Received: December 18, 2019. Revised: January 14, 2020.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Semantic Relation Detection based on Multi-task Learning and Cross-Lingual-View Embedding

Rizka Wakhidatus Sholikah[1]* Agus Zainal Arifin[1] Chastine Fatichah[1] Ayu Purwarianti[2]

[1]*Informatics Department,*
Faculty of Information and Communication Technology,
Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
[2]*School of Informatics and Electrical Engineering,*
Institut Teknologi Bandung, Bandung, Indonesia
* Corresponding author's Email: rizka16@mhs.if.its.ac.id

Abstract: Semantic relation extraction automatically is an important task in NLP. Various methods have been developed using either pattern-based approach or distributional approach. However, existing research only focuses on single task modeling without considering the possibility of generalization with other tasks. Besides, the methods that exist only use one view from task language as an input representation that might lack of features. This happens especially in languages that are classified as low resource language. Therefore, in this paper we proposed a framework for semantic relations classification based on multi-task architecture and cross-lingual-view embedding. There are two main stages in this framework, data augmentation based on pseudo parallel corpora and multi-task architecture with cross-lingual-view embedding. Further, extensive experiment of the proposed framework has been conducted. The results show that the use of rich resource language in cross-lingual-view embedding is able to support low-resource languages. This is shown by the results with accuracy and F1-scores of 85.8% and 87.6%, respectively. The comparison result also shows that our proposed model outperforms another state-of-the art.

Keywords: Semantic relation, Multi-task learning, Cross-lingual-view embedding, Distributional approach.

1. Introduction

Semantic relation is the relationship that exists between term based on their meaning. Semantic relation resources provide a list of terms and the relation that corresponds to them. The availability of these resources in large quantities can improve the performance of various tasks in information retrieval (IR) and natural language processing (NLP) such as query expansion [1, 2], text categorization [3, 4], taxonomy generation [5, 6], and summarization [7]. However, not every language has this kind of resources in large quantities. Manually create this resource requires a lot of time and effort. Therefore, automatically identify semantic relations is needed.

Automatically identify semantic relations, has long been an important task. The intended semantic relation can be semantic relations at the word level, phrase level, and sentence level. This discussion will focus only for semantic relations at the word level. From existing research, there are two main approach that used to identify semantic relation, pattern-based and distributional-based approaches. Pattern-based has been proposed by Hearst [8], the main idea is to create a lexico-syntactic pattern that is able to detect "is-a" relations or hypernym-hyponym relations. An example of a pattern used is "NP*y* such as NP*x*" where *y* is hypernym of *x* and *x* is hyponym of *y*. This method is one of the influential approaches in detecting semantic relations. In pattern-based, initialization of patterns can be done through manual creation by native or extracted automatically. Other studies that utilize patterns are Snow et al. [9], Simanovsky and Ulanov [10], Nityasya et al. [11] and Roller et al. [12]. Although it produces satisfying results, this approach has weaknesses in sparsity.

This happens because each pair of words have to be in accordance with the available pattern, otherwise there will be no relation detected. Besides this, approach is language specific, which for different languages must have a different pattern. The second category is the distributional approach [13-15]. Distributional-based approach extracts the relation between *x* and *y* based on the representation of their vectors. Current methods utilize word embedding to be used as a vector representation [16-18]. This method is able to overcome problems related to sparsity and language dependence.

However, existing methods focus on modeling a single problem. Like extracting a synonym relation [19-21], extracting a hypernym relation [13, 14, 22, 23], and extracting an antonym relation [21]. This method does binary classification whether a pair of words has a semantic relationship or not. The single model only focuses on a data set about certain problems and does not considered to be generalizing with another problem that has correlation. Research that has been conducted by Santus et al. [24] shows that two tasks can improve each other performance if both has correlation. In this case, semantic relations such as synonym can improve performance for detecting hypernym [24]. Another research also find that co-hyponym can improve hypernym detection [25]. Based on these facts there are several studies that utilize correlation to improve the performance of the model. One of them is the research from Shwartz et al. [26] which does multi-class classification on several semantic relations at a time. This method tries to detect which semantic relation a pair of words has. Other studies conducted classification based on multi-task neural networks [27]. Multi-task architecture can solve more than one problem in one neural network model. The model is created by learning parameters sharing between tasks. In multi-task learning, the features that feed into the network came only from single view representation. In this case, view is interpreted as language. The use of single view can lead to improper representation, especially when dealing with low resource language. In low resource languages, the embedding vectors that commonly use as features come from the collection of documents, which is not as much as in rich resource languages. Lack of training data can reduce the representation quality of the embedding. Low quality of vector embedding leads to poor performance of classifier. Hence it requires collaboration with another view from rich resource language to increase the performance of multi-task learning.

