Query Recommendation to Draw a Laugh from Web Searchers

doi: 10.1145/3366030.3366045

Query Recommendation to Draw a Laugh from Web Searchers

Hiroo Umeda*
Shizuoka University
Hamamamtu, Shizuoka, Japan
umeda@design.inf.shizuoka.ac.jp

Yusuke Yamamoto
Shizuoka University
Hamamamtu, Shizuoka, Japan
yusuke_yamamoto@acm.org

ABSTRACT

This study proposes a system, which shows funny term pairs when searchers issue a query into Web search engines. The proposed system analyzes the following two factors for a term pair in a given query: the unexpectedness and the semantic conflict. The experimental result showed that the proposed method provided a larger number of funny term pairs for queries than the baseline methods. Although the proposed method was not the best based on the average value, it can still offer opportunities for searchers to laugh and feel cheery when issuing queries into the Web search engines.

CCS CONCEPTS

• Human-centered computing → Human computer interaction(HCI); **• Information systems → Information retrieval** ;

KEYWORDS

human-computer interaction, information retrieval, query recommendations, laughter

ACM Reference format:
Hiroo Umeda and Yusuke Yamamoto. 2019. Query Recommendation to Draw a Laugh from Web Searchers. In *Proceedings of The 21st International Conference on Information Integration and Web-based Applications Services, Munich, Germany, December 2–4, 2019 (iiWAS2019)*, 8 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

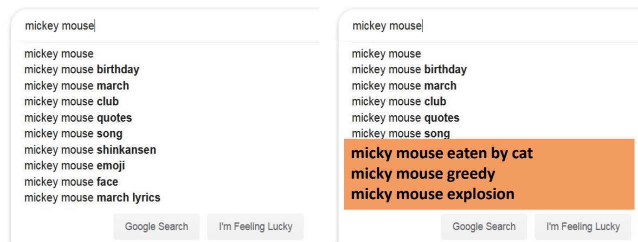

Figure 1: Comparison of the conventional query recommendation (left) and proposal system (right)

1 INTRODUCTION

People often desire to laugh and look for opportunities or information to laugh. At present, the Web is a gigantic resource to obtain various information and provides rich information on hobbies and entertainment. Generally, the Web offers more information than other media such as television and newspaper. Several people spend time viewing humorous/parody contents on the Web such as YouTube and the Onion[1]. Moreover, people often post humorous/funny contents on the Web and reply to those who provide feedback on the contents. Thus, the passion and desire for laughing have invented new terms in our culture. One of the word examples is "w," an abbreviation of the Japanese term "warai," which means laugh and often used to represent when people feel something funny. This phenomenon indicates that people enjoy laughing and want to laugh. Studies have also shown that laughing has a positive effect on our minds and bodies. For example, laughing can reduce stress, pain, and improve immunity[1][3]. Therefore, increasing opportunities for laughing in daily life is important.

Web search is a popular resource to collect information; however, it is still challenging to search the laughable information from the Web. For example, to find a funny web page about the "Mickey Mouse," a simple way is to issue a query like the "mickey mouse laughing" and the "Mickey mouse funny" into Web search engines. However, conventional Web search engines offer webpages that produce the terms "laughing" or "funny," which do not necessarily contain really-funny content. If people want to obtain funny contents from the Web search engines, they should input the appropriate queries.

In this present study, we propose a system to show terms that induce searchers to laugh in the query recommendation during a web search. Once searchers input a query into a search box, our system shows the terms whose combination with the query is humorous or funny on the Web. We build a hypothesis that if terms X and Y are a funny combination and the two terms meet the following condition: (1) the combination of X and Y is uncommon, and (2) the polarity of X is opposite to that of Y. This hypothesis is developed from some literature on humor. After the system estimated the humor of term combination, the system displayed the top-K humorous term combination in a query recommendation list.

