Graph-based Clustering for Semantic Classification of Onomatopoetic Words

Kenichi Ichioka
Interdisciplinary Graduate School of
Medicine and Engineering
University of Yamanashi, Japan
g07mk001@yamanashi.ac.jp

Fumiyo Fukumoto
Interdisciplinary Graduate School of
Medicine and Engineering
University of Yamanashi, Japan
fukumoto@yamanashi.ac.jp

Abstract

This paper presents a method for semantic classification of onomatopoetic words like "ひゅーひゅー (*hum*)" and "からんころん (*clip clop*)" which exist in every language, especially Japanese being rich in onomatopoetic words. We used a graph-based clustering algorithm called $Newman$ clustering. The algorithm calculates a simple quality function to test whether a particular division is meaningful. The quality function is calculated based on the weights of edges between nodes. We combined two different similarity measures, distributional similarity, and orthographic similarity to calculate weights. The results obtained by using the Web data showed a 9.0% improvement over the baseline single distributional similarity measure.

1 Introduction

Onomatopoeia which we call *onomatopoetic* word (*ono* word) is the formation of words whose sound is imitative of the sound of the noise or action designated, such as 'hiss' (McLeod, 1991). It is one of the linguistic features of Japanese. Consider two sentences from Japanese.

(1) 私は廊下のスリッパの音で起こされたので、とても眠い。
"I'm too sleepy because I awoke to the slippers in the hall."

(2) 私は廊下を ぱたぱた 走るスリッパの音で起こされたので、とても眠い。
"I'm too sleepy because I awoke to the *pit-a-pat* of slippers in the hall."

Sentences (1) and (2) are almost the same sense. However, sentence (2) which includes *ono* word, "ぱたぱた (*pit-a-pat*)" is much better to make the scene alive, or represents an image clearly. Therefore large-scale semantic resource of *ono* words is indispensable for not only NLP, but also many semantic-oriented applications such as Question Answering, Paraphrasing, and MT systems. Although several machine-readable dictionaries which are fine-grained and large-scale semantic knowledge like WordNet, COMLEX, and EDR dictionary exist, there are none or few onomatopoetic thesaurus. Because (i) it is easy to understand its sense of *ono* word for Japanese, and (ii) it is a fast-changing linguistic expressions, as it is a vogue word. Therefore, considering this resource scarcity problem, semantic classification of *ono* words which do not appear in the resource but appear in corpora is very important.

In this paper, we focus on Japanese onomatopoetic words, and propose a method for classifying them into a set with similar meaning. We used the Web as a corpus to collect *ono* words, as they appear in different genres of dialogues including broadcast news, novels and comics, rather than a well-edited, balanced corpus like newspaper articles. The problem using a large, heterogeneous collection of Web data is that the Web counts are far more noisy than counts obtained from textual corpus. We thus used a graph-based clustering algorithm, called $Newman$ clustering for classifying *ono* words. The algorithm does not simply calculate the number of shortest paths between pairs of nodes, but instead calculates a quality function

Coling 2008: Proceedings of 3rd Textgraphs workshop on Graph-Based Algorithms in Natural Language Processing, pages 33–40
Manchester, August 2008

of how good a cluster structure found by an algorithm is, and thus makes the computation far more efficient. The efficacy of the algorithm depends on a quality function which is calculated by using the weights of edges between nodes. We combined two different similarity measures, and used them to calculate weights. One is co-occurrence based distributional similarity measure. We tested mutual information (MI) and a χ^2 statistic as a similarity measure. Another is orthographic similarity which is based on a feature of *ono* words called "sound symbolism". Sound symbolism indicates that phonemes or phonetic sequences express their senses. As *ono* words imitate the sounds associated with the objects or actions they refer to, their phonetic sequences provide semantic clues for classification. The empirical results are encouraging, and showed a 9.0% improvement over the baseline single distributional similarity measure.

