Computational Semantics of Noun Compounds in a Semantic Space Model

Akira Utsumi
Department of Systems Engineering
The University of Electro-Communications
1-5-1, Chofugaoka, Chofushi, Tokyo 182-8585, Japan
utsumi@se.uec.ac.jp

Abstract

This study examines the ability of a semantic space model to represent the meaning of noun compounds such as "information gathering" or "weather forecast". A new algorithm, *comparison*, is proposed for computing compound vectors from constituent word vectors, and compared with other algorithms (i.e., predication and centroid) in terms of accuracy of multiple-choice synonym test and similarity judgment test. The result of both tests is that the comparison algorithm is, on the whole, superior to other algorithms, and in particular achieves the best performance when noun compounds have emergent meanings. Furthermore, the comparison algorithm also works for novel noun compounds that do not occur in the corpus. These findings indicate that a semantic space model in general and the comparison algorithm in particular has sufficient ability to compute the meaning of noun compounds.

1 Introduction

Noun compounds are short phrases consisting of two or more nouns such as "apple pie" and "information gathering". Research on noun compounds is important in any disciplines relevant to language, because they are very common not only in everyday language but also in technical documents [Costello *et al.*, 2006]. Recently, therefore, a number of computational studies have been made on interpretation of noun compounds [Costello *et al.*, 2006; Girju *et al.*, 2005; Kim and Baldwin, 2005; 2007].

According to computational lexical semantics, computing the meaning of noun compounds involves the following two processes:

- *Compound disambiguation*: the process of determining which sense of constituent words is used and identifying the semantic relation holding between (the senses of) constituent words in a noun compound.
- *Similarity computation*: the process of computing the semantic similarity between a noun compound and other words (and compounds), which is used for identifying taxonomic relations (e.g., synonym, hyponym) and associative relations.

These two processes are equally important, but unfortunately, the process of similarity computation has never been studied computationally in the field of NLP and AI; the existing studies on noun compounds such as those mentioned above have addressed only the compound disambiguation process. This problem becomes more serious when we consider a particularly intriguing aspect of noun compounds that they can yield *emergent properties or meanings* [Wilkenfeld and Ward, 2001]. Emergent properties are those that are made salient in the interpretation of a noun compound, but not salient either in the representation of the head noun or the modifier. For example, the sense INTELLIGENCE of "information gathering" is an emergent meaning since INTELLIGENCE is not likely to be listed as characteristic of either the head "gathering" or the modifier "information". Such non-compositional meanings cannot be yielded solely by the compositional process of compound disambiguation; they should be explained within the process of similarity computation. This paper, therefore, aims at proposing and evaluating a method of similarity computation for noun compounds.

For the purpose of similarity computation, this paper employs a semantic space model [Bullinaria and Levy, 2007; Landauer *et al.*, 2007; Padó and Lapata, 2007], in which each word is represented by a high-dimensional vector and the degree of semantic similarity between any two words can be easily computed as, for example, the cosine of the angle formed by their vectors. Semantic space models are computationally efficient as a way of representing meanings of words, because they take much less time and less effort to construct meaning representation and they can provide a more fine-grained similarity measure between words than other representation methods such as thesauri (e.g., WordNet). Semantic space models are also psychologically plausible; a number of studies have shown that vector-based representation achieves remarkably good performance for simulating human verbal behavior such as similarity judgment and semantic priming [Bullinaria and Levy, 2007; Landauer *et al.*, 2007]. Hence they are advantageous for similarity computation of emergent meanings of noun compounds.

The basic question to be answered, then, is how a proper vector representation of a noun compound should be computed in a semantic space model. One possible and simple way of doing this is to treat noun compounds as individual words; vectors for noun compounds are constructed directly from the corpus just as vectors for words are constructed. This method is expected to compute a proper semantic representation of noun compounds, but suffers from one serious limitation; it cannot deal with novel compounds which do not occur in the corpus. This drawback is all the more serious, given the empirical finding that people easily comprehend novel compounds [Gagné, 2000; Wisniewski, 1996].

