Trivia Score and Ranking Estimation Using Support Vector Regression and RankNet

Kazuya Niina **Kazutaka Shimada**
Kyushu Institute of Technology
680-4 Kawazu Iizuka Fukuoka 820-8502 Japan
{k_niina, shimada}@pluto.ai.kyutech.ac.jp

Abstract

Dialogue systems have been increasingly important these days. In particular, non-task-oriented dialogue systems are studied because of the success of neural network approaches such as seq2seq models. However, these models tend to generate simple responses such as "yes" and "ok." To construct a dialogue system that holds users' attention continuously, we need to generate utterances that capture the interest of the user. In this paper, we propose a method to extract trivia sentences for the purpose. Trivia information perhaps adds a surprise to users. Therefore, capturing trivia information is beneficial for dialogue systems. We estimate a trivia score of a sentence by using machine learning approaches, Support Vector Regression (SVR) and RankNet. We obtained 0.79 and 0.78 on SVR for the $nDCG$@5 and RankNet for the $nDCG$@10, respectively. We focus on the subject word in each sentence. The method with subject information outperformed that without subject information; 0.79 with subject information vs. 0.64 without subject information on the SVR for the $nDCG$@5.

1 Introduction

Dialogue systems, such as Siri and Alexa, have been increasingly important these days. In particular, non-task-oriented dialogue systems are studied because of the success of neural networks or reinforcement learning approaches such as seq2seq models (Vinyals and Le, 2015; Li et al., 2016). However, these models tend to generate simple responses such as "yes" and "ok". Therefore, users often get bored with the conversation based on such dialogue systems. To solve this problem, we need to generate utterances that stimulate users' interest.

In this paper, we focus on trivia to solve the problem. Here, we define trivia information as a lesser-known and interesting fact. For example, the following sentence is trivia information; "If red swamp crayfishes eat mackerels, the body color becomes blue." We believe that the trivia sentences can capture the interest of the users and are beneficial to construct a dialogue system that holds users' attention continuously.

To identify trivia information, we propose a method for estimating a trivia score of each sentence. In the method, we focus on a relation between the main topic and words in each sentence. We extract the main topic from a target sentence and then identify an important noun for estimating the trivia score. We compute feature values from the word pair and then apply them to machine learning approaches. We use the Support Vector Regression (SVR) and the RankNet as the machine learning approach.

2 Related Work

Prakash et al. (2015) have proposed a mining method for interesting trivia. They used trivia information in IMDB as the training data. They extracted named entities and superlative words as features for a machine learning method. They estimated interestingness as trivia by using SVMrank. Fatma et al. (2017) have proposed a method based on deep learning for a trivia classification task. They handled

Trivia	Score	Maximum
A statue of Buddha with the Afro haircut exists.	77	100
Largehead hairtails swim with the standing style	93	100
Tyrannosaurus cannot run	66	100
The blood type of all gorillas is B	110	200

Table 1: Examples of trivia sentences from "Hey! Spring of Trivia."

RDF triples from DBpedia[1] as the target data. They evaluated a Fusion Based CNN which learns combinations of features obtained by convolution and hand-crafted features. Tsurel et al. (2017) have extracted trivia information from Wikipedia by a scoring method. They focused on categories on each Wikipedia page and estimated a trivia score based on similarity and cohesiveness measures.

Ota et al. (2009) have proposed a method for extracting sentences with a surprise for a dialogue system. They computed the TFIDF value, the co-occurrence frequency, and the sentence length and extracted sentences with surprise from Wikipedia by using some rules. Niina and Shimada (2017) have proposed a method for extracting unusual facts from Wikipedia for a dialogue system. They also computed some scores from each sentence in Wikipedia and then detected unusual facts in a similar way to (Ota et al., 2009).

In this paper, we use a Japanese supervised dataset which contains trivia sentences with a trivia score. Our purpose is to estimate the trivia score and the ranking of each trivia sentence by using machine learning approaches based on word pair features.

3 Dataset

For the estimation of a trivia score of a sentence, we need a training dataset. We use sentences that appeared in "Hey! Spring of Trivia" that was a Japanese TV show. Most of the trivia on the show was sent by viewers. We extract trivia sentences from the Wikipedia page of the TV show.