In this research, we introduce a framework for semantic relation classification based on multi-task learning and cross-lingual-view embedding (CLVE). Our proposed CLVE enhanced the embedding representation of the source language (low resource) by adding the embedding of target language (rich resource). Thus, any language that has less data availability on the internet could achieve a good vector representation by using the proposed framework.

The remainder of this paper is organized as follows: in Section 2, we review the related work on automatically identification of semantic relation. In Section 3, we describe our proposed framework in detail. We discuss our experimental results in Section 4. In Section 5, we conclude our paper with a summary.

2. Related work

Semantic relation extraction in general can be divided into two categorize, pattern based approach and distributional approach. Pattern based is an approach that utilizes a certain pattern to extract semantic relations from a free text by matching the pattern and the sentences. Patterns can be obtained manually [8, 12] or automatically [9-11]. Manually extracting patterns are done by expertise or native language by gathering lexico-syntactic pattern that usually forms a relation. While the automatically pattern creation is done by using lexicon seeds of word pairs from a certain relation. Pattern-based research was first popularized by Hearst [8]. This research became a pioneer in extracting relations using patterns. In this research, hypernym-hyponym or "is-a" relation extraction is based on lexico-syntactic pattern, for example "NP*y* such as NP*x*". Another research conducted by Snow does pattern extraction automatically by using lexicon seeds [9]. The intended lexicon seed is a pair of related words (*x*, *y*). Then the pair of words in the lexicon seed will be used to get the pattern from the dependency path. After obtaining the path for all lexicon seeds, the next step is to classify whether *y* is a pair of *x* based on the path collection. The path that is included in the hypernym pattern is the path that gets high weight from the classification. The results of the pattern in Snow's study show that a main pattern that capable of covering many pairs of words is similar to the results of Hearst. In other languages such as Indonesian, research has also been conducted on the extraction of semantic relations based on Hearst-like patterns, one of them is research from Nityasya et al. [11]. This research automatically extracts patterns using lexicon seeds from WordNet. Unlike Snow,