Figure 1 shows an example of the proposed system output. The proposed system shows a funny term combination related to queries in a search box, such as "mickey mouse eaten by cat," "mickey mouse greedy," and "mickey mouse explosion." These term sets may evoke laughter when the searchers are issuing queries. The proposed system aims to provide searchers with opportunities to laugh when issuing a web search query that is unsupported when searching for funny web contents, although it is possible to use

*Both authors contributed equally to this research.
[1]The Onion: https://www.theonion.com/

iiWAS2019, December 2–4, 2019, Munich, Germany
2019. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

the term sets for funny content search. The rest of this paper is organized as follows. Section 2 describes related studies; Section 3 describes conditions for laughing. Section 4 describes the methods of queries recommendations to induce laughs. Section 5 describes the contents of the experiment; Section 6 presents the experimental results, and Section 7 discusses the results. Section 8 concludes the paper.

2 RELATED WORK

2.1 Concept of laughter

Many researchers have explored the concept of laughter and factors influencing laughter. Shibahara provides the mechanism of laughter from three perspectives: philosophical, semantic, and psychoanalytic [15]. Humor is one of the factors to generate laughter, and many researchers have explored the concept of humor. Yang et al. proposed a method to estimate a humorous sentence and to analyze four features behind sentences: *incongruity*, *ambiguity*, *interpersonal effect*, and *phonetic style* [20]. Hong and Ong proposed a system that learned a relation between words in puns and their sounds, which automatically generated new puns [6]. Kiddon and Brun developed a method to judge whether a sentence was sexually humorous, focusing on the appearance of sexual nouns and the sentence structure [8]. Purandare and Litman focused on prosodic and linguistic features for humor recognition and analyzed humorous conversations in comedy TV programs[14]. Mihalcea and Strapparava proposed an algorithm to classify humor sentence, focusing on stylistic characteristics such as alliteration, antonyms, and slang [12]. Zhang et al. examined the characteristics of humorous tweets in Twitter [21]. While these studies aim to detect humorous sentences, our study focuses on funny combinations of terms, referring to the definition of laughing from the semantic viewpoint of Shibahara's work [15].

2.2 Support for laughing

Several studies have been conducted to support laughing. Mancini et al. and Deniz et al. developed methods to recognize human laughing behaviors. Mancini et al. focused on the movements of heads, chests, and shoulder to determine whether people are laughing[10]. Deniz et al. developed a method to detect smiles through face analysis [4]. Studies have also developed systems to promote laughing. Khooshabeh et al. developed humorous virtual agents and examined their effects [7]. Cha et al. propose a social robot that greets people with a smile[2]. Tsujita and Rekimoto proposed a refrigerator, opened only when users laugh in its front to promote laughter in daily life[17]. Tsukada and Oki proposed a system showing images that evoke a natural smile when taking pictures [18]. B. Lee and W. Lee proposed a new camera concept to induce unconscious facial reactions in a photography subject. Their proposed camera will display a small facial expression icon to photography subjects [9]. Fushimi et al. proposed an application that can capture the natural smile of photography subjects by playing back the laughing[5]. These studies promote increased opportunities for laughing in daily life. In this study, we propose a method to promote laughing in web search.

2.3 Recommendations of unexpected information

Studies have developed systems that provide unexpected information unknown to users. Tsukuda et al. [19] and Noda et al. [13] proposed methods to discover unexpected words similar to input queries when using Wikipedia. Mejova et al. proposed a method that discovers unexpected information by comparing entity networks extracted from Wikipedia with entity networks extracted from Yahoo Answers[11]. In this study, we focus on the unexpected relation between the terms to find a funny/humorous combination of terms.

3 TERM COMBINATION TO EVOKE LAUGHTER

Given an input query, our system outputs a list of the terms to evoke laughter during a web search. For example, when input a query "Mickey Mouse", the system displays a combination of terms such as "mickey mouse greedy" and "mickey mouse eaten by a cat." These are displayed with conventional query recommendation results, as revealed in Figure 1.

We also describe the requirements of term combinations to promote laughing. Shibahara et al. [15] revealed that if a sentence evoked laughter, the two concepts that met the following conditions appeared in the sentence: (1) it is difficult to derive the image of one concept from other concepts, and (2) the two concepts of semantic conflicts. When we apply Shibahara's idea into our problem of funny term combination, we develop the hypotheses of which, when two terms meet the specified conditions, the combination of the terms evokes laughter: In the first condition, one term is related to other terms; however, it is still difficult to imagine the relation. For example, Disneyland is associated with Mickey Mouse because Mickey Mouse is one of the famous Disney characters. Therefore, the combination of the term "mickey mouse Disneyland" is easy to imagine. However, the terms "eaten by cat" are difficult to associate with the "mickey mouse" because people have fewer opportunities of seeing a cat eating Mickey mouse. We consider that an unexpected relation between terms to evoke laughs during a web search.