2 Previous Work

There are quite a lot of work on semantic classification of words with corpus-based approach. The earliest work in this direction are those of (Hindle, 1990), (Lin, 1998), (Dagan et al., 1999), (Chen and Chen, 2000), (Geffet and Dagan, 2004) and (Weeds and Weir, 2005). They used distributional similarity. Similarity measures based on distributional hypothesis compare a pair of weighted feature vectors that characterize two words. Features typically correspond to other words that co-occur with the characterized word in the same context. Lin (1998) proposed a word similarity measure based on the distributional pattern of words which allows to construct a thesaurus using a parsed corpus. He compared the result of automatically created thesaurus with WordNet and Roget, and reported that the result was significantly closer to WordNet than Roget Thesaurus was.

Graph representations for word similarity have also been proposed by several researchers (Jannink and Wiederhold, 1999; Galley and McKeown, 2003; Muller et al., 2006). Sinha and Mihalcea (2007) proposed a graph-based algorithm for unsupervised word sense disambiguation which combines several semantic similarity measures including Resnik's metric (Resnik, 1995), and algorithms for graph centrality. They reported that the results using the SENSEVAL-2 and SENSEVAL-3 English all-words data sets lead to relative error rate reductions of $5 - 8\%$ as compared to the previous work (Mihalcea, 2005).

In the context of graph-based clustering of words, Widdows and Dorow (2002) used a graph model for unsupervised lexical acquisition. The graph structure is built by linking pairs of words which participate in particular syntactic relationships. An incremental cluster-building algorithm using the graph structure achieved 82% accuracy at a lexical acquisition task, evaluated against WordNet 10 classes, and each class consists of 20 words. Matsuo *et al.* (2006) proposed a method of word clustering based on a word similarity measure by Web counts. They used Newman clustering for clustering algorithm. They evaluated their method using two sets of word classes. One is derived from the Web data, and another is from WordNet.[1] Each set consists of 90 noun words. They reported that the results obtained by Newman clustering were better than those obtained by average-link agglomerative clustering. Our work is similar to their method in the use of Newman clustering. However, they classified Japanese noun words, while our work is the first to aim at detecting semantic classification of onomatopoetic words. Moreover, they used only a single similarity metric, co-occurrence based similarity, while Japanese, especially "kanji" characters of noun words provide semantic clues for classifying words.

3 System Description

The method consists of three steps: retrieving co-occurrences using the Web, calculating similarity between *ono* words, and classifying *ono* words by using Newman clustering.

3.1 Retrieving Co-occurrence using the Web

One criterion for calculating semantic similarity between *ono* words is co-occurrence based similarity. We retrieved frequency of two *ono* words occurring together by using the Web search engine, Google. The similarity between them is calculated based on their co-occurrence frequency. Like much previous work on semantic classification of the lexicons, our assumption is that semantically similar words appear in similar contexts. A lot of strategies for searching words are provided in Google. Of these we focused on two methods: Boolean search AND and $phrase$-based search.

[1] They used WordNet hypernym information. It consists of 10 classes. They assigned 90 Japanese noun words to each class.

When we use AND boolean search, *i.e.*, (O_i O_j) where O_i and O_j are *ono* words, we can retrieve the number of documents which include both O_i and O_j. In contrast, $phrase$-based search, *i.e.*, ("O_i O_j") retrieves documents which include two adjacent words O_i and O_j.

3.2 Similarity Measures

The second step is to calculate semantic similarity between *ono* words. We combined two different similarity measures: the co-occurrence frequency based similarity and orthographic similarity measures.

3.2.1 Co-occurrence based Similarity Measure

We focused on two popular measures: the mutual information (MI) and χ^2 statistics.

1. Mutual Information

Church and Hanks (1990) discussed the use of the mutual information statistics as a way to identify a variety of interesting linguistic phenomena, ranging from semantic relations of the doctor/nurse type (content word/content word) to lexico-syntactic co-occurrence preferences between verbs and prepositions (content word/function word). Let O_i and O_j be *ono* words retrieved from the Web. The mutual information $MI(O_i, O_j)$ is defined as:

$$\begin{aligned} MI(O_i, O_j) &= \log \frac{S_{all} \times f(O_i, O_j)}{S_{O_i} \times S_{O_j}}, & (1) \\ \text{where } S_{O_i} &= \sum_{k \in O_{all}} f(O_i, O_k), & (2) \\ S_{all} &= \sum_{O_i \in O_{all}} S_{O_i}. & (3) \end{aligned}$$

In Eq. (1), $f(O_i, O_j)$ refers to the frequency of O_i and O_j occurring together, and O_{all} is a set of all *ono* words retrieved from the Web.