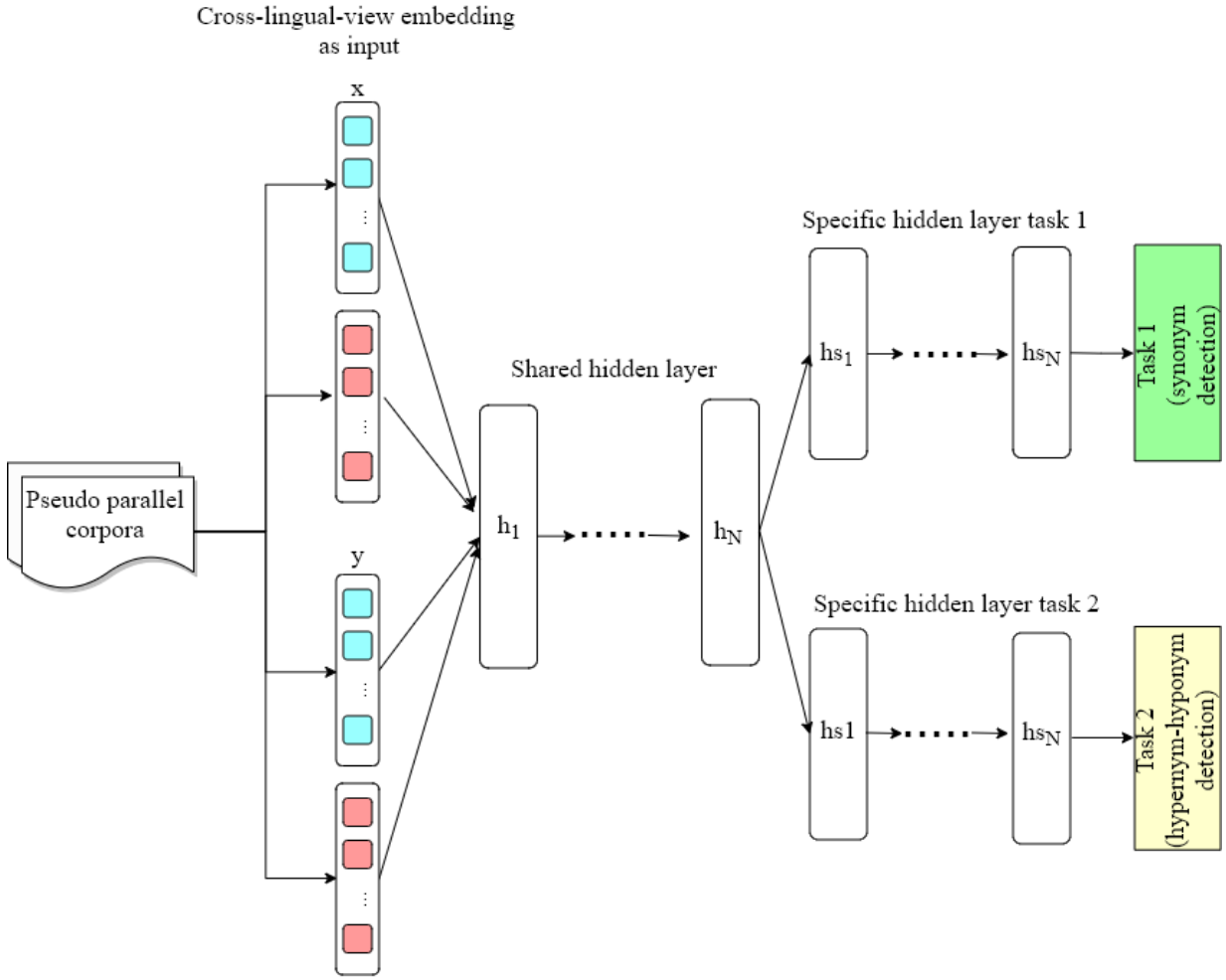

Figure. 1 Proposed framework consists of two parts pseudo parallel corpora and multi-task learning. For multi-task learning *x* and *y* are input that represented by cross-lingual-view embedding, the blue color target embedding and the red color source embedding

this research determines the final pattern based on the confident score of each pattern produced. Besides Indonesian, there are other languages that adopt the Hearst-like pattern to extract semantic relations, including Arabic [29], Chines [6], Turkish [30], etc. As we know that Hearst-like pattern has a weakness in terms of sparsity. Each pair of words must co-occur in a certain pattern otherwise it is considered unrelated. Besides, similar patterns can be written in different forms at the lexical level, so there is a need for generalization. In the latest research, generalization is done by representing the dependency pattern in continuous vectors obtained from LSTM-based neural networks. Various studies have shown that the use of LSTM-based is able to produce satisfactory performance [21, 26]. This is because the resulting pattern has a high level of coverage. However, the use of this pattern based strategy is the language dependent, which for each language must build its own pattern. Development of patterns by naively translating Hearst's patterns that build for English may be inappropriate for certain languages that have different morphology.

Distributional method detects the relation between *x* and *y* based on the distribution representation. Distributional approaches can be classified as supervised and unsupervised. The unsupervised method begins with the distributional similarity method. Distributional similarity utilizes the syntactic dependency-based vector space model. This vector space consists of a matrix with columns in the form of syntactic features and rows in the form of vocabularies. Meanwhile, the value of the matrix is weight (*w*, *f*) which shows the degree of association between word and syntactic features. Another research developed similarity by adding direction and calls it distributional inclusion hypothesis (DIH) [14]. DIH assumes that the context of more general word *y* will have a subset of the context of the more specific word *x*. Recent research related to semantic relations has shifted to a supervised approach. In this approach, the word pairs (*x*, *y*) are represented by vectors, which

are then inputted into the classifier to determine whether *x* and *y* has a relation or not. Several studies conducted the merging of *x* and *y* vectors using various approaches, such as concatenation [31], difference [22], and dot-product. Nowadays, research uses neural word embedding to produce distributional representations of each word for example Mikolov [16]. This approach is easier to build, simple, and produces good results. However, research conducted by Vylomova shows that for some tasks such as hypernym-hyponym more difficult to be modeled [32]. The same goes for the [21] in antonym-synonym distinction task.