The second requirement is a term that semantically conflicts with other terms. An example is the terms combination "mickey mouse" and "eaten by cat". Mickey Mouse is a rat that is afraid of cats because cats can eat the rats. This indicates that the term "mickey mouse" conflicts the term "cat" in the context of eating. A combination of the terms 'mickey mouse' and "eaten by cat" is unexpected and semantic conflicts. Therefore, we think this combination is funny. As explained in the subsequent section, we focus on differences between terms to compute a semantic conflict between terms.

4 ALGORITHM

In this section, we present an algorithm to evoke laughter by obtaining and ranking combination of terms when input a query. Our method works as follows:

(1) collects related words W_r combined for a query q.
(2) Calculates the unexpectedness of $w_r \in W_r$ for q.

(3) Calculates the polarity gap between q and $w_r \in W_r$
(4) $w_r \in W_r$, ranks the pair of w_r and q considering the unexpectedness and polarity gap for q.
(5) Displays the top K ranking pairs in query recommendation.

We explain each step in detail below.

4.1 Collecting related words

To save computational costs of ranking word in pairs, we must rank related words into a query. One way is to find candidate words with high co-occurrence. However, the co-occurrence and the unexpectedness between words offer a trade-off relation. To gather word candidates using only co-occurrence, we should miss unexpected word candidates. For this problem, we focus on the *Uncyclopedia*[2] as a resource to gather unexpected words and related words. The Uncyclopedia is an online encyclopedia that explains entities from a critical point of view. The Uncyclopedia contains many related topics about a target entity from an unexpected point of view. Thus, we consider that if two articles in the Uncyclopedia are linked to one another, both may be related and unexpectedly to the other. Thus, our system searches the Uncyclopedia for the articles linked to another article whose title is the query. After that, the system uses the names of the collected articles as word candidates to promote laughter.

4.2 Unexpectedness of word pairs

In this section, we focus on the difficulty in word association and word co-occurrence to compute the unexpectedness of word pairs.

If people are unable to imagine the word w_y from the word w_x, they imagine the word pair w_x and w_y. Thus, we assume that if w_x is unexpected for word w_y, it is difficult to associate w_y with w_x and its synonymous word w_r. For example, let us imagine the word pair "Mickey Mouse" and "greedy." If people could imagine that Mickey Mouse was greedy, the word pair "Mickey Mouse" and "greedy" would in an expected relation. Conversely, even if people failed to imagine that the mouse was greedy, if they imagine an image that other Disney main characters (synonyms of Mickey Mouse) were greedy, they could think that the word pair "Mickey Mouse - greedy" was possible. Therefore, we must analyze the difficulty in word association using a target word and its synonyms. Our system takes two steps to compute a difficulty in word association to collect the word pairs.

(1) Finds synonyms of a given word (query),
(2) Computes association of difficult candidate words (obtained in Section 4.1) for the query using the query's synonyms.

Our system computes association of difficult words for a given query from the two aspects: and (1) linkage of a word to word in Uncyclopedia and (2) co-occurrence of words in documents.

We hypothesize(1) that if word w_x links to word w_y in Uncyclopedia, w_x can associate w_y, and (2) the more synonyms of w_x link to w_y, the more w_y associates strongly with w_y. To use this hypothesis for computing the association of word difficulty to a given query, we must find the synonyms of the given query. Thus, we set an assumption that if the words w_x and w_y belong to the same category, w_y should be a synonym of w_x. Furthermore, we

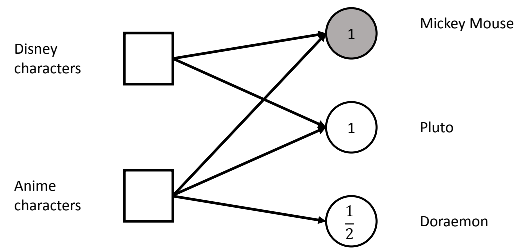

Figure 2: Likelihood of words to be synonym of a target word. In this figure, Mickey Mouse is a target word.

consider that the more categories w_y shares with w_x, the more likely w_y is a synonym of w_x. Based on this assumption, we compute the likelihood of w_y to be a synonym of w_x as follows:

$$Syno(w_y|w_x) = \frac{|Hype(w_x) \cap Hype(w_y)|}{|Hype(w_x)|} \quad (1)$$

Here, $Hype(w)$ represents the set of categories the word w belongs, and $|W|$ represents the number of elements of the set W.