An alternative way of computing compound vectors is to combine word vectors for constituent words (i.e., the head noun and the modifier) of a noun compound. Some algorithms (i.e., *centroid* or *predication*) have been devised for vector composition, but their semantic validity has never been examined in a systematic way. Hence this paper examines the applicability of these algorithms to noun compounds. Furthermore, this paper proposes a new algorithm for vector composition, i.e., *comparison*, and tests whether the proposed algorithm shows better performance on similarity computation of noun compounds.

2 Algorithm

2.1 Centroid

The standard method for vector composition in semantic space models is to compute the centroid of constituent word vectors. For a noun compound C consisting of the head noun H and the modifier M, the centroid algorithm computes the compound vector $\boldsymbol{v}_{cent}(C)$ as $(\boldsymbol{v}(H)+\boldsymbol{v}(M))/2$. However, the centroid algorithm has a serious drawback that word order is completely ignored; this algorithm wrongly computes the same vector for the different compounds, e.g., "apartment dog" and "dog apartment".

2.2 Predication

The predication algorithm has been proposed by Kintsch [2001] to compute the intuitively plausible and contextually dependent vectors of the proposition with the predicate argument structure. Given that a proposition $P(A)$ consisting of a predicate P (i.e., the modifier M in the case of a noun compound) and an argument A (i.e., the head H), the predication algorithm first chooses m nearest neighbors of a predicate P, i.e., m words with the highest similarity to P. The algorithm then picks up k neighbors of P that are also related to A. Finally the algorithm computes the centroid vector of P, A, and the k neighbors of P as a vector representation of $P(A)$. When the predication algorithm is applied to noun compounds, the set of neighbors of P relevant to A can be seen as representing the intended sense of the modifier M that is appropriate for describing the intended sense of the head noun H.

Formally, the predication algorithm of computing a compound vector $\boldsymbol{v}_{pred}(C)$ is given as follows.

1. Compute $N_m(M)$, which denotes a set of m neighbors of the modifier M.
2. Choose k words in $N_m(M)$ with the highest similarity to the head noun H.
3. Compute a vector $\boldsymbol{v}_{pred}(C)$ as the centroid of $\boldsymbol{v}(H)$, $\boldsymbol{v}(M)$, and k vectors of the words chosen at Step 2.

2.3 Comparison

The predication algorithm does not take fully into account the relevance (or similarity) between the head noun and its intended sense in a noun compound. It is quite likely that a more plausible vector can be computed by using the set of common neighbors of the head and the modifier. For this purpose, I propose a comparison algorithm which chooses k common neighbors and then computes the centroid vector of these neighbors and the head noun. The comparison algorithm can be seen as Gentner's [1983] comparison process consisting of alignment and projection [Utsumi, 2006], and Wisniewski [1996] empirically demonstrated that noun compound comprehension involves such comparison process.

Formally, the comparison algorithm of computing a compound vector $\boldsymbol{v}_{comp}(C)$ is given as follows:

1. Compute k common neighbors $N_i(H) \cap N_i(M)$ of the modifier M and the head H by finding the smallest i that satisfies $|N_i(H) \cap N_i(M)| \geq k$.
2. Compute a compound vector $\boldsymbol{v}_{comp}(C)$ as the centroid of $\boldsymbol{v}(H)$ and k vectors of the words chosen at Step 1.

3 Evaluation Experiment

3.1 Materials

As a corpus from which noun compounds were collected and the semantic space was constructed, one year's worth of Japanese Mainichi newspaper articles was used. This corpus consists of 523,249 paragraphs and 62,712 different words. Furthermore, a Japanese thesaurus "Nihongo Dai-Thesaurus" [Yamaguchi, 2006] was used for automatically identifying the meanings of words and compounds. This thesaurus consists of 1,044 basic categories which are divided into nearly 14,000 semantic categories. In this study, these semantic categories were used for representing word meanings. The thesaurus contains nearly 200,000 words (including compounds) most of which are classified into multiple semantic categories.

Noun compounds used for evaluation were chosen such that they occurred at least 20 times in the corpus and were included in the thesaurus. For each of the chosen compounds, its meanings (i.e., semantic categories that the compound belongs to) were automatically classified as emergent ones when the semantic categories included neither the head nor the modifier. As a result, 1,254 compounds were chosen for evaluation, and 606 out of them were judged to have emergent meanings.