2. χ^2 statistic

The $\chi^2(O_i, O_j)$ is defined as:

$$\begin{aligned} \chi^2(O_i, O_j) &= \frac{f(O_i, O_j) - E(O_i, O_j)}{E(O_i, O_j)}, & (4) \\ \text{where } E(O_i, O_j) &= S_{O_i} \times \frac{S_{O_j}}{S_{all}}. & (5) \end{aligned}$$

S_{O_i} and S_{all} in Eq. (5) refer to Eq. (2) and (3), respectively. A major difference between χ^2 and MI is that the former is a normalized value.

3.2.2 Orthographic Similarity Measure

Orthographic similarity has been widely used in spell checking and speech recognition systems (Damerau, 1964). Our orthographic similarity measure is based on a unit of phonetic sequence. The key steps of the similarity between two *ono* words is defined as:

1. Convert each *ono* word into phonetic sequences.

   The "hiragana" characters of *ono* word are converted into phonetic sequences by a unique rule. Basically, there are 19 consonants and 5 vowels, as listed in Table 1.

   Table 1: Japanese consonants and vowels

Consonant	–, N, Q, h, hy, k, ky, m, my, n, ny, r, ry, s, sy, t, ty, w, y
Vowel	a, i, u, e, o

   Consider phonetic sequences "hyu-hyu-" of *ono* word "ひゅーひゅー" (*hum*). It is segmented into 4 consonants "hy", "-", "hy" and "-", and two vowels, "u" and "u".

2. Form a vector in n-dimensional space.

   Each *ono* word is represented as a vector of consonants(vowels), where each dimension of the vector corresponds to each consonant and vowel, and each value of the dimension is frequencies of its corresponding consonant(vowel).

3. Calculate orthographic similarity.

   The orthographic similarity between *ono* words, O_i and O_j is calculated based on the consonant and vowel distributions. We used two popular measures, *i.e.*, the cosine similarity, and α-skew divergence. The cosine measures the similarity of the two vectors by calculating the cosine of the angle between vectors. α-skew divergence is defined as:

   $$\alpha div(x, y) = D(y \,||\, \alpha \cdot x + (1 - \alpha) \cdot y),$$

   where $D(x||y)$ refers to Kullback-Leibler and defined as:

   $$D(x||y) = \sum_{i=1}^{n} x_i * \log \frac{x_i}{y_i}. \qquad (6)$$

Lee (1999) reported the best results with α = 0.9. We used the same value. We defined a similarity metric by combining co-occurrence based and orthographic similarity measures[2]:

$$Sim(O_i, O_j) = MI(O_i, O_j) \times (Cos(O_i, O_j) + 1) \quad (7)$$

3.3 The Newman Clustering Algorithm

We classified *ono* words collected from the WWW. Therefore, the clustering algorithm should be efficient and effective even in the very high dimensional spaces. For this purpose, we chose a graph-based clustering algorithm, called $Newman$ clustering. The Newman clustering is a hierarchical clustering algorithm which is based on Network structure (Newman, 2004). The network structure consists of nodes within which the node-node connections are edges. It produces some division of the nodes into communities, regardless of whether the network structure has any natural such division. Here, "community" or "cluster" have in common that they are groups of densely interconnected nodes that are only sparsely connected with the rest of the network. To test whether a particular division is meaningful a quality function Q is defined:

$$Q = \sum_i (e_{ii} - a_i^2)$$

where e_{ij} is the sum of the weight of edges between two communities i and j divided by the sum of the weight of all edges, and $a_i = \sum_j e_{ij}$, *i.e.*, the expected fraction of edges within the cluster. Here are the key steps of that algorithm:

1. Given a set of n *ono* words $S = \{O_1, \cdots, O_n\}$. Create a network structure which consists of nodes $O_1, \cdots, O_n$, and edges. Here, the weight of an edge between O_i and O_j is a similarity value obtained by Eq. (7). If the "network density" of *ono* words is smaller than the parameter θ, we cut the edge. Here, "network density" refers to a ratio selected from the topmost edges. For example, if it was 0.9, we used the topmost 90% of all edges and cut the remains, where edges are sorted in the descending order of their similarity values.

