Natural Language Engineering **20** (2): 185–234. © Cambridge University Press 2013
doi:10.1017/S135132491200037X

A semantic space approach to the computational semantics of noun compounds

AKIRA UTSUMI

Department of Informatics, The University of Electro-Communications,
1-5-1 Chofugaoka, Chofushi, Tokyo 182-8585, Japan
e-mail: utsumi@inf.uec.ac.jp

(*Received 20 January 2011; revised 7 October 2012; accepted 3 December 2012; first published online 15 January 2013*)

Abstract

This study examines the ability of a semantic space model to represent the meaning of noun compounds such as 'information gathering' or 'heart disease.' For a semantic space model to compute the meaning and the attributional similarity (or semantic relatedness) for unfamiliar noun compounds that do not occur in a corpus, the vector for a noun compound must be computed from the vectors of its constituent words using vector composition algorithms. Six composition algorithms (i.e., centroid, multiplication, circular convolution, predication, comparison, and dilation) are compared in terms of the quality of the computation of the attributional similarity for English and Japanese noun compounds. To evaluate the performance of the computation of the similarity, this study uses three tasks (i.e., related word ranking, similarity correlation, and semantic classification), and two types of semantic spaces (i.e., latent semantic analysis-based and positive pointwise mutual information-based spaces). The result of these tasks is that the dilation algorithm is generally most effective in computing the similarity of noun compounds, while the multiplication algorithm is best suited specifically for the positive pointwise mutual information-based space. In addition, the comparison algorithm works better for unfamiliar noun compounds that do not occur in the corpus. These findings indicate that in general a semantic space model, and in particular the dilation, multiplication, and comparison algorithms have sufficient ability to compute the attributional similarity for noun compounds.

1 Introduction

Noun compounds are short phrases comprising two or more nouns such as 'apple pie' and 'information gathering.' Noun compounds are very common and ubiquitous in everyday language and technical documents. Baldwin and Tanaka (2004) reported that about 3 per cent of all word tokens are used as part of noun compounds in English and Japanese corpora. Research on noun compounds is important in any discipline relevant to language. In particular, computer understanding of noun compounds is important for many natural language processing (henceforth NLP) applications, such as question answering, machine translation, and information retrieval (Butnariu *et al.* 2010). For example, in order to improve the retrieval performance for the query 'heart disease,' an IR system would benefit from a semantic interpretation engine that determines what words are semantically related

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

to *heart disease*. Recently, a number of computational studies have been made on the interpretation of noun compounds (e.g., Girju *et al.* 2005; Kim and Baldwin 2005; Costello, Veale and Dunne 2006; Kim and Baldwin 2007; Butnariu and Veale 2008; Nakov 2008; Tratz and Hovy 2010).

According to computational lexical semantics (Jurafsky and Martin 2008), at least two approaches are involved in computing with word meanings: i.e., *word sense disambiguation* and *word similarity computation*. In the case of noun compounds, they are specified as follows:

- *Compound disambiguation* determines which sense of constituent words is used, and identifies the semantic relation holding between (the senses of) constituent words in a noun compound. For example, we must identify the semantic relation holding between the words (i.e., apple and pie) of the noun compound 'apple pie.'
- *Similarity computation* computes the semantic relatedness or similarity between a noun compound and other words (or compounds), which is used for identifying taxonomic (e.g., synonym and hyponym) and associative relations. For example, we compute the degree of semantic relatedness between a noun compound 'apple pie' and the word 'tart.'

Compound disambiguation has been a main topic of research in existing studies on noun compounds, while similarity computation has received only marginal attention. Some studies (e.g., Kim and Baldwin 2005; Ó Séaghdha and Copestake 2007; Nakov and Hearst 2008; Ó Séaghdha and Copestake 2009) have employed similarity computation to identify the semantic relation between the constituent words in a noun compound. However, the similarity computed in these studies differs from the similarity expected by the similarity computation involved in computational lexical semantics. This difference of similarity is clearly described by Turney (2006). According to Turney (2006), there are at least two kinds of similarities: *attributional similarity* [1] and *relational similarity*. Attributional similarity is the correspondence between attributes and its degree is computed between two words. In the case of complex nominals including noun compounds, attributional similarity is computed by treating complex nominals as single words. Relational similarity is the correspondence between relations and its degree is computed between two *word pairs*. For example, the compound 'chocolate milk' has a higher attributional similarity to the compound 'morning coffee' than to the compound 'plastic bag,' because both milk and coffee are beverages and thus share many attributes. On the other hand, 'chocolate milk' has a higher relational similarity to the compound 'plastic bag' than to the compound 'morning coffee,' because the relation between *chocolate* and *milk* and the relation between *plastic* and *bag* are highly similar or identical, i.e., N_1 is a material of N_2. Existing studies on noun compounds have addressed a method of computing relational similarity to identify semantic relations, but have paid little attention to attributional similarity.

[1] Note that, as Turney (2006) mentions, attributional similarity is the same concept as semantic relatedness.

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

The problem of lack of interest in attributional similarity becomes more serious when we consider a particularly intriguing aspect of noun compounds that they can yield *emergent properties* or *emergent meanings*. In the psychological literature (e.g., Hampton 1987; Murphy 1988; Wilkenfeld and Ward 2001), emergent properties are defined as those that are salient in interpreting a noun compound, but not salient in representing its constituent words (i.e., either the head noun or the modifier). For example, people often think that 'pet birds' have the property *talks*, but they do not think that pets and birds by themselves have this property. Hence, the property *talks* is an emergent property of 'pet birds.' On the other hand, 'pet birds' also have the property *sings*, but this property is not emergent because it is also a salient feature of the head noun 'birds.' Similarly, the sense *grid* of the compound 'power system' is an emergent meaning because *grid* is not likely to be listed as a characteristic of either the head 'system' or the modifier 'power.' Emergent properties or meanings have been examined extensively and observed repeatedly in the psychological studies of conceptual combinations (i.e., noun compounding) (Hampton 1987; Murphy 1988; Coulson 2001; Wilkenfeld and Ward 2001) and metaphors (Becker 1997; Utsumi 2005). These findings indicate that noun compounding is not a fully compositional process and involves some sort of non-compositionality. Such 'semi-compositional' meanings cannot be computed solely by the compositional process of compound disambiguation; they should be explained within the process of attributional similarity computation. Thus, this paper aims at proposing and evaluating a method of attributional similarity computation for noun compounds. (In the rest of this paper, we use the generic term 'similarity' to specifically refer to attributional similarity.)

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; Turney and Pantel 2010), in which each word is represented by a high-dimensional vector and the degree of semantic relatedness between any two words can be easily computed, for example, as the cosine of the angle formed by their vectors. Semantic space models are computationally efficient in 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 compared to other representation methods such as thesauri (e.g., WordNet). Semantic space models are also plausible as a psychological (or cognitive) model of mental lexicon or semantic memory; a number of studies have shown that the 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 computing the similarity of emergent meanings of noun compounds.

Therefore, the basic question to be answered 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:

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

it cannot deal with unfamiliar noun compounds that do not occur in the corpus. This drawback is all the more serious considering the corpus-based finding that many noun compounds occur only a few times (approximately 45–60 per cent occur only once) (Baldwin and Tanaka 2004) and the empirical finding that people easily comprehend unfamiliar compounds (e.g., Wisniewski 1996; Gagné 2000).

An alternative way of computing compound vectors is to combine word vectors for constituent words (i.e., the head noun and modifier) of a noun compound. Several algorithms (i.e., *centroid*, *predication*, *comparison*, and *dilation*) have been devised for vector composition (Kintsch 2001; Mitchell and Lapata 2008, 2010; Utsumi 2011), but their semantic validity has not been examined in a systematic way. Hence, this paper examines the applicability of these algorithms to noun compounds, by computing the similarity of noun compounds in English and Japanese and comparing their performances.

The rest of this paper is organized as follows. Section 2 reviews some previous studies relevant to this paper, focusing on noun compound interpretation and compositional semantics of semantic spaces. Section 3 explains two semantic space models used as a computational framework for computing the similarity of noun compounds. In addition, several algorithms for vector composition are introduced, which in Section 4 are compared in terms of the quality of noun compound vectors. Section 4 presents the method and results of the computer experiment conducted for evaluating the ability of the vector composition algorithms to compute the attributional similarity of noun compounds. In the experiment, three tasks (and measures) are used for evaluation: ranking of words semantically related to noun compounds, correlation between human similarity judgment and similarity computation, and semantic classification of compound-word pairs. Section 5 discusses some implications of the simulation results, as well as some issues for future studies. Finally, Section 6 provides concluding remarks for this study.

2 Related work

2.1 Noun compound interpretation

As mentioned in Section 1, noun compound interpretation involves two approaches, i.e., compound disambiguation and similarity computation.

Compound disambiguation refers to the task of determining the semantic relation holding between constituent words of a noun compound. For example, at the very least, possible semantic relations for the noun compound 'headache pill' are *cause* (i.e., 'pill that causes headaches'), and *purpose (prevent)* (i.e., 'pill that prevents headaches'). An algorithm that interprets noun compounds must determine possible relations for a given noun compound, and also rank these relations in order of plausibility.

Several previous studies (e.g., Rosario and Hearst 2001; Girju et al. 2005; Kim and Baldwin 2006, 2007) addressed the problem of compound disambiguation using a predefined set of semantic relations, and consequently they treated it as a multi-class classification problem, for which a number of efficient methods are available. For example, Nastase and Szpakowicz (2003) proposed a hierarchical

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

set of 30 relations grouped into five categories, while Girju et al. (2005) defined a flat list of 35 general semantic relations, only 21 of which turned out to be found in noun compounds. Recently, Tratz and Hovy (2010) presented a more fine-grained taxonomy of 43 semantic relations. Task 4 of SemEval-2007 (i.e., the fourth international workshop on semantic evaluations), aiming at testing the methods for classifying semantic relations between nominals (including noun compounds), includes seven relations (Girju *et al.* 2007, 2009). A wide variety of computational methods have been proposed for identifying appropriate semantic relations for a given noun compound: semantic scattering (Moldovan *et al.* 2004; Girju *et al.* 2010), word sense disambiguation (Kim and Baldwin 2007), the Web-based n-gram model (Lapata and Keller 2005), support vector machines (Girju et al. 2005), a maximum entropy classifier (Tratz and Hovy 2010), associating sense pairs and semantic relations (Rosario and Hearst 2001; Costello, Veale and Dunne 2006), verb semantics (Kim and Baldwin 2006), among others.

An alternative approach to compound disambiguation regards compound disambiguation as a process of paraphrasing according to which noun compounds are paraphrased by combinations of verbs and prepositions (Kim and Baldwin 2006; Nakov and Hearst 2006; Butnariu and Veale 2008; Nakov 2008; Butnariu et al. 2010; Kim and Nakov 2011). For example, the noun compounds 'apple pie' and 'apple store' can be paraphrased as 'a pie that is made of apples' and 'a store that sells Apple products.' This approach does not require a predefined set of semantic relations, which are sometimes criticized for being arbitrary and not able to provide a complete list. In addition, Nakov (2008) and Butnariu and Veale (2008) have demonstrated that Web mining methods are effective in collecting paraphrasing information. Due to these advantages, the paraphrase-based approach has recently become a dominant approach in the field of noun compound interpretation. This is reflected by the fact that noun compound paraphrasing was included as part of a set of tasks run in the recent workshop on semantic evaluation, SemEval-2010 (Butnariu et al. 2010).

Concerning similarity computation, some studies on compound disambiguation have employed relational similarity to determine whether two noun compounds are based on the same semantic relation (e.g., Kim and Baldwin 2005; Ó Séaghdha and Copestake 2007; Nakov and Hearst 2008; Ó Séaghdha and Copestake 2009). A number of methods for computing relational similarity have been devised: for example, the use of WordNet similarity between constituent words of noun compounds (Kim and Baldwin 2005) or between verbs that express semantic relations (Kim and Baldwin 2006), Dice coefficients of Web-based feature vectors based on context verbs and prepositions of noun compounds (Nakov and Hearst 2008), and a hybrid method that combines attributional similarity between constituent words and relational similarity between noun compounds (Ó Séaghdha and Copestake 2007, 2009; Nakov and Kozareva 2011).

However, these studies have not addressed a method for computing attributional similarity for noun compounds,[2] and thus, to the best of the author's knowledge,

[2] Some of these studies (e.g., Ó Séaghdha and Copestake 2008, 2009) used attributional features or similarity (e.g., lexical similarity between constituent words of noun compounds)

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

there are no studies investigating how attributional similarity of noun compounds can be computed. The lack of research on this topic may be due to the seemingly appropriate assumption that the computation of the attributional similarity of noun compounds does not require techniques or methods specific to noun compounds; a general method of similarity computation (i.e., word similarity) can be applied without modification to noun compounds by treating them as single words. However, as mentioned in Section 1, this assumption is incorrect, because noun compounding is a highly productive process, and thus an infinite number of unfamiliar compounds can be generated (e.g., Clark, Gelman and Lane 1985; Baldwin and Tanaka 2004; Ó Séaghdha and Copestake 2007). Furthermore, noun compounds often have semi-compositional meanings, which may not be possible to derive from a simple combination of meanings of the constituent nouns. Therefore, automatic interpretation of noun compounds requires a method for computing the attributional similarity of noun compounds, or complex nominals in general, that can deal with the emergence of meanings. This is what this study aims to accomplish.

Note that most of the previous studies on noun compounds (including this study) deal only with two-noun compounds (i.e., noun compounds comprising two nouns), but some noun compounds include longer noun sequences that comprise more than two nouns (e.g., 'information retrieval system' and 'world health organization'). Interpretation of these longer noun compounds requires the additional process of structural disambiguation, which is referred to as *bracketing* (Lauer 1994, 1995; Girju et al. 2005). This process determines which part of the compound is the head and which the modifier. For example, the compound 'information retrieval system' is left-bracketed as '(information retrieval) system,' which implies that 'system' is the head and 'information retrieval' is the modifier. Conversely, 'world health organization' is right-bracketed as 'world (health organization),' which implies that 'health organization' is the head and 'world' is the modifier. The bracketing problem is tackled in some studies (Lauer 1994, 1995; Girju et al. 2005; Lapata and Keller 2005) by a corpus-based model. Lauer (1994, 1995) demonstrated that corpus statistics for word adjacency and dependency are effective in bracketing noun compounds. Nakov and Hearst (2005) improved the corpus-based n-gram method by using search engine statistics, and Bergsma, Pitler and Lin (2010) examined the advantages of a Web-scale n-gram model.

2.2 *Semantic interpretation using semantic spaces*

In order to compute the similarity for larger linguistic units (e.g., typically phrases, sentences, or paragraphs) than those represented in semantic spaces (e.g., typically words), it is necessary to devise a method for generating a vector representation of a larger unit from its constituent words.

for the task of computing the relational similarity between noun compounds, and demonstrated that attributional features are useful for the task. Note, however, that this does not imply that they proposed or addressed *the problem of attributional similarity between noun compounds*.

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

The standard method for vector composition in semantic space models is to compute the centroid of the constituent word vectors. The centroid algorithm has been shown to yield very useful results in a number of applications of latent semantic analysis (e.g., Landauer and Dumais 1997; Foltz, Kintsch and Landauer 1998). However, it has the serious drawback that word order (and thus, a semantic role) is completely ignored, and thus the algorithm wrongly computes the same vector for a noun compound and its reversed compound (e.g., 'apartment dog' and 'dog apartment').

Kintsch (2000, 2001) proposed a predication algorithm for computing intuitively plausible and contextually dependent vectors of the proposition (i.e., the sentence) with the predicate argument structure. Kintsch demonstrated that the predication algorithm is effective for metaphor comprehension, semantic disambiguation, and other applications. Utsumi (2011) also proposed a comparison algorithm for computing the meaning of metaphors as a model of Gentner's (1983) comparison process comprising structural alignment and inference projection. However, it is not clear whether the predication and comparison algorithms are also effective in interpreting noun compounds, especially in processing the emergence of meanings.

Mitchell and Lapata (2010) compared several composition algorithms with respect to the correlation between two similarity judgments for the same pairs of phrases, namely human judgment and vector-based judgment. The phrases used in their study included noun compounds such as 'development plan' and 'oil industry.' Thus, their study can be regarded as an application of semantic space models to noun compounds, and their findings can be compared with the results of this study for the sake of discussion. However, their study was not intended to propose a method for noun compound interpretation, and thus, they did not directly examine the meanings of noun compounds including emergent meanings. They only employed the similarity between noun compounds (e.g., 'oil industry' and 'computer company') for evaluating composition algorithms. In addition, they did not test the comparison algorithm, which may be effective in interpreting noun compounds.

Although not addressing noun compounds specifically, a number of studies on multiword expressions (i.e., short phrases comprising two or more words such as 'kick the bucket,' which subsume noun compounds) have employed a semantic-space-based similarity computation to identify non-compositional multiword expressions (Schone and Jurafsky 2001; Baldwin *et al.* 2003; Katz and Giesbrecht 2006; Giesbrecht 2009; Guevara 2011; Reddy, McCarthy and Manandhar 2011). However, these studies assume that a correct vector for multiword expressions is constructed directly from the corpus by treating multiword expressions as individual words. This assumption implies that these studies cannot deal with unfamiliar compounds that do not occur in the corpus, which as mentioned in Section 1, is a serious problem when processing noun compounds. In addition, some of these studies also use vector composition – e.g., simple addition (Schone and Jurafsky 2001; Katz and Giesbrecht 2006; Guevara 2011; Reddy, McCarthy and Manandhar 2011), multiplication (Giesbrecht 2009; Guevara 2011; Reddy et al. 2011), and convolution (Giesbrecht 2009; Guevara 2011) – to compute the vector for multiword expressions, but they do not regard vector composition as a method for generating a plausible

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

vector representation for multiword expressions. They simply employed the combined vectors to measure decomposability (i.e., the degree of semantic similarity between a multiword expression and its constituent words). Hence, they did not examine (or intend to examine) the plausibility of the combined vectors as the meaning representation of multiword expressions.