Semantic spaces are generally constructed from large bodies of text by computing a word-context matrix whose (i, j)-th entry represents the distributional characteristics of the i-th word in the j-th linguistic context. The number of columns, i.e., the dimensionality of a semantic space, is often reduced. Several methods have been proposed for computing a word-context matrix and for reducing dimensions [Landauer *et al.*, 2007; Padó and Lapata, 2007]. Among them, latent semantic analysis (LSA) [Landauer and Dumais, 1997; Landauer *et al.*, 2007] is the most popular. LSA computes a word-context matrix based on the frequency of words in a paragraph (as a context), and then reduces the dimensionality of a semantic space by singular value decomposition.

Table 1: Accuracy (%) of the three algorithms on four test sets for familiar noun compounds in the corpus

Test	Centroid	Predication (m, k)	Comparison (k)	Compound Vector
Multiple Choice Synonym (All)	65.92	66.47 (250, 3)	66.26 (7)	**68.49**
Multiple Choice Synonym (Emergent)	66.08	66.27 (4, 1)	**70.32** (3)	68.66
Similarity Judgment (Emergent) *	28.85	49.59 (500, 20)	**53.92** (20)	20.73
Similarity Judgment (Suppressed) *	52.02	60.36 (7, 7)	38.47 (1)	**78.79**
Similarity Judgment (Harmonic Mean)	37.11	**49.23** (50, 20)	40.69 (8)	32.83

Note. An asterisk * indicates that the difference between algorithms is statistically significant ($p < .05$).

In this study, LSA was used for constructing a semantic space. A word-context matrix was constructed for 34,230 single words (excluding all compounds) that occurred ten times or more in the corpus. The dimensionality of the semantic space was reduced to be 300 because a 300-dimensional space usually yields best performance for simulating human behavior, e.g.,[Landauer and Dumais, 1997]. This 300-dimensional space was used in the evaluation experiment.

3.2 Method

In order to evaluate the semantic validity of the computed vectors, this study employed the following two tests for noun compounds: multiple-choice synonym (MCS) test and similarity judgment (SJ) test on discriminative meaning.

Multiple Choice Synonym Test

This test is very similar to the synonym portion of TOEFL, which has been used as a performance measure by many studies on semantic space. Each item of a synonym test consists of a stem word (i.e., a noun compound) and five alternative words (excluding noun compounds) from which the algorithm must choose one with the most similar meaning to the stem word.

In this study, two test sets were constructed. One test set was constructed to evaluate the overall performance of the algorithms to understand noun compounds. On the other hand, another test set was constructed for the specific purpose of testing the ability to represent emergent meanings, and thus it contains only test items for emergent meanings. For each noun compound of both test sets, two test items were automatically constructed in such a way that one correct alternative word was chosen randomly from the semantic categories of the target compound, or from the emergent categories in the case of the test set for emergent meanings, and other four alternatives were chosen randomly from the words that belonged to none of the basic categories of the compound, the head noun or the modifier. All alternative words were chosen such that they occurred more than 50 times in the corpus. As a result, MCS test for all compounds had 2,374 test items and MCS test for emergent compounds had 1,085 test items. For example, the MCS test set included the item consisting of the noun compound "information gathering" as a stem word and five alternative words *intelligence* (correct answer), *technology*, *guard*, *magazine*, and *dis-election*.

The computer's choices were determined by computing cosine similarity between the stem word (i.e., the target compound) and each of the five alternative words and choosing the word with the highest similarity. As a performance measure of the algorithms, the percentage of correct answers (i.e., accuracy) was computed for two test sets.

Similarity Judgment Test on Discriminative Meaning

This test directly examines how properly emergent meanings are represented in the vector of noun compounds. It measures how often emergent meanings are more similar to the noun compound than to the head noun and the modifier. For each of 606 emergent compounds, two words were chosen randomly from the emergent semantic categories such that they were included in none of the basic categories of the head or the modifier, and their frequency was 50 times or more. As a result, 1,085 words were chosen as emergent meanings. The performance of the algorithm was calculated as the percentage of emergent meanings that were more similar to the noun compound than to the head noun and the modifier.