Table 1 shows some examples of trivia sentences in the dataset. Each trivia sentence was evaluated by some judges by pushing a "hey[2]" button on the TV show. The judges pushed the button when they felt that the trivia sentence was interesting. In Table 1, the score denotes the number of "hey", namely the interestingness of the trivia. Since the maximum value of "hey" depends on the TV episodes, the table contains the maximum values for each trivia sentence. In this paper, we normalize the score by the maximum value. We regard the normalized score as the trivia score in this paper.

Trivia score	# of sentences
0.0 - 0.1	3
0.1 - 0.2	1
0.2 - 0.3	3
0.3 - 0.4	6
0.4 - 0.5	17
0.5 - 0.6	105
0.6 - 0.7	258
0.7 - 0.8	323
0.8 - 0.9	251
0.9 - 1.0	64
Total	1031

Table 2: The distribution of normalized trivia scores.

Table 2 shows the distribution of the trivia score from "Hey! Spring of Trivia." The distribution is unbalanced because the trivia sentences on the show were submitted by viewers as trivia and were selected by the production team.

4 Features

The purpose of this paper is to estimate a trivia score of a sentence by using machine learning. Therefore, we need features for the machine learning methods. In this section, we describe our feature extraction process. The outline of our method is shown in Figure 1. The feature extraction process consists of two phases; identification of a word pair and calculation of feature values.

[1]https://wiki.dbpedia.org/

[2]The meaning of this word in English is similar to "really" in this context.

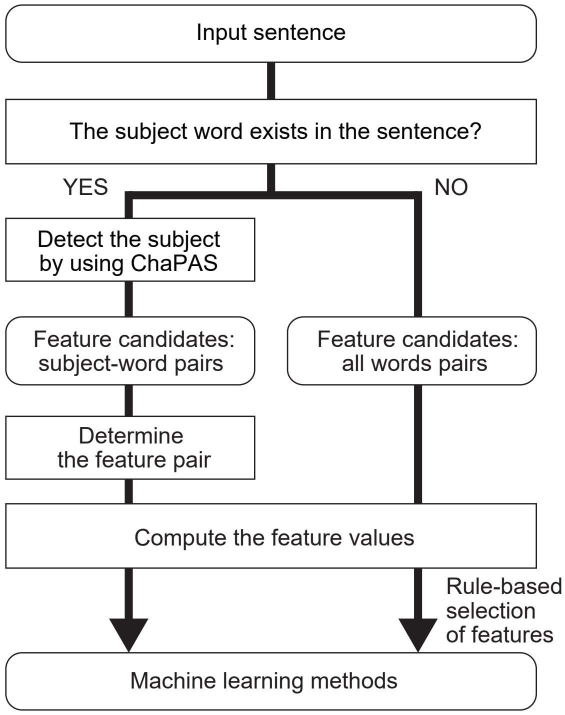

Figure 1: The outline of the feature selection.

4.1 Word Pair Extraction

In this section, we describe the word pair extraction for the calculation of feature values. This process is divided into two processes on the basis of one rule; presence of the subject word in a sentence. We select feature candidates via this process.

4.1.1 Subject Extraction

Each trivia sentence contains a topic. People find interesting and humorous if a sentence contains a gap between the topic and the mention in the sentence. Assume that the following trivia sentence, "If red swamp crayfishes eat mackerels, the body color becomes blue." The interesting point of this trivia sentence is the unexpected fact, "blue", against the common sense, "Crayfishes are red." The important point, in this case, is a relation between "crayfishes" and "blue." This point is what makes it trivia. On the other hand, there is no surprise for a relation between "mackerels" and "blue." These results indicate the significance of the main topic in the sentence for understanding the trivia. The topic of this sentence is crayfishes, namely the subject word in the sentence. Therefore, extraction of the subject word has a significant role to estimate a trivia score of each sentence.

