Automatic Narrative Humor Recognition Method Using Machine Learning and Semantic Similarity Based Punchline Detection

Rafal Rzepka, Yusuke Amaya, Motoki Yatsu and Kenji Araki
Graduate School of Information Science and Technology,
Hokkaido University, Sapporo, Japan
{rzepka,motoki.yatsu,araki}@ist.hokudai.ac.jp

Abstract

In this paper we introduce our method for recognizing jokes written in Japanese language by where the punchline is detected using WordNet. The results showed that when compared to method based on Bayesian posterior probability baseline, the proposed system achieved 5.3 point increase in recall and 2.6 point increase in classification accuracy. Our work[1] is the first challenge to detect humor in Japanese language and this ability can be utilized not only for more natural reactions while perceiving user's utterance, but also for discovering funny stories to be uttered by an agent.

1 Introduction

Humor is an important part of our lives and it is good for our health increasing immunity [Bennett and Lengacher, 2009] or lowering the stress level [Berk *et al.*, 1989]. Using it in a right moment, except its relaxing properties [Martin, 2010], is often treated as a manifestation of someone's intelligence. The same can be said about an ability of spotting a delicate joke hidden in everyday conversations. For a machine discovering a chance for using humor or detecting it opens new possibilities for maintaining natural flow of a conversation. This could be useful to enhance methods aiming at finding critical points in a dialog [Kubo and Abe, 2014] or to deal with moments, when there is a sudden need to change the topic as the conversation fades [Montero *et al.*, 2005]. Creating a companion robot that can deal with humor is regarded as important but very challenging task. For that reason many AI researchers have been developing various methods for both automatic generating [Taylor and Mazlack, 2004] [Tinholt and Nijholt, 2007] [Ritchie, 2005] [Strapparava *et al.*, 2011] and discovering humor for human-machine interaction. The most known research in the latter area was done by [Mihalcea and Pulman, 2007] [Mihalcea *et al.*, 2010] who introduced joke-specific features to increase efficiency of individual semantic relatedness measures which replaced stylistic features as used in their earlier work [Mihalcea, 2005].

In the case of Japanese language, there is a quite few examples of generating puns [Takizawa *et al.*, 1997] [Dybala *et al.*, 2008], but there is no, to the authors' best knowledge, research on recognizing humorous texts. To tackle this complicated problem it would be preferable to use deep semantic analysis, however the available tools and dictionaries are still not efficient enough, therefore we decided to use machine learning and similarity measures to see how helpful they are in the task of telling what is funny and what is not. Such ability would be preferred in companion machines, especially elderly-care robots which will most probably stay with us at homes in 2060, the year when the population of 60-years and older people in Japan is estimated to increase from 31.4% (2010) to 46.5%[2]. We believe that conversational skills of such agents will be crucial to their usability and for gaining trust, therefore both discovering a chance of joke in the utterance and finding humorous plots online for reusing it are important skills in a future robot toolbox. The ultimate goal of this research is use laughter when it should be used and cause laughter when a user needs it. In the following sections we introduce what material and methods we used to create the system and how efficient it was in empirical tests.

1.1 Type of Jokes We Used

Narrative jokes, not puns based mostly on acoustic similarities were used because we find it more challenging and useful in usual everyday conversation and depend on semantic dependencies between words. We extracted 987 texts from various sites found by a query "funny stories". Two examples of such humorous text are shown below.

> *She always asks me to ask her what she would like to eat. So I ask her what she would like to eat. But every time I do it, she answers "some food".*

> *Couple days ago a gas station guy asked me if I want some engine antifreeze coolant. I asked him how much it was and he answered "2000 yen for you, our valuable customer". So I asked how much was the price for a standard customer. He obviously didn't have that in his manual so after*

[1] The second author was the main contributor building the system and perform experiments. Because he never had a chance to share his work with the international audience before graduation, we decided to write a paper in English introducing his research.

[2] Governmental White Paper on Aging Society (2013): `http://www8.cao.go.jp/kourei/whitepaper/w-2013/zenbun/25pdf_index.html`, in Japanese, accessed April 2015.

few seconds he just said: "I think it is also 2000 yen, sir".

