An End-to-end Tag-based Recommendation System for Verbal Reasoning Questions

Z. Yue
Southern University of Science and Technology
China
yuezx@mail.sustc.edu.cn

Y. Jiang
Southern University of Science and Technology
China
jiangyh@mail.sustc.edu.cn

D. Pan
Harbin Institute of Technology
China
m18345155020@163.com

Z. Luo
Southern University of Science and Technology
China
luozw@sustc.edu.cn

ABSTRACT

Developing a verbal reasoning question[1] recommendation system is an ideal way to help the GRE® test takers improve their verbal reasoning abilities by practicing questions more efficiently. As there are a great number of verbal reasoning practice questions and limited practice time for test takers, it is impossible to practice all kinds of questions at the same time. Personalized referral systems should be built based on the characteristics of specific respondents, and forming professional recommendation systems for different questions. Based on the examinee's current practicing accuracy and fallible difficulties, we propose an End-to-end Tag-based Recommendation System (ETRS) for task takers to optimize practice effect. Code of this paper can be found on *https://github.com/Oliver-Q/ETRS-for-Verbal-Reasoning-Questions*.

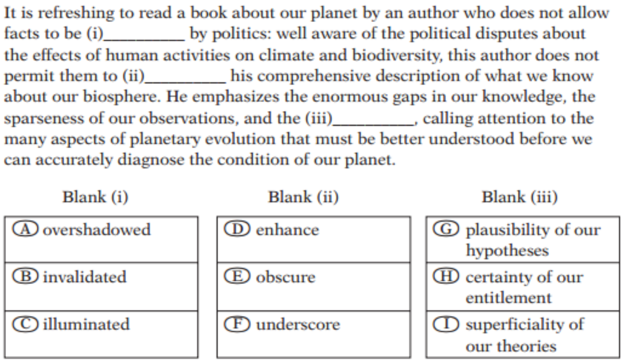
KEYWORDS

Recommendation system, Personalization service, User tagging, Text tagging, Cold-start problem, Nature language processing

1 INTRODUCTION

A total of 584,677 examinees took the GRE® General Test between July 1, 2015, and June 30, 2016.[2] The verbal reasoning abilities are huge uneven for different task taker. Since everyone needs to do exercise before they go to take a real exam, one simple recommendation system cannot provide a personalized service for everyone.

1.1 Verbal Reasoning Question



It is refreshing to read a book about our planet by an author who does not allow facts to be (i)________ by politics: well aware of the political disputes about the effects of human activities on climate and biodiversity, this author does not permit them to (ii)________ his comprehensive description of what we know about our biosphere. He emphasizes the enormous gaps in our knowledge, the sparseness of our observations, and the (iii)________, calling attention to the many aspects of planetary evolution that must be better understood before we can accurately diagnose the condition of our planet.

Blank (i)	Blank (ii)	Blank (iii)
Ⓐ overshadowed	Ⓓ enhance	Ⓖ plausibility of our hypotheses
Ⓑ invalidated	Ⓔ obscure	Ⓗ certainty of our entitlement
Ⓒ illuminated	Ⓕ underscore	Ⓘ superficiality of our theories

Figure 1.1 A Sample Text Completion Question of GRE® General Test

Conventionally, as Official Guide of GRE [1] mentioned, GRE Verbal Reasoning skills can be sharpened by working your way through these question sets. Traditional training way which guides the test taker to exercise the questions in a specific or random order is becoming less effective because it lacks personalization. As years pass since the birth of GRE® General Test, the practicing question sets are becoming increasingly larger. Practice questions are infinite while the examinee's time is limitedFor such frequently-examining knowledge points like clauses, adversatives, and pronouns, a recommendation system can be developed to compensate the blind spots of the candidate who will be

[1] Text Completion question is one type of verbal reasoning question in GRE® revised General Test. GRE® is a registered trademark of Educational Testing Service (ETS).