At the same time, another SJ test was designed to explore the degree to which the vector representation of noun compounds suppressed irrelevant meanings of the head noun. For example, the senses MEETING and ABSCESS of "gathering" are irrelevant to the noun compound "information gathering", and thus it is desirable that they are suppressed in understanding the compound. This test was constructed in such a way that, for each of 1,254 noun compounds, at most two words were chosen from the semantic categories that contained the head noun but not the compound. As a result, 2,051 words were chosen for 1,034 compounds. The performance of this test was the percentage of suppressed words that were assessed as less similar to the compound than to the head noun.

3.3 Results

For each of the three algorithms, four tests (two tests for MCS and another two for SJ) were conducted and their accuracy values were calculated. In computing the accuracy of the predication and comparison algorithms, the parameter m was varied between 1 and 20, between 25 and 50 in steps of 5, and between 100 and 500 in steps of 50, and the parameter k was varied between 1 and 20. (Of course, any combinations of m and k such that $m < k$ were removed.) Moreover, for purpose of comparison, another semantic space was constructed in which 1,254 target compounds were added to the original space as individual words, and the performance of such compound vectors was also calculated for four tests.

Table 1 shows the results of four test sets. For predication and comparison algorithms, optimal values are shown with the parameter values in parentheses at which the algorithm achieves the best performance.

For MCS test for all compounds, the predication algorithm yielded the best performance among the three algorithms, but its accuracy was only slightly higher than that of the comparison algorithm. Although these vector composition algorithms did not outperform the compound vector computed directly from the corpus, the difference of accuracy was not so large and not significant. On the other hand, the comparison algorithm showed the highest accuracy on MCS test for emergent compounds and outperformed the original compound vector. These results suggest that *a vector space model, especially the comparison algorithm, is useful for computing the meaning of noun compounds*, and that the resultant vector computed by the composition algorithms can yield better performance than the original vector constructed from the corpus. Note that, perhaps a bit surprisingly, the centroid algorithm worked better than we had expected.

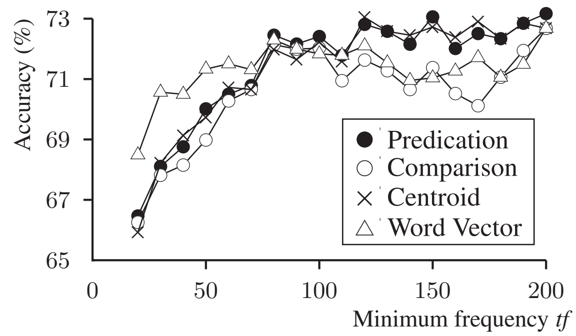
Concerning similarity judgment, as shown in Table 1, the comparison algorithm also achieved the highest accuracy for emergent meanings, but showed the lowest accuracy for suppressed meanings. Taken together with the finding of MCS tests, this result indicates that *the comparison algorithm is best suited to highlight emergent meanings (i.e., to increase the similarity to emergent meanings)*, but at the same time fails to suppress irrelevant meanings of the head noun. On the contrary, the accuracy of the original compound vector was highest in similarity judgment for suppressed meanings, while it was lowest for emergent meanings. When noun compounds are vectorized directly as individual words, these compound vectors appear to suppress emergent meanings as well as irrelevant head meanings.

In addition, since two SJ accuracies seem to trade off against one another, the harmonic mean of these two accuracies is calculated which is shown in the last row of Table 1. Harmonic mean of accuracy shows that the predication algorithm achieved the most balanced performance between activation of emergent meanings and suppression of irrelevant head meanings.

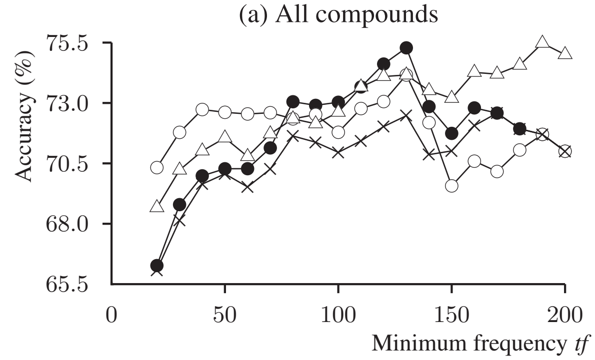
Effect of Frequency of Noun Compounds

In order to examine to what degree the algorithm's performance was affected by the frequency of the compounds in the corpus, I calculated the accuracy when target compounds were limited to those that occur at least *tf* times.