However, most of the existing research is only focused on solving single task such as hypernym-hyponym detection [13, 14, 22, 23], synonym detection [19-21] and antonym detection [21]. So it cannot accommodate the use of tasks from other relations to make generalizations. This was answered by research conducted by Shwartz [26], that modeled the task of extracting semantic relations into a multi-class classification problem. Shwartz uses a hybrid model that combines path based methods (pattern-based) with distributional methods. The proposed method improves the pattern quality from the path based using continuous vector based on LSTM network architecture. Then the integration is performed between path based and distributional methods. The results show that the use of hybrid models can produce better performance. Another research from Balikas et al. [27] conducted modeling of the synonym and hypernym-hyponym task using a multi-task architecture. Mukti-task architecture allows shared information between tasks at the shared hidden layer. This makes related tasks able to improve each other's performance.

In this research, a multi-task architecture model is proposed to solve the synonym and hypernym task. Different from Balikas [27], this paper focuses on the hypothesis of using cross-lingual-view embedding to enhance the important features of each word so that it can improve the performance of the classifier. Cross-lingual-view embedding utilizes embedding vectors from languages that are classified as rich resource languages as additional features.

3. Proposed framework

The proposed framework consists of two main process, as shown in Fig. 1. First, pseudo parallel corpora are performed to get augmentation data set. The second step is multi-task learning to classify the relation of the word pairs. The multi-task learning architecture in this paper was built to solve two problems, synonym classification and hypernym-hyponym classification.

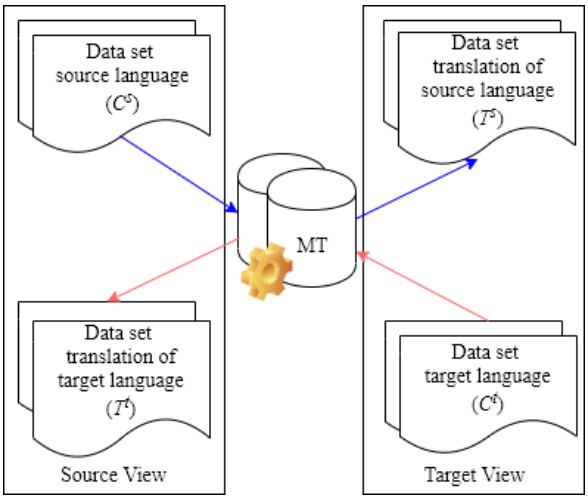

Figure. 2 Pseudo parallel corpora

3.1 Pseudo parallel corpora

In this paper, to enrich the training data, pseudo parallel corpora approach is used. Pseudo parallel corpora or translated corpora is a way to augment data by translating from the source language into the target language. Translation can be done in various ways, such as automatic translation from monolingual corpus, automatic learning translation, and translation using available machine translation. In this study we use google translate as an available machine translation resource to perform translation. Fig. 2 shows an illustration of the pseudo parallel corpora. Let $C^s = \{(x_i^s, y_i^s, rel^s)\}_{i=1}^{i=N}$, $C^t = \{(x_i^t, y_i^t, rel^t)\}_{i=1}^{i=M}$ defines as source corpora and target corpora, respectively. Each pair of words in C^s is translated to the target language producing $T^s = \{(x_{trans_i}^s, y_{trans_i}^s, rel^s)\}_{i=1}^{i=N}$. Similarly in C^t which produce $T^t = \{(x_{trans_i}^t, y_{trans_i}^t, rel^t)\}_{i=1}^{i=M}$ after translation. In pseudo parallel corpora, the result of translation is added to the original data set by concatenation $S^s = (C^s, T^t)$ and $S^t = (C^t, T^s)$. Then all instances of the original data set plus augmented data set are used as input in the training process. Fig. 2 shows that the result produce two kinds of views, the view of the source language and the view of the target language. In this study, the language used as an additional language is a language that originates from rich resource languages, for example, English. This is because English has a large collection of gold standard data sets. Besides

embedding vector representations in English is also more representative.