For experiments, we also retrieved 1,211 tweets with #twnovel hashtag as the non-humorous counterparts. Translation of an example is shown below.

I tweeted that I want to quit my job. "It's OK, do it", somebody answered, and his profile said he was 55. I kept tweeting "I want to quit" every day and suddenly my father came to Tokyo. "It's OK" he said hugging me for the first time in 20 years.

Because most of the "funny stories" were not ranked by readers and tweet novels could be amusing, we needed to ensure that more than one person agrees that a given story is humorous or not for the classification. We have cleaned texts removing duplicates, handle names, retweets and limited online jokes to shorter ones (less than 140 characters) to make both sets as similar as possible. We asked 11 students of our university (6 males, 5 females, 5 with background in humanities and 6 with science background). The survey task was to label a given text as laughable or not laughable. If more than half of participants agreed, the text was categorized as humorous (438) on non-humorous (362). An example of "funny stories" text which was not considered funny is given below.

When my son was in high school he told me not to approach him during the open house. Now he has long hair and no job. And he's told not to approach me when he sees me on the street.

2 Hybrid Approach for Recognizing Humor in Text

To investigate the ability to tell humorous text from non-humorous one, we decided to utilize two most popular classifying methods which are Support Vector Machine (SVM) [Cortes and Vapnik, 1995] and Naïve Bayes [Maron, 1961] together, i.e. adding a majority vote step. Because voting between only two classifiers is not effective if both outputs differ, we used one of the Naïve Bayes parameter – posterior probability – as an additional voter. This parameter describes classification probability of each category; if it is higher than certain threshold, it means the text contains funny elements and it is classified as humorous and non-humorous when the value is lower than the threshold (Figure 1 shows learning process and Figure 2 demonstrates how it is used further for recognition). More details on both methods are introduced in the next two subsection.

2.1 Classifier 1: Naïve Bayes

Naive Bayes classifiers are based on applying Bayes' theorem with strong (naïve) independence assumptions between the features. To classify text x it is needed to compare the posterior probability distributions of every category and to choose the one with biggest biggest value. The calculation are done by using the following formula,

$$P(x_i|c_k) = \frac{T(c_k, x_i)}{\sum_{x' V} T(c_k, x')} \tag{1}$$

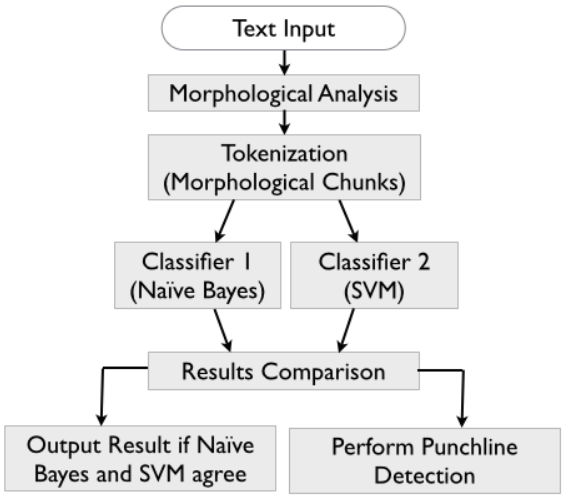

Figure 1: Learning part of our system.

where $P(x_i|c_k)$ is the conditional probability of x_i, V is the set of all words, and $T(c_k, x_i)$ is the number of occurrences of word x_i in category c_k. According to the Bayes theorem, we use posterior probability $P(B)$ (probability distribution of phenomena B occurring) and $P(B|A)$ (probability of phenomena B occurring after phenomena A), as shown below (with a condition that $P(A) > 0$).