3 Computational model

3.1 Semantic space model

The semantic space model is based on two main assumptions. One assumption is that the meaning of each word w_i can be represented by a high-dimensional vector $\boldsymbol{u}(w_i) = (u_{i1}, u_{i2}, \ldots, u_{in})$, that is, *a word vector*. These n real-valued components define the lexical meaning of the word. The second assumption is that the degree of semantic relatedness $sim(w_i, w_j)$ between any two words w_i and w_j can be computed using a similarity function of their word vectors. This study uses as the similarity function the cosine of the angle formed by their vectors $\boldsymbol{u}(w_i)$ and $\boldsymbol{u}(w_j)$, which is the most widely used in semantic space models.

Semantic spaces (or word vectors) are constructed from large bodies of text (i.e., corpus) by observing distributional statistics of word occurrence. The method for constructing semantic spaces generally comprises the following three steps.

(1) Initial matrix construction: n_w content words in a given corpus are represented as r-dimensional initial vectors whose elements are frequencies (i.e., the number of times a certain event relevant to the words occurs), and an n_w by r matrix $\boldsymbol{A}$ is constructed using n_w word vectors as rows.
(2) Weighting: The elements of the matrix $\boldsymbol{A}$ are weighted.
(3) Smoothing: The dimension of the row vectors of $\boldsymbol{A}$ is reduced from the initial dimension r to n.

As a result, an n-dimensional semantic space including n_w words is generated. The last step of smoothing is optional: When the matrix is not smoothed, $n = r$ (i.e., the initial dimension r becomes equal to the dimension n of the semantic space).

Numerous methods have been proposed for these steps of constructing semantic spaces (for an overview, see Padó and Lapata 2007; Turney and Pantel 2010). Among them, latent semantic analysis (henceforth LSA; Landauer and Dumais 1997; Landauer et al. 2007) is relatively popular, and is also adopted in this study to construct semantic spaces. LSA uses the frequency of words in a document to compute an initial matrix (Step 1), whose dimension r is equal to the number of documents. For Step 2, LSA often uses the tf-idf family of weighting schemes. In the tf-idf family, the weight is calculated by the product of the local weight based on the term frequency and the global weight based on the inverse document frequency. Although many functions have been proposed for calculating local and global weights (for an empirical comparison of different weighting functions, see Nakov, Popova and Mateev 2001), in this paper we use the following function, i.e., the product of the logarithm of the word frequency and the entropy (Quesada

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

2007):

$$u_{ij} = \log(tf_{ij} + 1) * \left(1 + \frac{\sum_{k=1}^{r} P_{ik} \log P_{ik}}{\log r}\right) \quad (1)$$

$$P_{ij} = \frac{tf_{ij}}{\sum_{k=1}^{r} tf_{ik}} \quad (2)$$

where u_{ij} is the weight for the jth element of the word vector for the ith word w_i, tf_{ij} is the frequency of word w_i in the jth document, and r is the number of documents. Finally, in Step 3, LSA reduces the number of dimensions using singular value decomposition (henceforth SVD).

Another family of methods, which relies on word co-occurrence frequency, has also been widely used for constructing semantic spaces (Burgess and Lund 1997; Schütze 1998; Bullinaria and Levy 2007; Recchia and Jones 2009). The most important difference from LSA is that a word–word matrix is constructed as an initial matrix at Step 1, whose elements are frequencies of word co-occurrences within a 'window' spanning any number of words. If we assume that the context is given by the set of words in the window, a word–word matrix can be regarded as a word–context matrix. The weighting scheme for a word–word matrix that is widely used and generally achieves good performance is pointwise mutual information (henceforth PMI; Church and Hanks 1990; Bullinaria and Levy 2007), which is defined as

$$pmi(w_i, w_j) = \log_2 \frac{p(w_i, w_j)}{p(w_i)p(w_j)} \quad (3)$$

$$p(w_i, w_j) = \frac{cf_{ij}}{\sum_{i=1}^{n_w} \sum_{j=1}^{n_c} cf_{ij}} \quad (4)$$

$$p(w_i) = \frac{\sum_{j=1}^{n_c} cf_{ij}}{\sum_{i=1}^{n_w} \sum_{j=1}^{n_c} cf_{ij}} \quad (5)$$

$$p(w_j) = \frac{\sum_{i=1}^{n_w} cf_{ij}}{\sum_{i=1}^{n_w} \sum_{j=1}^{n_c} cf_{ij}} \quad (6)$$

where cf_{ij} is the number of times two words w_i and w_j co-occur in a context, and n_c is the number of context words. The PMI between two words w_i and w_j, denoted by $pmi(w_i, w_j)$, expresses the amount of information we can gain about the occurrence of a word w_i, if it is known that a context includes the word w_j. Hence, positive PMI values indicate that two words are likely to co-occur and thus they are semantically related. PMI values close to zero indicate that they are independent and thus unrelated. On the other hand, negative PMI values theoretically imply that one word tends to occur where the other does not, suggesting that they are somewhat (and inversely) related. In fact, however, this is not the case; if two words co-occur less than expected by chance in a corpus, it is simply because of data sparseness, not because of the semantically inverse relation. Therefore, positive PMI (henceforth PPMI), a variation of PMI, is proposed in which negative PMI values are replaced with zero. Bullinaria and Levy (2007) empirically demonstrated that for a word–word matrix PPMI performs better than PMI and other weighting methods.

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

Hence, in this paper we use PPMI, which is formally defined as

$$u_{ij} = \begin{cases} pmi(w_i, w_j) & (\text{if } pmi(w_i, w_j) > 0) \\ 0 & (\text{otherwise}). \end{cases} \tag{7}$$

A popular smoothing technique for PMI or a word–word matrix is to reduce the dimension by choosing the context words using a certain criterion (e.g., high similarity to more words and highly weighted words). However, in this paper we do not smooth the matrix, given that some studies (e.g., Bullinaria and Levy 2007; Recchia and Jones 2009) demonstrated that PMI without smoothing performs better than LSA.

In summary, we use two methods for constructing semantic spaces, namely, LSA and PPMI. The reason for comparing two kinds of semantic spaces in this paper is that we would like to examine whether and how vector composition algorithms depend on the characteristics of semantic spaces.

3.2 Computing the vector for noun compounds: Algorithms for vector composition

The vector for a noun compound can be computed by using vector composition algorithms. For a noun compound C comprising two nouns, i.e., the head noun H and the modifier noun M, the vector $\boldsymbol{p} = (p_1, \ldots, p_n)$ for the noun compound C is computed by

$$\boldsymbol{p} = f(\boldsymbol{h}, \boldsymbol{m}), \tag{8}$$

where f is a vector composition function (i.e., algorithm) that takes two constituent vectors $\boldsymbol{h}$ and $\boldsymbol{m}$ as variables. The variable $\boldsymbol{h} = (h_1, \ldots, h_n)$ is a vector for the head noun H, and $\boldsymbol{m} = (m_1, \ldots, m_n)$ is a vector for the modifier noun M. In Section 4, several composition functions or algorithms are compared in terms of the semantic appropriateness of the compound vector $\boldsymbol{p}$.

In this study, as well as in some other studies (e.g., Baldwin and Tanaka 2004; Kim and Baldwin 2005), the right noun of a compound (e.g., *pie* of 'apple pie') is referred to as the head noun, and the left noun (e.g., *apple* of 'apple pie') is referred to as the modifier noun. This definition of head is based on the syntactic concept of headedness (i.e., the syntactic head of a compound is almost always the final constituent of that compound). In most cases, it coincides with the semantic definition of headedness that the head is the word that determines the semantic category of the compound.[3]

In the rest of this section, six composition algorithms that are compared in an evaluation experiment are presented with an example of computation using the

[3] Some compounds (i.e., dvandva) such as 'player coach' seem to be semantically headless (or have multiple heads) because two nouns can be connected by the coordinating conjunction 'and' (Moldovan et al. 2004). On the other hand, these compounds can also be treated in syntactically the same way as other compounds whose right noun is the semantic head; 'coach' is the head of the compound 'player coach,' and conversely, 'player' is the head of the compound 'coach player.' This study, as well as almost all other studies, adopts this view.

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

following hypothetical vectors.

$$\boldsymbol{h} = (0,\ 10,\ 4,\ 1,\ 6), \qquad \boldsymbol{m} = (1,\ 8,\ 4,\ 4,\ 0). \tag{9}$$

3.2.1 Centroid

The simplest but most widely used method for vector composition in semantic space models is to compute the centroid of constituent word vectors by

$$\boldsymbol{p} = \frac{\boldsymbol{h} + \boldsymbol{m}}{2}. \tag{10}$$

For example, the centroid of $\boldsymbol{h}$ and $\boldsymbol{m}$ shown in(9) is $\boldsymbol{p} = ((0, 10, 4, 1, 6) + (1, 8, 4, 4, 0))/2 = (0.5,\ 9.0,\ 4.0,\ 2.5,\ 3.0)$. Note that, when the cosine is used as a similarity measure, this algorithm always yields the same similarity computation result as the simple addition (i.e., $\boldsymbol{p} = \boldsymbol{h} + \boldsymbol{m}$), because the cosine does not depend on the length of the vectors.

3.2.2 Multiplication

An alternative and simpler method for vector composition is to use the multiplicative function (11), in which each component of one vector is multiplied by the corresponding component of another vector (Mitchell and Lapata 2010):

$$\boldsymbol{p} = \boldsymbol{h} \odot \boldsymbol{m} \tag{11}$$

where the symbol $\odot$ denotes component-wise multiplication defined as

$$p_i = h_i \cdot m_i. \tag{12}$$

For example, the multiplication of the two example vectors $\boldsymbol{h}$ and $\boldsymbol{m}$ in(9) is $\boldsymbol{p} = (0, 10, 4, 1, 6) \odot (1, 8, 4, 4, 0) = (0 \cdot 1, 10 \cdot 8, 4 \cdot 4, 1 \cdot 4, 6 \cdot 0) = (0, 80, 16, 4, 0)$.

Mitchell and Lapata (2010) demonstrated that the multiplication algorithm better predicts human similarity judgment between two-word phrases. Hence, this study tests whether their result is due to the ability of the multiplication algorithm to compute a vector for a noun compound of high semantic quality, or it is an artifact of the specific task or semantic spaces used.

3.2.3 Circular convolution

Circular convolution is a function that compresses the tensor product of two vectors, which is defined as

$$\boldsymbol{p} = \boldsymbol{h} \circledast \boldsymbol{m} \tag{13}$$

$$p_i = \sum_{j=0}^{n-1} h_{j \bmod n} \cdot m_{(i-j) \bmod n}. \tag{14}$$

Note that, in (14) that describes how each component of a compound vector is computed from constituent vectors, the indexes i and j for a component of a vector

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

take integer values modulo n (i.e., from 0 to $n-1$), rather than integer values ranging from 1 to n. Circular convolution combines the two example vectors to give $\boldsymbol{p} = (0, 10, 4, 1, 6) \circledast (1, 8, 4, 4, 0) = (68, 38, 108, 73, 70)$. For example, the second component $p_1 = 38$ is calculated as $h_0 \cdot m_1 + h_1 \cdot m_0 + h_2 \cdot m_4 + h_3 \cdot m_3 + h_4 \cdot m_2 = 0 \cdot 8 + 10 \cdot 1 + 4 \cdot 0 + 1 \cdot 4 + 6 \cdot 4 = 38$.

Circular convolution is often used as a method of generating semantic representations of individual words that contain syntactic or word-order information (Plate 2003; Jones and Mewhort 2007). In this study, similar to Mitchell and Lapata (2010), we regard circular convolution as an alternative method of vector composition.

Also note that the three algorithms (i.e., centroid, multiplication, and circular convolution) presented so far are insensitive to word order (or syntactic structure), which is a serious drawback of vector composition methods. On the other hand, the three composition algorithms explained below can generate different vectors depending on the order of the constituent words.

3.2.4 Predication

The predication algorithm was proposed by Kintsch (2001) to compute the intuitively plausible and contextually dependent vectors of propositions with the predicate argument structure. Given a proposition $P(A)$, where P is a predicate and A is an argument (in the case of a noun compound, the modifier M is the predicate and the head H is the argument), the predication algorithm first chooses the m nearest neighbors of a predicate P, i.e., the m words with the highest cosine to P. The algorithm then selects 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, and forms the vector representation of $P(A)$. When the predication algorithm is applied to noun compounds, the set of neighbors of P relevant to A represents the intended sense of the modifier M that is appropriate for describing the intended sense of the head noun H.

Formally, the predication algorithm computes a compound vector $\boldsymbol{p}$ by the following formula:

$$\boldsymbol{p} = \frac{\boldsymbol{m} + \boldsymbol{h} + \sum_{i=1}^{k} \boldsymbol{q_i}}{k+2} \tag{15}$$

where $\boldsymbol{q_i}$ denotes the vector with the ith highest cosine to the head vector $\boldsymbol{h}$ among the m vectors with the highest cosine to the modifier vector $\boldsymbol{m}$. Note that because the cosine similarity is insensitive to the length of vectors, $\boldsymbol{p} = \boldsymbol{m} + \boldsymbol{h} + \sum_{i=1}^{k} \boldsymbol{q_i}$ suffices.

For example, Table 1 shows how the predication algorithm calculates the composition vector of our example vectors $\boldsymbol{h}$ and $\boldsymbol{m}$ when $m = 5$ and $k = 2$. The leftmost column of Table 1 lists the neighbors of the modifier vector $\boldsymbol{m}$ in descending order of cosine similarity, whose values are listed in the second column. The predication algorithm first chooses the five nearest neighbors $\boldsymbol{w_5}, \boldsymbol{w_3}, \boldsymbol{w_1}, \boldsymbol{w_2}, \boldsymbol{w_9}$ (because $m = 5$), and then selects from them the two neighbors with the highest cosine to the head vector $\boldsymbol{h}$ (because $k = 2$). As shown in the third column, the vector $\boldsymbol{w_2}$ has the highest cosine to $\boldsymbol{h}$ and the vector $\boldsymbol{w_3}$ has the second highest cosine; thus, they correspond

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

Table 1. *Example of the step-by-step behavior of the predication algorithm for the hypothetical vectors*

Nearest neighbors of the modifier vector $\boldsymbol{m}$	Cosine values with the modifier vector $\boldsymbol{m}$	Cosine values with the head vector $\boldsymbol{h}$	
$\boldsymbol{w_5} = (1, 5, 3, 3, 0)$	0.995	0.792	
$\boldsymbol{w_3} = (2, 10, 9, 7, 7)$	0.881	0.889	$\rightarrow \boldsymbol{q_2}$
$\boldsymbol{w_1} = (9, 10, 6, 7, 1)$	0.876	0.678	
$\boldsymbol{w_2} = (2, 7, 7, 4, 5)$	0.866	0.892	$\rightarrow \boldsymbol{q_1}$
$\boldsymbol{w_9} = (7, 10, 5, 3, 4)$	0.857	0.842	
$\boldsymbol{w_6} = (6, 5, 2, 6, 2)$	0.773		
$\vdots$			

to $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$ in (15). As a result, the predication algorithm yields the composition vector $\boldsymbol{p} = \boldsymbol{m}+\boldsymbol{h}+\boldsymbol{q_1}+\boldsymbol{q_2} = (1, 8, 4, 4, 0)+(0, 10, 4, 1, 6)+(2, 7, 7, 4, 5)+(2, 10, 9, 7, 7) = (5, 35, 24, 16, 18)$.

3.2.5 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 modifier, i.e., words that are semantically related to both the head and modifier.

This study proposes that we can take fully into account the relevance by applying Utsumi's (2011) comparison algorithm, which chooses the k common neighbors with the higher cosine both to the head and modifier, and then computes the centroid of these common vectors and the head vector. The comparison algorithm can be seen as a computational model of Gentner's (1983) comparison (or analogical mapping) process consisting of alignment and projection (Utsumi 2011). In addition, the effectiveness of the comparison algorithm as a vector composition method for noun compounds may be supported by the empirical findings of Wisniewski (1996) stating that the comprehension of noun compounds involves such a comparison process.

Formally, the comparison algorithm computes a compound vector $\boldsymbol{p}$ using the following formula:

$$\boldsymbol{p} = \frac{\boldsymbol{h} + \sum_{\boldsymbol{c_i} \in S} \boldsymbol{c_i}}{k+1}. \tag{16}$$

In (16), S denotes a set of k vectors for the common neighbors of the head H and modifier M. The set of k common neighbors can be computed by finding the smallest i that satisfies $|N_i(H) \cap N_i(M)| \geq k$, where $N_i(w_j)$ denotes the set of the top i nearest neighbors of the word w_j, i.e., the set of i words with the highest cosine values to w_j. As in the case of predication, if the cosine is used as a similarity measure, multiplying the vector in (16) by $(k+1)$ (i.e., $\boldsymbol{p} = \boldsymbol{h} + \sum_{\boldsymbol{c_i} \in S} \boldsymbol{c_i}$) does not change the result of the similarity computation. Table 2 shows an example of

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

Table 2. *Example of common neighbors computed by the comparison algorithm for the hypothetical vectors*

Nearest neighbors of the modifier vector $\boldsymbol{m}$ and the cosines with $\boldsymbol{m}$		Nearest neighbors of the head vector $\boldsymbol{h}$ and the cosines with $\boldsymbol{h}$	
$\boldsymbol{w_5} = (1, 5, 3, 3, 0)$	0.995	$\boldsymbol{w_2} = (2, 7, 7, 4, 5)$	0.892
$\boldsymbol{w_3} = (2, 10, 9, 7, 7)$	0.881	$\boldsymbol{w_3} = (2, 10, 9, 7, 7)$	0.889
$\boldsymbol{w_1} = (9, 10, 6, 7, 1)$	0.876	$\boldsymbol{w_7} = (4, 6, 7, 2, 9)$	0.854
$\boldsymbol{w_2} = (2, 7, 7, 4, 5)$	0.866	$\boldsymbol{w_4} = (3, 6, 8, 0, 10)$	0.850
$\boldsymbol{w_9} = (7, 10, 5, 3, 4)$	0.857	$\boldsymbol{w_9} = (7, 10, 5, 3, 4)$	0.842
⋮		⋮	

common neighbor vectors using the hypothetical vectors in (9). Assuming $k = 1$, the comparison algorithm chooses one common neighbor $\boldsymbol{w_3}$, and then calculates the composition vector as $\boldsymbol{p} = \boldsymbol{h} + \boldsymbol{w_3} = (0, 10, 4, 1, 6) + (2, 10, 9, 7, 7) = (2, 20, 13, 8, 13)$.

