhe

Research Scholar

shashankdudhe373@gmail.com

Mayur Agrawal

Research Scholar

agrawalms2005@gmail.com

Dr. A. A. Tayade

Guide, PRPCEM

akhileshtayade18@gmail.com

Abstract

Professional networking is a big part of how entrepreneurs, investors, mentors, and business leaders actually end up finding the right opportunities and the right people. But as organizations grow, going through thousands of member profiles by hand stops being practical, and a lot of good connections simply get missed. This paper presents an AI-Powered Member Matchmaking Agent built for TiE (The Indus Entrepreneurs), a global non-profit that supports entrepreneurship and innovation, to deal with exactly this problem. The system lets a member type their requirement in plain English — the kind of entrepreneur, industry, location, company size, or business interest they are looking for

— and the backend takes it from there. It processes member profile information using TF-IDF based keyword matching along with sentiment analysis, and returns the most relevant matches ranked by profile similarity and business context. The system is built to keep working as the member database keeps growing, the system doesn't need any major changes to keep up with it. Down the line, it can also pull in things like company websites, news pieces, and other publicly available business info to fill in the gaps in existing profiles. Member privacy isn't an afterthought here either — confidential data stays locked down the whole time, even as the matching itself keeps getting sharper. Overall, the proposed solution cuts down manual effort and makes matches more accurate, which should help TiE members network and collaborate a lot more effectively.

Keywords – semantic matchmaking, natural language processing, TF-IDF, sentiment analysis, professional networking, recommendation systems.

Introduction

AI and NLP have really changed the way organizations deal with information and connect people to the right opportunities. Most professional networking platforms carry thousands of member profiles, and once a directory reaches that size, plain keyword search stops being enough — people either don't find the right match, or they find too many irrelevant ones. TiE (The Indus Entrepreneurs) runs into this exact problem. With chapters spread across different cities and a member base that's pretty diverse, actually tracking down the right business contact still mostly comes down to manual browsing, or just getting lucky at an event.

This is the gap the AI-Powered Member Matchmaking Agent is meant to fill. Instead of forcing members to scroll through directories or rely on filters that only understand exact keywords, the system takes a conversational query in plain English and figures out what the member actually needs. It reads member profile data, applies NLP techniques to understand the query, and uses TF-IDF based matching along with sentiment analysis to rank the most suitable members.

Recommendation quality is expected to keep improving as the profile database grows and gets enriched with information pulled from public sources like company websites and news articles. Throughout all of this, data privacy and secure access stay core design requirements, since member information at an organization like TiE is sensitive by nature.

Aims and Objectives

The proposed AI-Powered Member Matchmaking Agent aims to build an intelligent, NLP-driven system that helps TiE members discover relevant professional connections without having to manually search through the member directory.

1. To develop an AI-powered matchmaking system that lets TiE members find relevant connections using simple, conversational English queries.
2. To apply Natural Language Processing techniques for understanding user queries and analyzing member profile information.
3. To design a recommendation engine based on TF-IDF keyword matching and sentiment analysis that improves the relevance of suggested matches.
4. To build an efficient, user-friendly platform that reduces the manual effort involved in searching for suitable members within the TiE community.
5. To ensure secure handling of member data while keeping the system private, scalable, and accurate as it grows.

Literature Review

Recommendation systems and professional networking platforms have progressed beyond the era of plain keyword search. I remember when earlier recommendation systems mostly matched words, so a relevant result could be missed simply because the wording was slightly different. Today recommendation systems use technology to capture what a user really means rather than just scanning for matching words. These new approaches combine search and recommendation algorithms that try to understand the intent, behind the words. Several studies back this up — matchmaking that factors in user profiles, preferences, and context tends to noticeably outperform approaches that rely on keywords alone. There's also been some work around enriching member or user profiles using public data, which generally leads to recommendations that feel more current and more meaningful.