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} \tag{2}$$

Classification is performed with

$$classify = argmax_{c_k} P(c_k|x) \tag{3}$$

and then the following Bayesian theorem

$$P(c_k|x) = \frac{P(x|c_k)P(c_k)}{P(x)} \tag{4}$$

is used, where $P(c_k)$ is a ratio of c_k category in learning set texts and $P(x|c_k)$ is the likelihood. Because Naïve Bayes assumes conditional independency between words, the formula is

$$P(x|c_k) = P(x_1 \dots x_K|c_k) = \prod_{i=1}^{K} P(x_i|c_k) \tag{5}$$

where the denominator is needed for normalizing the posterior probability distribution. This is done with

$$P(x) = \sum_{j=1}^{M} \left(P(c_j) \prod_{i=1}^{K} P(x_i|c_j) \right) \tag{6}$$

and the final formula for classifying texts with Naïve Bayes becomes as follows.

$$classify = argmax_{c_k} \frac{P(c_k) \prod_{i=1}^{K} P(x_i|c_k)}{\sum_{j=1}^{M} \left(P(c_j) \prod_{i=1}^{K} P(x_i|c_j) \right)} \tag{7}$$

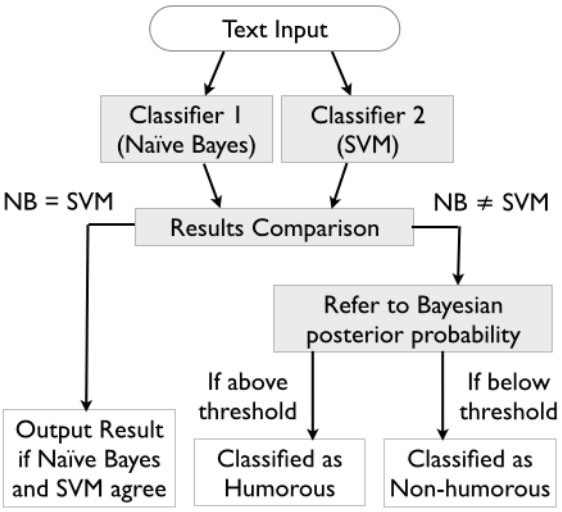

Figure 2: Algorithm of the proposed hybrid approach for recognizing humorous texts.

2.2 Classifier 2: Support Vector Machine

SVM is a popular non-probabilistic binary linear classifier utilizing a set of training examples marked for belonging to one of two categories, and an SVM training algorithm building a model to assign new examples into one of the categories. SVM constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space and he separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class. This point is called functional margin and is needed to assign an example x to one of the classes c_1 or c_2. In our case example x is represented as a feature vector $\boldsymbol{x}^T = (x_1, x_2, \ldots, x_M)$ made from morphological chunks (as in case of Naïve Bayes, we use bag-of-word approach here). Input feature vector is used for calculating two values of discrimination function as follows:

$$y = sign(\boldsymbol{w}^T \boldsymbol{x} - h) \tag{8}$$

As the learning set we used N feature vectors $x_1, \ldots, x_N$ and their training labels $y_1, \ldots, y_N$. If the set is linearly separable, there are parameters satisfying the following equation:

$$y_i(\boldsymbol{w}^T \boldsymbol{x}_i - h) \geq 1 \quad (i = 1, \ldots, N) \tag{9}$$

Two hyperplanes $\boldsymbol{w}^T \boldsymbol{x} - h = 1$ and $\boldsymbol{w}^T \boldsymbol{x} - h = -1$ divide the sets and the size of the margin becomes $\frac{1}{\|w\|}$ and the task is to find the biggest margin.

$$L(w) = min \frac{1}{2} \|w\|^2 \tag{10}$$

After Lagrangian multipler is used for optimization, the following conditions

$$\sum_{i=1}^{N} {}_i y_i = 0 \tag{11}$$

$$0 \leq \lambda_i \leq C, \quad (i = 1, \ldots, N) \tag{12}$$

objective function $L_D(\boldsymbol{\lambda})$ becomes

$$max_{\lambda_1, \ldots, \lambda_N} L_D(\boldsymbol{\lambda}) = \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{N} \lambda_i \lambda_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j \tag{13}$$

However, we assume that the set is linearly separable, essentially it is impossible to apply the method to non-linear cases, therefore the nonlinear transformation of feature vectors is performed and linear identification is carried out in their space. Now the original feature vector $\boldsymbol{x}_i$ is transformed by nonlinear map $\phi(\boldsymbol{x}_i)$ and because the inner product of input data from Equation 13 exists, the inner product of nonlinear map $\phi(\boldsymbol{x}_i)\phi(\boldsymbol{x}_j)$ is calculated by

$$\phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j) = K(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{14}$$

and when kernel $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is also calculated, we can construct optimized nonlinear map from $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ without using high-dimension $\phi(\boldsymbol{x}_i)\phi(\boldsymbol{x}_j)$.