3.2.6 Dilation

The dilation algorithm was proposed by Mitchell and Lapata (2010) as a basis-independent method for stretching or 'dilating' the head (or argument) vector along the direction of the modifier (or predicate) vector. This is achieved by decomposing the head vector $\boldsymbol{h}$ into two orthogonal vectors (i.e., a vector $\boldsymbol{x}$ parallel to $\boldsymbol{m}$ in (17) and a vector $\boldsymbol{y}$ orthogonal to $\boldsymbol{m}$ in (18)), and then stretching the parallel component $\boldsymbol{x}$ so that the modified vector $\boldsymbol{p}$ of $\boldsymbol{h}$ is more like $\boldsymbol{m}$:

$$\boldsymbol{x} = \frac{\boldsymbol{m} \cdot \boldsymbol{h}}{\boldsymbol{m} \cdot \boldsymbol{m}} \boldsymbol{m} \tag{17}$$

$$\boldsymbol{y} = \boldsymbol{h} - \boldsymbol{x} \;=\; \boldsymbol{h} - \frac{\boldsymbol{m} \cdot \boldsymbol{h}}{\boldsymbol{m} \cdot \boldsymbol{m}} \boldsymbol{m} \tag{18}$$

$$\boldsymbol{p} = \lambda \boldsymbol{x} + \boldsymbol{y} \;=\; (\lambda - 1) \frac{\boldsymbol{m} \cdot \boldsymbol{h}}{\boldsymbol{m} \cdot \boldsymbol{m}} \boldsymbol{m} + \boldsymbol{h} \tag{19}$$

In (19), λ is a parameter that represents the degree of dilation. Hence, $\lambda = 1$ means no dilation, and thus the resulting vector $\boldsymbol{p}$ is identical to the head vector $\boldsymbol{h}$. Note that, as in the case of predication and comparison, when the cosine is used as a similarity measure, the vector computed by (20) yields the same result of similarity computation as (19),

$$\boldsymbol{p} = (\lambda - 1)(\boldsymbol{m} \cdot \boldsymbol{h})\, \boldsymbol{m} + (\boldsymbol{m} \cdot \boldsymbol{m})\, \boldsymbol{h}. \tag{20}$$

In the case of our example, the dot products are calculated as $\boldsymbol{m} \cdot \boldsymbol{h} = (1, 8, 4, 4, 0) \cdot (0, 10, 4, 1, 6) = 100$, and $\boldsymbol{m} \cdot \boldsymbol{m} = (1, 8, 4, 4, 0) \cdot (1, 8, 4, 4, 0) = 97$. When we assume $\lambda = 3$, the dilation of $\boldsymbol{h}$ and $\boldsymbol{m}$ is $\boldsymbol{p} = (3 - 1) \times 100(1, 8, 4, 4, 0) + 97(0, 10, 4, 1, 6) = (200, 2570, 1188, 897, 582)$.

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

Table 3. *Semantic spaces used in this study*

	Corpus size (million words)	No. of different words $n_w (= n_c)$	No. of documents r	Matrix density (%)	
				word–doc	word–word
English	54.74	73,422	491,106	0.067	0.818
Japanese	26.20	63,875	604,058	0.063	1.100

Note. The columns of 'word–doc' and 'word–word' show the matrix densities of the word–document matrix for the LSA-based space and the word–word matrix for the PPMI-based space, respectively.

4 Evaluation experiment

4.1 Materials

The evaluation experiment was conducted in two languages, i.e., English and Japanese, to test whether different languages show consistent results.

In the experiment, we used two kinds of language resources, namely, a corpus and a thesaurus. Corpora were used to collect noun compounds and construct semantic spaces, while thesauri were used to collect unfamiliar noun compounds and to identify the meanings of words and compounds that are necessary for constructing test data. In this study, the written and non-fiction parts of the British National Corpus (henceforth BNC), and English WordNet 3.0 were used as an English corpus and thesaurus, respectively. Similarly, Japanese newspaper corpora (i.e., four years' worth of Mainichi newspaper articles and two years' worth of Nikkei newspaper articles), and the Japanese thesaurus 'Nihongo Dai-Thesaurus,'[4] (Yamaguchi 2006) were used as Japanese language resources.

4.1.1 Semantic spaces

Two semantic spaces, namely LSA-based and PPMI-based spaces, were constructed from these corpora. In order to exclude short documents that may adversely affect the quality of the constructed semantic spaces, we used as a unit of documents, English paragraphs that included 40 or more words and Japanese paragraphs that included 30 or more words. As a result, the English corpus contained 54.74 million words (73,422 types of words) and 491,106 documents (i.e., paragraphs), while the Japanese corpus contained 26.20 million words (63,875 word types) and 604,058 documents, as shown in Table 3.

The LSA-based semantic space was constructed from the word–document matrix $\boldsymbol{A}$ whose dimensions were 73,422 × 491,106 for English and 63,875 × 604,058 for Japanese. Following previous work (Mitchell and Lapata 2010), we determined

[4] This Japanese thesaurus consists of 1,044 basic categories that are divided into nearly 14,000 semantic categories. In this study, these semantic categories are used for representing the meanings of words. The thesaurus contains nearly 200,000 words (including compounds), most of which are classified into multiple semantic categories.

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

the number of dimensions n of the semantic space using a word-based similarity computation task. The task was identical to the task of related word ranking used in the evaluation experiment, which will be explained in Section 4.2.1, except that word–word pairs were used instead of compound–word pairs. One thousand English word–word pairs were selected from the University of South Florida Free Association Norms (Nelson, McEvoy and Schreiber 1998) and 1,000 Japanese word–word pairs were selected from a Japanese word association norm 'Renso Kijunhyo' (Umemoto 1969). We performed the task with different dimensions ranging from 100 to 2,000 in steps of 100 and compared the performance separately for English and Japanese word–word pairs. Both English and Japanese semantic spaces achieved the best performance when the number of dimensions was 300. Hence, 300-dimensional LSA spaces were used in the evaluation experiment.

The PPMI-based semantic space was constructed from the word–word matrix $\boldsymbol{A}$ with the same number of rows and columns (i.e., 73,422 × 73,422 for English, and 63,875 × 63,875 for Japanese), because in this study the word–word matrix was not smoothed. A context window of size 5 (i.e., five words on either side of the target word) was used for calculating the word co-occurrence frequency.

The density of the word–document or word–word matrix in Table 3 shows the percentage of non-zero elements of the matrix (after weighting), which is formally defined as (the number of non-zero elements) / $(l \times n) \times 100$. Given that the density of the initial matrix affects the quality of the semantic space, the English and Japanese semantic spaces may be equal in quality, because the two matrices are almost equal in density.

4.1.2 Compound-word pairs

In the evaluation experiment, we used two types of noun compounds, *familiar compounds* that occur in the corpus from which semantic spaces are constructed, and *unfamiliar compounds* that do not occur in the corpus. We used familiar compounds to evaluate how well composition algorithms can explicitly extract semantic information of noun compounds implicitly involved in the semantic space. Conversely, unfamiliar compounds were used to evaluate how well composition algorithms predict the semantic representation of noun compounds from semantic information of constituent words. Note that the ability to interpret unfamiliar compounds is particularly important for the algorithms not only as an NLP technique but also as a cognitive model, because people can easily interpret unfamiliar or novel noun compounds (Wisniewski 1996).

We chose familiar noun compounds that occur at least twenty times in the corpus and are also included in the thesaurus. Similarly, we selected unfamiliar noun compounds that do not occur in the corpus but are included in the thesaurus. All noun compounds comprised only two nouns (i.e., the head and modifier noun). As a result, 1,200 English and 2,191 Japanese familiar compounds, and 943 English and 427 Japanese unfamiliar compounds were chosen for evaluation.

For each of these chosen noun compounds, two semantically related words were selected from the English WordNet synsets (i.e., sets of synonyms) or the semantic

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

Table 4. *Examples of noun compounds and their related words used in the evaluation*

Familiar compound		
English	Non-emergent	turning point – landmark, opinion poll – canvass, power station – powerhouse
	Emergent	turning point – milestone, love affair – romance, heart disease – hypertension
Japanese	Non-emergent	information gathering – intelligence (*jouhou syuusyuu – chouhou*), traffic accident – disaster (*koutsuu jiko – saigai*), flag raising – Japanese flag (*kokki keiyou – hinomaru*)
	Emergent	interest-rate policy – rate reduction (*kinri seisaku – risage*), artificial intelligence – computer (*jinkou chinou – konpyûta*), flag raising – ceremony (*kokki keiyou – girei*)
Unfamiliar compound		
English	Non-emergent	divine law – principle, mass murder – massacre, ruling class – elite
	Emergent	bath towel – flannel, curtain call – bow, driving force – launch,
Japanese	Non-emergent	cover girl – beauty (*kabâ gâru – bijo*), career aspiration – ambition (*syusse yoku, – yashin*), subdivision – detail (*kai bunrui – saimoku*)
	Emergent	dress code – manner (*doresu kôdo – manâ*), animal protection – ecology (*kankyou hogo – ekorojî*), girlish taste – sentiment (*syoujo syumi – kansyou*)

Note. The original Japanese expressions are shown in parentheses, preceded by their literal English translations.

categories of the Japanese thesaurus that include the target compound. If none or only one word could be selected, then related words were selected randomly from hypernyms or coordinate words of the target compound. (This additional selection procedure was not necessary for Japanese words, because the semantic categories of the Japanese thesaurus are represented by fewer levels than in the English WordNet, and thus contain more words as synonyms.) In this procedure, the head and modifier themselves were prohibited from being selected as related words. As a result, this procedure generated 2,166 English and 4,296 Japanese compound-word pairs for familiar compounds, and 1,698 English and 833 Japanese pairs for unfamiliar compounds. Table 4 shows some examples of pairs of a noun compound and its related word used in the evaluation experiment.

Finally, these compound-word pairs were classified as emergent or non-emergent by the following criterion:

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

- A compound-word pair is judged to be emergent if semantic categories – synsets, hypernyms, hyponyms, and coordinates of the English thesaurus or the deepest categories of the Japanese thesaurus – of the paired word include neither the head nor the modifier of the compound.

Simply put, this criterion means that the word of a compound-word pair expresses the emergent feature (or meaning) of the compound if both the head and modifier of the compound are unrelated to the paired word. By this criterion, 276 English and 1,037 Japanese pairs for familiar compounds and 277 English and 292 Japanese pairs for unfamiliar compounds were judged to be emergent. Furthermore, if a noun compound had none or only one emergent pair, additional emergent pair(s) were generated using the same procedure as explained above. As a result, 534 English and 2,032 Japanese emergent pairs for familiar compounds (including 315 English and 1,055 Japanese compounds) and 542 English and 643 Japanese emergent pairs for unfamiliar compounds (including 313 English and 333 Japanese compounds) were generated. Table 4 shows some examples of emergent and non-emergent pairs. It must be noted here that emergence is not a property for a noun compound alone, but a property of a compound-word pair. Hence, the same noun compound (e.g., 'turning point' in Table 4) appears in both emergent and non-emergent pairs. Note also that this emergence judgment procedure provides only an approximation of feature emergence that involves a more complex phenomenon.

In order to confirm the appropriateness of this emergence judgment procedure, we conducted an experiment in which human ratings of semantic relatedness were collected. In the experiment, we chose 25 emergent and 25 non-emergent compound-word pairs from the 4,296 Japanese compound-word pairs generated above. Fourteen participants were asked to rate the semantic relatedness between the compound and the word in all 50 compound-word pairs (e.g., semantic relatedness between 'heart disease' and 'hypertension') on a seven-point scale of one (*unrelated*) to seven (*related*). (These data were also used for the evaluation experiment in Section 4.2.) They were also asked to rate the semantic relatedness of the head or the modifier of the compound to the word paired with the compound (e.g., semantic relatedness between 'disease' and 'hypertension' or between 'heart' and 'hypertension'). The presentation order of those pairs was randomized for each participant. If the emergence judgment procedure using a thesaurus is appropriate, it is expected that, in the case of an emergent compound-word pair, the paired word would be rated as semantically more related to the compound than to the constituent words (i.e., the head and modifier) of the compound. On the other hand, non-emergent compound-word pairs are not expected to exhibit such differences. The results of the evaluation experiment were consistent with the prediction. We compared the relatedness rating of a compound with the relatedness rating of the constituent words of that compound. The relatedness rating value of constituent words was determined as the higher rating between the head and modifier ratings. For emergent pairs, the relatedness rating of compounds to the paired words ($M = 5.27$) was significantly higher than the relatedness rating of constituent words ($M = 4.99$), $t(24) = 2.65$, $p < .01$. For non-emergent pairs, the relatedness rating of compounds ($M = 5.96$)

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

was not significantly different from the relatedness rating of constituent words ($M = 6.15$), $t(24) = 1.48$, $p > .10$. These results confirm that our classification of emergent and non-emergent pairs is appropriate for an approximation of feature emergence.[5]

4.2 Method

To evaluate which composition algorithms are appropriate for computing the attributional similarity of noun compounds, this study employs the following three tests for noun compounds: related word ranking, similarity correlation, and semantic classification. The first two tests (i.e., related word ranking and similarity correlation) are used to evaluate the relative performance of similarity computation, while the semantic classification test is designed to explore the absolute performance of the similarity computation.

4.2.1 Related word ranking

This test examines the degree of attributional similarity between the compound and its related word of a compound-word pair by finding the rank of the related word under the ordering imposed by the cosine. This kind of word ranking has been used as a performance test by many studies on semantic spaces (Griffiths, Steyvers and Tenenbaum 2007; Baroni and Zamparelli 2010).

Figure 1 illustrates the overall procedure of the word ranking test. First, for each compound-word pair (c_i, w_i) comprising a noun compound c_i and its related word w_i in a given set of pairs P described in Section 4.1.2, the vector for the noun compound c_i was computed by a given composition algorithm. Afterward, the rank r_i of the paired word w_i was assessed by computing the cosine similarity between (the vector for) c_i and (the vectors for) all words (including w_i) in a semantic space, and sorting all words in descending order of the cosine. A higher rank implies that the word w_i is semantically more related to the compound c_i. For example, in the case of the compound-word pair (c_1, w_1) = (heart disease, hypertension) shown in Figure 1, the word 'hypertension' has the eighth highest cosine similarity to 'heart disease' in the semantic space. Hence, for the pair (c_1, w_1), the rank r_1 is 8. Similarly, for the pair 'love affair – romance' the rank r_i of the word 'romance' is 3. Next, all compound-word pairs in the set P were sorted in ascending order of the rank r_i, as depicted on the right side of Figure 1. The sorted list of the rank r_i is denoted as $r'_1, \ldots, r'_{|P|}$. (Hence, for example, r'_1 is the minimum rank in $r_1, \ldots, r_{|P|}$.) In Figure 1, the pair 'love affair – romance' ranks fifth (i.e., r'_5 = (the rank r_1 of that pair) = 3), while the pair 'heart disease – hypertension' ranks twentieth (i.e., $r'_{20} = 8$).

Finally, the overall performance of each composition algorithm was measured by the median rank of the sorted list $r'_1, \ldots, r'_{|P|}$ for the set of pairs P. Formally, the

[5] The inter-rater agreement was calculated as the mean of the Spearman's rank correlation coefficients over all possible pairs of 14 raters. The result was that $\rho = 0.450$ for the compound-word similarity and $\rho = 0.516$ for the word–word similarity.

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

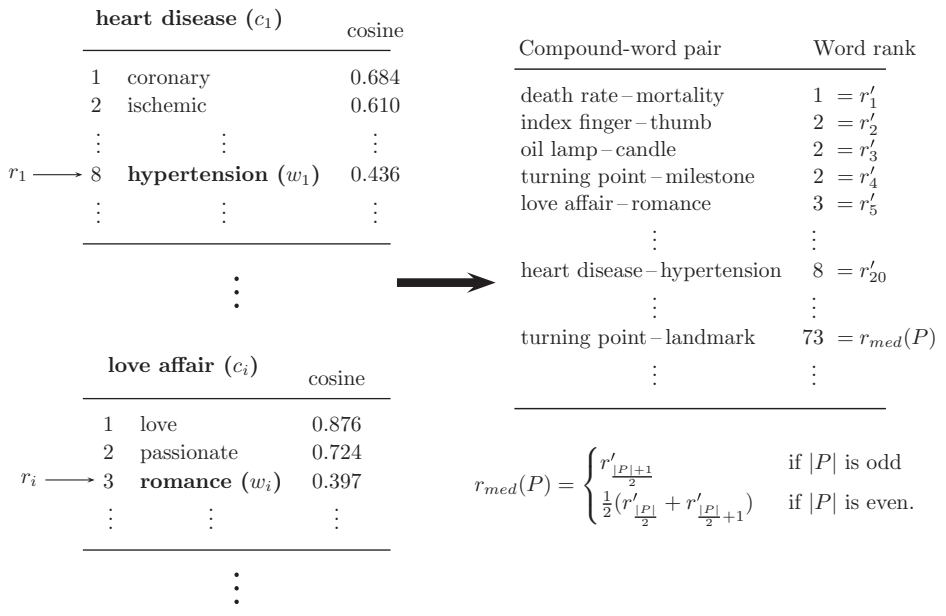


Fig. 1. Overall procedure of the task of ranking related words.

median rank $r_{med}(P)$ of a set P of compound-word pairs is defined as

$$r_{med}(P) = \begin{cases} r'_{\frac{|P|+1}{2}} & \text{if } |P| \text{ is odd} \\ \frac{1}{2}\left(r'_{\frac{|P|}{2}} + r'_{\frac{|P|}{2}+1}\right) & \text{if } |P| \text{ is even.} \end{cases} \tag{21}$$

In addition, we used as a secondary measure first quartile (i.e., 25th percentile $\simeq r'_{0.25|P|}$) and third quartile (i.e., 75th percentile $\simeq r'_{0.75|P|}$).