Figure 2 shows the computation of how likely some words are synonyms of a target word "Mickey Mouse". In the figure, "Mickey Mouse" and "Pluto[3]" belong to the categories "anime character" and "Disney character". Moreover, "'Doraemon[4]" belongs to "anime character". In this example, the likelihood of "Draemon" to be a synonym of "Mickey Mouse" is 1/2 because "Draemon" shares one of the two categories with "Mickey Mouse". To implement the above proposition, we used the Hyponymy extraction tool provided by the Advanced Language Information Forum [5]. The tool extracts the relation between Wikipedia entities and categories. Moreover, we explain a method to compute a degree of word association with a query. As discussed previously, we hypothesize (1) that if word w_x links to word w_y in Uncyclopedia, w_x can associate with w_y, and (2) the more synonyms of w_x link to w_y, the strongly w_y associate with w_y. Considering these hypotheses and the likelihood of words to be a synonym of a word, we formalize the computation of the degree of word association with a query. Where q is a given query and a related word w_r, q is obtained via the Uncyclopedia, which is the association degree of w for q. Thus, $Asso(w|q)$ is computed using the following equation:

$$Asso(w_r|q) = \frac{1 + \sum_{w_s \in W_s(q)} Syno(w_r|q) \cdot Link(w_r|w_s)}{1 + |W_s(q)|} \quad (2)$$

Here, $W_s(q)$ represents the synonyms of q, and $Link(a|b)$ provides one if word a links to word b in Uncyclopedia otherwise zero.

Figure 3 illustrates an example of association degree of computation, given a query "Mickey Mouse". In this figure, left-side nodes are the target query and its synonyms, whereas right-side nodes are the related words of the query in Uncyclopedia. Links

[2] https://en.uncyclopedia.info/wiki

[3] Pluto: https://en.wikipedia.org/wiki/Pluto_(Disney)

[4] Draemon: https://en.wikipedia.org/wiki/Doraemon

[5] https://alaginrc.nict.go.jp/hyponymy/index.html

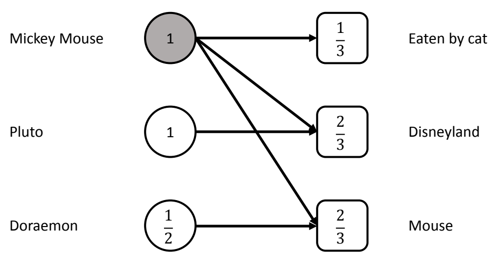

Figure 3: Example of association degree calculation.

from node X to node Y are X links to Y in Uncyclopedia. The numbers in the left-side nodes represent the synonym likelihood of the calculation in Fig. 2. In this example, the synonym likelihood of "Pluto" is 1; then, the association degree of term "Disneyland" for "Mickey Mouse", *Also*(Disneyland|Mickey Mouse) is equal to (1 + 1)/(1 + 2) = 2/3. In the proposed method, the co-occurrence degree $corr(w_x|w_y)$ between the words w_x and w_y is defined using the following equation.

$$Corr(w_x|w_y) = \frac{H(w_x \wedge w_y)}{H(w_x)} \tag{3}$$

Here, $H(w)$ is the number of webpages that contain w. We used the Bing Web Search API to obtain the number. From the above formula, the proposed system that computes the co-occurrence $Corr(w_r|q)$ of the query q and its related word w_r.