Such implicitly mapping inputs into high-dimensional feature spaces is called a kernel trick and because the kernel's inner product $\phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j)$ must be definable and calculations as simple as possible, multinomial expression kernel or Gaussian kernel are commonly utilized. p is a parameter that determines the order of the kernel function, σ is a parameter that determines the spread of the kernel function.

Using the kernel trick,

$$max_{\lambda_1, \ldots, \lambda_N} L_D(\boldsymbol{\lambda}) = \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{N} \lambda_i \lambda_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{15}$$

$$y = sign \left(\sum_{i=1}^{N} \lambda_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}_j) - h \right) \tag{16}$$

decision space parameter h is determined and acquiring discrimination function y becomes possible without the need for calculating w directly. Learning is performed by setting the parameter h that properly divides the learning data and the Equations (11)(12)(15)(16) are used. Classification is performed by deciding if the input text belongs to category c_1 or c_2. If in Equation (16)$y = 1$, c_1 is chosen and if $y = -1$, c_2 is decided. Parameters are set according to Equations (11) (12)(15)(16.

3 Experiment for Classification Accuracy

We divided experiments into three parts. In the first experiment we evaluated classification accuracy on the set of texts which were classified in the same way by both Naïve Bayes and SVM classifiers. The second test used texts that gave different outputs and it helped us to set the posterior probability threshold for humorous ones. The probability was normalized to become 1 and we experimented with threshold values of 0.001, 0.1, 0.3, 0.7, 0.9 and 1 to find the best one. In the third experiment we used the best treshold confirmed in the second test to compare our proposed method with the baseline methods. In all experiments we used 10-fold cross-validation.

3.1 Experiment Results

We used Naïve Bayes and SVM separately as the baseline methods using all 724 texts. To measure effectiveness we we used standard precision/recall/accuracy calculations:

$$precision = \frac{TP}{TP + FP} \quad (17)$$

$$recall = \frac{TP}{TP + FN} \quad (18)$$

$$f_{score} = \frac{TP + TN}{TP + TN + FP + FN} \quad (19)$$

where TP, FP, TN, FN being number of classification true positives, false positives, true negatives and false negatives, respectively.

The hybrid method precision was 7.5 points better and recall 12.0 points better than Naïve Bayes as the baseline (see Table 1). When compared to SVM, proposed method achieved 8.4 higher precision and 6.8 higher recall. In future, we want to compare our method also with other classification methods, but they may be more time consuming which is a big obstacle when the method is utilized in a dialog processing system.

The second experiment was run on 162 sentences which did not match in the first one. The precision (54.7%) and recall (64.1%) of the classification was highest when 0.001 threshold was used (see Table 2). We set this treshold in the third experiment and compared the proposed system again with baselines, achieving further improvement – better precision and recall in both cases, Naïve Bayes (0.4 points and 6.1 points) and SVM (1.3 points and 0.9 points). The final results are shown in Table 3.

Table 1: Hybrid vs. Baseline – results comparison

Method (number of texts)	Precision(%)	Recall(%)	F-score
NB (724)	79.0	78.3	0.786
SVM (724)	78.1	83.5	0.807
If both classifiers agreed (562)	**86.5**	**90.3**	**0.883**

Table 2: Accuracy of different thresholds

Threshold	Precision (%)	Recall(%)	F-score
0.001	**54.7**	**64.1**	**0.590**
0.1	50.9	45.4	0.480
0.3	51.2	41.9	0.461
0.7	49.9	36.4	0.421
0.9	54.2	36.4	0.436
1	49.8	23.3	0.317

However, results of the second experiment when the 0.001 threshold was used are not too high (precision 54.7%, recall 64.1%). The case when both methods disagreed was influencing the overall evaluation, results of the third experiments were worse than the results of the first one (7.1 points drop in precision and 5.9 points decrease in recall). One probable reason for the lower efficiency is that in the case of humor posterior probability becomes 0 or 1 and does not differ much even if the threshold is set. It was obviously necessary to deal with the disagreeing outputs.