2. Starting with a state in which each *ono* word is the sole member of one of n communities, we repeatedly joined communities together in pairs, choosing at each step the join that results in the greatest increase.

3. Suppose that two communities are merged into one by a join operation. The change in Q upon joining two communities i and j is given by:

$$\begin{aligned} \triangle Q_{ij} &= e_{ij} + e_{ji} - 2a_i a_j \\ &= 2(e_{ij} - a_i a_j) \end{aligned}$$

4. Apply step 3. to every pair of communities.

5. Join two communities such that $\triangle Q$ is maximum and create one community. If $\triangle Q < 0$, go to step 7.

6. Re-calculate e_{ij} and a_i of the joined community, and go to step 3.

7. Words within the same community are regarded as semantically similar.

The computational cost of the algorithm is known as $O((m + n)n)$ or $O(n^2)$, where m and n are the number of edges and nodes, respectively.

4 Experiments

4.1 Experimental Setup

The data for the classification of *ono* words have been taken from the Japanese *ono* dictionary (Ono, 2007) that consisted of 4,500 words. Of these, we selected 273 words, which occurred at least 5,000 in the document URLs from the WWW. The minimum frequency of a word was found to be 5,220, while the maximum was about 26 million. These words are classified into 10 classes. Word classes and examples of *ono* words from the dictionary are listed in Table 2.

"Id" denotes id number of each class. "Sense" refers to each sense of *ono* word within the same class, and "Num" is the number of words which should be assigned to each class. Each word

[2] When we used χ^2 statistic as a co-occurrence based similarity, MI in Eq. (7) is replaced by χ^2. In a similar way, $Cos(O_i, O_j)$ is replaced by $max - \alpha div(x, y)$, where max is the maximum value among all $\alpha div(x, y)$ values.

Table 2: Onomatopoetic words and # of words in each class

Id	*Sense*	Num	Onomatopoetic words
1	*laugh*	63	あっはっは (a,Q,h,a,Q,h,a), あはは (a,h,a,h,a), わはは (w,a,h,a,h,a) あはあは (a,h,a,a,h,a), いひひ (i,h,i,h,i), うっしっし (u,Q,s,i,Q,s,i), · · ·
2	*cry*	34	あーん (a,–,N), うわーん (u,w,a,–,N), あんあん (a,N,a,N), えんえん (e,N,e,N) うるうる (u,r,u,u,r,u), うるるん (u,r,u,r,u,N), うるっ(u,r,u,Q), えーん (e,–,N), · · ·
3	*pain*	34	いがいが (i,k,a,i,k,a), ひりひり (h,i,r,i,h,i,r,i), がじがじ (k,a,s,i,k,a,s,i) がんがん (k,a,N,k,a,N), · · ·
4	*anger*	33	かーっ(k,a,–,Q), かちん (k,a,t,i,N), かつん (k,a,t,u,N), かっ(k,a,Q), かっか (k,a,Q,k,a), がみがみ (k,a,m,i,k,a,m,i), かりかり (k,a,r,i,k,a,r,i), かんかん (k,a,N,k,a,N), · · ·
5	*spook*	31	あわわ (a,w,a,w,a), うぎゃー (u,ky,a,–), がーん (k,a,–,N), ぎく (k,i,k,u) ぎくっ(k,i,k,u,Q), ぎくり (k,i,k,u,r,i), ぎくん (k,i,k,u,N), · · ·
6	*panic*	25	あくせく (a,k,u,s,e,k,u), あたふた (a,t,a,h,u,t,a), あっぷあっぷ (a,Q,h,u,a,Q,h,u), あわあわ (a,w,a,a,w,a)· · ·
7	*bloodless*	27	かくっ(k,a,k,u,Q), がくっ(k,a,k,u,Q), がっかり (k,a,Q,k,a,r,i), がっくり (k,a,Q,k,u,r,i) かくん (k,a,k,u,N), ぎゃふん (ky,a,h,u,N), ぎゅー (ky,u,–), · · ·
8	*deem*	13	うっとり (u,Q,t,o,r,i), きゅーん (ky,u,–,N), きゅん (ky,u,N) つくづく (t,u,k,u,t,u,k,u), · · ·
9	*feel delight*	6	うしうし (u,s,i,u,s,i), きゃぴきゃぴ (ky,a,h,i,ky,a,h,i) うはうは (u,–,h,a,–,u,–,h,a), ほいほい (h,o,i,h,o,i), るんるん (r,u,N,r,u,N), · · ·
10	*balk*	7	いじいじ (i,s,i,i,s,i), うじうじ (u,s,i,u,s,i), おずおず (o,s,u,o,s,u) ぐだぐだ (k,u,t,a,k,u,t,a), もじもじ (m,o,s,i,m,o,s,i), · · ·
Total		273	