4.2.2 Similarity correlation

In this test, we used correlation analysis to examine how well composition algorithms (and their computed similarity values) predict human similarity judgment.

In order to collect a manageable size of human judgment data, we chose one hundred Japanese compound-word pairs that comprised twenty-five non-emergent and twenty-five emergent pairs for familiar compounds, and twenty-five non-emergent and twenty-five emergent pairs for unfamiliar compounds.[6] We then asked fourteen participants, who were all native speakers of Japanese, to rate the semantic relatedness between the compounds and words of the 100 pairs stated above. These pairs were rated on a seven-point scale ranging from one (*unrelated*) to seven (*related*). The presentation order of those pairs was randomized for each participant. The mean ratings for each pair were used in the correlation analysis.

The performance of each composition algorithm was measured by Spearman's correlation coefficient. First, for each of the one hundred compound-word pairs,

[6] English compound-word pairs were not used for the correlation analysis, because a sufficient number of native English speakers could not be recruited.

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

the vector for the compound was generated by a given algorithm. Next, the cosine similarity between the compound vector and the vector for the paired word was calculated. Lastly, a correlation coefficient was calculated between the computed cosine values and mean human ratings of the compound-word pairs.

4.2.3 Semantic classification

In this test, each compound of the compound-word pairs generated in Section 4.1.2 was paired with a semantically unrelated word. During the test, a given set of pairs was classified according to the cosine similarity into two groups, namely semantically related and unrelated pairs. (Hence, this test can be regarded as a binary classification problem.)

Semantically unrelated pairs were constructed by combining the compound (e.g., 'love affair') of semantically related pairs (e.g., 'love affair–romance') with an unrelated (or less related) word (e.g., 'finding'). Unrelated words were chosen randomly from the words that were unrelated to the compound, the head noun, and the modifier noun. Specifically, words were judged to be unrelated if none of their synsets, hypernyms, hyponyms, and coordinates included the target compound, head noun, and modifier noun for English tests, or if none of their basic categories included the target compound, head noun and modifier noun for Japanese tests. In addition, to largely reduce the undesirable effect of the word frequency on the cosine similarity, all unrelated words were chosen to occur more than fifty times in the corpus. Hence, the number of unrelated pairs generated was equal to the number of related pairs. As a result, the number of pairs that had to be classified was twice the number of compound-word pairs used in the word ranking test. For example, 4,332 English and 8,592 Japanese pairs were used for familiar compounds, and 3,396 English and 1,666 Japanese pairs were used for unfamiliar compounds.

In the evaluation experiment, all compound-noun pairs were sorted in descending order of cosine similarity, and the top k pairs were retrieved as semantically related. The performance for this binary classification was measured by average precision, which has been widely used for evaluating the ranked retrieval results (Manning, Raghavan and Schütze 2008). Average precision (AP) is the average of the precision value obtained for each recall level (i.e., after each 'related' pair is retrieved). For a ranked set of compound-noun pairs P, the average precision AP(P) was calculated by

$$\mathrm{AP}(P) = \frac{1}{N} \sum_{i=1}^{N} \mathrm{Precision}(P_i) \tag{22}$$

where P_i is the set of compound-word pairs whose cosine similarity is equal to or more than the cosine of the ith related pair, and N is the number of related compound-word pairs. Higher average precision values imply better performance. Note that the average precision approximates the area under the uninterpolated precision–recall curve.

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

4.3 Results

For each of the six composition algorithms, both versions (i.e., all and emergent compounds) of the three tests were conducted in both English and Japanese and using LSA-based and PPMI-based semantic spaces, and their performance measures were calculated.

In order to compute the performance of the predication, comparison, and dilation algorithms, we estimated the optimal parameters using a tenfold cross-validation procedure. The cosine similarity (and the rank) of compound-word pairs in each group was calculated, with the optimal parameters estimated using the other nine groups as training data. Optimal parameters were estimated based on the median rank for the word ranking and similarity correlation tests, while they were estimated based on the mean average precision (henceforth MAP) over training groups for the semantic classification task. The parameter space was given such that the parameter m (for predication) varied between 1 and 50 and between 100 and 500 in steps of 50, the parameter k (for predication and comparison) varied between 1 and 10, and the parameter λ (for dilation) varied between 1.1 and 10.0 in steps of 0.1. Note that we removed any combinations of m and k for which $m < k$.

Furthermore, for purpose of comparison, two non-compositional methods were considered for obtaining compound vectors. The first method, regarded as the baseline, simply used the vector representing the head noun as the compound vector. (Note that for $\lambda = 1.0$, the dilation algorithm is identical to this baseline algorithm.) The second method computed compound vectors directly from the corpus; in other words, semantic spaces were constructed in which all the target noun compounds were added as single words. This method was expected to achieve better and probably the best performance, which suggests a target level for the composition algorithms. Note that this method could not be applied to unfamiliar compounds because they were not included in the corpus.

4.3.1 Overall performance

In this section, the results of the three tasks for familiar noun compounds are shown. The results of the unfamiliar noun compounds will be shown later in Section 4.3.4.

Related word ranking. Tables 5 and 6 respectively show the results (i.e., median rank) of the word ranking test when LSA-based and PPMI-based semantic spaces are used for vector composition. In addition, Figures 2 and 3 also show the first and third quartile ranks of the ranking of a word, as well as the median ranks. (Because the third quartile rank is much lower than the median rank and the first quartile rank, it is not used as a performance measure in this paper.)

Concerning the results of the LSA-based semantic space (shown in Table 5 and Figure 2), it is not surprising that the original compound vector computed directly from the corpus yields the best performance (i.e., both median rank and first quartile rank) for Japanese compounds. In the case of English compounds, the original compound vector also yields higher performance. Among the six composition algorithms, the dilation algorithm achieves the best performance in

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

Table 5. *Median rank of words semantically related to familiar noun compounds in the ordering computed in the LSA-based semantic space*

Algorithm	All pairs		Emergent pairs	
	English	Japanese	English	Japanese
Centroid	2,925.0 (5)	1,534.0 (3)	6,833.0 (6)	3,826.0 (6)
Multiplication	17,074.5 (8)	18,711.0 (8)	16,715.0 (8)	20,770.5 (8)
Circular convolution	7,883.0 (7)	5,940.5 (7)	13,885.0 (7)	8,304.5 (7)
Predication	2,894.0 (4)	1,597.5 (4)	6,436.5 (5)	3,724.5 (5)
Comparison	2,831.5 (3)	1,639.5 (5)	3,995.5 (1)	2,446.0 (3)
Dilation	2,315.0 (1)	1,216.5 (2)	4,127.0 (2)	2,379.5 (2)
Original vector	2,385.0 (2)	1,183.0 (1)	4,491.0 (3)	2,017.5 (1)
Head noun	2,981.0 (6)	1,737.0 (6)	5,249.0 (4)	2,857.0 (4)

Note. Underlined numbers indicate the lowest median rank values (i.e., the best performance) among the composition algorithms. Numbers in parentheses denote the ranks of the algorithms in ascending order of median rank.

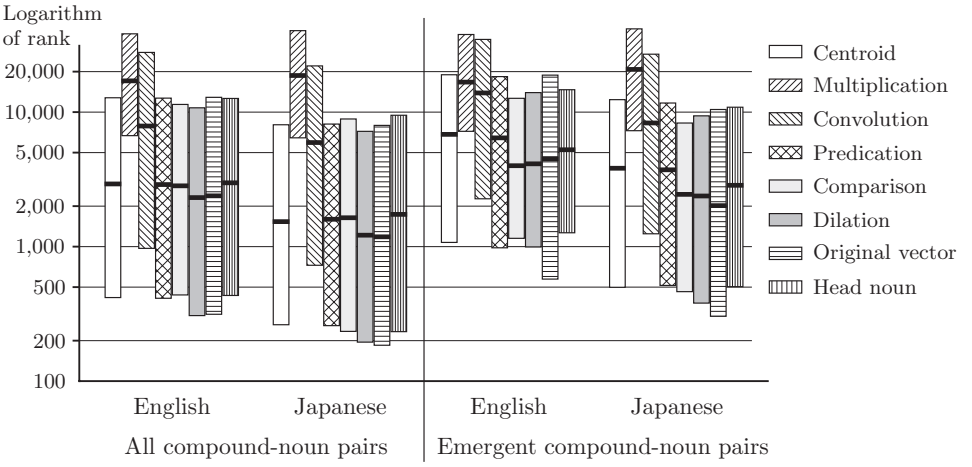


Fig. 2. Boxplots of the rank of semantically related words to familiar noun compounds in the ordering computed in the LSA-based semantic space. The bottom and top of the box represent the first and third quartile ranks, respectively, and the band near the middle of the box represents the median rank. Note that a lower rank value implies better performance.

both languages, although the comparison algorithm achieves the best performance for English emergent pairs. In addition, only the dilation and comparison algorithms perform better than the head noun vector (i.e., baseline) in both languages and in both types of pairs. These results suggest that the dilation and comparison algorithms may be effective in computing the meaning of noun compounds. Note that the circular convolution and multiplication algorithms have much worse performance than the other four algorithms.

The results of the PPMI-based semantic space (shown in Table 6 and Figure 3) are radically different from those of the LSA-based space in one point. The multiplic-

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

Table 6. *Median rank of words semantically related to familiar noun compounds in the ordering computed in the PPMI-based semantic space*

Algorithm	All pairs		Emergent pairs	
	English	Japanese	English	Japanese
Centroid	1,237.0 (2)	679.5 (3)	2,147.5 (5)	1,572.0 (4)
Multiplication	796.5 (1)	408.5 (1)	1,674.0 (1)	1,003.5 (1)
Circular convolution	6,361.5 (8)	8,217.5 (8)	6,148.5 (8)	9,678.0 (8)
Predication	1,432.5 (5)	822.5 (5)	2,196.5 (6)	1,768.5 (7)
Comparison	1,524.0 (6)	889.0 (7)	2,135.0 (4)	1,729.5 (6)
Dilation	1,247.5 (3)	645.5 (2)	1,953.0 (2)	1,422.0 (3)
Original vector	2,527.5 (7)	852.5 (6)	3,409.0 (7)	1,340.5 (2)
Head noun	1,289.0 (4)	773.5 (4)	2,113.0 (3)	1,683.0 (5)

Note. Underlined numbers indicate the lowest median rank values (i.e., the best performance) among the composition algorithms. Numbers in parentheses denote the ranks of the algorithms in ascending order of median rank.

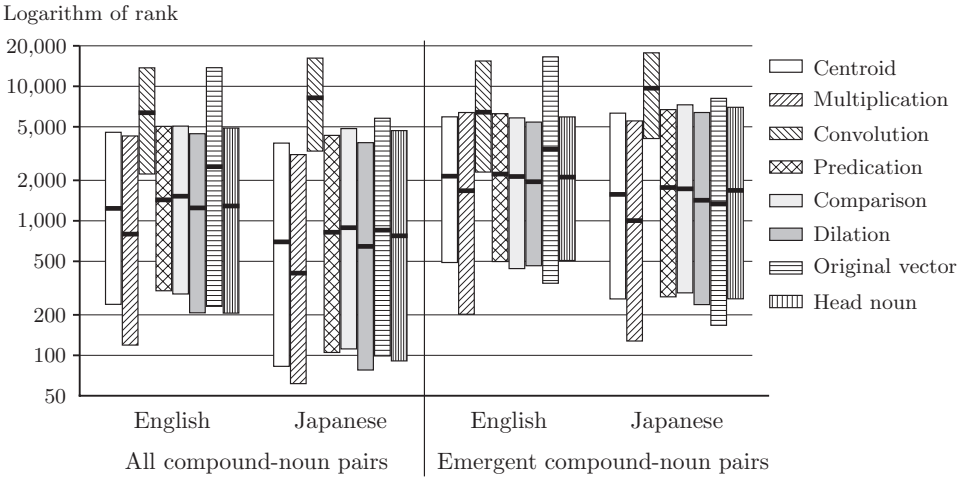


Fig. 3. Boxplots of the rank of semantically related words to familiar noun compounds in the ordering computed in the PPMI-based semantic space.

ation algorithm achieves the highest performance among all the algorithms in the PPMI-based semantic space, even though it achieves the lowest performance when the LSA-based semantic space is used. These opposing results clearly indicate that the performance of the multiplication algorithm depends heavily on the characteristics of semantic spaces. Possible reasons for these results will be discussed in Section 5. The relative performances of all composition algorithms other than multiplication do not differ largely between the PPMI-based and LSA-based semantic spaces. The dilation algorithm performs better than the other algorithms across languages and types of pairs. The comparison algorithm performs worse in the PPMI-based space than in the LSA-based space; it underperforms the baseline in both types of

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

Table 7. *Correlation coefficients between human similarity ratings and the cosine similarity computed by the composition algorithms for Japanese familiar compounds*

Algorithm	All pairs				Emergent pairs			
	LSA		PPMI		LSA		PPMI	
Centroid	.367*	(3)	.376*	(3)	.213	(6)	.179	(4)
Multiplication	−.149	(8)	.393*	(2)	.032	(8)	.240	(2)
Circular convolution	.216	(7)	.294	(7)	.345†	(1)	.098	(6)
Predication	.365*	(4)	.397**	(1)	.173	(7)	.209	(3)
Comparison	.328*	(6)	.360*	(4)	.241	(5)	.133	(5)
Dilation	.447**	(1)	.355*	(5)	.324	(2)	.095	(7)
Original vector	.416**	(2)	.331*	(6)	.266	(3)	.248	(1)
Head noun	.361*	(5)	.287	(8)	.255	(4)	.009	(8)

Note. Underlined numbers indicate the highest coefficients among the composition algorithms. Numbers in parentheses denote the ranks in descending order of the correlation coefficients. † $p < .10$. *$p < .05$. **$p < .01$.

pairs, and the predication and centroid algorithms in the case of all pairs. Circular convolution also achieves much worse performance.

Concerning effects that may occur due to the use of different languages, the relative performances among the composition algorithms do not differ between English and Japanese. This consistency in English and Japanese results increases the generality of the findings in this paper. However, note that a difference is observed regarding the absolute ranks; as shown in Figures 2 and 3, the median and first quartile ranks of Japanese compound-word pairs are lower (thus better) than those of English pairs (i.e., the boxes for Japanese compound-word pairs are positioned lower than the boxes for English pairs). One possible reason for this difference may be that the difficulty levels for English and Japanese tests are different, owing to the fact that different kinds of thesauri are used for English and Japanese. As mentioned in Section 4.2, the distance in the hierarchical structure of the thesaurus determines which words are semantically related to the target compounds, and thus different thesauri may generate test items with different difficulty levels.

Similarity correlation. Table 7 shows correlation coefficients between human similarity rating and cosine similarity computed in both semantic spaces.

Let us first consider the result of all compound-word pairs ($n = 50$) shown in the second and third columns of Table 7. When the LSA-based semantic space is used, the four composition algorithms (i.e., centroid, predication, comparison, and dilation) are significantly correlated with human judgments, while the multiplication and circular convolution algorithms are not. The dilation algorithm outperforms all other algorithms including the original compound vector. These results are almost consistent with the results of the word ranking test. The results of the correlation analysis of the PPMI-based semantic space are also consistent with the results of the word ranking test. The multiplication algorithm shows a significant correlation with

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

Table 8. *Mean average precision of semantic classification for familiar compounds using the LSA-based semantic space*

	All pairs		Emergent pairs	
Algorithm	English	Japanese	English	Japanese
Centroid	.9202 (4)	.9264 (3)	.9358 (4)	.9427 (6)
Multiplication	.5919 (8)	.5618 (8)	.6805 (8)	.5908 (8)
Circular convolution	.8229 (7)	.8304 (7)	.7865 (7)	.8286 (7)
Predication	.9207 (3)	.9254 (4)	.9321 (5)	.94547 (5)
Comparison	.9140 (5)	.9185 (5)	.9555 (1)	.9597 (1)
Dilation	.9283 (2)	.9355 (2)	.9502 (2)	.9590 (2)
Original vector	.9482 (1)	.9541 (1)	.9128 (6)	.94550 (4)
Head noun	.9121 (6)	.9169 (6)	.9395 (3)	.9480 (3)

Note. Underlined numbers indicate the highest MAP values among the composition algorithms. Numbers in parentheses denote the ranks of the algorithms in descending order of the MAP values.

human judgment and performs better than all other algorithms except predication. The high performance of the predication algorithm is not observed in the word ranking test. The other three composition algorithms (i.e., dilation, comparison, and centroid) are significantly correlated with human judgments, but the circular convolution algorithm is not.

Correlations calculated for emergent pairs ($n = 25$) are not statistically significant, and thus it may not be appropriate to derive a reliable conclusion from the results of the emergent pairs. However, the results of the emergent pairs are almost consistent with the results of all the pairs; the correlation coefficients of the dilation and multiplication algorithms are relatively high in the LSA-based and PPMI-based semantic spaces, respectively. A notable difference is that the correlation coefficient of circular convolution is rather high and marginally significant in the LSA-based space.

Semantic classification. Tables 8 and 9 respectively show the results (i.e., mean average precision) of the semantic classification test in the LSA-based and PPMI-based semantic spaces.