Using the association degree and co-occurrence, we obtain unexpected related words for a given query. With the unexpectedness of a related word w_r for query q, $Unexp(w_r|q)$ is computed as follows:

$$Unexp(w_r|q) = \frac{1}{\frac{Asso(w_r|q)+Corr(w_r|q)}{2}} \tag{4}$$

4.3 Semantic conflict

Several approaches are used to analyze the semantic conflict between two words. We focus on the polarity gap between the two words as one of the approximations of the semantic conflict. We also observe that semantic conflicted concept pairs conflict from the aspect of affected polarities like "peace (positive) and war (negative)" and "good (positive) and evil (negative)". Affected polarity indicates whether a word is positive or negative. Thus, the polarity range is 1 (positive) to -1 (negative). In this study, we calculate the polarity gap of two words w_x and w_y, $Conf(w_x, w_y)$ as the semantic conflict between w_x and w_y. The calculation formula is revealed as follows:

$$Conf(w_x, w_y) = |pol(w_x) - pol(w_y)| \tag{5}$$

Here, $pol(w_x)$ represents the polarity of the word w_x. We used the database developed by Takamura et al. to obtain the word polarity [16]. Some words are unavailable in the Takamura's database. In this case, we compute the polarity of such words w using the polarity of their co-occurring words in web documents. The procedure is as follows:

(1) The system searches for webpages that contain word w and extracts nouns from the webpages.
(2) The system computes the average polarity of the extracted nouns listed in the Takamura's table.

The Bing Web Search API[6] was used to search for webpages. The number of obtained search results was 20. The frequency of nouns is not considered when averaging the polarity of nouns.

4.4 Ranking of funny word pairs

Finally, the system ranks word candidates W_r (obtained in Section 4.1) by considering the unexpectedness and the semantic conflict of a given query q. The ranking function *Rank* for $w_r \in W_r$ is defined as follows:

$$Rank(w_r|q) = Conf(q, w_r) \cdot Unexp(q, w_r) \tag{6}$$

5 EXPERIMENT

We perform an experiment to evaluate the performance of our method when ranking a combination of funny terms. As discussed in Section 1, our proposed system aims to provide opportunities for people to laugh when they issue queries into Web search engines. This shows that the system unrecommends queries useful to search for funny web information. In this experiment, we evaluated how funny people felt with the term pairs that the system provided.

5.1 Baseline method

A thorough search of the literature shows that no methods find funny term pairs are available. For this experiment, we used the following methods to modify the proposed method:

- **Unexp**: The method where only the unexpectedness of term pairs in Equation 6 is considered.
- **Conf**: The method where only the semantic conflict of term pairs in Equation 6 is considered.

5.2 Evaluation metrics

We used *precision* and *mean average precision (MAP)* to evaluate the ranked lists for giving queries. The definition of precision is as follows:

$$P@k = \frac{|Pair_s(k) \cap Pair_h|}{k} \tag{7}$$

Here, $Pair_s(k)$ represents a set of the top k word pairs returned by the system, whereas $Pair_h$ represents a set of the word pairs that human evaluators judge as funny. In this experiment, if more than half of the evaluators judged a word pair as funny, the pair was treated as "funny".

MAP is defined as follows:

$$MAP = \frac{\sum_{q \in Q} AP(q)}{|Q|} \tag{8}$$

Here, q represents the query entered into the system, and Q represents the set of q. Furthermore, *AP(q)* is defined as follows:

$$AP(q) = \frac{1}{|Pair_s \cap Pair_h|} \sum_{k=1}^{|Pair_s|} I(k) P@k \tag{9}$$

[6]Bing Web Search API: https://azure.microsoft.com/en-us/services/cognitive-services/bing-web-search-api/

$I(k)$ returns one if the k-th pair in the ranking is funny otherwise zero.

5.3 Query set

For the experiment, we used a total of 15 queries listed in Table 1. We prepared five queries from three categories: anime character, sport, and food. Each query term is popular in Japan.