3.2 Multi-task with cross-lingual-view embedding

The next stage is the process of classifying relations by using multi-task learning. In this research, a multi- task neural network architecture was built to solve two relations classification problems; synonym relations and hypernym-hyponym relations. The multi-task architecture used hard parameter sharing approach. Hard parameter sharing is an approach that utilizes hidden layers as parameter shares among all tasks. In addition, hard parameters are one way to regularize thereby reducing the risk of over fitting. The detail architecture of multi-task learning can be seen in Fig. 1.

In Fig. 1, we introduce CLVE as input of the network. CLVE combines embedding from two views, the source language s and the target language t. The target language is chosen from rich resources language so that the vector representation is better than the source language. The hypothesis of this research is the use of CLVE can enrich the features of each word that are used as input. By enriching important features, each word can be better represented so that it can improve performance. The first step is obtaining CLVE. CLVE can be done by aligning each pair word in the source view and target view. The pairing is simply done by matching their index. Let $\{(x_1^s, y_1^s), (x_2^s, y_2^s), \dots, (x_{(N+M)}^s, y_{(N+M)}^s)\}$ as list of word pairs in source view and $\{(x_1^t, y_1^t), (x_2^t, y_2^t), \dots, (x_{(N+M)}^t, y_{(N+M)}^t)\}$ as list of word pairs in target view, the result of alignment is $\{(x_1^s, x_1^t, y_1^s, y_1^t), (x_2^s, x_2^t, y_2^s, y_2^t), \dots, (x_{(N+M)}^s, x_{(N+M)}^t, y_{(N+M)}^s, y_{(N+M)}^t)\}$. For example, in the source view index 1, there is a word pair (*hewan*, *kucing*) and in the same index in the target view, there is a word pair (animal, cat), then after the alignment we get (*hewan*, animal, *kucing*, cat).

After the results of alignment are obtained, the second step is representing the word into embedding vector. Each word is converted into a vector by looking at the embedding matrix. The embedding matrix is obtained from pre-trained Fasttext with the dimension 300 [28][18]. The embedding representation E_x and E_y then feed into the network as an input. For each task specific $\mathcal{T}$, the concatenation vector embedding $E = [E_x, E_y]$ is mapped into shared hidden layer as shown in Eq. (1).

$$h = f(W^T[E_x, E_y] + b) \quad (1)$$

where $f()$ is rectified linear units (ReLU) activation function, W is weight of hidden layer and b is bias. ReLU is chosen as activation function by considering the computationally efficient compared to other nonlinear activation function, such as Sigmoid. In shared hidden layer, each task can share parameter and exchange information to improve their performance. The next layer is specific hidden layers that used only by the specific task without being influenced by other tasks. The specific hidden layer also consists of non-linear fully connected layer. The input of this layer is the output from shared hidden layer h. For task $\mathcal{T}_j$ the mapping can be seen in Eq. (2).

$$hs_j = f(W^T h + b) \quad (2)$$

The last layer in each task predicts the relationship by using the sigmoid function as written in Eq. (3).

$$f(x) = \frac{1}{1+e^{-x}} \quad (3)$$

In this paper, we used Sigmoid as classifier because our tasks are binary classification. For binary classification, the used of Sigmoid is similar with Softmax.

On hard parameter sharing, the back propagation process updates parameters at the shared hidden layer which is affected by the error rate of all tasks. While at specific hidden layer, the parameter updates only depend on the error of the certain tasks. For all the tasks, we adopt binary cross entropy as the loss function. Whereas RMSprop optimizer is used for learning parameter.

4. Result and discussion

4.1 Data

In this experiment, we use 3 kinds of languages, English (EN), Indonesian (ID), and Arabic (AR). The data set used contains a list of word pairs consisting of 3 classes; synonym relations, hypernym-hyponym relations, and random relations. Table 1 shows an example of the data set used. For English, the RUMEN

Table 1. Statistics of data set

Language	Hypernym-hyponym	Synonym	Random
English (EN)	6,325	6,325	6,325
Indonesian (ID)	8,042	8,890	8,000
Arabic (AR)	4,885	6,080	4,000

Received: December 18, 2019. Revised: January 14, 2020.