[2] According to *A Snapshot of the Individuals Who Took the GRE® General Test*, an official investigation published by ETS®

[3] The sample question set is fetched from http://gre.kmf.com/practisenew/tc/3/54

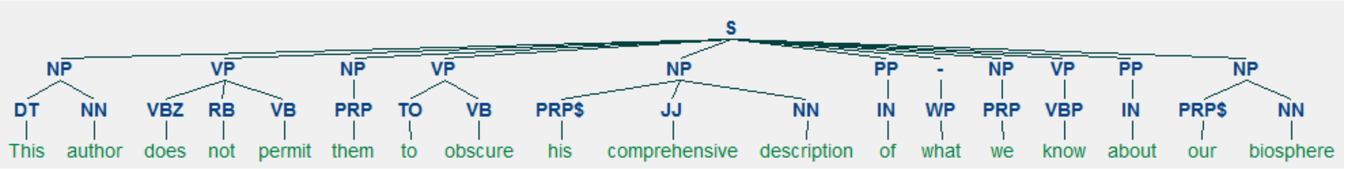

S
NP
VP
NP
VP
NP
PP
-
NP
VP
PP
NP
DT NN VBZ RB VB PRP TO VB PRP$ JJ NN IN WP PRP VBP IN PRP$ NN
This author does not permit them to obscure his comprehensive description of what we know about our biosphere

Figure 3.1.1 Visual representation of a shallow parsed tree for a sample question text.

undertaking the GRE exam. This can be done by analyzing the past trend of the mistakes committed by the candidate.

Verbal reasoning questions appear in several formats, text completion question is what we focused on and discussed in detail below. See a sample of text completion question in Figure1.1. The question shown is composed of three sentences and has three blanks. Three answer choices per blank function independently. And the answers for this sample question are overshadowed, obscure, and superficiality of our theories.

-

1.2 End-to-end Tag-based Recommendation System (ETRS)

We designed a tag-based recommendation system for those examinees, especially for, who don't know their missing knowledge points. When a test taker is practicing, some logical features of questions may become fallible difficulty. In the first place, compared his verbal reasoning abilities with other practicer, the historical practicing accuracy is objective and precise. As the new practice question comes out, we managed to add knowledge points tags for the question via nature language processing tools. Next, with the increasing of practiced questions, a great number of personal fallible difficulties come forth to personalize the user tags. Even though one knows the details of the missing knowledge points, he still does not know whether he has compensated his blind spots. Last, we recommend relative questions leveraging the common tags among user and questions until the user no longer makes mistake of the same knowledge points.

The proposed system aims to assist an examinee to navigate the questions feature space in an interactive way in which the examinee has his own fallible difficult in each feature dimension so that the examinee can find the optimal question to compensate his blind spots. We have also built up an end-to-end system of this kind for GRE® verbal reasoning practice questions. For user who may not have many historical practicing, the ETRS will manage to know the user's tag first. In this situation, practicing accuracy can work as a guidance for new users. The rest of this paper is organized as follows: Research background is expatiated in Section2, including nature language processing, cold-start problem and recommendation system. Section 3 gives detail information and algorism of the recommendation system. Section 4 reports the detail implement and the results of the ETRS. In the last part, conclusion and future work are given in Section 5.

2 RESEARCH BACKGROUND

In order to add tags for questions automatically, we introduced some nature language processing approach. To optimize the recommendation effect for both new questions and new users, we cleverly used the accuracy matching mechanism to solve the cold-start problem in our system. At last, we showed our tag-based feature compared to traditional recommend strategy.

2.1 Nature Language Processing (NLP)

Text completion question of verbal reasoning is nothing more but nature language of English speakers. Not to mention the question's a high degree of correspondence to the reality. Some significant words with logical meaning represent the solution points for different questions. These words are predictable and easy to extract from the context of the questions. We used nature language processing tools to analyze the question and obtain the keywords for tagging automatically.