marked with bracket denotes phonetic sequences consisting of consonants and vowels.

We retrieved co-occurrences of *ono* words shown in Table 2 using the search engine, Google. We applied Newman clustering to the input words. For comparison, we implemented standard k-means which is often used as a baseline, as it is one of the simplest unsupervised clustering algorithms, and compared the results to those obtained by our method. We used Euclidean distance (L_2 norm) as a distance metric used in the k-means.

For evaluation of classification, we used Precision($Prec$), Recall(Rec), and $F\text{-}measure$ which is a measure that balances precision and recall (Bilenko et al., 2004). The precise definitions of these measures are given below:

$$Prec = \frac{\#PairsCorrectlyPredictedInSamecluster}{\#TotalPairsPredictedInSameCluster} \tag{8}$$

$$Rec = \frac{\#PairsCorrectlyPredictedInSameCluster}{\#TotalPairsInSameCluster} \tag{9}$$

$$F - measure = \frac{2 \times Prec \times Rec}{(Prec + Rec)} \tag{10}$$

4.2 Results

The results are shown in Table 3. "Co-occ. & Sounds" in Data refers to the results obtained by our method. "Co-occ." denotes the results obtained by a single measure, co-occurrence based distributional similarity measure, and "Sounds" shows the results obtained by orthographic similarity. "θ" in Table 3 shows a parameter θ used in the Newman clustering.[3] Table 3 shows best performance of each method against θ values. The best result was obtained when we used $phrase$-based search and a combined measure of co-occurrence(MI) and sounds (cos), and F-score was 0.451.

4.2.1 AND vs $phrase$-based search

Table 3 shows that overall the results using $phrase$-based search were better than those of AND search, and the maximum difference of F-score between them was 20.6% when we used a combined measure. We note that AND boolean search did not consider the position of a word in a document, while our assumption was that semantically similar words appeared in similar contexts. As a result, two *ono* words which were not semantically similar were often retrieved by AND boolean search. For example, consider two antonymous words, "a,h,a,h,a" (grinning broadly) and "w,a,–,N" (Wah, Wah). The co-occurrence frequency obtained by AND was 5,640, while that of $phrase$-based search was only one. The observation shows that we find phrase-based search to be a good choice.

[3] In case of k-means, we used the weights which satisfies network density.