Table 3: Hybrid vs. Baseline – overall evaluation

Method	Precision (%)	Recall (%)	F-score
NB	79.0	78.3	0.786
SVM	78.1	83.5	0.807
Both (Hybrid)	**79.4**	**84.4**	**0.818**

4 Adding Punchline Detection

To deal with the problem described above, we decided to extend our system adding a punchline detection module which could help deciding if an input is humorous or not when the classifiers disagree. We based this idea on Incongruity Theory, which was being developed by various thinkers throughout the ages from Aristotle to Kant, but it was [Schopenhauer, 1907] who shaped it philosophically. The theory says that it is the perception of something incongruous, i.e. something that violates our mental patterns and expectations. This approach was successfully adopted for humor detection in English [Mihalcea *et al.*, 2010] and we decided to take the same approach and use knowledge-base semantic relatedness to see how surprisingly (incongruously) different is one part of the text when compared with another. We also used WordNet [Miller, 1995] which is widely used for calculating similarity between texts and tested six popular methods described later.

4.1 Processing Steps

As the text is input, the system checks if it consists of more than two sentences, if there is two or less, the posterior probability from Naïve Bayes is used because the comparison between only two sentences is inefficient and in case of one is impossible. After cleaning the sentences from curly brackets and other unnecessary symbols, morphological analysis is performed with MeCab analyzer [Kudo, 2005] to separate words (nouns, verbs, adjectives and adverbs that exist in Japanese WordNet). With these words the semantic similarity between two consecutive sentences is calculated. After the average similarity values are obtained, the system repeats the procedure for the next pair and this way the standard values for comparison is acquired. We utilize two types of such values – the first one is an average of all sentences excluding similarity between the last sentence and its precedent, the second is the smallest value of all sentences also without the last sentence precedent. In the end the set average similarity is compared with the last and the preceding sentences similarity. If the two last sentences gave similarity lower than the average one, the system estimates that incongruity is big enough to label the text as humorous. If it becomes higher, then the text is classified as non-humorous (see Figure 3).

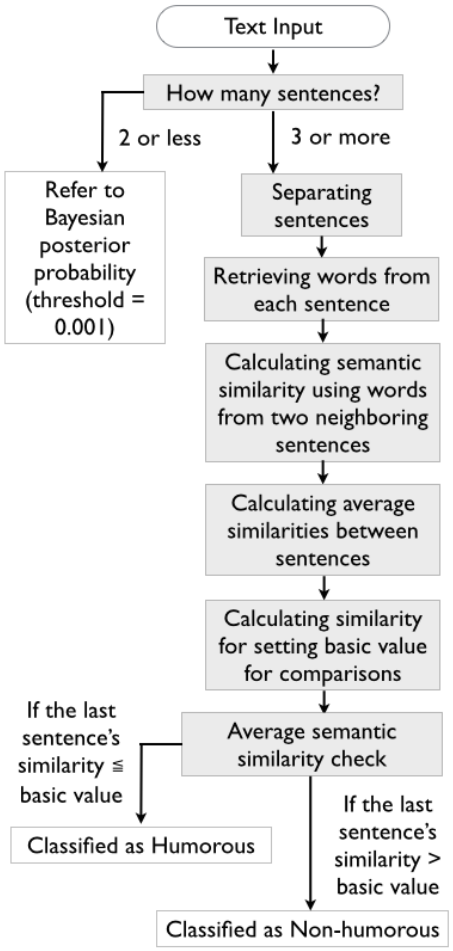

Figure 3: Algorithm for classification using punchline detection.

4.2 Methods for Calculating Lexical Similarities

For measuring semantic similarities we used six popular methods described briefly below.