2.2 Cold-start problem

Without a large amount of user data and items featured, it hard to allow the users to be satisfied with the recommendation results and willing to use the recommendation system. Andrew et al. [9] said that one common but difficult problem for a recommender system is the cold-start problem. The cold start problem is mainly divided into three categories: new items, new users and new system. For new released questions, we automatically extract the tags with nature language processing tools and statistic the overall exercise accuracy. For new users, we recommend the question matching their historical accuracy even though only several questions have been done. In this way, the whole system can quickly get data to perform better and have a fair recommend effect at the very beginning.

2.3 Tag-based Recommendation

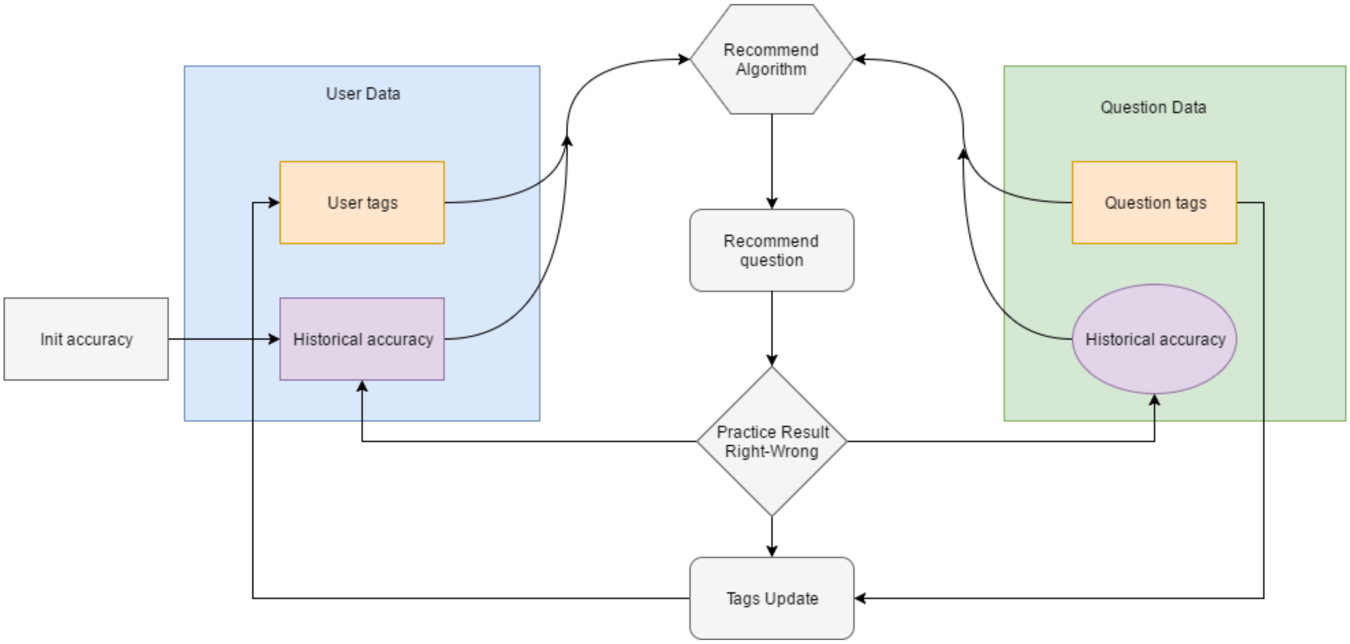


Figure 3.2.1 Conceptual overview of accuracy matching user tag generation pipeline.

The recommendation system can provide personalized information services in different ways; depending on whether the system has recorded and analyzed the user's previous preferences. Traditional collaborative filtering recommendation methods [11,12,13] focused their effort on user data (e.g., user ratings and user similarity), ignoring the common content similarity between user and item. Like Sarwar et al. [2] said, once the most similar items are found, the prediction is then computed by taking a weighted average of the target user's ratings on these similar items. Moreover, content-based filtering [10] simply extract feature through overall content material without focusing on the insight knowledge.