Table 3: Classification results

Data	Algo.	Sim (Co-occ.)	Sim (Sounds)	Search method	θ	Prec	Rec	F	# of clusters
Co-occ. & Sounds	k-means	χ^2	cos	AND	.050	.134	.799	.229	10
				Phrase	.820	.137	.880	.236	10
		MI		AND	.050	.134	.562	.216	10
				Phrase	.150	.190	.618	**.289**	10
		χ^2	αdiv	AND	.680	.134	.801	.229	10
				Phrase	.280	.138	.882	.238	10
		MI		AND	.040	.134	.602	.219	10
				Phrase	.140	.181	.677	.285	10
	Newman	χ^2	cos	AND	.170	.182	.380	.246	9
				Phrase	.100	.322	.288	.304	14
		MI		AND	.050	.217	.282	**.245**	13
				Phrase	.080	.397	.520	**.451**	7
		χ^2	αdiv	AND	.130	.212	.328	.258	9
				Phrase	.090	.414	.298	.347	17
		MI		AND	.090	.207	.325	.253	6
				Phrase	.160	.372	.473	.417	8
Co-occ.	k-means	χ^2	–	AND	.460	.138	.644	.227	10
				Phrase	.110	.136	.870	.236	10
		MI		AND	.040	.134	.599	.219	10
				Phrase	.150	.191	.588	.286	10
	Newman	χ^2	–	AND	.700	.169	.415	.240	8
				Phrase	.190	.301	.273	.286	14
		MI		AND	.590	.159	.537	.245	3
				Phrase	.140	.275	.527	**.361**	5
Sounds	k-means	–	cos	–	.050	.145	.321	.199	10
			αdiv	–	.020	.126	.545	.204	10
	Newman	–	cos	–	.270	.151	.365	**.213**	4
			αdiv	–	.350	.138	.408	.206	3

4.2.2 A single vs combined similarity measure

To examine the effectiveness of the combined similarity measure, we used a single measure as a quality function of the Newman clustering, and compared these results with those obtained by our method. As shown in Table 3, the results with combining similarity measures improved overall performance. In the $phrase$-based search, for example, the F-score using a combined measure "Co-occ(MI) & Sounds(cos)" was 23.8% better than the baseline single measure "Sounds(cos)", and 9.0% better a single measure "Co-occ(MI)".

Figure 1 shows F-score by "Co-occ(MI) & Sounds(cos)" and "Co-occ(MI)" against changes in θ. These curves were obtained by $phrase$-based search. We can see from Figure 1 that the F-score by a combined measure "Co-occ(MI) & Sounds(cos)" was better than "Co-occ(MI)" with θ value ranged from .001 to .25. One possible reason for the difference of F-score between them is the edges selected by varying θ. Figure 2 shows the results obtained by each single measure, and a combined measure to examine how the edges selected by varying θ affect overall performance, F-measure. "Precision" in Figure 2 refers to the ratio of correct *ono* word pairs (edges) divided by the total number of edges. Here, correct *ono* word pairs were created by using the Japanese *ono* dictionary, *i.e.*, we extracted word pairs within the same sense of the dictionary. Surprisingly, there were no significant difference between a combined measure "Co-occ(MI) & Sounds(cos)" and a single measure "Co-occ(MI)" curves, while the precision of a single measure "Sounds" was constantly worse than that obtained by a combined measure. Another possible reason for the difference of F-score is due to product of MI and Cos in Eq. (7). Further work is needed to analyze these results in detail.

4.2.3 k-means vs Newman algorithms

We examined the results obtained by standard k-means and Newman clustering algorithms. As can be seen clearly from Table 3, the results with Newman clustering were better than those of the standard k-means at all search and similarity measures, especially the result obtained by Newman clustering showed a 16.2 % improvement over the k-means when we used Co-occ.(MI) & Sounds(cos) & $phrase$-based search. We recall that we used 273 *ono* words for clustering. However, Newman clustering is applicable for a large number of nodes and edges without decreasing accuracy too much, as it does not simply calculate the number of short-