1. **Deep Learning-Based Recommendation:** Zhang, Lu, and Jin (2021) went through a bunch of AI and deep learning approaches used across recommender systems and found they pretty consistently beat the older, simpler rule-based methods on accuracy.
2. **Adaptive Matchmaking:** Wang et al. (2015) built Friendbook around behavioral and lifestyle signals instead of just explicit profile fields, and it ended up matching people a lot better than systems that only look at what someone typed into their profile.
3. **Semantic Matching Over Keyword Search:** this is where Reimers and Gurevych (2019) come in— sentence embeddings pick up on meaning way better than keyword comparison does, so two people can still get matched even when they've worded things completely differently.
4. **Team Recommendation Systems:** Tuarob et al. (2021) put together RECAST, which recommends software teams based on both the skills needed and whether people would actually work well together. It beat older team-recommendation methods by a wide margin.
5. **Recommendation in Professional Networks:** Masciari, Umair and Ullah (2024) went through a lot of recent AI recommendation research and noticed that professional and business-networking use cases haven't gotten nearly as much attention as e-commerce or media recommendation has.
6. **Semantic Search for Idea and Knowledge Management:** Poveda, Westerski and Iglesias (2012) applied semantic search to an idea-management system. Meaning-based retrieval just works better than keyword matching once people start phrasing things in natural, open-ended language.

Table 1: Comparative Literature Review

Ref.	Author(s) & Year	Method / Technique	Application	Key Findings	Limitations	Research Gap
1	Zhang et al. (2021)	AI / Deep Learning Survey	Recommender Systems	Reviews accuracy gains from deep learning techniques	Broad survey, not business-specific	Needs domain-specific business recommendations
2	Wang et al. (2015)	Behavioral / Semantic Matching	Social Friend Recommendation	Behavioral signals outperform explicit-only profiles	Built for personal, not business, matching	Business-oriented recommendation required
3	Reimers & Gurevych (2019)	Sentence Embeddings (Semantic NLP)	General NLP / Semantic Search	Embeddings capture meaning beyond exact keywords	Not applied to business profile matching	Extend to business networking
4	Tuarob et al. (2021)	ML-Based Team Recommendation	Software Team Formation	Improves team-fit recommendations significantly	Focused on software teams only	Needs richer business context
5	Masciari et al. (2024)	Systematic Review	AI Recommendation Trends	Professional/business use cases remain underexplored	Review only, no new system proposed	Business opportunity matching needed
6	Poveda et al. (2012)	Semantic Search	Idea Management Systems	Meaning-based search outperforms keyword search	Applied to ideas, not people/profiles	Adapt for entrepreneurship and TiE

Limitations of Existing Systems

1. **Domain Mismatch:** Most existing recommendation work targets recruitment, academic collaboration, or e-commerce. Very little of it is built specifically for business and entrepreneurial networking.
2. **Limited Business Context:** Several systems handle explicit or implicit user preferences reasonably well but don't really model business-specific context such as industry fit or investment stage.
3. **No Integration with External Data:** Most systems work purely off internal data and don't combine private organizational records with public business information, which limits how rich a recommendation can get.
4. **High Resource Requirements:** A number of the deep-learning-based approaches need large training datasets and significant computing power, which isn't always practical for a mid-sized networking platform.
5. **Conclusion:** These gaps point to a clear need for a matchmaking system built specifically for business networking — one that understands conversational queries, works with the kind of data organizations like TiE actually have, and stays lightweight enough to run without heavy infrastructure. The proposed AI-Powered Member Matchmaking Agent is designed around addressing exactly this gap.

Proposed Methodology

Each module in this system was built, tested, and tweaked on its own before it got folded into the full platform — the whole thing came together through a fairly iterative, trial-and-error kind of process. Here's how that broke down stage by stage:

1. Requirement Analysis and Data Collection

Member profile data — including industry, role, company details, and business interests — is collected and organized into a structured format suitable for processing. Requirements from actual TiE use cases, such as the types of queries members are likely to ask, are studied to shape how the query-understanding module is built.

2. Data Preparation and Profile Enrichment

Collected member data is cleaned and structured before being fed into the system. Where possible, publicly available business information — company websites, news mentions, industry data — is used to enrich existing profiles, so that recommendations aren't limited to whatever fields a member filled in manually.

3. Natural Language Query Processing

When a member writes a question in English NLP works behind the scenes to discover what the member really means. NLP pulls out details such as the entrepreneur type, industry, location or business interest that are hidden in the sentence. In short NLP takes a worded question and turns it into a form that the recommendation engine can use.

4. Recommendation Engine Development

The core matching logic uses TF-IDF based keyword matching to measure similarity between the query and member profiles, combined with sentiment analysis to gauge the tone and context behind a member's stated interests. Together, these produce a ranked list of the most relevant member matches.

5. System Integration and User Interface

The query processing module, recommendation engine and member database are all integrated into a working platform. The platform provides an interface where members can type a query and view ranked results.