Our target in this paper is Japanese. In Japanese, the subject word in sentences is often omitted, zero-pronouns. Thus, we need to identify the subject word in each sentence. We use a Japanese Predicate Argument Structure Analyzer, ChaPAS[3], for the subject extraction. ChaPAS is a modified model of (Watanabe et al., 2010). In this paper, we extract a nominative case, ga-case in Japanese, as the subject.

4.1.2 Word Pair Candidates

The pair of the subject word and other words in a sentence is important for the calculation of feature values. On the other hand, there are many combinations of the subject and words in a sentence. In this process, we handle nouns and verbs for generating word pair candidates. For example, we obtain a word pair (mailbox, bottom-of-the-ocean)[4] from the trivia sentence " There is a mailbox on the bottom of the ocean. "

As mentioned above, zero-pronouns frequently appear in Japanese sentences. In other words, sentences do not always contain the subject. In this situation[5], we create all pairs about (noun, noun) and (noun, verb) from the sentences.

4.1.3 Feature Pair Determination

In the previous sub-section, we obtain some (subject, word) pairs from each sentence. Hence, we need to determine the feature pair for the calculation of feature values that are used in machine learning.

We assume that a good feature for the trivia score estimation rarely appears in real-world texts because the important point about trivia is a gap between words. Therefore, we compute a co-occurrence value, $coFreq$, as follows:

$$coFreq(s, w) = \frac{pair\text{-}freq(s, w)}{freq(s)} \quad (1)$$

where s is the subject extracted Section 4.1.1. w is the pair word with s. $pair\text{-}freq(s, w)$ is the co-occurrence frequency of the pair (s, w). $freq(s)$ is

[3]https://sites.google.com/site/yotarow/chapas

[4]Note that we ignore some verbs, such as " do (suru in Japanese) " and " be (" desu " in Japanese), as stop-words.

[5]Hereinafter, this is called "non-subject situation."

the frequency of s. $pair\text{-}freq(s, w)$ and $freq(s)$ are computed from 7-grams of Japanese Google N-grams (Kudo and Kazawa, 2007).

As mentioned above, trivia sentences in the dataset perhaps do not contain the subject word. In this situation (non-subject situation), we do not determine the feature pair from (noun, noun) and (noun, verb) in this process. On the basis of some rules in the calculation process (the next section), we determine which pair should be used for the machine learning.

4.2 Feature Value Calculation

In this paper, we compute the following feature values for machine learning methods.

- IDF
- Similarity
- Inverse Entity Frequency (IEF)
- Word embeddings

4.2.1 IDF

We assume that a trivia sentence with well-known words contains a higher trivia score than that with less-known words. Therefore, we use the IDF value of the subject as the feature value. In other words, we assume that the trivia score of a sentence is high in the case that the IDF value of the subject in the sentence is low, namely a well-known word. The IDF value is computed from all Wikipedia pages.

In the non-subject situation, we compute the IDF values for all nouns in the target trivia sentence and then use the minimum value as the feature value.

4.2.2 Similarity

Trivia sentences tend to contain word combinations that are rare. For example, (grave-marker, printer) for the trivia sentence "There is a dedicated printer for grave makers." Generally, the combination is rare. This implies that the similarity of words in the feature pair is small, as compared with ordinary word pairs. Therefore, we apply a word similarity measure to the feature values. We generate a word-embedding model by using word2vec (Mikolov et al., 2013). Then, we compute the cosine similarity of words in the target feature pair, as the feature value.

In the non-subject situation, we compute the cosine for all (noun, noun) pairs in the target trivia sentence, and then use the minimum value as the feature value.

4.2.3 Inverse Entity Frequency (IEF)

In the previous sub-section, "Similarity", we compute a similarity value between words. In this feature, we extend this point to a category-level. In a similar way to the word similarity, we assume that a word in the feature pair rarely appears in documents that related to another word. For example, we obtain (mummy, fuel) from "Mummies were used as a fuel for the train in the 18th century." Here we obtain documents that related to mummies, such as "corpse" and "ancient Egypt." There is obviously a gap between the word "fuel" and these documents.

