Classifying Lexical-semantic Relationships by Exploiting Sense/Concept Representations

Kentaro Kanada, Tetsunori Kobayashi and **Yoshihiko Hayashi**
Waseda University, Japan
kanada@pcl.cs.waseda.ac.jp
koba@waseda.jp
yshk.hayashi@aoni.waseda.jp

Abstract

This paper proposes a method for classifying the type of lexical-semantic relation between a given pair of words. Given an inventory of target relationships, this task can be seen as a multi-class classification problem. We train a supervised classifier by assuming that a specific type of lexical-semantic relation between a pair of words would be signaled by a carefully designed set of relation-specific similarities between the words. These similarities are computed by exploiting "sense representations" (sense/concept embeddings). The experimental results show that the proposed method clearly outperforms an existing state-of-the-art method that does not utilize sense/concept embeddings, thereby demonstrating the effectiveness of the sense representations.

1 Introduction

Given a pair of words, classifying the type of lexical-semantic relation that could hold between them may have a range of applications. In particular, discovering typed lexical-semantic relation instances is vital in building a new lexical-semantic resource, as well as for populating an existing lexical-semantic resource. As argued in (Boyd-Graber et al., 2006), even Princeton WordNet (henceforth PWN) (Miller, 1995) is noted for its sparsity of useful internal lexical-semantic relations. A distributional thesaurus (Weeds et al., 2014), usually built with an automatic method such as that described in (Rychlý and Kilgarriff, 2007), often comprises a disorganized semantic network internally, where a variety of lexical-semantic relations are incorporated without having proper relation labels attached. These issues could be addressed if an accurate method for classifying the type of lexical-semantic relation is available.

A number of research studies on the classification of lexical-semantic relationships have been conducted. Among them, Necsuleşcu et al. (2015) recently presented two classification methods that utilize word-level feature representations including word embedding vectors. Although the reported results are superior to the compared systems, neither of the proposed methods exploited "sense representations," which are described as *the fine-grained representations of word senses, concepts, and entities* in the description of this workshop[1].

Motivated by the above-described issues and previous work, this paper proposes a supervised classification method that exploits sense representations, and discusses their utilities in the lexical relation classification task. The major rationales behind the proposed method are: (1) a specific type of lexical-semantic relation between a pair of words would be indicated by a carefully designed set of relation-specific similarities associated with the words; and (2) the similarities could be effectively computed by exploiting sense representations.

More specifically, for each word in the pair, we first collect relevant sets of sense/concept nodes (*node sets*) from an existing lexical-semantic resource (PWN), and then compute similarities for some designated pairs of node sets, where each node is represented by an embedding vector depending on its type (sense/concept). In terms of its design, each node set pair is constructed such that it is associated with a specific type of lexical-semantic relation. The resulting array of similarities, along with the underlying word/sense/concept embedding vectors is finally

[1] https://sites.google.com/site/senseworkshop2017/background

Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications, pages 37–46,
Valencia, Spain, April 4 2017.

fed into the classifier as features.

The empirical results that use the BLESS dataset (Baroni and Lenci, 2011) demonstrate that our method clearly outperformed existing state-of-the-art methods (Necsuleşcu et al., 2015) that did not employ sense/concept embeddings, confirming that properly combining the similarity features also with the underlying semantic/conceptual-level embeddings is indeed effective. These results in turn highlight the utility of "the sense representations" (the sense/concept embeddings) created by the existing system referred to as AutoExtend (Rothe and Schütze, 2015).

The remainder of the paper first reviews related work (section 2), and then presents our approach (section 3). As our experiments (section 4) utilize the BLESS dataset, the experimental results are directly compared with that of (Necsuleşcu et al., 2015) (section 5). Although our methods were proved to be superior through the experiments, our operational requirement (sense/concept embeddings should be created from the underlying lexical-semantic resource) could be problematic especially when having to process *unknown* words. We conclude the present paper by discussing future work to address this issue (section 6).