On the whole, the results of semantic classification are consistent with the results of the word ranking and similarity correlation tests. When the LSA-based semantic space is used, the dilation algorithm performs consistently well; its MAP value is higher than the other composition algorithms in the case of all pairs, and also higher than the original vector for emergent pairs. The comparison algorithm achieves the best performance particularly for emergent pairs; this result is partially consistent with the result of the word ranking test. Only the dilation and comparison algorithms perform better than the baseline, regardless of language and type of pair. The multiplication and circular convolution algorithms perform worse than the other algorithms. In particular, the multiplication algorithm performs very poorly. In the PPMI-based semantic space, the multiplication algorithm achieves the best

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

Table 9. *Mean average precision of semantic classification for familiar compounds using the PPMI-based semantic space*

Algorithm	All pairs		Emergent pairs	
	English	Japanese	English	Japanese
Centroid	.8998 (3)	.9153 (4)	.9130 (5)	.9099 (4)
Multiplication	.9306 (1)	.9427 (1)	.9310 (1)	.9396 (1)
Circular convolution	.6460 (8)	.5441 (8)	.6798 (8)	.5382 (8)
Predication	.8911 (5)	.9048 (6)	.9052 (6)	.9001 (6)
Comparison	.8870 (7)	.8991 (7)	.9136 (4)	.8988 (7)
Dilation	.9003 (2)	.9162 (3)	.9169 (2)	.9136 (3)
Original vector	.8893 (6)	.9289 (2)	.8642 (7)	.9176 (2)
Head noun	.8933 (4)	.9050 (5)	.9150 (3)	.9079 (5)

Note. Underlined numbers indicate the highest MAP values among the composition algorithms. Numbers in parentheses denote the ranks of the algorithms in descending order of the MAP values.

performance, regardless of language and type of pair, as in the case of the word ranking test. The dilation algorithm also performs consistently well, although its MAP value is lower than that of the original vector when Japanese pairs are considered. The comparison algorithm performs somewhat worse than in the LSA-based space; it underperforms not only the baseline, but also the predication and centroid algorithms. The circular convolution algorithm has the lowest performance, as in the case of the word ranking test.

Summarizing, based on the above results, we conclude that the dilation algorithm is generally most effective in computing the meaning of noun compounds in the LSA-based semantic space, while the multiplication algorithm is best suited to compute the meaning of noun compounds in the PPMI-based semantic space. In particular, the finding that these algorithms can exceed the performance obtained by the original vector suggests that vector composition not only approximates the original vector but also actively elicits (or mines) the implicit knowledge of compound meanings behind the corpus. Furthermore, these results are not specific to a certain language, thus indicating some generality of the findings.

4.3.2 *Effect of frequency of noun compounds*

To assess to what degree an algorithm's performance is affected by the frequency of the compounds in the corpus, we also examined the performance scores (i.e., median rank and mean average precision) calculated when the target compounds were limited to those that occur at least *tf* times in the corpus. Figures 4 and 5 show the median rank of the word ranking test for a threshold *tf* ranging from 20 (i.e., 'unlimited') to 400 in steps of 10. Graphs of mean average precision for the semantic classification test calculated with the same thresholds are shown in the Appendix (i.e., Figures 11 and 12). (Note that the graphs for the multiplication

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

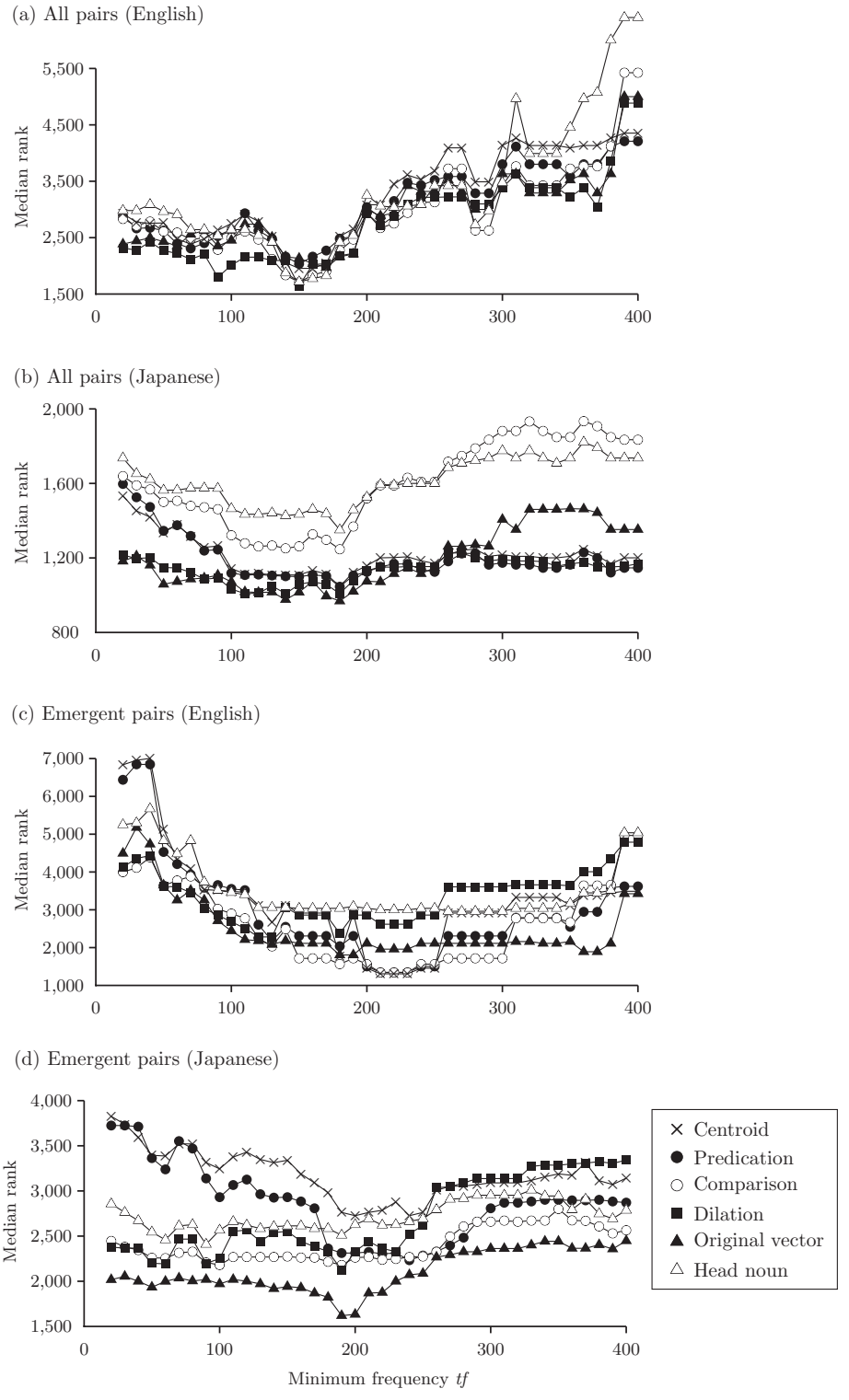


Fig. 4. Median rank as a function of minimum frequency of noun compounds in the case of the LSA-based semantic space.

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

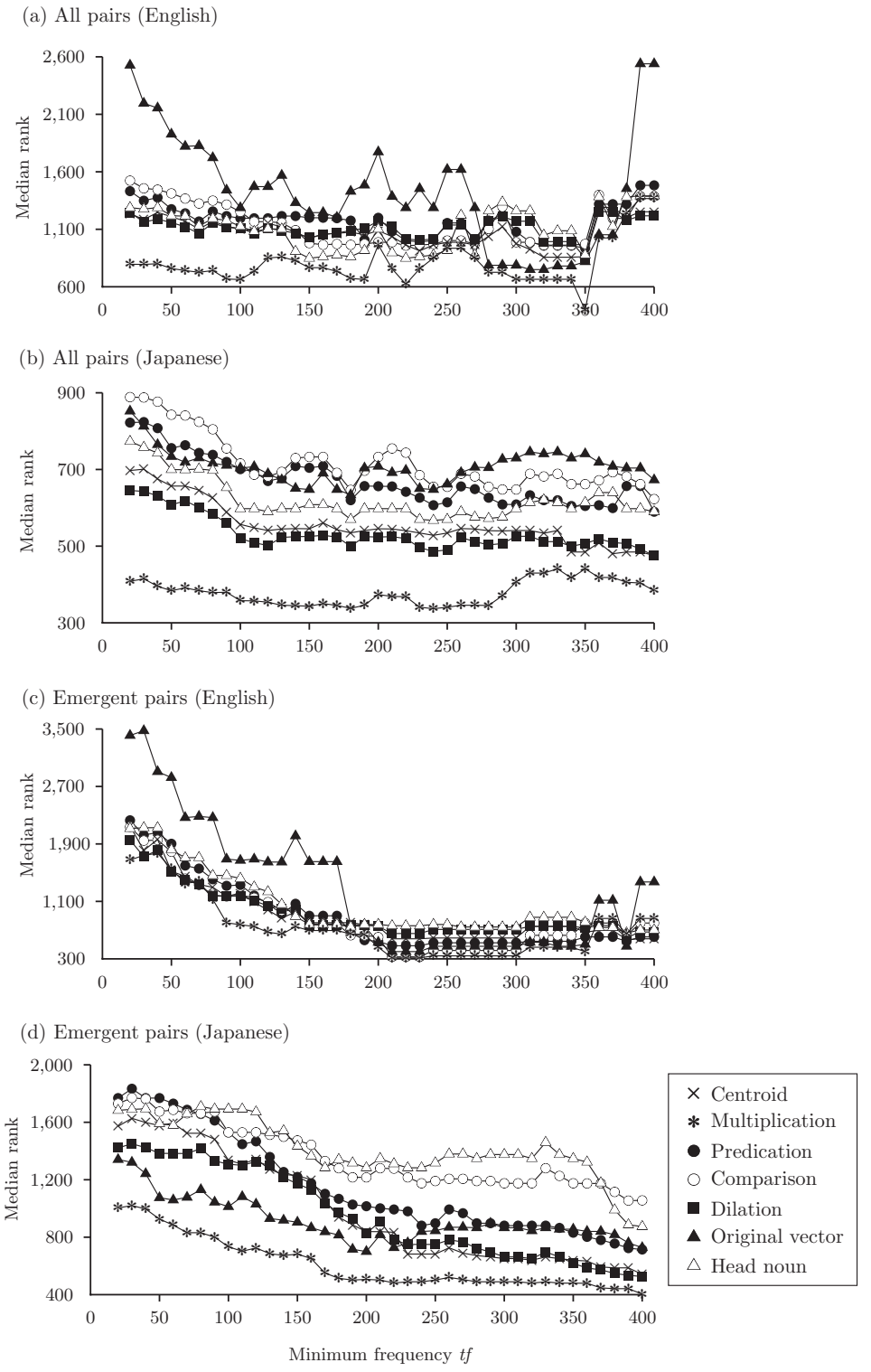


Fig. 5. Median rank as a function of minimum frequency of noun compounds in the case of the PPMI-based semantic space.

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

and circular convolution algorithms in the LSA-based space, and graphs for the circular convolution algorithm in the PPMI-based space are not shown in these figures because their performance is much lower than that of the other algorithms regardless of frequency.)

As expected, both median rank and mean average precision are proportionally dependent on the frequency threshold *tf* ; they get better (i.e., lower rank value and higher MAP value) as target compound-word pairs are limited to more frequent compounds. One plausible explanation for the result is that frequent compounds are used as if they are lexicalized, and thus the vectors for the constituent words (i.e., head and modifier) of frequent compounds implicitly involve the semantic knowledge for the compounds, which can be accurately extracted by the composition algorithms. On the other hand, the constituent words of less frequent compounds are not likely to involve the semantic information of the compounds.

However, one unexpected result is obtained that the performance of some compound-word pairs drops for highly frequent compounds. For example, Figure 4 depicts that the performance of all compound-word pairs in the LSA-based space drops (i.e., the median rank rises) for compounds with corpus frequency > 150 (English) and > 180 (Japanese). This tendency is less pronounced in the PPMI-based space, as shown in Figure 5, but the performance of English pairs becomes worse for compounds with corpus frequency > 350. One possible explanation is that highly frequent compounds (and their constituent words) are semantically related to a number of words, and consequently it is more difficult for a semantic space model to compute the vectors of highly frequent compounds that accurately grasp the meaning of the compounds than the vectors of less frequent compounds.

Concerning whether the relative performances of the composition algorithms change over frequency, no drastic changes are observed, as shown in Figures 4 and 5 or in Figures 11 and 12. Regardless of the frequency threshold, the dilation algorithm performs consistently better than the other algorithms (excluding the original vector) in the LSA-based semantic space, while the multiplication algorithm outperforms the other algorithms in the PPMI-based semantic space. This result suggests that the compound frequency does not explain the observed performance differences among the composition algorithms.

4.3.3 Effect of constituent frequency

The quality of a composition vector depends not only on compound frequency, as shown in Section 4.3.2, but also on the frequency of constituent words (i.e., head and modifier), because word frequency has a great effect on the quality of word vectors. Hence, in this section, we examine the relationship between constituent frequency and the performance of composition algorithms.

Figures 6 and 7 show the median rank of all compound-word pairs for a threshold *tf* of constituent frequency ranging from 10 to 10^4. (Note that the data for high constituent frequency are not plotted in some graphs because their values of median rank are too high to be plotted.) The performance of composition algorithms slightly increases (i.e., the median rank decreases) in the case of the LSA-based space, or

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

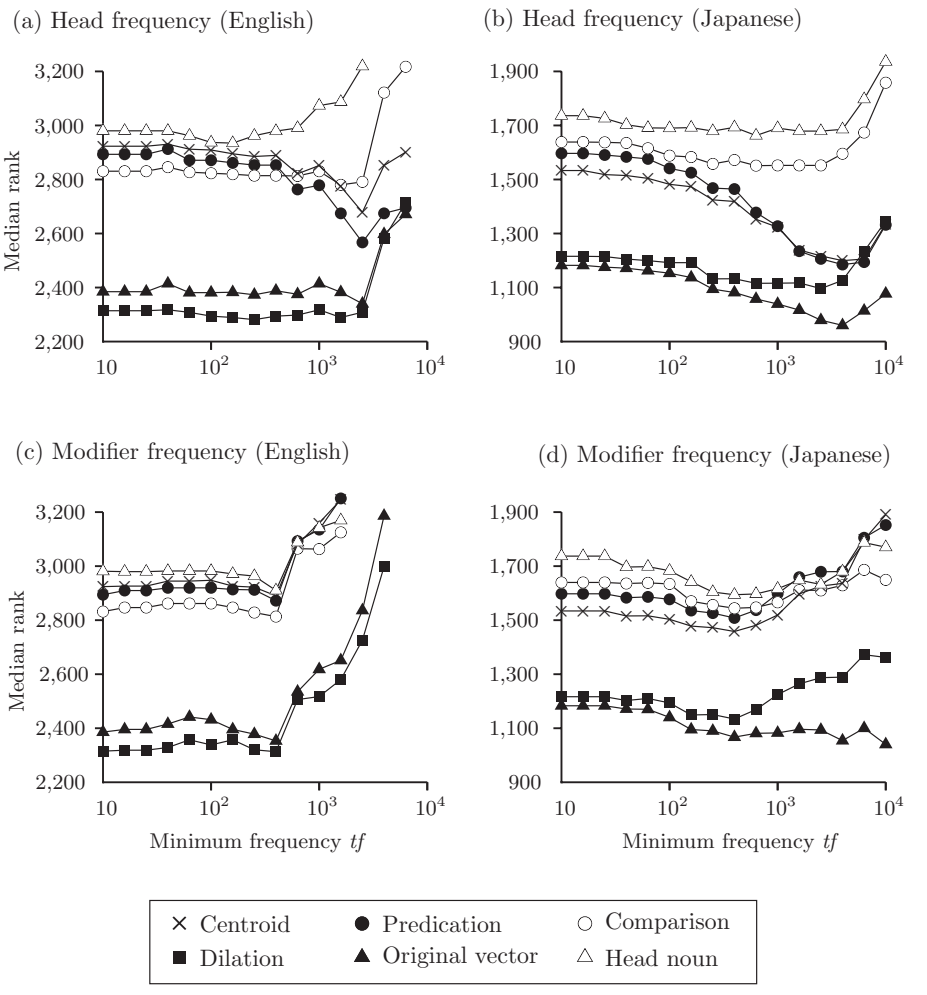


Fig. 6. Median rank of all compound-word pairs as a function of minimum frequency of constituent words (head or modifier) in the case of the LSA-based semantic space.

does not change in the case of the PPMI-based space, regardless of whether head frequency or modifier frequency is considered. This result suggests that constituent frequency has little effect on the performance of composition algorithms. The quality of composition algorithms may depend on the degree to which the vectors for the constituent words implicitly involve the semantic knowledge of the compounds, but such the degree may have nothing to do with constituent frequency (and depends on compound frequency).

In addition, the performance drops for compounds with very high constituent frequency (i.e., about 10^3 or more), as in the case of the relation between compound frequency and algorithms' performance described in Section 4.3.2. The reason may be the same; highly frequent words are semantically related to many words, and thus it is more difficult for a composition algorithm to extract the specific meaning appropriate for noun compounds.