Unlike the above system which needs lots of users' and items' data and quantities of calculations requiring for a powerful machine. ETRS aims to assist an examinee to practice more effectively, when he can simply practice the type of questions and detect his fallible difficulty.

3 EXPERIMENTAL AND COMPUTATIONAL DETAILS

3.1 Text Tokenization and Question Tagging

We applied shallow parsing Natural Language Tool kit(NLTK) to acquire logical meaningful phrases and observe logical relations among them. Figure 3.1.1 is the preceding output is the raw shallow-parsed sentence tree for our sample question. We leveraged the pattern package to implement a shallow parser to extract logical chunks out of question context.

It should be noted that we only focus on the words that are required to master in GRE test, in this case, a GRE Word Book. After go through the shallow parser, the gender of each word is detected as showing above. We can see from Figure 3.1.1, for instance, the word 'obscure' is defined as a verb. The next step is to find the synonym set of 'obscure' using the synset method in wordnet [6] module from NLTK and the gender of the word. The definition of the word 'obscure' as a verb is 'make less visible or unclear'. So, we got synonyms such as 'blur', 'confuse' and 'hidden'. We set up such synonym set for each GRE word that are required to master for examinee. In this case, we leveraged GRE Word Book and tried to match the word appeared in question text and add a synonym tag for this word. Furthermore, for ubiquitous grammatical marker words such as adversative words 'however', 'but' and 'although', clause leading words 'which' 'who' and 'that', we used featured pattern of regular expressions to find them and then add tags of grammatical meaning correspondently. From above we leveraged some modules provided by NLTK to build our own auto-tagging system. In Figure 3.1.2, there are several oral texts that can be assigned to various types of opponents, terms, and so on. Initially, these problems are all together, just as there are various texts in the text corpus. After passing through the text classification system, each logical keyword is divided into a specific logical meaning category. Thus, we can achieve content-based tags of questions automatically.

3.2 Accuracy Matching and User Tag Generation

As we have mentioned above, the whole system can quickly get data to perform better and have a fair recommend effect at the very beginning. In order to deal with the user tags shortage, we introduced the practice accuracy matching technic. This accuracy can be initialized by user himself or overall questions average accuracy. The score function is shown in equation below.

$$R_{matching} = 1 - ABS\left(a_{question} - a_{user}\right) \qquad (1)$$

First recommendation based on this approach will be transported to the user in a rather smooth way. Then the user practices the recommended question and tell the system right-wrong about the result. Immediately, both historical accuracy will be updated based on this result so do the tags. For instance, if the result is right, the user tags will be fetched from question tags and marked as a positive weight. Otherwise, they will be marked as negative weight. More details of this process can be found in figure 3.2.1 which shows the overall pipeline. After cold-start and finished several rounds of recommendation, the number of user's tags will strikingly increase and corresponding weight will be allocated unevenly. The system has been warmed up and we can apply some more personalized recommendation.

3.3 Making Recommendations: Tag-Based Recommend

As seen in figure 3.2.1, we implement the recommend algorithm based on the common tags between user and questions.

ALGORITHM 1: Tag-based Recommend Algorithm

```
user_tags ← user
question_set ← overall questions
question is inside question_set
for question is inside question_set, do
        common_tags ← user & question tags in common
        for each tag in common_tags, do
                question_score ← each question recommendation rate from question_set
                convert weight to score
                scale score by weight / number_of_all_same_tag
                question_score ← question_score + tag_score
        end
end
question_score ← question_score + accuracy_score
sorted question_set by question_score
return question_set[0]
end
```

Above is our tag-based recommend algorithm. It is seen that the overall algorithm focuses on the common tags between user and questions. Question score is first summed up by the scaled weight of common tags. In order to prevent the fact that some question may have too many tags and get unfair high recommending score, we scaled the weight by dividing the number of all the same name tags. Considered the initial state, we merged the score of matching rate from the last part of the system. We finally achieve the evaluating score for each question in the question set. The score function is shown in (2).