2 Related work

A lexical-semantic relationship is a fundamental relationship that plays an important role in many NLP applications. A number of research efforts have been devoted to developing an automated and accurate method to type the relationship between an arbitrary pair of words. Most of these studies (Fu et al., 2014; Kiela et al., 2015; Shwartz et al., 2016), however, concentrated on the *hypernymy* relation, since it is the most fundamental relationship that forms the core taxonomic structure in a lexical-semantic resource. In comparison, fewer studies considered a broader range of lexical-semantic relations, e.g., (Necsuleşcu et al., 2015) and our present work.

Lenci and Benotto (2012), among the hypernymy-centered researches, compared existing directional similarity measures (Kotlerman et al., 2010) to identify hypernyms, and proposed a new measure that slightly modified an existing measure. The rationale behind their work is: as hypernymy is a prominent asymmetric semantic relation, it might be detected by the higher similarity score yielded by an asymmetric similarity measure. Their idea of exploiting a specific type of similarity to detect a specific type of lexical-semantic relationship is highly feasible.

Recently, distributional and distributed word representations (word embeddings) have been widely utilized, partly because the offset vector simply brought about by vector subtraction over word embeddings can capture some relational aspects including a lexical-semantic relationship. Given these useful resources, Weeds et al. (2014) presented a supervised classification approach that employs a pair of distributional vectors for a given word pair as the feature, arguing that concatenation and subtraction were almost equally effective vector operations. Similar lines of work were presented by (Necsuleşcu et al., 2015) and (Vylomova et al., 2016): the former suggested concatenation might be slightly superior to subtraction, whereas the latter especially highlighted the subtraction. Here it should be noted that Necsuleşcu et al. (2015) employed two kinds of vectors: one is a CBOW-based vector (Mikolov et al., 2013b), and the other involves word embeddings with a dependency-based skip-gram model (Levy and Goldberg, 2014).

The present work exploits semantic/conceptual-level embeddings, which were actually derived by applying the AutoExtend (Rothe and Schütze, 2015) system. Among the recent proposals for deriving semantic/conceptual-level embeddings (Huang et al., 2012; Pennington et al., 2014; Neelakantan et al., 2014; Iacobacci et al., 2015), we adopt the AutoExtend system, since it elegantly exploits the network structure provided by an underlying semantic resource, and naturally consumes existing word embeddings. More importantly, the underlying word embeddings are directly comparable with the derived sense representations. In the present work, we applied the AutoExtend system to the Word2Vec CBOW embeddings (Mikolov et al., 2013b) by referring to PWN version 3.0 as the underlying lexical-semantic resource. As far as the authors know, AutoExtend-derived embeddings have been evaluated in the tasks of similarity measurements and word sense disambiguation: they are yet to be applied to a semantic relation classification task.

There are a few datasets (Baroni and Lenci, 2011; Santus et al., 2015) available that were prepared for the evaluation of lexical-semantic rela-

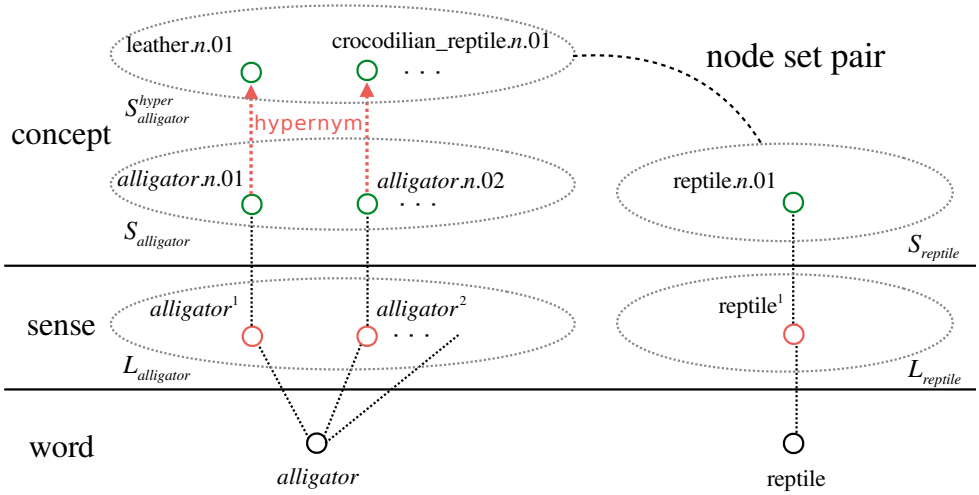

Figure 1: Creating node sets and node set pairs (in *hypernymy* relation).

tion classification tasks. We utilized the BLESS dataset (Baroni and Lenci, 2011) in order to directly compare our proposed method with the related methods given in (Necsuleşcu et al., 2015).