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

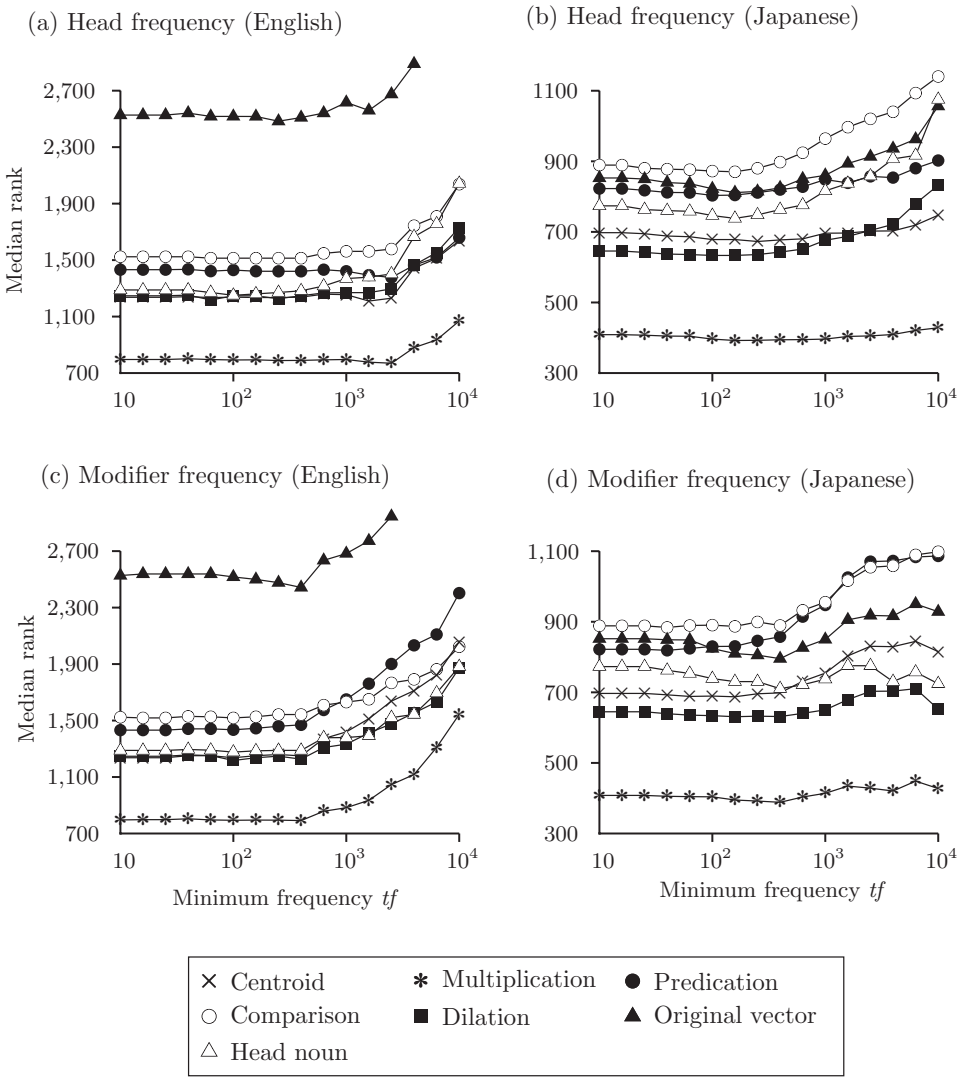


Fig. 7. Median rank of all compound-word pairs as a function of minimum frequency of constituent words (head or modifier) in the case of the PPMI-based semantic space.

4.3.4 Performance for unfamiliar noun compounds

In this section, we show the results of the three tests for unfamiliar compounds, and compare them with those of familiar compounds.

Related word ranking. Tables 10 and 11 show the results of the word ranking test for unfamiliar compounds in the LSA-based and PPMI-based semantic spaces. Figures 8 and 9 show boxplots of the ranks (i.e., the first and third quartile ranks as well as the median rank) of semantically related words to unfamiliar compounds. (Note again that because unfamiliar compounds are not included in the corpus, their original vector cannot be computed.)

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

Table 10. *Median rank of words semantically related to unfamiliar noun compounds in the ordering computed in the LSA-based semantic space*

Algorithm	All pairs		Emergent pairs	
	English	Japanese	English	Japanese
Centroid	4,163.0 (5)	6,453.0 (5)	7,480.5 (5)	8,287.0 (5)
Multiplication	14,682.0 (7)	15,024.0 (7)	12,400.0 (7)	15,941.0 (7)
Circular convolution	8,364.5 (6)	11,461.0 (6)	11,704.0 (6)	13,942.0 (6)
Predication	4,077.5 (4)	6,364.0 (4)	6,771.5 (4)	7,753.0 (4)
Comparison	3,571.5 (2)	5,097.0 (2)	4,693.0 (1)	5,166.0 (1)
Dilation	3,127.5 (1)	4,574.0 (1)	5,283.5 (2)	5,781.0 (2)
Head noun	3,748.0 (3)	5,464.0 (3)	6,627.0 (3)	6,757.0 (3)

Note. Underlined numbers indicate the lowest median ranks (i.e., the best performance) among the composition algorithms. Numbers in parentheses denote the ranks of the algorithms in ascending order of the median rank.

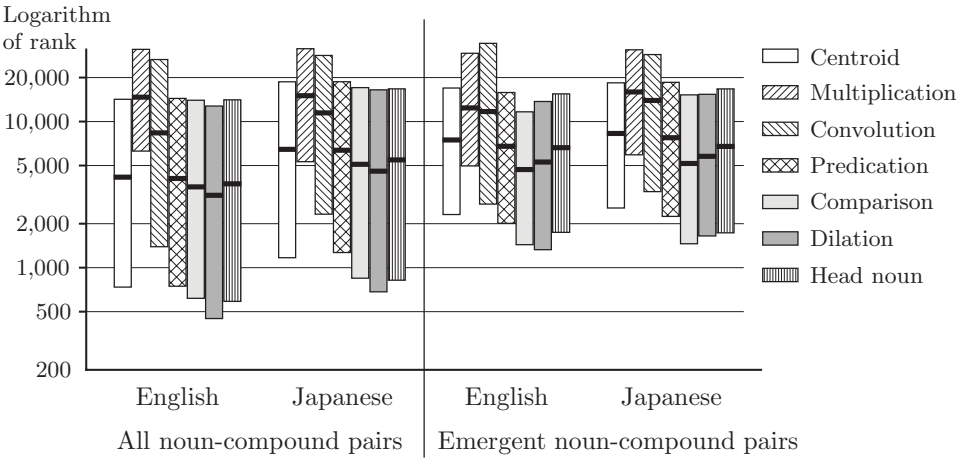


Fig. 8. Boxplots of the rank of semantically related words to unfamiliar noun compounds in the ordering computed in the LSA-based semantic space. The bottom and top of the box represent the first and third quartile ranks, respectively, and the band near the middle of the box represents the median rank. Note that a lower rank value implies better performance.

The performance of unfamiliar compounds still remains at a high level, although lower than the performances of familiar compounds (as compared with Tables 5 and 6). This result demonstrates the ability of semantic space models to comprehend unfamiliar compounds.

As shown in Tables 10 and 11, and in Figures 8 and 9, the relative performances of the composition algorithms do not change noticeably between familiar and unfamiliar compounds. In the LSA-based semantic space, regardless of whether compounds are familiar or unfamiliar, the dilation algorithm shows the best performance (i.e., lowest first quartile rank in all cases and lowest median rank in the case of all pairs). Only two algorithms, dilation and comparison, perform better than the baseline regardless of language and type of pair. On the PPMI-based

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

Table 11. *Median rank of words semantically related to unfamiliar noun compounds in the ordering computed in the PPMI-based semantic space*

Algorithm	All pairs		Emergent pairs	
	English	Japanese	English	Japanese
Centroid	1,801.5 (4)	2,999.0 (4)	2,745.0 (4)	4,438.0 (2)
Multiplication	1,357.0 (1)	2,900.0 (3)	2,691.0 (2)	4,485.0 (3)
Circular convolution	6,540.5 (7)	10,675.0 (7)	6,594.5 (7)	11,163.0 (7)
Predication	2,123.0 (6)	3,433.0 (6)	2,681.5 (1)	5,056.0 (5)
Comparison	1,976.5 (5)	3,360.0 (5)	2,850.5 (5)	5,558.0 (6)
Dilation	1,684.5 (3)	2,659.0 (1)	2,730.0 (3)	4,360.0 (1)
Head noun	1,612.5 (2)	2,850.0 (2)	3,114.5 (6)	4,678.0 (4)

Note. Underlined numbers indicate the lowest median rank values (i.e., the best performance) Numbers in parentheses denote the ranks of the algorithms in ascending order of the median rank.

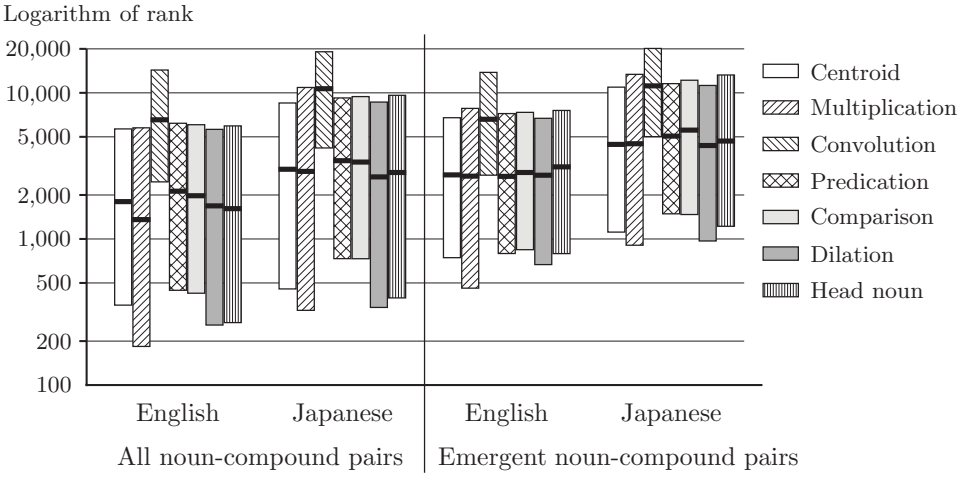


Fig. 9. Boxplots of the rank of semantically related words to familiar noun compounds in the ordering computed in the PPMI-based semantic space. The bottom and top of the box represent the first and third quartile ranks, respectively, and the band near the middle of the box represents the median rank. Note that a lower rank value implies better performance.

semantic space, the multiplication algorithm also performs consistently better than the other algorithms in terms of first quartile rank. Among the other algorithms, the dilation algorithm performs best as in the case of familiar pairs. The consistency of these results suggests that the results obtained are intrinsic to vector composition algorithms.

However, it must be noted that some notable differences are observed between the results of familiar and unfamiliar compounds. When the LSA-based space is used, the comparison algorithm achieves a lowest median rank for emergent pairs and a lowest first quartile rank for Japanese emergent pairs. This result may suggest that the comparison algorithm is more appropriate for representing emergent

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

Table 12. *Correlation coefficients between human similarity ratings and the cosine similarity computed by the composition algorithms for Japanese unfamiliar compounds*

Algorithm	All pairs				Emergent pairs			
	LSA		PPMI		LSA		PPMI	
Centroid	.319*	(5)	.381*	(2)	.002	(6)	.261	(2)
Multiplication	.005	(7)	.414**	(1)	.303	(2)	.223	(4)
Circular convolution	.347*	(4)	.123	(7)	.183	(4)	.121	(5)
Predication	.308	(6)	.346*	(4)	−.019	(7)	.292	(1)
Comparison	.416**	(2)	.312*	(5)	.304	(1)	.241	(3)
Dilation	.431**	(1)	.360*	(3)	.173	(5)	.107	(6)
Head noun	.411**	(3)	.260	(6)	.221	(3)	.046	(7)

Note. Underlined numbers indicate the highest coefficients among the composition algorithms. Numbers in parentheses denote the ranks of the algorithms in descending order of the correlation coefficients.
† $p < .10$. *$p < .05$. **$p < .01$.

meanings of unfamiliar compounds. Another difference is that the performance of the multiplication algorithm in the PPMI-based space seems to be lower for unfamiliar compounds; the dilation algorithm has a lower median rank than the multiplication algorithm for Japanese pairs, and the predication algorithm has a lower median rank for English emergent pairs.

Similarity correlation. Table 12 shows the correlation coefficients between human similarity rating and cosine similarity computed in both semantic spaces.

The results of the correlation analysis for unfamiliar noun compounds are also essentially the same as the results for familiar compounds; when all compound-word pairs are considered, the three composition algorithms (i.e., centroid, comparison, and dilation) are significantly correlated with human judgment in both semantic spaces, and the multiplication algorithm is significantly correlated only in the PPMI-based space. In addition, the dilation algorithm outperforms all other algorithms in the LSA-based semantic space, while the multiplication algorithm outperforms all other algorithms in the PPMI-based space. One difference is that the predication algorithm does not have a significant correlation with human judgment for unfamiliar compounds, although it has for familiar compounds.

Concerning emergent pairs, the result that no algorithms achieve a significant correlation with human judgment is common to both familiar and unfamiliar compounds, suggesting that emergent meanings of compounds are more difficult to represent in a semantic space model. In both spaces, the comparison algorithm has a higher correlation for unfamiliar compounds than for familiar compounds, while the dilation algorithm has a lower correlation for unfamiliar compounds. The multiplication algorithm shows an unexpectedly higher correlation in the LSA-based space.

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

Table 13. *Mean average precision of semantic classification for unfamiliar compounds using the LSA-based semantic space*

Algorithm	All pairs		Emergent pairs	
	English	Japanese	English	Japanese
Centroid	.9433 (4)	.9031 (4)	.9192 (4)	.9240 (4)
Multiplication	.7187 (7)	.6382 (7)	.7296 (7)	.6790 (7)
Circular convolution	.8402 (6)	.8102 (6)	.8104 (6)	.8119 (6)
Predication	.9411 (5)	.9024 (5)	.9216 (3)	.9232 (5)
Comparison	.9584 (2)	.9344 (1)	.9372 (1)	.9531 (1)
Dilation	.9601 (1)	.9337 (2)	.9329 (2)	.9460 (2)
Head noun	.9580 (3)	.9276 (3)	.9115 (5)	.9311 (3)

Note. Underlined numbers indicate the highest MAP values among the composition algorithms. Numbers in parentheses denote the ranks of the algorithms in descending order of the MAP values.

Semantic classification. Tables 13 and 14 respectively show the results (i.e., mean average precision) of the semantic classification test for unfamiliar compounds using the LSA-based and PPMI-based semantic spaces.

Overall performance of unfamiliar compounds also remains at a high level, although it is worse than the performance of familiar compounds. As in the case of the word ranking test, the relative performance of the composition algorithms does not differ greatly between familiar and unfamiliar compounds. The dilation algorithm yields consistently better performance in the LSA-based space, and the multiplication algorithm yields the best performance in the PPMI-based space. However, one notable difference is that the comparison algorithm performs best in more cases than for familiar compounds when the LSA-based space is used; the comparison algorithm performs best for Japanese compounds or emergent compounds. Taken together with the results of the word ranking test, this result suggests that the comparison algorithm may be more appropriate for deriving the meanings of unfamiliar compounds, which are neither explicitly nor implicitly involved in the semantic space, from the semantic knowledge of the constituent words involved in the semantic space.

5 Discussion

5.1 Implications of the experimental results

In this section, we discuss the characteristics of each composition algorithm that are implied from the results of the evaluation experiment.

5.1.1 Dilation algorithm and predication algorithm

The evaluation experiment demonstrated that the dilation algorithm shows consistently better performance than the other vector composition algorithms for

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

Table 14. *Mean average precision of semantic classification for unfamiliar compounds using the PPMI-based semantic space*

Algorithm	All pairs		Emergent pairs	
	English	Japanese	English	Japanese
Centroid	.9129 (4)	.8683 (4)	.8649 (5)	.8441 (3)
Multiplication	.9339 (1)	.8931 (1)	.8908 (1)	.8722 (1)
Circular convolution	.6716 (7)	.5101 (7)	.6469 (7)	.5012 (7)
Predication	.9029 (6)	.8449 (6)	.8581 (6)	.8274 (6)
Comparison	.9066 (5)	.8650 (5)	.8671 (4)	.8397 (5)
Dilation	.9221 (2)	.8780 (2)	.8747 (2)	.8512 (2)
Head noun	.9220 (3)	.8777 (3)	.8728 (3)	.8437 (4)

Note. Underlined numbers indicate the highest MAP values among the composition algorithms. Numbers in parentheses denote the ranks of the algorithms in descending order of the MAP values.

computing the attributional similarity of noun compounds. This finding is consistent with the finding reported by Mitchell and Lapata (2010) stating that the dilation algorithm performed consistently well when computing the similarities between noun compounds, as well as between adjective–noun combinations and between verb–object combinations. This finding suggests that an appropriate compound vector can be computed by moderately modifying the head vector. This also explains why the predication algorithm does not perform well, although it embodies the head–modifier interaction. The interaction in the predication algorithm is more modifier-directed and thus the predication algorithm is less likely to capture the relevance between the meanings of a noun compound and its head noun. (Hence, it can be presumed that the predication algorithm may be suitable for noun compounds in which the modifier plays a more important role in determining the compound meaning than the head noun.)

5.1.2 Multiplication algorithm

The multiplication algorithm, a simple multiplicative model with no free parameters, is found to perform quite differently depending on the type of semantic spaces; it performs best in the PPMI-based semantic space, while it performs worst in the LSA-based space. One possible reason for this sharply contrasting result lies in the meaningfulness of vector dimensions or components in a semantic space. The function of the component-wise multiplication is to highlight dimensions relevant to both vectors and downplay irrelevant dimensions. For example, in the case of the hypothetical vectors in (9), component-wise multiplication of the two vectors $\boldsymbol{h} = (0, 10, 4, 1, 6)$ and $\boldsymbol{m} = (1, 8, 4, 4, 0)$ results in $(0, 80, 16, 4, 0)$, in which the second dimension is highlighted and the first and fifth dimensions are downplayed (or zeroed). Hence, if the dimensions of a semantic space represent distinctive features or meanings, component-wise multiplication is expected to behave as a meaningful

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

function for the semantic interaction between words. In the PPMI-based semantic space, each component of the word vectors has a clear meaning; the jth component represents the number of co-occurrence with the context word w_j, giving the semantics for component-wise multiplication of two word vectors. If two constituent words of a compound are likely to co-occur with the same context word, that compound is also likely to occur with that context word. If either of the constituent words do not co-occur with a certain context word, that compound is also unlikely to occur with that context word. Therefore, as demonstrated in Section 4, when the PPMI-based semantic space is used, the multiplication algorithm performs much better than the other algorithms for vector composition. On the other hand, if vector dimensions do not represent clear meanings or distinctive features in its own right, component-wise multiplication seems not to work as intended. The LSA-based semantic space corresponds to this case; dimensions of the low-dimensional space reduced by SVD do not have a clear semantics. For example, the fact that the ith component of an LSA-based vector $\boldsymbol{x}$ is zero does not necessarily imply that the ith component of a composition vector of $\boldsymbol{x}$ and $\boldsymbol{y}$ should be zero. In other words, we cannot judge whether the ith component of the composition vector should be zero or not, unless the ith dimension represents a certain distinctive feature or meaning. Therefore, the multiplication algorithm performs much worse than the other composition algorithms in the LSA-based semantic space. Note that our results for the multiplication algorithm are consistent with the findings reported by Mitchell and Lapata (2010) finding that the multiplication algorithm performed best on a simple semantic space, because their semantic space was based on the word–context frequency matrix used in the PPMI-based space of this paper. (They did not use a PPMI weighting scheme, but both their and our spaces were not reduced by SVD and, thus, both dimensions had clear semantics.)