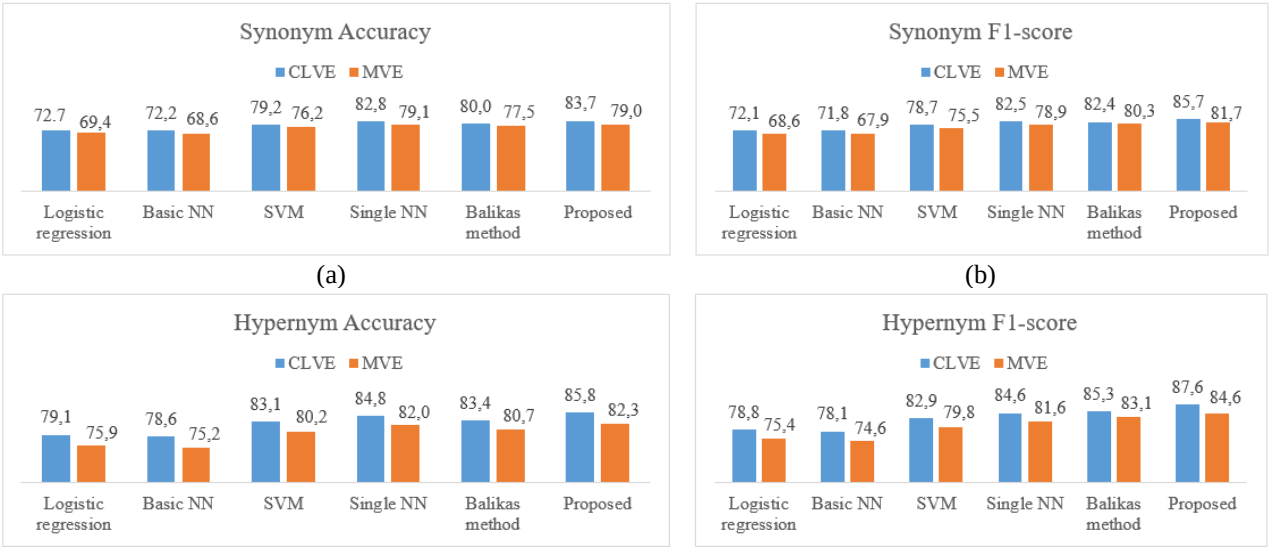

Figure. 3 The comparison between the use of MVE and CLVE as input in several methods: (a) shows the accuracy of synonym task, (b) shows the F1-score of synonym task, (c) shows the accuracy of hypernym task, and (d) shows the F1-score of hypernym task

data set from Balikas [27] is used. Meanwhile for Arabic and Indonesian, we created in-house data sets from universal word net. Table 1 shows the statistics of the data set. However, for Indonesia, we use 8,000 records for all relations, as well as for Arabic, we use 4,000 records for each relation.

In-house data set preparation for ID and AR is done by following the steps below:

1. Using the lemma collection from universal word net. Lemma collection from Indonesian is 106,688 and Arabic is 37,335.
2. Filtering lemmas, selecting lemmas that consist of single term and not beginning with capital letters (i.e. name of location and person).
3. Selecting lemma by randomizing the collection of lemma from filtering results.
4. Looking for a synonym, hypernym-hyponym, and random pair of lemmas. In synonyms, the pair only done using the top-3 of synonyms. In hypernym, we also used top-3 hypernym, while for random is done by randomly selecting word pairs.
5. The result is a pair of words (x, y), where x is a word from the collection of lemma and y is a pair based on universal word net semantic relations.

After collecting all data set, out of vocabulary (OOV) check is performed based on the existing pre-trained word embedding. Word pairs in data sets that are not contained in pre-trained embedding will be removed. Meanwhile, in the proposed method if a word has an embedding in the source language but the results of the translation in the target language are not contained in the target language embedding or vice versa, then the word pairs are also removed. The embedding algorithm used in this paper is Fasttext [18]. Pre-trained word embedding is used for English, Indonesian and Arabic with a number of dimensions of 300 [28].