$$S(\mathrm{q}) = R_{matching} * 100 + \sum_{tag}^{all\ tags} \left(\frac{W_{tag}}{N_{same\ tag}}\right) \qquad (2)$$

$S(\mathrm{q})$ means the evaluated score for each question in question set, $R_{matching}$ is what we got from the accuracy matching process, W_{tag} represents the weight of each tag in common tags and $N_{all\ same\ tag}$ means the number of all same name tags. Note that the more tags we have, the more personalized recommendation we would have. In other words, with the usage of the system, the recommendation of the system shows increasingly personalization.

4 RESULTS AND DISCUSSION

4.1 Question Tagging Results of Verbal Reasoning Questions

Table 1. Some sample questions and their tags

Question-id	Accuracy	Auto-Tags
f2azxj	0.39	Clause, Adversative, Refer, Repeat
82b0xj	0.67	Refer, Repeat, Reverse
f2b1dj	0.22	Negative, Repeat, Refer
72b0yj	0.54	Positive, Repeat
b2b1nj	0.15	Positive, Negative, Repeat, Refer

Total 78 sample questions have been added into our system3. All of our test questions can be found on http://gre.kmf.com/question/%s.html (%s should be replaced by question-id). Each question went through the auto-tagging process and has their tags shown in Auto-Tags column. Tags are generated by following pattern:

Table 2. Part of sample pattern used in tagging

Regular Expressions	Auto-Tags
r'.*who$'	'Clause'
r'.*:$'	'Repeat'

r'.*not$'	'Reverse'
r'.*this$'	'Refer'
r'.*dispute$'	'Negative'

From the auto tags result of each question, different questions' tags are in high degree of diversity and highly matching with the feature of the question context. This indicates that our process of NLP works good enough.

4.2 Recommendation Result of ETRS

We first let user who is preparing for GRE test practice in ETRS system 7 times. The result of his practice is shown in Table 3. "T" means that the result is right, "F" means that the result is different from the official answer. For the 8th recommendation, the recommend list gained by our ETRS is represented in Table 4. Part of questions in the question set have been shown in the sort tag-based recommendation algorithm detailed in equation (2).

Table 3. Result of one user in 7 practices

Question-id	22b0oj	22b23j	22b1xj	62b0pj	82b04j	f2azxj	02azzj
Right	T				T		T
Wrong		F	F	F		F	

Table 4. Recommendation List of one user after 7 times practice

Question-id	Auto-Tags	Recommend-score
c2b05j	Positive, Clause, Repeat	97.03
72b1zj	Refer, Repeat, Reverse	89.33
891jwk	Negative, Repeat, Refer	87.28
62b0qj	Positive, Repeat	77.31
72b1yj	Positive, Negative, Repeat, Refer	69.92

From the tables and results above, we can tell that our tag-based recommend algorithm is not been trapped by the number of tags that a single question has and have a good diversity of tags and questions in recommendation list.

5 CONCLUSION

In this paper, we first dissected the key points of question text in text completion of verbal reasoning. Then we achieved auto tagging for questions text via nature language processing. After we get tags for question, we suggested an accuracy matching approach to add tags for users and warm up the whole system. Finally, with enough tags and tag-based algorithm, we made this system from end-to-end and performed an ETRS for verbal reasoning questions. Compared to item-based or user-based recommendation system, ETRS does not require copious user data to achieve precise recommendation. In other words, ETRS works well enough for an initial state of recommend known as cold-start problem. Moreover, unlike other recommendation systems asked their users to tag items manually, ETRS achieved tagging questions automatically. However, some recommendatory field experts' work may require before a good quality tag set can derive while user-based recommendation only needs enough user data. To guarantee tags' precision, tagging process and patterns need to be well-designed.