6. Testing and Evaluation

The system is tested using sample member datasets, with performance measured on recommendation accuracy, relevance of results, response time, and overall user satisfaction.

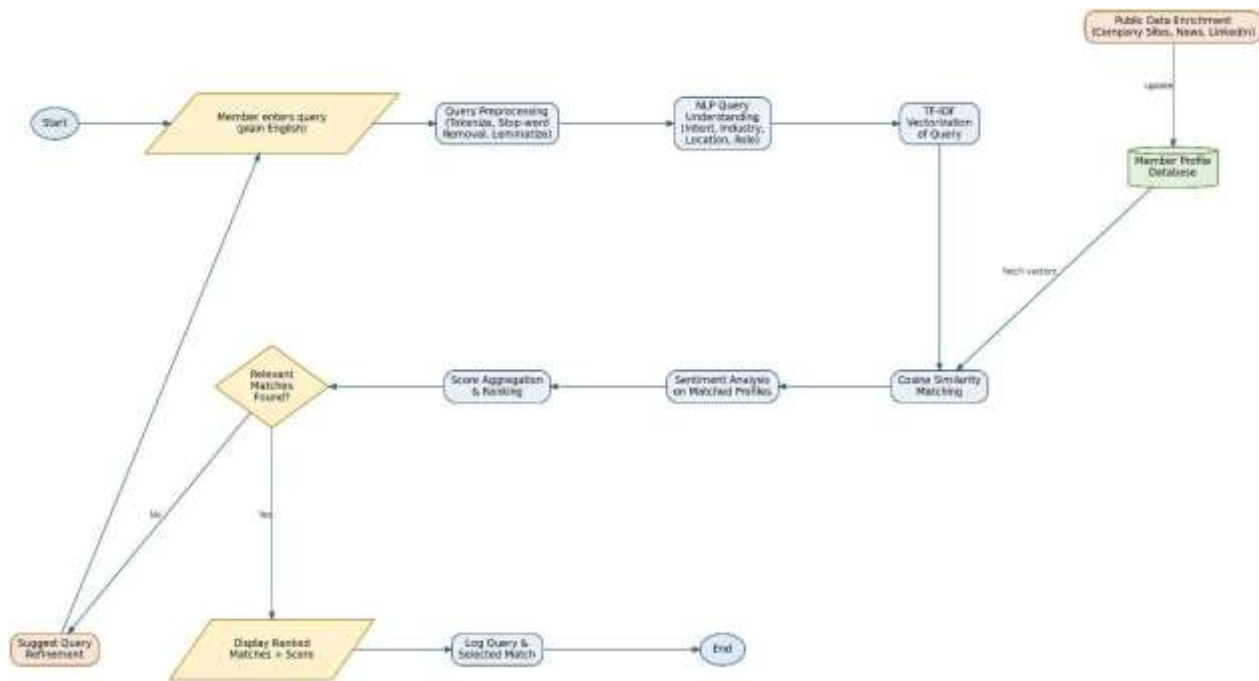


Figure 1: AI-Powered Member Matchmaking Agent

Start and Query Input: Member types their requirement in plain English, such as "fintech founder in Pune," instead of using fixed filters.

Query Preprocessing: I see how the system splits the query into words removes filler words and reduces words to their base form so that only important terms remain.

NLP Query Understanding: I notice that the system pulls out the main details from the query such, as the industry the job role and the location.

TF-IDF Vectorization: The cleaned query is turned into a number pattern that can be compared with member profiles.

Member Profile Database & Public Data Enrichment: Member records are refreshed regularly with information from company websites, news articles and LinkedIn.

Cosine Similarity Matching: The query vector is matched against stored profile vectors to locate the matches.

Sentiment Analysis, on Matched Profiles: The matched profiles are examined for tone and meaning, not for shared keywords.

Score Ranking: Similarity scores and sentiment scores are merged to create a ranked list of candidates.

Relevant Matches Found? (Decision): System checks if top matches clear a relevance threshold.

- **No** → member refines query, loops back to query input.
- **Yes** → system displays results.

Display Ranked Matches + Score: Member sees ranked matches with relevance scores. **Log Query & Selected**

Match: Query and chosen match are logged for future accuracy. **End:** Cycle completes; member can start a new query anytime.