We use the categories on Wikipedia for this process. We compute the Inverse Entity Frequency (IEF) value from the category of the subject on Wikipedia and another word (pair-word) in the feature pair. The process is as follows:

1. Extract the Wikipedia page of the subject
2. Extract the category of the Wikipedia page
3. Extract the page set which belongs to the category
4. Compute the IDF of the pair-word in the page set
5. Compute the IEF by using the following equations

$$IEF(s, w) = \frac{IDF_{C_s}(w)}{log(|C_s| + 1)} \tag{2}$$

$$IDF_{C_s}(w) = log\frac{|C_s| + 1}{df_{C_s}(w) + 1}$$

where s and w are the subject and the pair-word, respectively. C_s is the page set of the category that s belongs to. $IDF_{C_s}(w)$ is the IDF of w in C_s. $df_{C_s}(w)$ is the document frequency of w in C_s.

In the non-subject situation, we compute the IEF value of all pairs in the target trivia sentence and then use the maximum value as the feature value.

4.2.4 Word embeddings

In recent years, word embedding is often used as a feature for machine learning. Hence, we also apply a word embedding model into the feature set. We use the embedding of the subject as the feature.

In the non-subject situation, we select three words from the beginning of the sentence and use the embedding of them as the features. If the number of target words is less than 3, we add the zero vector to the feature space[6].

5 Experiment

In this section, we evaluate our features with the dataset described in Section 3. We apply the features into two machine learning methods, a regression model and a ranking model. First, we evaluate the trivia score estimation with the regression model. Then, we compare the ranking model with the regression model and a baseline based on the previous work, in terms of ranking estimation.

As common settings of the experiment, we used the data dump provided by Wikimedia Foundation on May 21, 2017[7]. We also used Japanese DBpedia[8] for the IEF calculation. We generated a word embedding model by using Word2Vec[9] with skip-gram. The number of dimensions was 200.

5.1 Trivia Score Estimation

In the experiment, we evaluated a regression model based on features extracted in Section 4 for the trivia score estimation. We used Support Vector Regression (SVR) (Drucker et al., 1997) with the RBF kernel. SVR is a linear regression method based on SVM. We implemented the model with default parameters by scikit-learn. We used the RBF kernel. The parameters, C and γ, were default values.

Here we have a problem with the dataset. As mentioned in Section 3, the distribution of the dataset was unbalanced. The model generated from the dataset probably estimates approximately 0.75 as the trivia score of many instances. It is not suitable for the estimation model. Therefore, we reconstructed the dataset by the following process.

Method	MAE	MSE	R2
Proposed	**0.2490**	0.0835	-0.0108
Baseline	0.2503	**0.0827**	**0.0000**

Table 3: The result of the trivia score estimation task by SVR.

1. Sort the dataset in descending order by the original trivia score
2. Set 0.95 to a new pseudo trivia score
3. Assign the new pseudo trivia score for the top 10% sentences
4. Delete the current top 10% sentences.
5. Decrease the pseudo trivia score by 0.1.
6. Repeat 3 to 5 until 0.05 about the pseudo trivia score.

As a result, we obtained a balanced dataset. In other words, the trivia score of the top 103 trivia sentences[10] was 0.95 and that of the next 103 trivia sentences was 0.85. Likewise, a new pseudo trivia score is assigned to trivia sentences.

For evaluation criteria, we used Mean Absolute Error (MAE), Mean Squared Error (MSE) and Coefficient of Determination (R2) in 10-fold cross-validation. As a naive baseline, we use the model that regarded the trivia score as the average value on the dataset, namely approximately 0.5.

Table 3 shows the estimation result by SVR and the baseline. We found that the proposed method just slightly exceeded the baseline in terms of MAE. Although the baseline obtained better results about MSE and R2, they were also very few differences. In addition, our method estimated that the trivia scores of most sentences were within the range of 0.45 to 0.65. In other words, the range that our method can estimate is insufficient. Therefore, we need to discuss new features for identifying trivia sentences with high trivia scores (0.85 to 0.95).

5.2 Ranking Estimation

Our final goal is to apply trivia sentences into our dialogue system. Thus, our motivation is to extract

[6]Note that we always add zero vectors of two words in the subject situation because we just use one subject in this process.