Table 2. Confusion matrix

		Predicted class	
		Yes	**No**
Actual class	**Yes**	TP	FN
	No	FP	TN

4.2 Matrix evaluation

The performance of the proposed framework is measured by using accuracy and F1-score based on the standard confusion matrix as shown in Table 2. The calculation of accuracy and F1-score can be done by following Eq. (4) and Eq. (5).

$$accuracy = \frac{TP+TN}{(TP+FN+FP+TN)}. \quad (4)$$

$$F1score = 2 \times \frac{(precision \times recall)}{(precision+recall)}. \quad (5)$$

$$precision = \frac{TP}{(TP+FP)}. \quad (6)$$

$$recall = \frac{TP}{(TP+FN)}. \quad (7)$$

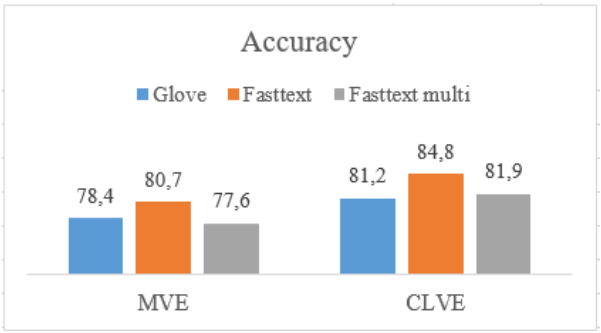

Figure. 4 The comparison of different embedding methods

TP (true positive) is a condition when the classifier correctly predicted the instance as "Yes" and the actual class is "Yes". FN (false negative), when the classifier incorrectly predicts the class as "No", whereas the actual class is "Yes". FP (false positive) is when the classifier incorrectly predicts the class as "Yes", but the actual class is "No". TN (true negative), the classifier correctly predicts the instance as "No" and the actual class also "No".

4.3 Experimental result

In the first scenario, the experiment was conducted to compare the use of mono-view embedding (MVE) and CLVE as input. Comparisons are made by using several methods, including state-of-the art and proposed method. Pseudo parallel corpora of ID-EN, AR-EN and AR-ID are used as data set in this scenario. The result shown in Fig. 3 is the average result of the three data sets. In Fig. 3, it can be seen that for all tasks (synonyms and hypernym) the use of CLVE as input produces better results compared to MVE. In Fig. 3a, we can see that by using CLVE the accuracy of the synonym task increased up to 4.7%. Fig. 3b also shows that the F1 score of each method has improved by an average of 3.4%. In addition, Fig. 3c and Fig. 3d present the accuracy and F1-score of hypernym task. In line with the synonym task, CLVE also can boost performance of hypernym task with the average of accuracy and F1-score, 82.5% and 82.9%, respectively. This can indicate that the additional features of embedding from rich-resource language is able to improve the representation of the intended word. The better the representation of the words used, the better the results of the classification. In this research, embedding representations for source language and target language have the same dimensions, 300. In CLVE, there are no constraints that require dimensions between source and target to be the same.

The second scenario is comparing a variety of word embedding. The comparison is done with global vector (GloVe) [17], Fasttext [18] and Fasttext multi. Fasttext multi is multi-lingual word embedding that project each mono lingual embedding into general vector space [33]. The experiment was conducted using our proposed method with CLVE and without CLVE. The result in Fig. 4 shows that Fasttext gets higher performance compare to GloVe and Fasttext multi. Neither MVE nor CLVE Fasttext gets higher performance with 80.7% and 84.8% each.

The third scenario is done by comparing the use of views on input. In this experiment, the use of view of the source language, views of the source and target languages (CLVE), and view of the target language are compared. Each view is obtained after the pseudo-parallel corpora process. The experiment was conducted using multi-task learning architecture. The data sets used are Indonesian data sets as the source and English data sets as the target. The results of the experiment show that for all tasks, the used of two views can outperform other strategies. This can be seen in Table 3 where the accuracy and F1-score of the CLVE for all tasks are higher than the others. The accuracy obtained for synonyms and hypernym reached 85.7% and 87.5% respectively, while for F1-scores of the synonym and hypernym tasks reached 87.1% and 88.8% respectively. From this result, it can be seen that the use of source view or target view alone produces accuracy values and F1-scores that are almost similar. This shows that the vector representation obtained from embedding from the source language and the target language alone are less able to represent important features. Meanwhile, the result of combining the two views (CLVE) is able to provide additional important information to each input, so that it can improve the quality of the classifier.