Table 1: Query set

Category	Queries
Character	Anpanman,[a] Moomin, Mickey Mouse, Rilakkuma,[b] Thomas the tank engine
Sport	baseball, soccer, boxing, tennis, golf
Food	curry rice, ramen, hamburger, nikujaga,[c] fried rice

[a] Anpanman: https://en.wikipedia.org/wiki/Anpanman
[b] Rilakkuma: https://en.wikipedia.org/wiki/Rilakkuma
[c] Nikujaga (potato with meat): https://en.wikipedia.org/wiki/Nikujaga

5.4 Evaluator

We used a Japanese crowdsourcing service, Lancers.jp[7], to recruit evaluators. We asked each evaluator to judge whether each pair in a list of a query word pairs was funny. Six evaluators were allocated for each query as revealed in Table 1. We recruited a total of 90 workers for the experiment. We paid 50 Japanese yen (approximately 50 cents) for each query.

5.5 Evaluation procedure

We asked the crowdsourcing evaluators to evaluate word pairs as follows. We showed the following description to explain an evaluation task procedure. Then, we asked each evaluator to evaluate whether each of the shown word pairs was funny.

> *For the task evaluation, please judge whether a combination of words "Mickey Mouse" and another word are funny. Several word pairs are presented in a next page. Please evaluate them. Select "funny" if you feel a word pair is funny, otherwise select "not funny".*
> *For example, with a combination of the word "traffic accident" and the word "Banana", imagine a traffic accident is caused by a car slipping a banana. If you feel the image funny because it could be practically impossible, select "funny."*

After the task introduction, we displayed the output results of the proposed method and the baseline methods for one of the queries in Table 1 presented to each evaluator.

6 RESULT

Table 2 shows P@5, P@10, P@15, AP, and MAP of our method and the baseline methods. The MAP values of our method was 0.247, unexpectedness factor only (**Unexp**) was 0.288, and semantic conflict factor only (**Conf**) was 0.222. The MAP results show that a method that considers only the unexpectedness could provide funny pairs for an average query.

[7] https://www.lancers.jp/

However, the AP values in Table 2 indicate that the proposed method performed better in some queries than the two baseline methods. For example, our method provided the best results for the six queries such as "Rilakkuma (*AP*=0.313)", "Tennis (*AP*=0.315)", and "Ramen (*AP*=0.380)." However, the table shows that our method fails to yield good results for some queries. In the case of the query "Thomas the Tank Engine", the proposed method provided the lowest value among the three methods (*AP*=0.243). Although the **Conf** method provided the lowest MAP, the method provided the better AP results than others queries such as "Mickey Mouse" (*AP*=0.364), "Soccer" (*AP*=0.25) or "Curry Rice" (*AP*=0.1).

Table 3 lists the top five results of the four queries for which the proposed method fails to perform better. As "Asopaso Maso", "One Pun Man" for the query "Anpanman", and "Pikmin" for the query "Moomin", we guess that the evaluators judged several term pairs as funny because their pronunciations and spellings were similar to those of the queries. Furthermore, we guessed that some results were funny because they cannot be imagined from the queries such as "intimidation." "Anpanman[8]."

Conversely, the evaluators judged that most of the results of some queries were unfunny. One possible reason was that the evaluators had few ideas about the provided results and they did not understand why the term pairs were funny. Although other people feel them as funny (e.g., "waraimeshi"[9] for the query "fried rice").

7 DISCUSSION

The MAP scores in Table 2 show that the method considers that the unexpectedness is better than the proposed method. However, once we checked the AP scores query by query, we found that the proposed method was the best for some queries. Several possible reasons for that are as follows:

The first reason is the computation of *unexpectedness*. To measure the unexpectedness of term pairs, the system computed the association degree and co-occurrence between two terms. The proposed method relies on the link structure of Wikipedia and term usage in webpages. If a given topic is unusual on the Web, the system may fail to measure the unexpectedness of term pairs for the topic appropriately.

The second reason is a result of the computation of the *semantic conflict*. The proposed method measures the polarity gap between terms to approximate the semantic conflict. However, we think that this approximation was too rough for the calculation of the semantic conflict because positive terms do not always conflict negative terms, and vice versa. In the future, we plan to use word embedding techniques like Word2Vec to check whether two terms are in a semantic conflict relation in a vector space.

8 CONCLUSION

In this paper, we propose a system to show funny term pairs when searchers issue a query into Web search engines. The proposed system analyzes the two factors of term pairs: the unexpectedness and the semantic conflict. For example, when the query "Mickey

[8] Anpanman is one of the famous anime characters in Japan and works for justice. This shows that it is unlikely that Anpaman is a threat to anyone.
[9] Waraimeshi is a Japanese comedian. The term "meshi" refers to rice in Japanese. The term "warai" refers to laughter in Japanese. If "waraimeshi" is to order "fried rice", we think some people might feel this as funny.