Path Length

Path method computes the semantic relatedness of word senses by counting the number of nodes along the shortest path between the senses in the 'is-a' hierarchies of WordNet. Since a longer path length indicate less relatedness, the relatedness value returned is the multiplicative inverse of the path $length$ (distance) between the two concepts: relatedness = 1 / distance (see Equation 20). If the two concepts are identical, then the distance between them is one; therefore, their relatedness is also 1 .

$$Sim_{path} = \frac{1}{1 + length} \tag{20}$$

Leacock & Chodorow Similarity

[Leacock and Chodorow, 1998] proposed the following method:

$$Sim_{lch} = -log\frac{length}{2 * D} \tag{21}$$

where $length$ is the shortest distance between the two concepts (using node-counting) and D is the maximum depth of the taxonomy.

Wu & Palmer Measure

Measure proposed by [Wu and Palmer, 1994] calculates similarity by considering the depths of the two concepts along with the depth of the LCS (Least Common Superconcept). The formula, shown below, means that $0 < score \leqslant 1$. The score can never be zero because the depth of the LCS is never zero (the depth of the root of a taxonomy is one). The score is one if the two input concepts are the same.

$$Sim_{wup} = \frac{2 * depth(LCS)}{depth(concept_1) + depth(concept_2)} \tag{22}$$

Resnik Similarity

In the method of [Resnik, 1995] is based on the hypothesis saying that two similar concepts are probably more similar when they share information content. The related value (see Equation 23) is equal to the information content (IC) of the Least Common Subsumer (LCS) (most informative subsumer), which means that the value will always be greater-than or equal-to zero.

$$Sim_{res} = IC(LCS) \tag{23}$$

Lin Similarity

In the calculation using LCS and IC proposed by [Lin, 1998], the similarity value returned by the measure is a number shown in Equation 24, where $IC(concept)$ is the information content of $concept$. The similarity value becomes greater-than or equal-to zero and less-than or equal-to one.

$$Sim_{lin} = \frac{2 * IC(LCS)}{IC(concept_1) + IC(concept_2)} \tag{24}$$

Jiang & Conrath Similarity

[Jiang and Conrath, 1997] proposed a method which returns a similarity value calculated as follows.

$$Sim_{jcn} = \frac{1}{IC(concept_1) + IC(concept_2) - 2 * IC(LCS)} \tag{25}$$

The core of the idea, based on Resnik [Resnik, 1995] is to consider the information content of lowest common subsumer (LCS) and the two compared concepts to calculate the distance between the two concepts and the distance is then used to compute the similarity measure.

5 Evaluation Experiments

We performed two following experiments. The baseline in the first one was posterior probability of Naïve Bayes and the aimed to evaluate the efficiency of punchline detection. The second one was to test if adding punchline detection module to the hybrid method was useful and the hybrid system became the baseline method. In both experiments the used

threshold was for posterior probability was set to 0.001, the value experimentally confirmed in the preliminary tests. For validation tests we again used 10-fold cross-validation using 124 texts on which both Naïve Bayes and SVM disagreed. For the second experiment 726 previously described texts were utilized.

Results

Table 4 shows results for classifying 124 texts achieved by the baseline and punchline detection method. Recall was very high (96.9% in case of Lin) and all the semantic similarity calculation methods significantly surpassed the baseline. Also precision appeared to be better for all methods (Leacock & Chodorow abd Lin in the top with 60.4%). F-score was highest for Lin's method acquiring 0.726 and it also exceeded the baseline which was 0.590. From the results we can observe that it is better to use the average similarity as the base, not the lowest similarity value when detecting incongruity (in the case of Lin's method, precision increase is 4.0 points and recall increase is 16.6 points).

Table 5 shows performance results of hybrid method equipped with punchline detection ability. The comparison showed that, depending on semantic similarity measure, there was a recall increase between 2.1 and 5.3 points, and precision increase between 2.0 and 2.6 points. When f-score is considered and all sentences average is the base, Lin's method surpassed the baseline by 3.8 points showing that the system enhancement was succesfull.