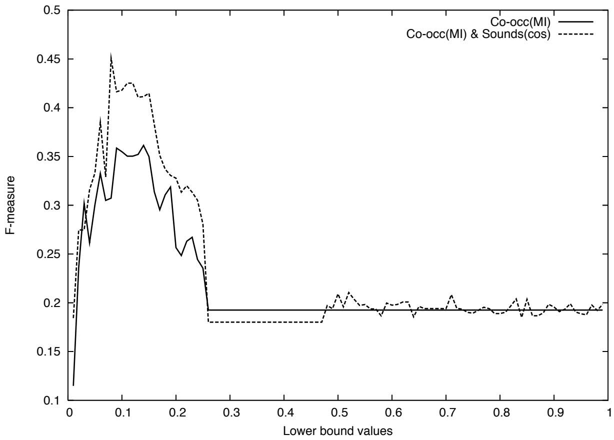

Figure 1: F-score against θ values

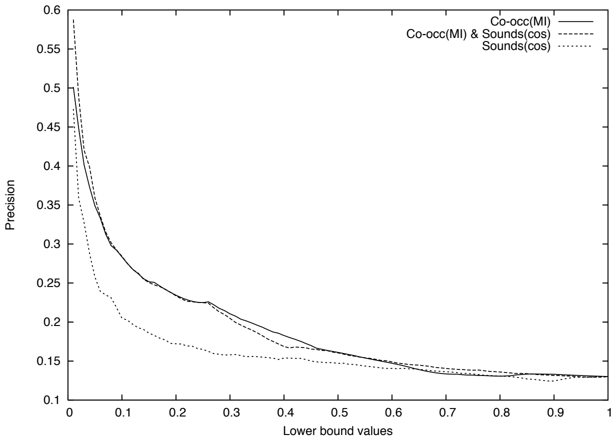

Figure 2: Precision against θ values

est paths between pairs of nodes, but instead calculates a simple quality function. Quantitative evaluation by applying the method to larger data from the Web is worth trying for future work.

4.3 Qualitative Analysis of Errors

Finally, to provide feedback for further development of our classification approach, we performed a qualitative analysis of errors. Consider the following clusters (the Newman output for Co-occ.(MI), Sounds(cos) and $phrase$-based search), where each parenthetic sequences denotes *ono* word:

A1: (t,o,Q) (t,o,Q,t,o) **(t,o,Q,k,i,N,t,o,Q,k,i,N)**
A2: (o,h,o,h,o), (e,h,e,h,e), (h,e,h,e,h,e), **(o,-,o,-)**
A3: (u,s,i,u,s,i), (m,o,s,i,m,o,s,i), **(m,o,s,o,m,o,s,o)**

Three main error types were identified:

1. Morphological idiosyncrasy: This was the most frequent error type, exemplified in **A1**, where "(t,o,Q,k,i,N,t,o,Q,k,i,N)" ($pain$ sense) was incorrectly clustered with other two words ($laugh$ sense) merely because orthographic similarity between them was large, as the phonetics sequences of "(t,o,Q,k,i,N,t,o,Q,k,i,N)" included "t" and "o".
2. Sparse data: Many of the low frequency *ono* words performed poorly. In **A2**, "(o,-,o,-)" (cry sense) was classified with other three words ($laugh$ sense) because it occurred few in our data.
3. Problems of polysemy: In **A3**, "(m,o,s,o,m,o,s,o)" ($pain$ sense) was clustered with other two words ($balk$ sense) of its gold standard class. However, the *ono* word has another sense, $balk$ sense when it co-occurred with action verbs.

5 Conclusion

We have focused on onomatopoetic words, and proposed a method for classifying them into a set of semantically similar words. We used a graph-based clustering algorithm, called Newman clustering with a combined different similarity measures. The results obtained by using the Web data showed a 9.0% improvement over the baseline single distributional similarity measure. There are number of interesting directions for future research.

The distributional similarity measure we used is the basis of the *ono* words, while other content words such as verbs and adverbs are also effective for classifying *ono* words. In the future, we plan to investigate the use of these words and work on improving the accuracy of classification. As shown in Table 2, many of the *ono* words consist of duplicative character sequences such as "h" and "a" of "a,h,a,h,a", and "h" and "i" of "i,h,i,h,i". Moreover, characters which consist of *ono* words within the same class match. For example, the hiragana character "は" (h,a) frequently appears in *laugh* sense class. These observations indicate that integrating edit-distance and our current similarity measure will improve overall performance.

Another interesting direction is a problem of polysemy. It clearly supports the classification of (Ono, 2007) to insist that some *ono* words belong to more than one cluster. For example, "(i,s,o,i,s,o)" has at least two senses, *panic* and *feel delight* sense. In order to accommodate this, we should apply an appropriate soft clustering technique (Tishby et al., 1999; Reichardt and Bornholdt, 2006; Zhang et al., 2007).

Acknowledgments

We would like to thank the anonymous reviewers for their helpful suggestions. This material is supported in part by the Grant-in-aid for the Japan Society for the Promotion of Science(JSPS).