As for future work, text tokenization and question tagging can be more intelligent with new technique in NLP such as machine learning algorithm such as LSTM & Recurrent Neural Network. Questions can be understood automatically and given relative difficulty suggestions. In addition, we can design an expert system in knowledge-based reasoning [17] to analyze and evaluate the question based on knowledge tags. Furthermore, with more usage and user data, we can also implement other recommendation algorithm collaboration filtering as an approach to achieve more precise recommendation or hybrid other recommendation algorithms combine content-based and item-based techniques [13,14,15,16].

ACKNOWLEDGMENTS

This work was partially supported by SUSTech fund (05/Y01051814, 05/Y01051827, 05/Y01051830, 05/Y01051839).

REFERENCES

[1] Educational Testing Service. The Official Guide to the GRE Revised General Test, 2nd Edition. McGraw-Hill Education.

[2] Sarwar, B., Karypis, G., Konstan, J. and Riedl, J., 2001, April. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web* (pp. 285-295). ACM.

[3] tp:/ Bird, Steven, Edward Loper and Ewan Klein (2009) *Natural Lqueanguage Processing with Python.* O'Reilly Media Inc.

[4] Segaran, T., 2007. *Programming collective intelligence: building smart web 2.0 applications.* O'Reilly Media, Inc.

[5] Cao, Y. and Li, Y., 2007. An intelligent fuzzy-based recommendation system for consumer electronic products. *Expert Systems with Applications*, *33*(1), pp.230-240.

[6] Miller, G.A., 1995. WordNet: a lexical database for English. *Communications of the ACM*, *38*(11), pp.39-41.

[7] Basu, C., Hirsh, H. and Cohen, W., 1998, July. Recommendation as classification: Using social and content-based information in recommendation. In *Aaai/iaai* (pp. 714-720).

[8] Pazzani, M. and Billsus, D., 2007. Content-based recommendation systems. *The adaptive web*, pp.325-341.

[9] Schein, A.I., Popescul, A., Ungar, L.H. and Pennock, D.M., 2002, August. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR*

conference on Research and development in information retrieval (pp. 253-260). ACM.

[10] R. J. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In Proceedings of the Fifth ACM Conference on Digital Libraries, pages 195–204, 2000.

[11] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, pages 43–52, 1998.

[12] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. GroupLens: Applying collaborative filtering to Usenet news. Communications of the ACM, 40(3):77–87, 1997.

[13] C. Basu, H. Hirsh, and W. Cohen. Recommendation as classification: Using social and content-based information in recommendation. In Proceedings of the Fifteenth National Conference on Artificial Intelligence, pages 714–720, 1998.

[14] M. Claypool, A. Gokhale, and T. Miranda. Combining content-based and collaborative filters in an online newspaper. In Proceedings of the ACM SIGIR Workshop on Recommender Systems—Implementation and Evaluation, 1999.

[15] M. K. C ondliff, D. D. Lewis, D. Madigan, and C. Posse. Bayesian mixed-effect models for recommender systems. In ACM SIGIR '99 Workshop on Recommender Systems: Algorithms and Evaluation, 1999.

[16] N. Good, J. B. Schafer, J. A. Konstan, A. Borchers, B. M. Sarwar, J. L. Herlocker, and J. Riedl. Combining collaborative filtering with personal agents for better recommendations. In Proceedings of the Sixteenth National Conference on Artificial Intelligence, pages 439–446, 1999.

[17] Chandrasekaran, B., 1986. Generic tasks in knowledge-based reasoning: High-level building blocks for expert system design. *IEEE expert*, *1*(3), pp.23-30.

[18] Steedman, M., 2000. *The syntactic process* (Vol. 24). Cambridge: MIT press.

[19] Jäschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L. and Stumme, G., 2007, September. Tag recommendations in folksonomies. In *European Conference on Principles of Data Mining and Knowledge Discovery* (pp. 506-514). Springer Berlin Heidelberg.