Table 2: Evaluation Results for P@5, P@10, P@15, AP and MAP

	P@5			P@10			P@15			AP		
Query	**Proposed**	**Unexp**	**Conf**	**Proposed**	**Unexp**	**Conf**	**Proposed**	**Unexp**	**Conf**	**Proposed**	**Unexp**	**Conf**
Anpanman	0.6	0.6	0.6	0.5	0.5	0.5	0.467	0.533	0.467	0.543	0.546	0.513
Mickey Mouse	0.2	0.2	0.2	0.2	0.3	0.4	0.200	0.200	0.400	0.266	0.272	0.364
Thomas the tank engine	0	0.4	0	0.2	0.2	0.2	0.267	0.133	0.333	0.243	0.253	0.313
Moomin	0.4	0.2	0.2	0.2	0.2	0.2	0.200	0.200	0.200	0.424	0.443	0.355
Rilakkuma	0.2	0.2	0.2	0.3	0.1	0.2	0.267	0.200	0.267	0.313	0.228	0.290
baseball	0.2	0	0	0.1	0.1	0.1	0.067	0.067	0.067	0.159	0.137	0.121
soccer	0.2	0	0.2	0.2	0.1	0.2	0.133	0.067	0.133	0.236	0.118	0.250
boxing	0	0	0.2	0.2	0.1	0.2	0.133	0.133	0.133	0.171	0.143	0.200
tennis	0.4	0.2	0.2	0.2	0.2	0.2	0.200	0.200	0.200	0.315	0.236	0.249
golf	0	0.2	0	0.1	0.1	0	0.067	0.067	0.067	0.125	1	0.091
curry rice	0	0	0.2	0.1	0	0.1	0.067	0	0.067	0.086	0.058	0.100
ramen	0.6	0.4	0	0.4	0.4	0	0.333	0.333	0.067	0.380	0.321	0.132
humberger	0	0	0	0.2	0	0.2	0.133	0	0.133	0.162	0.135	0.162
nikujaga	0.2	0.2	0	0.1	0.1	0	0.067	0.133	0.067	0.163	0.321	0.073
fried rice	0	0	0	0.1	0.1	0	0.133	0.067	0.067	0.119	0.111	0.112
MAP										0.247	0.288	0.222

Table 3: Top five results of the proposed method (✓is the combination judged to laugh. × is the combination judged not to laugh).

	Good result		Bad result	
Rank	Anpanman	Moomin	curry rice	fried rice
1	make threat ✓	Nagoya Republic ×	saint ×	waraimeshi ×
2	Asopaso Maso ✓	pikmin ✓	Bhaisajyaguru ×	Umi Natori ×
3	One Pun Man ✓	hippopotamus ×	Syashinshiko ×	sleepingness ×
4	Cardboard meat bun ×	Okehazama ✓	amphibian ×	spider ×
5	think with common sesne ×	Santa Claus ×	virtue ×	tabasco ×

Mouse" is input into the proposed system, the system shows such funny term pairs as "Mickey Mouse - eaten by cat" and "Mickey Mouse - greedy" in a list of query recommendations. We expect that the proposed system can help searchers to have opportunities to laugh and to feel cheery when issuing and thinking about queries into Web search engines. The experimental result showed that the proposed method offered a much larger number of funny term pairs for some queries than the baseline methods; although the proposed one was not the best average method. However, there remains a big problem in our method, particularly in the semantic conflict computation process. In the future, we plan to use word embedding techniques like Word2Vec to check whether two terms are in a semantical conflict relation with a vector space. Furthermore, we want to study effective searcher user interfaces to display funny term pairs and algorithms to search for funny webpages using the term pairs.

ACKNOWLEDGMENTS

This work was supported in part by Grants-in-Aid for Scientific Research (18H03243, 18H03244, 18H03494, 18KT0097, 18K18161, 16H02906) from MEXT of Japan.