In order to justify our argument that the poor performance of the multiplication algorithm in the LSA-based space is due to the lack of dimension semantics, we conducted two additional experiments. In the first experiment, we compared the performance of the multiplication algorithm among eight semantic spaces, which were obtained from all combinations of two initial matrices (i.e., word–document matrix or word–word matrix), two weighting schemes (i.e., the product of log word frequency and the entropy defined in (1) and PPMI defined in (7)), and whether or not a matrix is smoothed by SVD. This experiment aims at testing whether the observed performance difference between the LSA-based and PPMI-based spaces is due to SVD smoothing or otherwise, because the initial matrix and the weighting scheme are also different between these two spaces. Table 15 shows the result of the word ranking test when eight semantic spaces were used for vector composition. Note that the combination of a word–document matrix, a logtf-entropy weighting method, and the presence of SVD smoothing (i.e., first row of Table 15) corresponds to the LSA-based space used for the evaluation experiment in Section 4, while the combination of a word–word matrix, a PPMI weighting method, and the absence of SVD smoothing (i.e., last row of Table 15) corresponds to the PPMI-based space. Table 15 clearly shows that SVD smoothing greatly reduces the performance of the multiplication algorithm regardless of the initial matrix and the weighting scheme.

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

Table 15. *Comparison of the eight semantic spaces including the LSA-based and PPMI-based spaces in terms of median rank of words semantically related to familiar noun compounds*

Semantic Space	Multiplication		Centroid	
	English	Japanese	English	Japanese
Word–document matrix				
logtf-entropy				
with SVD (LSA)	17,074.5	18,711.0	2,925.0	1,534.0
without SVD	2,087.0	1,675.0	1,251.0	870.5
ppmi				
with SVD	18,277.5	15,561.5	2,425.0	1,402.0
without SVD	1,915.5	1,642.0	1,163.0	837.5
Word–word matrix				
logtf-entropy				
with SVD	7,072.5	8,354.0	3,973.5	3,296.5
without SVD	3,602.5	2,285.5	4,788.0	4,429.5
ppmi				
with SVD	11,793.5	27,051.5	1,179.0	592.5
without SVD (PPMI)	796.5	408.5	1,237.0	679.5

Note. logtf-entropy = the product of the logarithm of the word frequency and the entropy defined in (1); ppmi = positive PMI defined in (7).

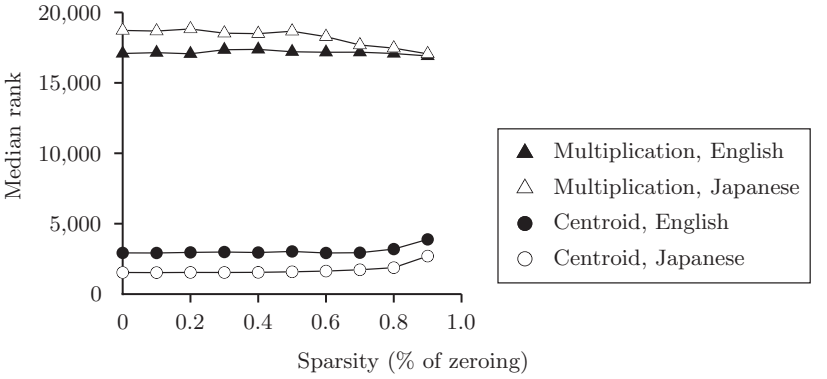


Fig. 10. Zeroing effect on the performance (i.e., median rank for familiar compounds) of the LSA-based semantic space.

Therefore, it can be concluded from the result that the poor performance of the multiplication algorithm in the LSA-based space is attributed to SVD smoothing.

In SVD smoothing, an original word vector is projected onto the low-dimensional reduced space, and as a result an initial sparse matrix usually becomes dense. If a word vector has many zero dimensions, the multiplication algorithm seems to work well because it can downplay such the dimensions as irrelevant semantic information. Hence, it is uncertain whether the poor performance in the LSA-based space is due to the loss of dimension semantics (caused by dimensionality reduction of SVD

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

smoothing) or simply due to sparsity reduction. The second additional experiment was conducted to examine which is the main cause of the poor performance of the multiplication algorithm. If sparsity reduction is the main cause, the performance of the multiplication algorithm would be improved simply by zeroing out small values in LSA-based word vectors. On the other hand, if the poor performance essentially comes from SVD smoothing, such the simple 'zeroing out' method would not improve the performance. In this experiment, we examined whether the performance of the multiplication algorithm is improved by zeroing out elements with smaller absolute values than a certain threshold. Figure 10 shows the median rank of the multiplication algorithm (and the centroid algorithm) in the sparsified LSA-based space where smaller values are zeroed out. In this figure, the horizontal axis denotes the percentage of zeroed elements (in other words, the degree of sparsity), and thus its value of zero indicates the original LSA-based space. The result was that, as shown in Figure 10, the performance of the multiplication algorithm was not improved by zeroing out LSA vectors, although a very slight improvement was observed in the Japanese space. This finding indicates that zeroing has little effect on the performance of the multiplication algorithm. It can thus be concluded that the poor performance of the multiplication algorithm is definitely due to the loss of dimension semantics by SVD smoothing.

5.1.3 Centroid algorithm

The centroid algorithm, a simple additive model, does not perform well overall; its performance is lower than the dilation algorithm. This result seems to suggest that the centroid algorithm is a less useful method for vector composition.

On the other hand, it is also a fact that the centroid algorithm has been widely used in semantic space models. One possible reason may lie in its robustness; as shown in Table 15, the performance of the centroid algorithm is rather stable independent of the type of semantic spaces, as compared to the multiplication algorithm. For example, the centroid algorithm achieves fairly good performance regardless of whether vector dimensions represent distinctive features or not.

5.1.4 Comparison algorithm

The performance of the comparison algorithm also depends on the type of semantic spaces and the compound familiarity. The overall tendency is that the comparison algorithm performs better in the LSA-based space than in the PPMI-based space; for example, the relative performance of the comparison algorithm shown in Tables 5 and 8 is better than that in Tables 6 and 9. In the LSA-based semantic space, the comparison algorithm is more appropriate for representing the meanings of unfamiliar compounds than for representing the meanings of familiar compounds. In some cases (especially in the case of emergent pairs shown in Tables 5, 10, and 13), the comparison algorithm outperforms the dilation algorithm. This finding suggests that common neighbors of the head and modifier,

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

Table 16. *Optimal number of common neighbors estimated in the evaluation experiments using the tenfold cross-validation procedure*

	All pairs	Emergent pairs
English		
Median rank, familiar	7, 5, 5, 10, 3, 7, 9, 10, 10, 6 (7.2)	7, 9, 8, 9, 9, 8, 7, 7, 9, 8 (8.1)
MAP, familiar	1, 1, 1, 1, 1, 1, 1, 1, 1, 1 (1.0)	8, 8, 7, 8, 9, 7, 8, 8, 8, 8 (7.9)
Median rank, unfamiliar	5, 6, 5, 5, 4, 5, 5, 6, 5, 5 (5.1)	10, 9, 8, 7, 10, 10, 10, 10, 10, 8 (9.2)
MAP, unfamiliar	1, 1, 1, 1, 1, 1, 1, 1, 1, 1 (1.0)	10, 10, 10, 10, 10, 10, 10, 10, 10, 10 (10.0)
Japanese		
Median rank, familiar	4, 3, 4, 3, 5, 4, 3, 3, 3, 4 (3.6)	8, 9, 10, 9, 7, 8, 8, 9, 10, 10 (8.8)
MAP, familiar	1, 1, 1, 1, 1, 1, 1, 1, 1, 1 (1.0)	8, 8, 8, 8, 8, 8, 8, 8, 8, 8 (8.0)
Median rank, unfamiliar	1, 1, 1, 1, 1, 1, 1, 1, 1, 1 (1.0)	2, 2, 2, 2, 3, 2, 2, 2, 2, 2 (2.1)
MAP, unfamiliar	1, 1, 1, 1, 1, 1, 1, 1, 1, 1 (1.0)	3, 3, 3, 3, 3, 3, 3, 3, 3, 3 (3.0)

Note. Numbers in parentheses show the average numbers of common neighbors.

which are not taken into account by the dilation algorithm, provide a basis for representing or mining the implicit knowledge (i.e., emergent meanings) behind the corpus. In addition, the better performance for unfamiliar compounds (or emergent meanings) in the LSA-based space may be due to a smoothing effect; extracting the implicit knowledge benefits more from SVD smoothing, and the comparison algorithm may benefit more from the smoothing effect than other algorithms.

The effect of common neighbors on the computation of emergent meanings is also supported by the results of the optimal values of the parameter k (i.e., the number of common neighbors). Table 16 lists the optimal values and their averages of the parameter k estimated by the tenfold cross-validation procedure. This table clearly demonstrates that the optimal number of common neighbors is larger for the emergent pairs than for all pairs. This result indicates that more common neighbors are required for computing the emergent meanings than for computing the non-emergent meanings.

5.1.5 *Circular convolution algorithm*

The circular convolution algorithm performs consistently much worse than the other composition algorithms regardless of language, semantic space, and compound familiarity. This result of circular convolution is not surprising. As mentioned in Section 3.2, circular convolution is simply designed to bind together pairs of vectors so that the resulting vector can be decoded back into the original vectors using its syntactic (or word-order) information; it is not concerned with the process of semantic composition. Therefore, it is likely that the circular convolution algorithm is not adequate as a method for computing the meaning of noun compounds. Note

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

that the experiment of Mitchell and Lapata (2010) yielded the same result for the circular convolution algorithm.

5.1.6 Composition algorithm versus original vector

Interestingly, it is also found that in some cases (e.g., median rank and first quartile rank for English pairs shown in Table 5 and Figure 2) certain composition algorithms (especially the dilation and multiplication algorithms) outperform the non-compositional compound vector constructed directly from the corpus by treating noun compounds as single words. This finding is also consistent with the result of Mitchell and Lapata (2010) that some compositional vectors performed better than the original non-compositional vectors. This finding indicates a high representational ability of the vector composition algorithms, particularly for moderate interactions between the head and modifier. It also suggests that, in general, the semantic space model provides a powerful framework for computational lexical semantics.

5.2 Compound disambiguation and semantic space model

This study has examined the ability of the semantic space model to compute the attributional similarity of a noun compound without considering semantic relations holding between the head and modifier. However, comprehending noun compounds 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 differences would affect the performance of similarity computations.

One promising way of utilizing semantic relations is to use words or phrases expressing the semantic relation (e.g., 'made of' and 'cause') when vectors for constituent words are combined for vector composition. For example, when a noun compound 'apple pie' is used to refer to 'pie made of apple,' the vector representation of 'apple pie' can be reasonably assumed to be the vector of its paraphrase 'pie made of apple.' To compute such a vector, first the predicate vector of 'be made of apples' is computed from the vectors for 'be made of' and 'apple,' and the sentence vector is then computed from the vectors of the argument 'a pie' and the predicate 'be made of apples.' (A similar approach is taken by Kintsch 2008 for solving analogy problems.)

To this end, the process of identifying semantic relations is necessary. As mentioned in Section 1, computational methods for compound disambiguation have been extensively studied, and semantic space models can also be applied to identify semantic relations. For example, a number of studies have attempted to identify semantic relations by computing the attributional similarity between constituent nouns of two noun compounds (e.g., Rosario and Hearst 2001; Rosario, Hearst and Fillmore 2002; Nastase and Szpakowicz 2003; Kim and Baldwin 2005; Ó Séaghdha and Copestake 2009). The rationale behind this method is that constituent words (i.e., head or modifier) of noun compounds involving the same semantic relation are

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

likely to belong to the same category. For example, if one knows that the semantic relation 'made-of' holds true for noun compounds 'apple pie' and 'strawberry jam,' it is quite reasonable to assume that 'orange juice' encodes the same semantic relation, because these head nouns can be classified into the same semantic category 'foods,' and the modifiers can be classified as 'fruits.' A semantic space model can compute the attributional similarity between words, which can be used to classify each constituent word into an appropriate semantic category. Turney (2006) also addresses this problem within the framework of a semantic space model, but his method, the so-called latent relational analysis, computes the relational similarity using a pair–pattern matrix, which is different from a word–document matrix and a word–context matrix used in this study.

6 Conclusion

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

- The dilation algorithm is generally most effective in computing the attributional similarity of noun compounds. This suggests that the dilation algorithm is in general a versatile vector composition algorithm.
- The multiplication algorithm performs quite differently depending on the type of the semantic space. It is best suited for vector composition in PPMI-based (or word co-occurrence-based in general) semantic spaces. This suggests that component-wise multiplication is effective for semantic spaces whose dimensions represent distinctive features or semantics.
- The comparison algorithm is suited for representing the (emergent) meanings of unfamiliar noun compounds in the LSA-based semantic space. Its performance is equal or superior to the performance of the dilation algorithm; this suggests that the comparison algorithm is effective in eliciting the implicit knowledge of compound meanings.
- The predication and centroid algorithms are not, on the whole, superior to the dilation, multiplication, and comparison algorithms. This may be largely due to the lack of the moderate interaction between the head and modifier.
- The circular convolution algorithm shows much worse performance than any of the other composition algorithms. It is not appropriate for semantic composition.
- The relative performances among the vector composition algorithms considered in this study, do not significantly differ between English and Japanese tests. This indicates that the findings obtained are not specific to a particular language; they are likely to be common to any language.

It would be interesting and vital for further work to develop a method for computing the attributional similarities that utilizes semantic relations holding between the head and modifier, as well as a more efficient method for vector

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

composition. Additionally, we are working toward extending this study to longer noun compounds consisting of three or more nouns.

Acknowledgments

This study was supported in part by Grants-in-Aid for Scientific Research C (No. 20500234) and Scientific Research B (No. 23300098) from the Japan Society for the Promotion of Science. I would like to thank three anonymous reviewers for their insightful comments and suggestions, which helped me improve the article. Of course, all remaining errors are of my own responsibility.

References

Baldwin, T., Bannard, C., Tanaka, T., and Widdows, D. 2003. An empirical model of multiword expression decomposability. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, pp. 89–96. Sapporo, Japan: Association for Computational Linguistics.

Baldwin, T., and Tanaka, T. 2004. Translation by machine of complex nominals: getting it right. In *Proceedings of the 2nd ACL Workshop on Multiword Expressions: Integrating Processing*, pp. 24–31. Barcelona, Spain: Association for Computational Linguistics.

Baroni, M., and Zamparelli, R. 2010. Nouns are vectors, adjectives are matrices: representing adjective–noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP2010)*, pp. 1183–93. Cambridge, MA: Association for Computational Linguistics.

Becker, A. 1997. Emergent and common features influence metaphor interpretation. *Metaphor and Symbol* **12**(4): 243–59.

Bergsma, S., Pitler, E., and Lin, D. 2010. Creating robust supervised classifiers via Web-scale N-gram data. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL-10)*, pp. 865–74. Uppsala, Sweden: Association for Computational Linguistics.

Bullinaria, J., and Levy, J. 2007. Extracting semantic representations from word co-occurrence statistics: a computational study. *Behavior Research Methods* **39**(3): 510–26.

Burgess, C., and Lund, K. 1997. Modelling parsing constraints with high-dimensional context space. *Language and Cognitive Processes* **12**: 177–210.

Butnariu, C., Kim, S. N., Nakov, P., Ó Séaghdha, D., Szpakowicz, S., and Veale, T. 2010. SemEval-2010 Task 9: the interpretation of noun compounds using paraphrasing verbs and prepositions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pp. 39–44. Uppsala, Sweden: Association for Computational Linguistics.

Butnariu, C., and Veale, T. 2008. A concept-centered approach to noun-compound interpretation. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008)*, pp. 81–8. Manchester, UK: Association for Computational Linguistics.

Church, K., and Hanks, P. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics* **16**: 22–9.

Clark, E. V., Gelman, S. A., and Lane, N. M. 1985. Compound nouns and category structure in young children. *Child Development* **56**: 84–94.

Costello, F. J., Veale, T., and Dunne, S. 2006. Using WordNet to automatically deduce relations between words in noun–noun compounds. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pp. 160–7. Sydney, Australia: Association for Computational Linguistics.

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

Coulson, S. 2001. *Semantic Leaps: Frame-Shifting and Conceptual Blending in Meaning Construction*. New York: Cambridge University Press.

Foltz, P., Kintsch, W., and Landauer, T. 1998. The measurement of textual coherence with latent semantic analysis. *Discourse Processes* **25**: 285–307.

Gagné, C. 2000. 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**: 365–89.

Gentner, D. 1983. Structure mapping: a theoretical framework for analogy. *Cognitive Science* **7**: 155–70.

Giesbrecht, E. 2009. In search of semantic compositionality in vector spaces. In S. Rudolph, F. Dau, and S. Kuznetsov (eds.), *Conceptual Structures: Leveraging Semantic Technologies (Proceedings of the 17th International Conference on Conceptual Structures, ICCS2009)*, pp. 173–84. Berlin: Springer.

Girju, R., Beamer, B., Rozovskaya, A., Fister, A., and Bhat, S. 2010. A knowledge-rich approach to identifying semantic relations between nominals. *Information Processing and Management* **46**: 589–610.

Girju, R., Moldovan, D., Tatu, M., and Antohe, D. 2005. On the semantics of noun compounds. *Computer Speech and Language* **19**: 479–96.

Girju, R., Nakov, P., Nastase, V., Szpakowicz, S., Turney, P., and Yuret, D. 2007.SemEval-2007 Task 04: classification of semantic relations between nominals. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, pp. 13–8. Prague, Czech Republic: Association for Computational Linguistics.