3 Proposed method

We adopt a supervised learning approach for classifying the type of semantic relationship between a pair of words (w_1, w_2), expecting that the plausibility of a specific semantic lexical-relationship can be measured by the similarity between the senses/concepts associated with each of the words.

Figure 1 exemplifies the fundamental rationale behind the proposed method. We assume that the plausibility of the *hypernymy* relation between "alligator" (w_1) and "reptile" (w_2) can be mainly measured by the similarity between the set of hypernym concepts of "alligator" ($S_{w_1}^{hyper}$) and the set of concepts of "reptile" (S_{w_2}). Based on this assumption, we calculate the similarities by the following steps. Recall that these similarities are assumed to measure the plausibilities of relationships that could hold between a given word pair.

1. Collect pre-defined types of node sets for each word (five types; detailed in section 3.1).

2. Build some useful pairs of node sets by considering the possible relationships assumed to be held between the words (7 pair types; detailed in section 3.2).

3. Calculate the similarities for each node set pair by three types of calculation methods (detailed in section 3.3).

In total we calculate 21 (7 pairs $\times$ 3 methods) similarities per word pair along with the cosine similarity between word embeddings. We use these similarities and vector pairs that yielded the similarities as feature.

3.1 Collecting node sets for each word

By consulting PWN, we collect the following five types of node sets for each word. These node set types are selected so as to characterize relevant lexical-semantic relationships in the target inventory detailed in section 4.1.

- L_w: a set of senses that a word w has
- S_w: a set of concepts each denoted by a member of L_w
- S_w^{hyper}: a set of concepts whose member is directly linked from a member of S_w by the PWN `hypernymy` relation
- S_w^{attri}: a set of concepts whose member is directly linked from a member of S_w by the PWN `attribute` relation
- S_w^{mero}: a set of concepts whose member is directly linked from a member of S_w by the PWN `meronymy` relation

3.2 Building node set pairs

Given a pair of words (w_1, w_2), we build seven types of node set pairs as shown in Table 1. Each row in the table defines the combination of node sets and presents the associated mnemonic.

Table 1: Node set pairs built for (w_1, w_2).

Node set pair		Mnemonic
L_{w_1}	L_{w_2}	sense
S_{w_1}	S_{w_2}	concept
$S_{w_1}^{hyper}$	S_{w_2}	hyper
$S_{w_1}^{hyper}$	$S_{w_2}^{hyper}$	coord
$S_{w_1}^{attri}$	S_{w_2}	attri_1
S_{w_1}	$S_{w_2}^{attri}$	attri_2
$S_{w_1}^{mero}$	S_{w_2}	mero

These types of node set pairs are defined in expecting that:

- sense, concept: captures semantic similarity/relatedness between the words;
- hyper: captures *hypernymy* relation between the words;
- coord: dictates if the words share a common hypernym;
- attri_1, attri_2: dictates if w_1 describes some aspect of w_2 (attri_1) or vice versa (attri_2);
- mero: captures the *meronymy* relation between the words.

Note that the italicized words indicate lexical relationships often used in linguistic literature.

3.3 Similarity calculation

In a pair of node sets, each node set could have a different number of elements, meaning that we cannot apply element-wise computation (e.g., cosine) for measuring the similarity between the node sets. We thus propose the following three similarity calculation methods and compare them in the experiments.

In the following formulations: c indicates a certain node set pair type defined in Table 1; (X_{w_1}, X_{w_2}) is the node set pair for (w_1, w_2) specified by c; and $sim(\vec{x_1}, \vec{x_2})$ is the cosine similarity between $\vec{x_1}$ and $\vec{x_2}$.

sim^c_{max} method:

$$sim^c_{max}(w_1, w_2) = \max_{x_1 \in X_{w_1}, x_2 \in X_{w_2}} sim(\vec{x_1}, \vec{x_2}) \quad (1)$$

As the formula defines, this method selects a combination of the node sets that yield the maximum similarity, implying that it achieves a disambiguation functionality.