Figure 1 shows the accuracy of two MCS tests for a threshold *tf* ranging from 20 (i.e., "unlimited") to 200 in steps of 10. As expected, accuracy proportionally depended on frequency threshold *tf*; accuracy was higher as test items were limited to more frequent compounds. A more interesting finding was that the predication algorithm outperformed the original compound vector and gave the highest accuracy when the minimum frequency was 80 or above (although the accuracy of the predication algorithm was lower for emergent meanings when $tf \geq 140$). This finding indicates that the predication algorithm can serve to interpret highly frequent (i.e., familiar) compounds. (Note that, although not shown in Figure 1 because of its triviality, the result for SJ tests was that accuracy was also proportional to frequency threshold and the relative performance of the four algorithms did not change over frequency threshold.)



(a) All compounds



(b) Emergent compounds

Figure 1: Accuracy of MCS tests as a function of minimum frequency of compounds

Effect of Which Noun Contributes More to Compounds

I have implicitly assumed that the head noun contributes more to the compound meaning than the modifier. The predication and comparison algorithms are devised or employed mainly for such head-centered compounds. However, in some compounds and probably in many compounds with very abstract head, the modifier plays a more important role in determining the compound meaning than the head noun [Kim and Baldwin, 2005]. For example, according to the Japanese thesaurus, the compound "weather forecast" has many senses related to the modifier "weather", such as EARTH SCIENCE and METEOROLOGICAL PHENOMENON. These "modifier-centered" compounds may be unable to be interpreted properly by the algorithms for head-centered compounds.

In order to test this possibility, I calculated the accuracy of four test sets separately for three types of compounds. The type of a noun compound was determined by assessing which of the head or the modifier shares more semantic categories with the compound in the thesaurus. Noun compounds that shared more semantic categories with the head were judged as head-centered, and those that shared more categories with the modifier were judged as modifier-centered. Otherwise they were judged as equally-weighted compounds.

Table 2 shows the accuracy of three types of compounds. The result of predication and comparison algorithms shows that modifier-centered compounds had lower accuracy than head-centered or equally-weighted compounds, indicating that these algorithms were indeed inadequate for interpreting modifier-centered compounds. (One exception is that the predication algorithm had higher accuracy for modifier-

Table 2: Accuracy (%) of the three algorithms for different types of compounds

Test / Compound Type	Cent	Pred	Comp	Vctr
MCS (All)	65.92	66.47	66.26	**68.49**
Head-Centered	68.87	69.63	**72.09**	69.63
Modifier-Centered	61.02	59.41	55.11	**65.05**
Equally-Weighted	64.55	65.71	64.13	**68.57**
MCS (Emergent)	66.08	66.27	**70.32**	68.66
Head-Centered	60.21	61.78	**64.92**	62.30
Modifier-Centered	57.96	59.24	**61.15**	59.24
Equally-Weighted	69.34	68.93	**73.68**	72.32
SJ (Emergent)*	28.85	49.59	**53.92**	20.74
Head-Centered*	29.32	49.74	**52.88**	18.85
Modifier-Centered*	25.48	50.32	**52.23**	15.29
Equally-Weighted*	29.44	49.39	**54.55**	22.39
SJ (Suppressed)*	52.02	60.36	38.47	**78.79**
Head-Centered*	50.86	60.56	38.78	**76.76**
Modifier-Centered*	49.87	58.40	37.33	**79.47**
Equally-Weighted*	53.85	61.00	38.68	**80.17**

Note. * denotes a significant difference ($p < .05$).

centered compounds in SJ test for emergent meanings.)

One notable finding is that, on MCS test for all meanings, the comparison algorithm achieved the highest accuracy for head-centered compounds. When the comparison algorithm processed modifier-centered compounds by exchanging the head and the modifier, its accuracy increased to 59.95%, and thereby the total accuracy increased to 67.02%. On the other hand, the predication algorithm did not increase the accuracy of modifier-centered compounds even if their word order was reversed. This finding suggests a possibility that the comparison algorithm may be more appropriate for interpreting all noun compounds.