Error analysis

The first observable problem was the insufficient number of words in the knowledge base. In erroneous classifications there were cases where not existing in Japanese WordNet "after that" or "such" had low similarity so the incongruity was detected where there was actually not much of a surprise. Another problem, which was obvious from the start, was the lack of deeper understanding, because not all funny stories have a surprising punchline, every sentence can be evenly amusing.

6 Conclusions and Future Work

In this paper we presented the first trial to automatically detect humor in Japanese texts. First, we introduced a hybrid system utilizing both Support Vector Machine and Naïve Bayes classifiers which performed better than both methods used separately. To deal with the problem of both methods disagreement and to enable voting we decided to add a module that is capable of detecting punchlines. For this we used six popular semantic similarity measures and discovered that Lin's method is most useful for our approach. We experimentally confirmed that the performance of the enhanced hybrid system achieved better results in both precision and recall. The experiments showed that even without deeper language understanding algorithms, our statistical approach is able to recognize humor in Japanese texts which is important for detecting funniness in utterances and finding humorous stories online for reusing them later in own utterances to amuse users. In the next step we plan to improve the recognition performance, especially the precision, by modeling the human humor cognition and adding techniques for deeper language

Table 4: Performance comparison between punchline detection module and posterior probability.

algorithm	precision (%)	recall (%)	F-score
NB Post	54.7	64.1	0.590
Case where last and preceding sentences similarity is omitted and whole text average similarity is the base			
Path	60.3	91.7	0.708
Leacock	**60.4**	91.7	0.708
Wu	59.3	90.2	0.699
Resnik	60.1	94.3	0.717
Lin	**60.4**	**96.9**	**0.726**
Jiang	60.1	94.8	0.714
Case where last and preceding sentences similarity is omitted and whole text lowest similarity is the base			
Path	58.4	88.2	0.684
Leacock	58.4	88.2	0.684
Wu	57.0	83.0	0.655
Resnik	56.4	80.3	0.642
Lin	56.4	80.3	0.642
Jiang	56.9	86.1	0.666

Table 5: Performance comparison between hybrid system alone and hybrid system with punchline detection module.

Algorithm	Precision (%)	Recall (%)	F-score
NB Post	79.4	84.4	0.818
Case where last and preceding sentences similarity is omitted and whole text average similarity is the base			
Path	**82.0**	88.7	0.852
Leacock	**82.0**	88.7	0.851
Wu	81.8	88.4	0.850
Resnik	**82.0**	89.2	0.854
Lin	**82.0**	**89.7**	**0.856**
Jiang	81.8	89.2	0.853
Case where last and preceding sentences similarity is omitted and whole text lowest similarity is the base			
Path	81.7	88.1	0.848
Leacock	81.7	88.1	0.848
Wu	81.4	87.0	0.841
Resnik	81.4	86.5	0.839
Lin	81.4	86.5	0.839
Jiang	81.4	87.6	0.844

understanding. To achieve this goal it is necessary to analyze human joking strategies and discover further features which could be used for automatic classification.

References

[Bennett and Lengacher, 2009] Mary Payne Bennett and Cecile Lengacher. Humor and laughter may influence health iv. humor and immune function. *Evidence-based Complementary and Alternative Medicine : eCAM*, 6(2):159–164, 06 2009.

[Berk *et al.*, 1989] L S Berk, S A Tan, W F Fry, B J Napier, J W Lee, R W Hubbard, J E Lewis, and W C Eby. Neuroendocrine and stress hormone changes during mirthful laughter. *Am J Med Sci*, 298(6):390–396, Dec 1989.

[Cortes and Vapnik, 1995] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, September 1995.

[Dybala *et al.*, 2008] Pawel Dybala, Michal Ptaszynski, Shinsuke Higuchi, Rafal Rzepka, and Kenji Araki. Humor prevails! – implementing a joke generator into a conversational system. In *AI 2008: Advances in Artificial Intelligence*, pages 214–225. Springer Berlin Heidelberg, 2008.

[Jiang and Conrath, 1997] Jay J. Jiang and David W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *CoRR*, 1997.