The vector pair from the most similar node sets $(\vec{x_1}, \vec{x_2})$ is also used as feature. The actual usage of this pair in the experiments is detailed in section 4.2.

sim^c_{sum} method:

$$sim^c_{sum}(w_1, w_2) = sim(\sum_{x_1 \in X_{w_1}} \vec{x_1}, \sum_{x_2 \in X_{w_2}} \vec{x_2}) \quad (2)$$

As defined by the formula, this method firstly makes a holistic meaning representation by summing all embeddings of the nodes contained in each node set. We devised this method with the expectation that it could dictate *semantic relatedness* rather than *semantic similarity* (Budanitsky and Hirst, 2006). The pair of the summed embeddings $(\sum_{x_1 \in X_{w_1}} \vec{x_1}, \sum_{x_2 \in X_{w_2}} \vec{x_2})$ is also used as feature.

sim^c_{med} method:

$$sim^c_{med}(w_1, w_2) = \operatorname{median}_{x_1 \in X_{w_1}, x_2 \in X_{w_2}} sim(\vec{x_1}, \vec{x_2}) \quad (3)$$

The method is expected to express the similarity between mediated representations of each node set. Instead of the arithmetic average, we employ the median to select a representative node in each node set, allowing us to use the associated vector pair as feature.

4 Experiments

We evaluated the effectiveness of the proposed supervised approach by conducting a series of classification experiments using the BLESS dataset (Baroni and Lenci, 2011). Among the possible learning algorithms, we adopted the Random Forest™algorithm as it maintains a balance between performance and efficiency. The results are assessed by using standard measures such as Precision (P), Recall (R), and $F1$. We employed the pre-trained Word2Vec embeddings[2].

[2]300-dimensional vectors by CBoW, available at https://code.google.com/archive/p/word2vec/

We trained sense/concept embeddings by applying the AutoExtend system[3] (Rothe and Schütze, 2015) while using the Word2Vec embeddings as the input and consulting PWN 3.0 as the underlying lexical-semantic resource.

4.1 Dataset

We utilized the BLESS dataset, which was developed for the evaluation of distributional semantic models. It provides 14,400 tetrads of (w_1, w_2, lexical-semantic relation type, topical domain type): where the topical domain type designates a semantic class from the coarse semantic classification system consisting of 17 English concrete noun categories (e.g., tools, clothing, vehicles, and animals). The lexical-semantic relation types defined in BLESS and their counts are described as follows:

- COORD (3565 word pairs): they are co-hyponyms (e.g., alligator-lizard).
- HYPER (1337 word pairs): w_2 is a hypernym of w_1 (e.g., alligator-animal).
- MERO (2943 word pairs): w_2 is a component/organ/member of w_1 (e.g., alligator-mouth).
- ATTRI (2731 word pairs): w_2 is an adjective expressing an attribute of w_1 (e.g., alligator-aquatic).
- EVENT (3824 word pairs): w_2 is a verb referring to an action/activity/happening/event associated with w_1 (e.g., alligator-swim).

Note here that these lexical-semantic relation types are not completely concord with the PWN relations described in section 3.1.

Data division: In order to compare the performance for the present task we divided the data in three ways: *In-domain*, *Out-of-domain* (as employed in (Necsuleşcu et al., 2015)), and *Collapsed-domain*. For the *In-domain* setting, the data in the same domain were used both for training and testing. We thus conducted a five-fold cross validation for each domain. For the *Out-of-domain* setting, one domain is used for testing and the remaining data is used for training. In addition, we prepared the *Collapsed-domain* setting, where we conducted a 10-fold cross validation for the entire dataset irrespective of the domain.

[3]The default hyperparameters were used.