[7]https://dumps.wikimedia.org/jawiki/

[8]http://ja.dbpedia.org/

[9]https://radimrehurek.com/gensim/models/word2vec.html

[10]10% of 1031 instances.

Method	$nDCG@5$	$nDCG@10$
Scoring	0.722	0.739
Regression	**0.791**	0.776
RankNet	0.775	**0.782**

Table 4: The result of the ranking task.

sentences with high trivia scores for the purpose. In other words, we need trivia sentences in the higher rank in all sentences. Therefore we evaluated our features with a ranking task of trivia sentences.

We used RankNet that was proposed by (Burges et al., 2005). RankNet is a gradient descent method for learning ranking functions based on the pairwise approach. In this experiment, the number of hidden layers was 2 and the number of units was 1024. The activation function was ReLU and the loss function was cross entropy. We compared the RankNet-based method with the regression model (SVR) described in Section 5.1 and a baseline. The baseline proposed by (Niina and Shimada, 2017) was based on a scoring function to extract unusual facts from sentences in Wikipedia because the task was similar to our task, namely trivia sentence extraction.

In the experiment, we randomly divided the dataset into two parts; training and test. We used 90% as the training data and 10% as the test data. As a criterion, we use $nDCG@k$ ($k = 5, 10$).

Table 4 shows the experimental result of the ranking task. The RankNet-based method and SVR outperformed the baseline from the related work in terms of both settings, $k = 5, 10$. This result shows the effectiveness of our methods. In addition, the SVR described in Section 5.1 also outperformed the baseline based on a scoring function although the result of the SVR was not enough in the trivia score estimation task. This result shows that the proposed features were effective for the ranking task, as compared with a scoring function from the related work. In other words, the proposed features were effective to recognize a relationship between magnitudes of trivia scores although those were not sufficient to estimate actual trivia scores of sentences.

5.3 Discussion about Features

First, we evaluated the RankNet-based method by ablation test. Table 5 shows the result. ALL denotes

Feature	$nDCG@5$	$nDCG@10$
ALL	0.775	0.782
OUT-IDF	0.711	0.759
OUT-Similarity	0.790	0.799
OUT-IEF(s, n)	0.802	0.824
OUT-IEF(s, v)	0.765	0.770
OUT-wordEmb	0.722	0.754

Table 5: The result of the ablation test by RankNet.

Range	IDF	IEF(s, v)
top 5	7.594	0.1000
top 10	6.231	0.3195
bottom 10	3.371	0.5105
bottom 5	3.019	0.3309
average	4.556	0.3821

Table 6: The values of IDF and IEF values in the dataset.

the result of the method with all features, namely the same as Table 4. OUT-IDF denotes the result of the method without the IDF feature. IEF(s, n) and IEF(s, v) are IEF(subject, noun) and IEF(subject, verb), respectively. From the table, IDF, IEF(subject, verb), and vectors from the word embedding model were effective to recognize the ranks of each trivia sentence because deleting these features led to decrease of the accuracy. However, the results were based on our small dataset, namely 90% as training data and 10% as test data from 1031 instances. Increasing the dataset and evaluating the larger dataset are important for a reliable experiment.

Next, we discuss some features of our methods in detail. Here we focused on the IDF and IEF(s, v) features. Table 6 shows the feature values computed from the dataset for the actual top/bottom 5 and 10 trivia sentences. The IDF values in Table 6 were clearly arranged in descending order. This is one reason that the feature generated the good performance in Table 5. However, it should be in ascending order from the assumption described in Section 4.2.1. In other words, this was a reversal phenomenon; we expected that the values in the top ranks and the bottom ranks were low and high, respectively. Therefore the assumption itself might not be correct for the trivia identification although the IDF feature was effective. The IEF values were ex-

Method	$nDCG@5$	$nDCG@10$
SVR with Subject	0.791	0.776
SVR without Subject	0.637	0.669
RN with Subject	0.775	0.782
RN without Subject	0.713	0.745

Table 7: The effectiveness of the proposed features with the subject information for SVR and RankNet (RN).

pected in descending order from the assumption in Section 4.2.3. However, the values in Table 6 were out of order although the feature contributed to generating a little better performance in Table 5. In addition, the results, especially the trivia score estimation, were not always sufficient as compared with a naive baseline. To accomplish the higher accuracy, we need to consider new features and the combinations of the current features and them. Moreover, we need to apply other machine learning methods to the task.