Girju, R., Nakov, P., Nastase, V., Szpakowicz, S., Turney, P., and Yuret, D. 2009. Classification of semantic relations between nominals. *Language Resources and Evaluation* **43**(2): 105–21.

Griffiths, T., Steyvers, M., and Tenenbaum, J. 2007. Topics in semantic representation. *Psychological Review* **114**: 211–44.

Guevara, E. 2011. Computing semantic compositionality in distributional semantics. In *Proceedings of the 9th International Conference on Computational Semantics (IWCS-9)*, pp. 135–44. Oxford, UK: Association for Computational Linguistics.

Hampton, J. 1987. Inheritance of attributes in natural concept conjunctions. *Memory and Cognition* **15**: 57–71.

Jones, M. N., and Mewhort, D. J. 2007. Representing word meaning and order information in a composite holographic lexicon. *Psychological Review* **114**(1): 1–37.

Jurafsky, D., and Martin, J. H. 2008. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Second Edition*. Upper Saddle River, NJ: Prentice Hall.

Katz, G., and Giesbrecht, E. 2006. Automatic identification of non-compositional multi-word expressions using latent semantic analysis. In *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*, pp. 12–9. Sydney, Australia: Association for Computational Linguistics.

Kim, S., and Baldwin, T. 2005. Automatic interpretation of noun compounds using WordNet similarity. In R. Dale, K. Wong, J. Su, and O. Kwong (eds.), *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP2005)*, pp. 945–56. Berlin: Springer.

Kim, S., and Baldwin, T. 2006. Interpreting semantic relations in noun compounds via verb semantics. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL 2006) Main Conference Poster Sessions*, pp. 491–8. Sydney, Australia: Association for Computational Linguistics.

Kim, S., and Baldwin, T. 2007. Disambiguating noun compounds. In *Proceedings of the 22nd Conference on Artificial Intelligence (AAAI-07)*, pp. 901–6. Vancouver, Canada: AAAI Press.

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

Kim, S. N., and Nakov, P. 2011. Large-scale noun compound interpretation: using bootstrapping and the Web as a corpus. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP2011)*, pp. 648–58. Edinburgh, UK: Association for Computational Linguistics.

Kintsch, W. 2000. Metaphor comprehension: a computational theory. *Psychonomic Bulletin & Review* **7**: 257–66.

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

Kintsch, W. 2008. How the mind computes the meaning of metaphor: a simulation based on LSA. In R. Gibbs (ed.), *The Cambridge Handbook of Metaphor and Thought*, pp. 129–42. New York: Cambridge University Press.

Landauer, T. K., and Dumais, S. T. 1997. A solution to Plato's problem: the latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review* **104**: 211–40.

Landauer, T., McNamara, D., Dennis, S., and Kintsch, W. 2007. *Handbook of Latent Semantic Analysis*. Mahwah, NJ: Lawrence Erlbaum Associates.

Lapata, M., and Keller, F. 2005. Web-based models for natural language processing. *ACM Transactions on Speech and Language Processing* **2**(1): 1–31.

Lauer, M. 1994. Conceptual association for compound noun analysis. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL-94)*, pp. 337–9. Las Cruces, NM: Association for Computational Linguistics.

Lauer, M. 1995. Corpus statistics meet the noun compound: some empirical results. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL-95)*, pp. 47–54. Cambridge, MA: Association for Computational Linguistics.

Manning, C. D., Raghavan, P., and Schütze, H. 2008. *Introduction to Information Retrieval*. Cambridge, UK: Cambridge University Press.

Mitchell, J., and Lapata, M. 2008. Vector-based models of semantic composition. In *Proceedings of 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-08: HLT)*, pp. 236–44. Columbus, OH: Association for Computational Linguistics.

Mitchell, J., and Lapata, M. 2010. Composition in distributional models of semantics. *Cognitive Science* **34**: 1388–429.

Moldovan, D., Badulescu, A., Tatu, M., Antohe, D., and Girju, R. 2004. Models for the semantic classification of noun phrases. In *Proceedings of the Computational Lexical Semantics Workshop at HLT-NAACL'2004*, pp. 60–7. Boston, MA: Association for Computational Linguistics.

Murphy, G. 1988. Comprehending complex concepts. *Cognitive Science* **12**: 529–62.

Nakov, P. 2008. Noun compound interpretation using paraphrasing verbs: feasibility study. In *Proceedings of the 13th International Conference on Artificial Intelligence: Methodology, Systems, Applications (AIMSA'08)*, pp. 103–17. Berlin: Springer.

Nakov, P., and Hearst, M. 2005. Search engine statistics beyond the n-gram: application to noun compound bracketing. In *Proceedings of the 9th Conference on Computational Natural Language Learning (CoNLL)*, pp. 17–24. Ann Arbor, MI: Association for Computational Linguistics.

Nakov, P., and Hearst, M. 2006. Using verbs to characterize noun-noun relations. In *Proceedings of 12th International Conference on Artificial Intelligence: Methodology, Systems, Applications (AIMSA'06), LNCS vol. 4183*, pp. 233–44. Berlin: Springer.

Nakov, P., and Hearst, M. A. 2008. Solving relational similarity problems using the Web as a corpus. In *Proceedings of 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-08: HLT)*, pp. 452–60. Columbus, OH: Association for Computational Linguistics.

Nakov, P., and Kozareva, Z. 2011. Combining relational and attributional similarity for semantic relation classification. In *Proceedings of Recent Advances in Natural Language Processing (RANLP2011)*, pp. 323–30. Hissar, Bulgaria.

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

Nakov, P., Popova, A., and Mateev, P. 2001. Weight functions impact on LSA performance. In *Proceedings of the EuroConference Recent Advances in Natural Language Processing (RANLP2001)*, pp. 187–93. Tzigov Chark, Bulgaria.

Nastase, V., and Szpakowicz, S. 2003. Exploring noun-modifier semantic relations. In *Proceedings of the 5th International Workshop on Computational Semantics (IWCS-5)*, pp. 285–301. Tilburg, Netherlands: Association for Computational Linguistics.

Nelson, D., McEvoy, C., and Schreiber, T. 1998. The University of South Florida word association, rhyme, and word fragment norms. http://www.usf.edu/FreeAssociation/.

Ó Séaghdha, D., and Copestake, A. 2007. Co-occurrence contexts for noun compound interpretation. In *Proceedings of the ACL-07 Workshop on A Broader Perspective on Multiword Expressions*, pp. 57–64. Prague, Czech Republic: Association for Computational Linguistics.

Ó Séaghdha, D., and Copestake, A. 2008. Semantic classification with distributional kernels. In *Proceedings of the 22nd International Conference on Computational Linguisitcs (COLING 2008)*, pp. 649–56. Manchester, UK: Association for Computational Linguistics.

Ó Séaghdha, D., and Copestake, A. 2009. Using lexical and relational similarity to classify semantic relations. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL-09)*, pp. 621–9. Athens, Greece: Association for Computational Linguistics.

Padó, S., and Lapata, M. 2007. Dependency-based construction of semantic space models. *Computational Linguistics* **33**(2): 161–99.

Plate, T. A. 2003. *Holograhic Reduced Representation: Distributed Representation for Cognitive Structures*. Stanford, CA: CSLI Publications.

Quesada, J. 2007. Creating your own LSA spaces. In T. Landauer, D. McNamara, S. Dennis, and W. Kintsch (eds.), *Handbook of Latent Semantic Analysis*, pp. 71–85. Mahwah, NJ: Lawrence Erlbaum Associates.

Recchia, G., and Jones, M. N. 2009. More data trumps smarter algorithms: comparing pointwise mutual information with latent semantic analysis. *Behavior Research Methods* **41**: 647–56.

Reddy, S., McCarthy, D., and Manandhar, S. 2011. An empirical study on compositionality in compound nouns. In *Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP2011)*, pp. 210–8. Chiang Mai, Thailand: Asian Federation of Natural Language Processing.

Rosario, B., and Hearst, M. 2001. Classifying the semantic relations in noun compounds via a domain-specific lexical hierarchy. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP2001)*, pp. 82–90. Pittsburgh, PA: Association for Computational Linguistics.

Rosario, B., Hearst, M., and Fillmore, C. 2002. The descent of hierarchy, and selection in relational semantics. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics (ACL-02)*, pp. 247–54. Philadelphia, PA: Association for Computational Linguistics.

Schone, P., and Jurafsky, D. 2001. In knowledge-free induction of multiword unit dictionary headwords a solved problem? In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP2001)*, pp. 100–8. Pittsburgh, PA: Association for Computational Linguistics.

Schütze, H. 1998. Automatic word sense discrimination. *Computational Linguistics* **24**(1): 97–123.

Tratz, S., and Hovy, E. 2010. A taxonomy, dataset, and classifier for automatic noun compound interpretation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL-10)*, pp. 678–87. Uppsala, Sweden: Association for Computational Linguistics.

Turney, P. D. 2006. Similarity of semantic relations. *Computational Linguistics* **32**(3): 379–416.

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

Turney, P. D., and Pantel, P. 2010. From frequency to meaning: vector space models of semantics. *Journal of Artificial Intelligence Research* **37**: 141–88.
Umemoto, T. 1969. *Renso Kijunhyo (Free Association Norm)*. Tokyo: Tokyo Daigaku Shuppankai.
Utsumi, A. 2005. The role of feature emergence in metaphor appreciation. *Metaphor and Symbol* **20**(3): 151–72.
Utsumi, A. 2011. Computational exploration of metaphor comprehension processes using a semantic space model. *Cognitive Science* **35**(2): 251–96.
Wilkenfeld, M., and Ward, T. 2001. Similarity and emergence in conceptual combination. *Journal of Memory and Language* **45**(1): 21–38.
Wisniewski, E. 1996. Construal and similarity in conceptual combination. *Journal of Memory and Language* **35**: 434–53.
Yamaguchi, T. 2006. *Nihongo Dai-Thesaurus CD-ROM*. Tokyo: Taishukan Shoten.

Appendix: Effect of frequency of noun compounds on the task of semantic classification

Figures 11 and 12 show the MAP values of the semantic classification test for a threshold *tf* ranging from 20 to 400 in steps of 10. These figures are discussed in Section 4.3.2.

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

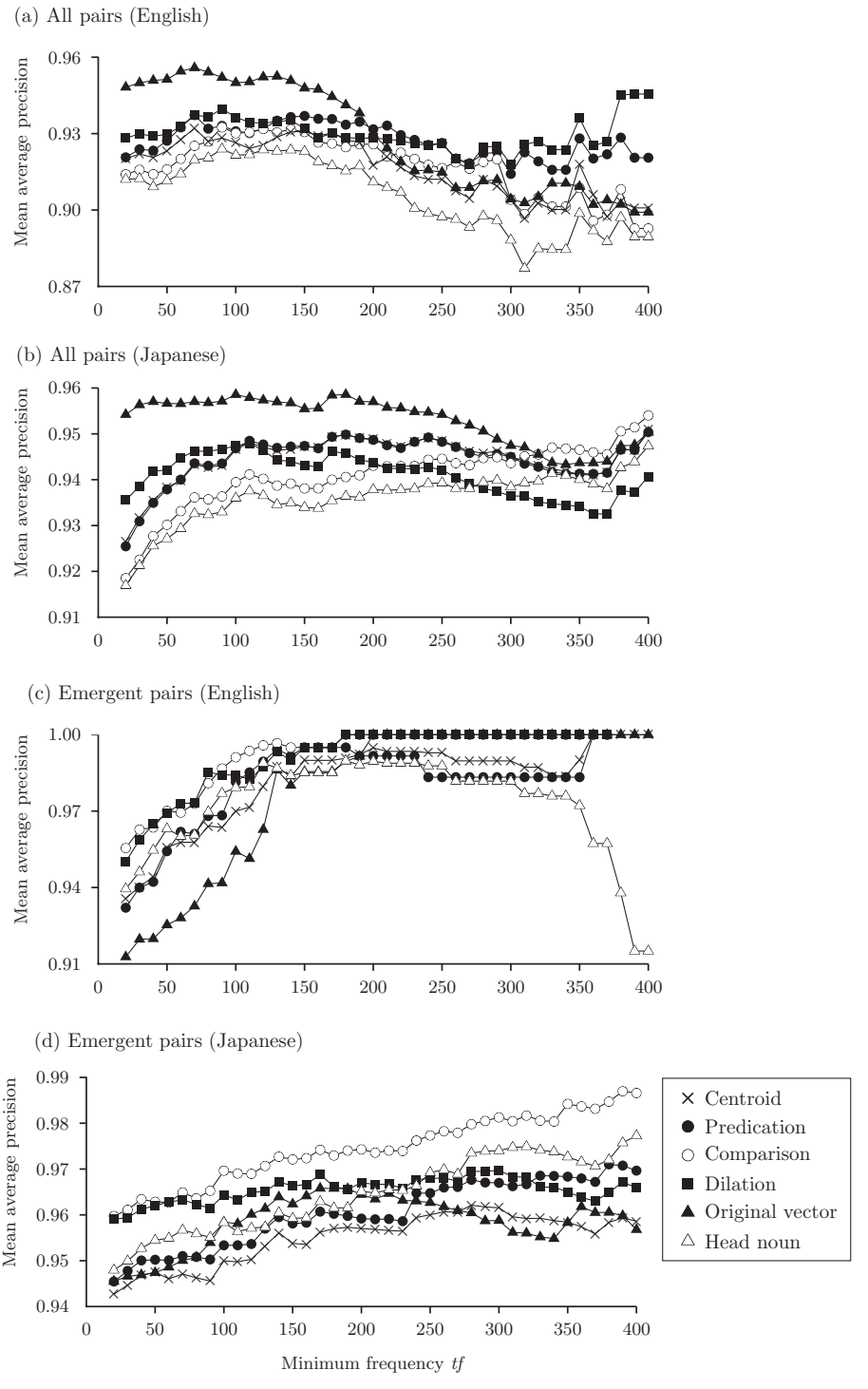


Fig. 11. Mean average precision of semantic classification as a function of minimum frequency of noun compounds in the case of the LSA-based semantic space.

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press

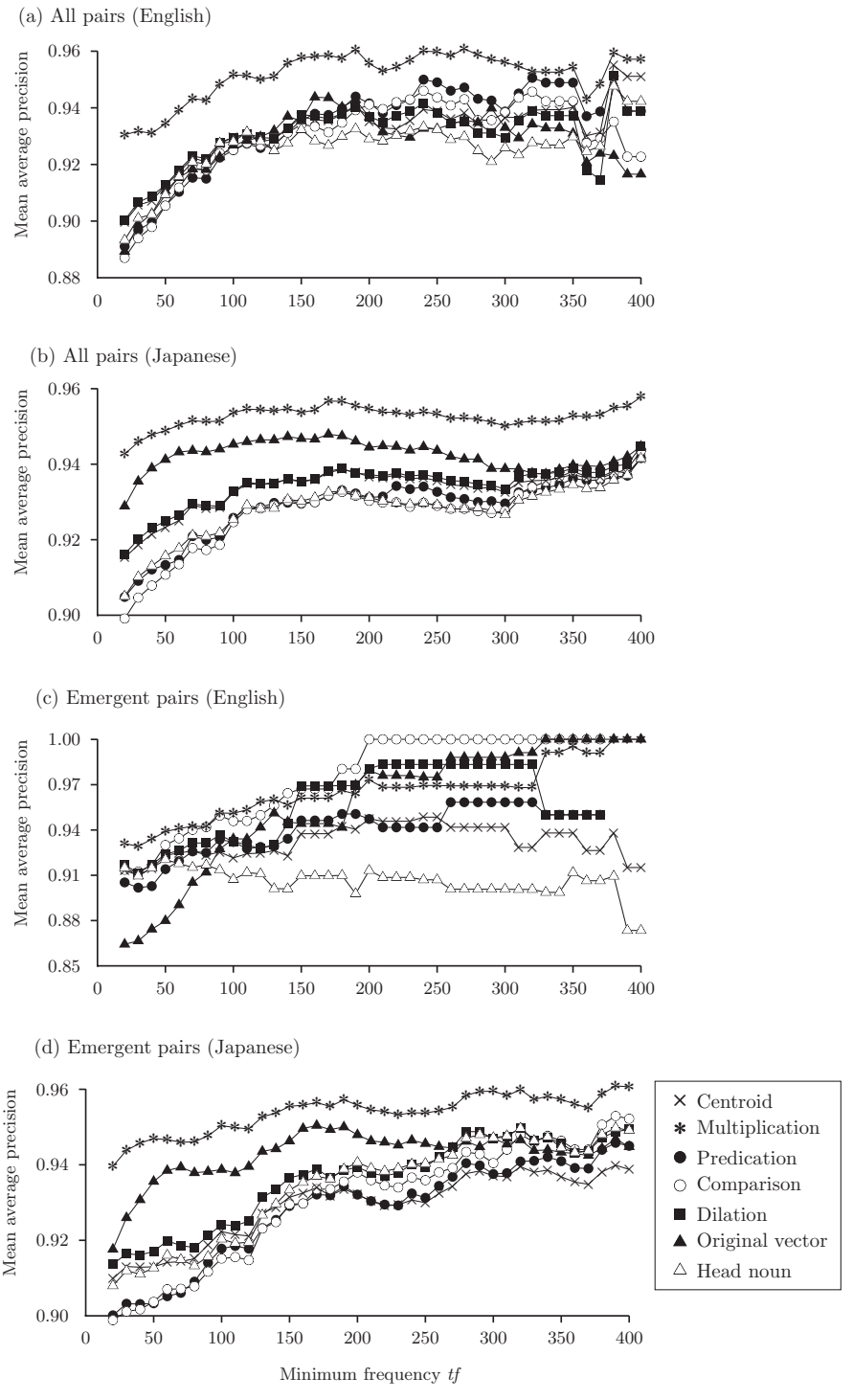


Fig. 12. Mean average precision of semantic classification as a function of minimum frequency of noun compounds in the case of the PPMI-based semantic space.

https://doi.org/10.1017/S135132491200037X Published online by Cambridge University Press