4.2 Comparing methods

A supervised relation classification system referred to as **WECE** (**W**ord **E**mbeddings **C**lassification syst**E**m) in (Necsuleşcu et al., 2015) was especially chosen for comparisons, since this method combines and uses the word embeddings of a given word pair (w_1, w_2) as feature. They compare two types of approaches described as $WECE_{offset}$ and $WECE_{concat}$: $WECE_{offset}$ uses the offset of the word embeddings ($\vec{w_2} - \vec{w_1}$) as the feature vector, whereas $WECE_{concat}$ uses the concatenation of the word embeddings. Moreover, they use two types of word embeddings: a bag-of-words model (BoW) (Mikolov et al., 2013a) and a dependency-based skip-gram model (Dep) (Levy and Goldberg, 2014). In summary, the WECE system has the following variations: $WECE^{offset}_{BoW}$, $WECE^{offset}_{Dep}$, $WECE^{concat}_{BoW}$ and $WECE^{concat}_{Dep}$.

As described in Section 3, we utilize 22 kinds of similarity and the underlying vector as features. In order to make reasonable comparisons, we compare two vector composition methods. In addition to the already described array of similarities, the $Proposal_{concat}$ method uses the concatenated vector of the underlying vectors, whereas the $Proposal_{concat}$ method employs the difference vector. As a result, the dimensionalities of the resulting vectors employed in these methods are 13,222 (22 similarities + 22 $\times$ 600 dimensions for concatenated vectors) and 6,622 (22 similarities + 22 $\times$ 300 dimensions for difference vectors), respectively.

Baseline: As detailed in section 3, our methods utilize PWN neighboring concepts linked by particular lexical-semantic relationships, such as (`hypernymy`, `attribute`, and `meronymy`). We thus set the baseline as follows while respecting the direct relational links defined in PWN.

- Given a word pair (w_1, w_2), if any concept in S_{w_1} and that in S_{w_2} are directly linked by a certain relationship in PWN, let w_1 and w_2 be in the relation.

Note that the baseline method cannot find any word pair that is annotated to have the EVENT relation in the BLESS dataset, because there are no links in PWN that share the same or a similar definition. Likewise it is not capable for the method to find any word pair with the ATTRI

	In-domain			Out-of-domain			Collapsed-domain		
	P	R	$F1$	P	R	$F1$	P	R	$F1$
$WECE^{offset}_{BoW}$	0.900	0.909	0.904	0.680	0.669	0.675	-	-	-
$WECE^{offset}_{Dep}$	0.853	0.865	0.859	0.687	0.623	0.654	-	-	-
$Proposal_{offset}$	0.913	0.907	0.906	0.766	0.762	0.753	0.867	0.867	0.865
$WECE^{concat}_{BoW}$	0.899	0.910	0.904	0.838	0.570	0.678	-	-	-
$WECE^{concat}_{Dep}$	0.859	0.870	0.865	0.782	0.638	0.703	-	-	-
$Proposal_{concat}$	**0.973**	**0.971**	**0.971**	**0.839**	**0.819**	**0.812**	0.970	0.970	0.970

Table 2: Comparison of the overall classification results.

relation in BLESS, because the definition of the `attribute` relationship in PWN differs from the definition of the ATTRI relation in the BLESS dataset. Thus, we can only compare the results for the relationships COORD, HYPER, and MERO with the common measures, P, R, and $F1$.

5 Results

5.1 Major results

Table 2 compares our results with that of the WECE systems in the three data set divisions (In/Out/Collapsed domains). The results show that $Proposal_{concat}$ performed best in all measures of each division (shown in bold font). We observe two common trends across the approaches including WECE: (1) Every score in the Out-of-domain setting was lower than that in the In-domain setting; and (2) The methods using vector concatenation achieved higher scores than those using vector offsets. The former trend is reasonable, since information that is also more relevant to the test data is contained in the training data in the In-domain settings. The latter trend suggests that concatenated vectors may be more informative than offset vectors, supporting the conclusion presented in (Necsuleşcu et al., 2015).