REFERENCES

[1] Mary P Bennett, Janice M Zeller, Lisa Rosenberg, and Judith McCann. 2003. The effect of mirthful laughter on stress and natural killer cell activity. (2003).

[2] Narae Cha, Inyeop Kim, Mingyu Park, Auk Kim, and Uichin Lee. 2018. HelloBot: Facilitating Social Inclusion with an Interactive Greeting Robot. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*. ACM, 21–24.

[3] Rosemary Cogan, Dennis Cogan, William Waltz, and Melissa McCue. 1987. Effects of laughter and relaxation on discomfort thresholds. *Journal of behavioral medicine* 10, 2 (1987), 139–144.

[4] Oscar Déniz, M Castrillon, J Lorenzo, L Anton, and Gloria Bueno. 2008. Smile detection for user interfaces. In *International Symposium on Visual Computing*. Springer, 602–611.

[5] Ryohei Fushimi, Shogo Fukushima, and Takeshi Naemura. 2015. Laughin'Cam: Active Camera System to Induce Natural Smiles. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 1959–1964.

[6] Bryan Anthony Hong and Ethel Ong. 2009. Automatically extracting word relationships as templates for pun generation. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*. Association for Computational Linguistics, 24–31.

[7] Peter Khooshabeh, Cade McCall, Sudeep Gandhe, Jonathan Gratch, and James Blascovich. 2011. Does it matter if a computer jokes. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*. ACM, 77–86.

[8] Chloe Kiddon and Yuriy Brun. 2011. That's what she said: double entendre identification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*. Association for Computational Linguistics, 89–94.

[9] Boram Lee and Woohun Lee. 2009. Cheese cam: unconscious interaction between humans and a digital camera. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*. ACM, 4285–4290.

[10] Maurizio Mancini, Giovanna Varni, Radoslaw Niewiadomski, Gualtiero Volpe, and Antonio Camurri. 2014. How is your laugh today?. In *Proceedings of the extended abstracts of the 32nd annual ACM conference on Human factors in computing systems*. ACM, 1855–1860.

[11] Yelena Mejova, Ilaria Bordino, Mounia Lalmas, and Aristides Gionis. 2013. Searching for interestingness in Wikipedia and Yahoo!: answers.. In *WWW (Companion Volume)*. 145–146.

[12] Rada Mihalcea and Carlo Strapparava. 2005. Making computers laugh: Investigations in automatic humor recognition. In *Proceedings of the Conference on*

Human Language Technology and Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 531–538.

[13] Yohei Noda, Yoji Kiyota, and Hiroshi Nakagawa. 2010. Discovering serendipitous information from wikipedia by using its network structure. In *Fourth International AAAI Conference on Weblogs and Social Media*.

[14] Amruta Purandare and Diane Litman. 2006. Humor: Prosody analysis and automatic recognition for f* r* i* e* n* d* s. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 208–215.

[15] Naoki Shibahara et al. 2006. Mechanisms of the Generation of a Smile and Laughter. *J.Kinki Welf* 7, 1 (2006), 1–11.

[16] Hiroya Takamura, Takashi Inui, and Manabu Okumura. 2005. Extracting semantic orientations of words using spin model. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 133–140.

[17] Hitomi Tsujita and Jun Rekimoto. 2011. Smiling makes us happier: enhancing positive mood and communication with smile-encouraging digital appliances. In *Proceedings of the 13th international conference on Ubiquitous computing*. ACM, 1–10.

[18] Koji Tsukada and Maho Oki. 2010. EyeCatcher: a digital camera for capturing a variety of natural looking facial expressions in daily snapshots. In *International Conference on Pervasive Computing*. Springer, 112–129.

[19] Kosetsu Tsukuda, Hiroaki Ohshima, Mitsuo Yamamoto, Hirotoshi Iwasaki, and Katsumi Tanaka. 2013. Discovering unexpected information on the basis of popularity/unpopularity analysis of coordinate objects and their relationships. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. ACM, 878–885.

[20] Diyi Yang, Alon Lavie, Chris Dyer, and Eduard Hovy. 2015. Humor recognition and humor anchor extraction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2367–2376.

[21] Renxian Zhang and Naishi Liu. 2014. Recognizing humor on twitter. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, 889–898.