Performance for Novel Noun Compounds

The results that have been presented so far concern *familiar* noun compounds that occur in the corpus from which the semantic space is constructed. In other words, the contextual information of these compounds is implicitly involved in the semantic space. Therefore, it is worth examining the performance of the algorithms for noun compounds that do not occur in the corpus, i.e., that are *novel* for the semantic space. The ability to interpret novel compounds is particularly important for the algorithms not only as a NLP technique but also as a cognitive model, because people can easily interpret novel noun compounds [Wisniewski, 1996].

For this purpose, I collected 413 noun compounds that did not occur in the corpus but were included in the thesaurus. For these novel compounds, four test sets (two MCS tests and two SJ tests) were constructed in the same way as described in Section 3.2. As a result, 753 items were included in MCS test for all 413 compounds, 539 items were in MCS and SJ tests for 287 emergent compounds, and 734 items were in SJ test for 371 compounds with suppressed meanings.

Table 3 lists accuracy of four test sets for novel compounds

Table 3: Accuracy (%) of the three algorithms for novel compounds

Test	Centroid	Predication	Comparison
MCS (All)	**54.45**	54.32	53.52
MCS (Emergent) *	56.03	61.60	**66.05**
SJ (Emergent) *	37.29	61.41	**64.94**
SJ (Suppressed) *	44.82	**61.31**	33.52
SJ (Harmonic)	40.71	**50.11**	40.89

Note. * denotes a significant difference ($p < .05$).

and harmonic mean of accuracy for two SJ tests. MCS's accuracy of novel compounds still remained high and was significantly above the chance level of 20%, although lower than that of "familiar" compounds shown in Table 1, decreasing by 4% to 14%. Moreover in the case of SJ tests, all the three algorithms achieved higher accuracy for novel compounds. These findings demonstrate fine ability of the semantic space model to comprehend novel compounds.

Moreover, relative performance did not change markedly among three algorithms, although the centroid unexpectedly yielded the highest accuracy for MCS for all meanings. The consistency of results suggest that the obtained results are intrinsic to the algorithms for vector composition.

4 Discussion

This study has examined the ability of the semantic space model to compute semantic similarity of a noun compound without considering semantic relations holding between the head and the modifier. However, noun compound comprehension is actually a more complicated process, in which semantic relations may have a large influence on similarity computation. For example, some types of relations may yield more emergent meanings than others, and such difference would affect the performance of similarity computation.

One promising way of utilizing semantic relations is to use words or phrases expressing the semantic relation (e.g., "made of", "cause") for vector composition. For example, the vector representation of "apple pie" can be seen as identical to the vector of its paraphrase "pie made of apples". Such vector can be computed in such a way that a predicate vector of "be made of apples" is computed first from vectors for "be made of" and "apple", and the sentence vector is then computed from vectors of the argument "a pie" and the predicate "be made of apples". (The similar approach is taken by Kintsch [2008] for solving analogy problems.)

To this end, the process of identifying semantic relations should be necessary. As mentioned in the introduction, computational methods for compound disambiguation have been extensively studied [Costello *et al.*, 2006; Kim and Baldwin, 2005; 2007], but a semantic space model can also identify semantic relations [Turney, 2005]. If one knows that the semantic relation "made-of" holds true of noun compounds "apple pie" and "strawberry jam", it is quite reasonable to assume that "orange juice" encodes the same semantic relation since these head nouns can be classified into the same semantic category `foods` and the modifiers can be classified as `fruits`.

Similarity judgment by a semantic space model can be used to classify each word into an appropriate semantic category.

The same supervised technique may also be applicable to classifying noun compounds into head-centered, modifier-centered, or equally-weighted ones. Such automatic classification can lead to a sophisticated method in which different algorithms are used for different types of compounds.