[Kubo and Abe, 2014] Masumi Kubo and Akinori Abe. Critical points in conversations from the perspective of chance discovery. *Procedia Computer Science*, 35(0):969 – 978, 2014. Knowledge-Based and Intelligent Information and Engineering Systems 18th Annual Conference, KES-2014 Gdynia, Poland, September 2014 Proceedings.

[Kudo, 2005] Taku. Kudo. Mecab : Yet another part-of-speech and morphological analyzer. *http://taku910.github.io/mecab/*, 2005.

[Leacock and Chodorow, 1998] Claudia Leacock and Martin Chodorow. Combining local context and wordnet similarity for word sense identification. In Christiane Fellfaum, editor, *MIT Press*, pages 265–283, Cambridge, Massachusetts, 1998.

[Lin, 1998] Dekang Lin. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine learning (ICML-98)*, 1998.

[Maron, 1961] M. E. Maron. Automatic indexing: An experimental inquiry. *Journal of the ACM*, 8(3):404–417, 1961.

[Martin, 2010] R.A. Martin. *The Psychology of Humor: An Integrative Approach*. Educational psychology. Elsevier Science, 2010.

[Mihalcea and Pulman, 2007] Rada Mihalcea and Stephen Pulman. Characterizing humour: An exploration of features in humorous texts. In *Proceedings of the 8thInternational Conference on Computational Linguistics and Intelligent Text Processing*, pages 337–347, 2007.

[Mihalcea *et al.*, 2010] Rada Mihalcea, Carlo Strapparava, and Stephen Pulman. Computational models for incongruity detection in humour. In *Computational Linguistics and Intelligent Text Processing Lecture Notes in Computer Science*, volume 6008, pages 364–374, 2010.

[Mihalcea, 2005] Rada Mihalcea. Making computers laugh: Investigations in automatic humor recognition. In *In Proc. of the Joint Conference on Human Language Technology / Empirical Methods in Natural Language Processing (HLT/EMNLP*, 2005.

[Miller, 1995] George A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38:39–41, 1995.

[Montero *et al.*, 2005] CalkinA.S. Montero, Yukio Ohsawa, and Kenji Araki. Modeling the discovery of critical utterances. In Rajiv Khosla, RobertJ. Howlett, and LakhmiC. Jain, editors, *Knowledge-Based Intelligent Information and Engineering Systems*, volume 3681 of *Lecture Notes in Computer Science*, pages 554–560. Springer Berlin Heidelberg, 2005.

[Resnik, 1995] Philip Resnik. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *Proceedings of the XI International Joint Conferences on Artificial*, pages 448–453, 1995.

[Ritchie, 2005] Graeme Ritchie. Computational mechanisms for pun generation. In *Proceedings of the 10th European Natural Language Generation Workshop*, pages 125–132. Citeseer, 2005.

[Schopenhauer, 1907] A. Schopenhauer. *The World as Will and Idea*. Kegan Paul, 1907.

[Strapparava *et al.*, 2011] Carlo Strapparava, Oliviero Stock, and Rada Mihalcea. Computational humour. In Roddy Cowie, Catherine Pelachaud, and Paolo Petta, editors, *Emotion-Oriented Systems*, Cognitive Technologies, pages 609–634. Springer Berlin Heidelberg, 2011.

[Takizawa *et al.*, 1997] Osamu Takizawa, Masuzo Yanagida, Akira Ito, and Hitoshi Isahara. A computational processing of rhetorical expressions : Puns, ironies, and tautologies. *Technical report of IEICE. Thought and language*, 97(79):9–16, may 1997.

[Taylor and Mazlack, 2004] J Taylor and L Mazlack. Computationally recognizing wordplay in jokes. *Proceedings of CogSci 2004*, 2004.

[Tinholt and Nijholt, 2007] Hans Wim Tinholt and Anton Nijholt. Computational humour: Utilizing cross-reference ambiguity for conversational jokes. In Francesco Masulli, Sushmita Mitra, and Gabriella Pasi, editors, *WILF*, volume 4578 of *Lecture Notes in Computer Science*, pages 477–483. Springer, 2007.

[Wu and Palmer, 1994] Zhibiao Wu and Martha Palmer. Verb semantics and lexical selection. In *Proc. of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138, 1994.