Development Tools and Technologies

- **Programming Language:** Python
- **NLP Libraries** – NLTK and spaCy handle text preprocessing and query understanding
- **Matching Technique** – TF-IDF through scikit-learn, paired with sentiment analysis
- **Backend Framework:** Flask / Django for API and application logic
- **Database:** MySQL / MongoDB for storing member profile data
- **Frontend** – built with plain HTML, CSS, and JavaScript
- **Testing Tools** – manual browser testing, plus Postman for hitting the APIs

These tools together cover query processing, matching logic, data storage, and the interface members interact with.

Experimental Procedure

The system is built module by module — first the data pipeline, then query processing, then the recommendation engine

— and each module is tested on its own before being combined into the full platform. Once individual modules are working correctly, they are tested together using realistic sample queries, such as a member looking for "an early-stage fintech founder in Pune" or "an investor interested in manufacturing." The system's output is checked against what a human reviewer would consider a reasonable match, to judge whether the ranking logic is actually producing useful result.

Performance Evaluation and Validation:

A handful of metrics were used to judge how well the system actually performs — recommendation accuracy, how relevant the matches are, response time, and how satisfied users seem to be. To check accuracy, system-suggested matches were compared against matches a person would pick manually for the same query. Response time was tracked from the second a query gets submitted to when ranked results show up, just to make sure things stay fast even as the member database keeps growing. User satisfaction wasn't measured with any formal survey — it came from informally asking a small group of test users to try out a few sample queries and share what they thought.

Overall Methodology

Here is a simple outline of the process: Requirement Analysis → Data Collection & Enrichment → System Design → NLP Query Processing → TF-IDF & Sentiment-Based Recommendation Engine → Integration → Testing → Evaluation → Deployment.

This gives a clear path for building and testing the system. The end goal is to cut down on manual search effort and make the connections people find within the TiE community genuinely better.

Limitations and Recommendations

Data Dependency: The system's recommendation quality depends heavily on how complete and accurate the underlying member profile data is. Missing or outdated information can directly reduce match quality.

Computational Load at Scale: As the member database grows, running TF-IDF matching and sentiment analysis across a larger pool of profiles will need more computing power and memory to keep response times reasonable.

Limited Matching Depth: TF-IDF based matching, while effective for keyword-level similarity, doesn't fully capture deeper semantic meaning the way more advanced embedding-based techniques could.

Recommendation: Move Toward Semantic Matching: Future versions of the system could incorporate embedding-based semantic search to catch matches that TF-IDF alone would miss due to different wording.

Recommendation: External Data Enrichment: Bringing in structured public data feeds (news, company databases) rather than relying only on internal profiles could noticeably improve match relevance over time.

Conclusion

The AI-Powered Member Matchmaking Agent addresses a real, practical problem inside professional networking organizations like TiE — finding the right connection inside a large and growing member base without relying on manual searching. By combining NLP-based query understanding with TF-IDF matching and sentiment analysis, the system turns plain English queries into ranked, relevant member recommendations. It is designed to work with a growing dataset and to improve further as profiles are enriched with external business information, all while keeping member data secure and private. Overall, the system is meant to reduce manual networking effort, improve the relevance of connections made, and support stronger collaboration within the TiE community. Future work can build on this foundation by adding semantic matching and richer data enrichment to push recommendation quality even further.

References

- [1] Q. Zhang, J. Lu, and Y. Jin, "Artificial Intelligence in Recommender Systems," *Complex & Intelligent Systems*, Springer, Heidelberg, vol. 7, no. 1, 2021, pp. 439–457.
- [2] Z. Wang, J. Liao, Q. Cao, H. Qi, and Z. Wang, "Friendbook: A Semantic-Based Friend Recommendation System for Social Networks," *IEEE Transactions on Mobile Computing*, IEEE, vol. 14, no. 3, 2015, pp. 538–551.
- [3] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3982–3992.
- [4] S. Tuarob, N. Assavakamhaenghan, W. Tanaphantaruk, P. Suwanworaboon, S. U. Hassan, and M. Choetkiertikul, "Automatic Team Recommendation for Collaborative Software Development," *Empirical Software Engineering*, Springer, vol. 26, no. 4, 2021, pp. 1–53.
- [5] E. Masciari, A. Umair, and M. H. Ullah, "A Systematic Literature Review on AI-Based Recommendation Systems and Their Ethical Considerations," *IEEE Access*, IEEE, vol. 12, 2024, pp. 121223–121241.
- [6] G. Poveda, A. Westerski, and C. A. Iglesias, "Application of Semantic Search in Idea Management Systems," *International Conference for Internet Technology and Secured Transactions*, IEEE Xplore, 2012, pp. 230–236.