Nevertheless, the results in the table clearly show that $Proposal_{offset}$ outperformed both $WECE^{concat}_{BoW}$ and $WECE^{concat}_{Dep}$ not only in Precision but also in Recall in the Out-of-domain setting. This may confirm that sense representations, acquired by exploiting a richer structure encoded in PWN, are richer in semantic content than word embeddings learned from textual corpora, and hence, even the offset vectors are capable of abstracting some characteristics of potential lexical-semantic relations between a word pair effectively.

Table 3 breaks down the results obtained by $Proposal_{concat}$ in the Out-of-domain setting ($Proposal^{OoD}_{concat}$), and compares them with the $Baseline$ results, showing that $Proposal_{concat}$ clearly outperformed the $Baseline$ in Recall and F1. This clearly confirms that the direct relational links defined in PWN are insufficient for classifying the BLESS relationships. With respect to the internal comparison of the $Proposal^{OoD}_{concat}$ results, a prominent fact is the high-performance classification of the ATTRI and EVENT relationships. By definition, these relationships link a noun to an adjective (ATTRI) or to a verb (EVENT), whereas the COORD, HYPER, and MERO relationships connect a noun to another noun. This may suggest that the information carried by part-of-speech plays a role in this classification task.

Relationship	P	R	F1
COORD	0.761	0.559	0.645
by *Baseline*	*0.550*	*0.108*	*0.180*
HYPER	0.767	0.654	0.706
by *Baseline*	*0.746*	*0.199*	*0.314*
MERO	0.625	0.809	0.705
by *Baseline*	*0.934*	*0.034*	*0.065*
ATTRI	0.913	0.995	0.952
EVENT	0.974	0.983	0.979

Table 3: Breakdown of the results obtained by $Proposal^{OoD}_{concat}$. The $Baseline$ results are shown in *italics*.

Table 4 further details the results obtained by $Proposal^{OoD}_{concat}$ by showing the confusion matrix, also endorsing that the fine-grained classification of inter-noun relationships (COORD, HYPER, and MERO) is far more difficult than distinguishing cross-POS relationships (ATTRI and EVENT). In particular, as suggested in (Shwartz et al., 2016), *synonymy* is difficult to distinguish from *hypernymy* even by humans.

	HYPER	COORD	ATTRI	MERO	EVENT
HYPER	875	157	10	282	13
COORD	189	1994	220	1118	44
ATTRI	1	13	2716	1	0
MERO	74	423	24	2380	42
EVENT	2	32	5	27	3758

Table 4: Confusion matrix for the results by $Proposal^{OoD}_{concat}$ in the Out-of-domain setting.

5.2 Ablation tests

This section more closely considers the results in the Out-of-domain setting. Table 5 shows the results of the ablation tests for the $Proposal^{OoD}_{concat}$ setting, comparing the effectiveness of the source of the similarities. Each row other than $Proposal^{OoD}_{concat}$ displays the results when the designated feature is ablated. The $-word$ row shows the result when ablating the 601-dimensional features created from the pair of word embeddings, and the other rows show the results when ablating the corresponding 1803-dimensional features (three similarities and the three vector pairs that yielded the similarities) generated from each node set pair.

	P	R	$F1$	$F1$ diff
$Proposal^{OoD}_{concat}$	0.839	0.819	0.812	-
$-word$	0.845	0.827	0.819	0.008
$-sense$	0.833	0.815	0.806	-0.006
$-concept$	0.826	0.809	0.802	-0.010
$-coord$	0.834	0.811	0.803	-0.009
$-hyper$	0.826	0.803	0.800	-0.012
$-attri_1$	0.826	0.806	0.798	-0.014
$-attri_2$	0.842	0.820	0.814	0.002
$-mero$	0.835	0.813	0.806	-0.006

Table 5: Ablation tests comparing the effectiveness of each node set.

This table suggests that the $concept$, $hyper$, and $attri_1$ node set pairs are effective, as indicated by the relatively large decreases in $F1$. Surprisingly, however, the features generated from the word embeddings affected the performance. This implies that abstract-level semantics encoded in sense/concept embeddings are more robust in the classification of the target lexical-semantic relationships. However, the utility of sense embeddings was modest.This may result from the learning method in AutoExtend: it tries to split a word embedding into the senses' embeddings without considering the virtual distribution of senses in the Word2Vec training corpus. It is a potential future work to address this issue.