The main contribution of our method is to handle a relationship with the subject in each trivia sentence. For the validation of this contribution, we compared our methods based on Figure 1 and methods that did not handle the relation, namely methods with only the "NO" process in Figure 1. Table 7 shows the experimental result of the ranking task. The methods with subject information outperformed those without subject information for both criteria; $nDCG@5$ and $nDCG@10$. This result shows the effectiveness of the features that incorporated the relation with the subject.

6 Conclusions

In this paper, we proposed some features for estimating a trivia score of each sentence. We focused on a relation between the main topic and the words in each sentence. We extracted the main topic, namely the subject word, from a target sentence and then identified an important noun for estimating the trivia score. We computed feature values from the word pair and then applied them to machine learning approaches. We used the Support Vector Regression (SVR) and the RankNet as the machine learning approach.

In the experiment about the trivia score estimation by using SVR, we did not always obtain the success as compared with the naive baseline. On the other hand, in the experiment about the ranking task, the SVR obtained 0.791 on $nDCG@5$ and the RankNet-based method obtained 0.782 on $nDCG@10$. The two methods outperformed the baseline from the related work. The IDF feature, IEF feature, and the vectors from the word embedding model were effective to recognize the rank of each trivia sentence. In addition, the method with subject information outperformed that without subject information. These results show the effectiveness of the proposed features with the subject information. However, our subject extraction method relied on a simple rule and an existing tool. The improvement of this process is important future work. Moreover, the size of the dataset was not enough for machine learning techniques. Increasing the dataset is also important future work.

Our final goal is to apply trivia sentences into a dialogue system that holds users' attention continuously. Therefore, we need to not only estimate the trivia score of a sentence but also extract trivia sentences from massive sentences. For the purpose, we need to incorporate non-trivia features although we focused on trivia features in this paper. In addition, we need to discuss the usage of the extracted trivia sentences in the dialogue system, such as selection of trivia sentences for the output process and output control of a trivia sentence in a dialogue.

References

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 89–96. ACM.

Harris Drucker, Christopher JC Burges, Linda Kaufman, Alex J Smola, and Vladimir Vapnik. 1997. Support vector regression machines. In *Advances in neural information processing systems*, pages 155–161.

Nausheen Fatma, Manoj Kumar Chinnakotla, and Manish Shrivastava. 2017. The Unusual Suspects: Deep Learning Based Mining of Interesting Entity Trivia from Knowledge Graphs. In *Thirty-First AAAI Conference on Artificial Intelligence*, pages 1107–1113.

Taku Kudo and Hideto Kazawa. 2007. Web Japanese n-gram version 1. published by Gengo Shigen Kyokai.

Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. 2016. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing,D16-112*, pages 1192–1202, Valencia, Spain, November.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781:1–12.

Kazuya Niina and Kazutaka Shimada. 2017. A dialogue system with miscellaneous knowledge from sizzle words. In *IEICE Tech. Rep, NLC2017-42 (in Japanese)*, volume 117, pages 77–82.

Tomohiro Ota, Fujio Toriumi, and Kenichiro Ishii. 2009. Sentence extraction from Wikipedia for making utterance for dialogue system. In *The 23rd Annual Conference of the Japanese Society for Articial Intel- ligence (in Japanese)*, number 2G1-NFC5-11.

Abhay Prakash, Manoj Kumar Chinnakotla, Dhaval Patel, and Puneet Garg. 2015. Did you know? - mining interesting trivia for entities from wikipedia. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, pages 3164–3170.

David Tsurel, Dan Pelleg, Ido Guy, and Dafna Shahaf. 2017. Fun facts: Automatic trivia fact extraction from wikipedia. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 345–354. ACM.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Yotaro Watanabe, Masayuki Asahara, and Yuji Matsumoto. 2010. A structured model for joint learning of argument roles and predicate senses. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 98–102. Association for Computational Linguistics.