References

Bilenko, M., S. Basu, and R. J. Mooney. 2004. Integrating Constraints and Metric Learning in Semi-Supervised Clustering. In *Proc. of 21st International Conference on Machine Learning*, pages 81–88.

Chen, K. J. and C. J. Chen. 2000. Automatic Semantic Classification for Chinese Unknown Compound Nouns. In *Proc. of 38th Annual Meeting of the Association for Computational Linguistics*, pages 125–130.

Church, K. and P. Hunks. 1990. Word Association Norms, Mutual Information and Lexicography. In *In Proc. of 28th Annual Meeting of the Association for Computational Linguistics.*, pages 76–83.

Dagan, I., L. Lee, and F. Pereira. 1999. Similarity-based Models of Cooccurrence Probabilities. *Machine Learning*, 34(1-3):43–69.

Damerau, F. 1964. A Technique for Computer Detection and Correction of Spelling Errors. *Communications of the ACM*, 7:171–176.

Galley, M. and K. McKeown. 2003. Improving Word Sense Disambiguation in Lexical Chaining. In *Proc. of 19th International Joint Conference on Artificial Intelligence*, pages 1486–1488.

Geffet, M. and I. Dagan. 2004. Feature Vector Quality and Distributional Similarity. In *Proc. of 20th International Conference on Computational Linguistics*, pages 247–253.

Hindle, D. 1990. Noun Classification from Predicate-argument Structures. In *Proc. of 28th Annual Meeting of the Association for Computational Linguistics*, pages 268–275.

Jannink, J. and G. Wiederhold. 1999. Thesaurus Entry Extraction from an On-line Dictionary. In *Proc. of Fusion'99*.

Lee, L. 1999. Measures of Distributional Similarity. In *Proc. of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 25–32.

Lin, D. 1998. Automatic Retrieval and Clustering of Similar Words. In *Proc. of 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 768–773.

Matsuo, Y., T. Sakaki, K. Uchiyama, and M. Ishizuka. 2006. Graph-based Word Clustering using a Web Search Engine. In *Proc. of 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP2006)*, pages 542–550.

McLeod, W. T. 1991. *The COLLINS Dictionary and Thesaurus*. HarperCollinsPublishers.

Mihalcea, R. 2005. Unsupervised Large Vocabulary Word Sense Disambiguation with Graph-based Algorithms for Sequence Data Labeling. In *Proc. of the Human Language Technology / Empirical Methods in Natural Language PRocessing Conference*, pages 411–418.

Muller, P., N. Hathout, and B. Gaume. 2006. Synonym Extraction Using a Semantic Distance on a Dictionary. In *Proc. of the Workshop on TextGraphs*, pages 65–72.

Newman, M. E. J. 2004. Fast algorithm for detecting community structure in networks. *In Physics Review E*, (69, 066133).

Ono, M. 2007. *Nihongo Omomatope Jiten (in Japanese)*. Shougakukan.

Reichardt, J. and S. Bornholdt. 2006. Statistical Mechanics of Community Detection. *PHYICAL REVIEW E*, (74):1–14.

Resnik, P. 1995. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *Proc. of 14th International Joint Conference on Artificial Intelligence*, pages 448–453.

Sinha, R. and R. Mihalcea. 2007. Unsupervised Graph-based Word Sense Disambiguation Using Measures of Word Semantic Similarity. In *Proc. of the IEEE International Conference on Semantic Computing*, pages 46–54.

Tishby, N., F. C. Pereira, and W. Bialek. 1999. The Information Bottleneck Method. In *Proc. of 37th Annual Allerton Conference on Communication Control and Computing*, pages 368–377.

Weeds, J. and D. Weir. 2005. Co-occurrence Retrieval: A Flexible Framework for Lexical Distributional Similarity. *Computational Linguistics*, 31(4):439–476.

Widdows, D. and B. Dorow. 2002. A Graph Model for Unsupervised Lexical Acquisition. In *Proc. of 19th International conference on Computational Linguistics (COLING2002)*, pages 1093–1099.

Zhang, S., R. S. Wang, and X. S. Zhang. 2007. Identification of Overlapping Community Structure in Complex Networks using Fuzzy C-means Clustering. *PHYSICA A*, (374):483–490.