Table 6 compares the effectiveness of the *types* of features. The *only similarities* row shows the results when ablating the vectoral features and only using the 22-dimensional similarity features (21 semantic/conceptual-level similarities along with a word-level similarity). On the other hand, the *only vector pairs* row shows the results from the adverse setting, using the 22 vector pairs (using 13,200-dimensional features).

	P	R	F1	F1 diff
$Proposal^{OoD}_{concat}$	0.839	0.819	0.812	-
only similarities	0.704	0.687	0.683	-0.129
only vector pairs	0.834	0.812	0.804	-0.007

Table 6: Effectiveness comparison of the *types* of features.

It is shown that using vectorial features would produce more accurate results than simply using the similarity features, confirming the general assumption: more features yield more accurate results. However, we would have to emphasize that, even only with the similarity features, our approach outperformed the comparable method in $Recall$ (shown in the Out-of domain columns of Table 2).

Table 7 shows the results of the other ablation tests, comparing the effectiveness of the similarity calculation methods. Each row in the table displays the result when ablating the 4207-dimensional features (seven similarities plus seven vector pairs that yielded these similarities).

As the results in the table show, the $F1$ scores did not change significantly in each ablated condition, showing that the effect provided by the ablated method is completed by the remaining methods. There exists some redundancy in preparing these three calculation methods.

	P	R	$F1$	$F1$ diff
$Proposal^{OoD}_{concat}$	0.839	0.819	0.812	-
$-sim_{max}$	0.835	0.812	0.805	-0.007
$-sim_{sum}$	0.843	0.822	0.816	0.004
$-sim_{med}$	0.838	0.811	0.805	-0.007

Table 7: Additional ablation tests comparing the similarity calculation methods.

6 Discussion

This section discusses two issues: the first is associated with the usage of PWN in the experiments using BLESS, and the other is concerned with the "lexical memorization" problem.

Usage of PWN: As detailed in Section 3, our methods utilize neighboring concepts linked by the particular lexical-semantic relations `hypernymy`, `attribute`, and `meronymy` defined in PWN. Some may consider that the HYPER, ATTRI, and MERO relationships can be estimated simply by consulting the above-mentioned PWN relationships. However, this is definitely NOT the case, since almost all of the semantic relation instances in the BLESS dataset are not immediately defined in PWN: Among the 14,400 BLESS instances, only 951 are defined in PWN. For an obvious example, there are no links in PWN that are labeled `event`, which is a type of semantic relation defined in BLESS. The low Recall results presented in Table 3 endorsed this fact, and clearly show the sparsity of useful semantic links in PWN. However, for some of the lexical-semantic relation types that exhibit transitivity, such as *hypernymy*, consulting the PWN indirect links could be effectively utilized to improve the results.

Lexical memorization: Levy et al. (2015) recently argued that the results achieved by many supervised methods are inflated because of the "lexical memorization" effect, which only learns "prototypical hypernymy" relations. For example, if a classifier encounters many positive examples such as (X, animals), where X is a hyponym of animal (e.g., dog, cat, ...), it only learns to classify any word pair as a *hypernym* as far as the second word is "animal." We argue that our method can be expected to be relatively free from this issue. The similarity features are not affected by this effect, since any similarity calculation is a symmetric operation, and independent of word order. Moreover, the sim^c_{max} or sim^c_{med} method selects a pair of sense/concept embeddings, where the combination usually differs depending on the combination of node sets. On one hand, the sim^c_{sum} method could be affected by the memorization effect, since the vectorial feature for the *prototypical hypernym* is invariable.

7 Conclusion

This paper proposed a method for classifying the type of lexical-semantic relation that could hold between a given pair of words. The empirical results clearly outperformed those previously obtained with the state-of-the-art results, demonstrating that our rationales behind the proposal are valid. In particular, it would be reasonable to assume that the plausibility of a specific type of lexical-semantic relation between the words could be chiefly recovered by a carefully designed set of relation-specific similarities. These results also highlight the utility of "the sense representations," since our similarity calculation methods rely on the sense/concept embeddings created by the AutoExtend system.