5 Conclusion

Through the evaluation experiment, this paper has shown the validity of a semantic space model for similarity computation of noun compounds. The findings are summarized as follows:

- The comparison algorithm, on the whole, achieved the best performance among the three algorithms. Its performance did not differ from, or in some cases was superior to, the performance of the original compound vector constructed directly from the corpus. In particular, the computed vectors were appropriate for representing emergent meanings.
- The predication algorithm was not, on the whole, superior to the comparison algorithm, but showed a well-balanced performance. It also outperformed the original compound vector and yielded the best performance when compounds were highly frequent.
- The centroid algorithm showed unexpectedly good performance despite its simplicity. This result may be largely due to the symmetric nature of the algorithm.
- These findings apply to novel compounds, indicating that the algorithms work for novel compounds, as well as for familiar compounds.

It would be interesting and vital for further work to develop a method for similarity computation which utilizes semantic relations holding between the head and the modifier, as well as a more efficient method for vector composition. Additionally, I am trying to extend this work to longer noun compounds consisting of three or more nouns.

Acknowledgments

This research was supported in part by Grant-in-Aid for Scientific Research(C) (No.20500234) from Japan Society for the Promotion of Science.

References

[Bullinaria and Levy, 2007] John A. Bullinaria and Joseph P. Levy. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39(3):510–526, 2007.

[Costello *et al.*, 2006] Fintan J. Costello, Tony Veale, and Simon Dunne. Using WordNet to automatically deduce relations between words in noun-noun compounds. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 160–167, 2006.

[Gagné, 2000] Christina L. Gagné. Relation-based combinations versus property-based combinations: A test of the CARIN theory and the dual-process theory of conceptual combination. *Journal of Memory and Language*, 42:365–389, 2000.

[Gentner, 1983] Dedre Gentner. Structure mapping: A theoretical framework for analogy. *Cognitive Science*, 7:155–170, 1983.

[Girju *et al.*, 2005] Roxana Girju, Dan Moldovan, Marta Tatu, and Daniel Antohe. On the semantics of noun compounds. *Computer Speech and Language*, 19:479–496, 2005.

[Kim and Baldwin, 2005] Su Nam Kim and Timothy Baldwin. Automatic interpretation of noun compounds using WordNet similarity. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP2005)*, pages 945–956, 2005.

[Kim and Baldwin, 2007] Su Nam Kim and Timothy Baldwin. Disambiguating noun compounds. In *Proceedings of the 22nd Conference on Artificial Intelligence (AAAI-07)*, pages 901–906, 2007.

[Kintsch, 2001] Walter Kintsch. Predication. *Cognitive Science*, 25(2):173–202, 2001.

[Kintsch, 2008] Walter Kintsch. How the mind computes the meaning of metaphor: A simulation based on LSA. In R.W. Gibbs, editor, *The Cambridge Handbook of Metaphor and Thought*, pages 129–142. Cambridge University Press, 2008.

[Landauer and Dumais, 1997] Thomas K. Landauer and Susan T. Dumais. A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 104:211–240, 1997.

[Landauer *et al.*, 2007] Thomas K. Landauer, Danielle S. McNamara, Simon Dennis, and Walter Kintsch. *Handbook of Latent Semantic Analysis*. Lawrence Erlbaum Associates, 2007.

[Padó and Lapata, 2007] Sebastian Padó and Mirella Lapata. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199, 2007.

[Turney, 2005] Peter D. Turney. Measuring semantic similarity by latent relational analysis. In *Proceedings of the 19th International Joint Conferences on Artificial Intelligence (IJCAI-05)*, pages 1136–1141, 2005.

[Utsumi, 2006] Akira Utsumi. Computational exploration of metaphor comprehension processes. In *Proceedings of the 28th Annual Meeting of the Cognitive Science Society (CogSci2006)*, pages 2281–2286, 2006.

[Wilkenfeld and Ward, 2001] Merry J. Wilkenfeld and Thomas B. Ward. Similarity and emergence in conceptual combination. *Journal of Memory and Language*, 45(1):21–38, 2001.

[Wisniewski, 1996] Edward J. Wisniewski. Construal and similarity in conceptual combination. *Journal of Memory and Language*, 35:434–453, 1996.

[Yamaguchi, 2006] Tasuku Yamaguchi. *Nihongo Dai-Thesaurus CD-ROM*. Taishukan Shoten, Tokyo, 2006.