Future work could follow two directions. First, we need to improve the classification of inter-noun semantic relations. We may particularly need to distinguish the *hypernym* relationship, which is asymmetric, from the symmetric *coordinate* relationship. In this regard, we would need to improve the creation of node sets and the combinations to capture the innate difference of the relationships.

Second, we need to address the potential drawback of our proposal, which comes from our operational requirement: a lack of sense/concept embeddings is crucial, as we cannot collect relevant node sets in this case. Therefore, we need to develop a method to assign some of the existing concepts to an *unknown* word, which is not contained in PWN, by seeking the nearest concept in the resource. A possible method would first seek the nearest concept in the underlying lexical-semantic resource for an *unknown* word, and then induce a revised set of sense/concept embeddings by iteratively applying the AutoExtend system.

Acknowledgments

The present research was supported by JSPS KAKENHI Grant Number JP26540144 and JP25280117.

References

Marco Baroni and Alessandro Lenci. 2011. How we blessed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 1–10. Association for Computational Linguistics.

Jordan Boyd-Graber, Christiane Fellbaum, Daniel Osherson, and Robert Schapire. 2006. Adding dense, weighted connections to wordnet. In *Proceedings of the third international WordNet conference*, pages 29–36.

Alexander Budanitsky and Graeme Hirst. 2006. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47.

Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning semantic hierarchies via word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1199–1209, Baltimore, Maryland, June. Association for Computational Linguistics.

Eric Huang, Richard Socher, Christopher Manning, and Andrew Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 873–882, Jeju Island, Korea, July. Association for Computational Linguistics.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. Sensembed: Learning sense embeddings for word and relational similarity. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 95–105, Beijing, China, July. Association for Computational Linguistics.

Douwe Kiela, Felix Hill, and Stephen Clark. 2015. Specializing word embeddings for similarity or relatedness. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2044–2048, Lisbon, Portugal, September. Association for Computational Linguistics.

Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. 2010. Directional distributional similarity for lexical inference. *Natural Language Engineering*, 16(04):359–389.

Alessandro Lenci and Giulia Benotto. 2012. Identifying hypernyms in distributional semantic spaces. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 75–79, Montréal, Canada, 7-8 June. Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland, June. Association for Computational Linguistics.

Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. 2015. Do supervised distributional methods really learn lexical inference relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 970–976, Denver, Colorado, May–June. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

George A. Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Silvia Necsuleşcu, Sara Mendes, David Jurgens, Núria Bel, and Roberto Navigli. 2015. Reading between the lines: Overcoming data sparsity for accurate classification of lexical relationships. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 182–192, Denver, Colorado, June. Association for Computational Linguistics.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1059–1069, Doha, Qatar, October. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October. Association for Computational Linguistics.

Sascha Rothe and Hinrich Schütze. 2015. Autoextend: Extending word embeddings to embeddings for synsets and lexemes. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint*

Conference on Natural Language Processing (Volume 1: Long Papers), pages 1793–1803, Beijing, China, July. Association for Computational Linguistics.

Pavel Rychlý and Adam Kilgarriff. 2007. An efficient algorithm for building a distributional thesaurus (and other sketch engine developments). In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 41–44, Prague, Czech Republic, June. Association for Computational Linguistics.

Enrico Santus, Frances Yung, Alessandro Lenci, and Chu-Ren Huang. 2015. Evalution 1.0: an evolving semantic dataset for training and evaluation of distributional semantic models. In *Proceedings of the 4th Workshop on Linked Data in Linguistics (LDL-2015)*, pages 64–69.

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving hypernymy detection with an integrated path-based and distributional method. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2389–2398, Berlin, Germany, August. Association for Computational Linguistics.

Ekaterina Vylomova, Laura Rimell, Trevor Cohn, and Timothy Baldwin. 2016. Take and took, gaggle and goose, book and read: Evaluating the utility of vector differences for lexical relation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1671–1682, Berlin, Germany, August. Association for Computational Linguistics.

Julie Weeds, Daoud Clarke, Jeremy Reffin, David Weir, and Bill Keller. 2014. Learning to distinguish hypernyms and co-hyponyms. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2249–2259, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.