The fourth scenario was carried out using a variety of languages. This experiment used Indonesian, Arabic and English data sets. Language testing is done to see the robustness of the method to other data. For Indonesian and Arabic, the original data set is used without using pseudo parallel corpora. Meanwhile in Indonesian + English translation, Arabic + English translation, and Arabic + Indonesian translation the pseudo parallel corpora method is used to produce the final data set. The experiment was conducted using CLVE as input and multi-task learning architecture. The results of the experiment are written in Table 4. Table 4 shows that the used of augmentation data using pseudo-parallel corpora decrease overall performance compared to the used of original data set. The result is consistent for Indonesian and Arabic. Meanwhile, the result in

Table 3. The comparison of different view as input

View	Accuracy		F1-score	
	Synonym	Hypernym	Synonym	Hypernym
Source view	82.0	84.4	84.5	85.8
Source + target view (CLVE)	**85.7**	**87.5**	**87.1**	**88.8**
Target view	83.5	84.3	86.3	86.6

Table 4. The comparison of various data set

Methods	Accuracy		F1-score	
	Synonym	Hypernym	Synonym	Hypernym
Indonesian	87.8	88.5	89.7	90.1
Indonesian-English	85.7	87.5	87.1	88.8
Arabic	85.1	87.8	88.3	89.9
Arabic-English	85.1	8.46	87.3	87.9
Arabic-Indonesian	80.2	83.4	82.5	86.1

Arabic data also shows that the use of rich resource language (English) as target views can produce better result compared to those from low resource language (Indonesia).

The next scenario is done by comparing activation function in shared hidden layer. The comparison is done in single task architecture and multi-task architecture. In this scenario, the architecture has a similar number of shared hidden layer and specific hidden layer (for multi-task learning). The ID-EN data set is used in this experiment with the splitting ratio 0.7 and 0.3 for training and testing, respectively. Table 5 shows that the used of ReLU in hidden layer outperform another activation. We also can see that the used of ReLU effect the performance of neural network in both architecture, single task architecture and multi-task architecture.

The final scenario is comparing the proposed method with the existing state-of-the art methods. In this experiment, the comparison is done with the logistic regression method, support vector machine (SVM), basic NN, single-task NN, multi-task learning from Balika's method [27] and multi-task learning Balikas with CLVE. Logistic regresion and SVM were applied using scikit-learn library. Both of these methods represent methods in the machine learning approach which generally produce good performance in binary classification problems. Basic NN is a simple NN architecture with linear hidden layers. Meanwhile single NN is a single task NN with non-linear hidden layer using ReLU. The architecture is similar with the proposed method but applied only for the single task without specific hidden layer. Basic NN and single NN show the state of the art approach of using a single task architecture on neural networks. Balika's method original is multi task architecture proposed by Balikas [27], whereas Balikas with CLVE is Balika's method original with CLVE as input. Balikas with CLVE aims to show that the proposed CLVE can be applied as input to other architectures. Experiments were conducted on data from the results of pseudo parallel corpora from ID-EN, AR-EN and AR-ID. All methods use the same data set with similar split setting of training and testing with 0.7 and 0.3, respectively. The result that display in Table 6 is the average result from the three data sets. In Table 6, it can be seen that both the synonym and hypernym tasks of the proposed method have higher accuracy and F1-scores compared to other methods. The accuracy of synonyms reaches 83.7% with F1 score 85.7%, while for hypernym the accuracy reaches 85.8% and F1-score 87.6%. The results from Table 6 show that the proposed method outperform the state-of-the art methods in terms of the classification of semantic relations.

4.4 Discussion

The experimental result shows that the proposed framework can overcome the problem of the low-quality representation of low resource language, thereby increasing the performance of multi-task learning. Table 6 shows that the proposed framework is able to outperform the state of the art. The proposed framework as long as we know is the first method that can overcome the problem of low resource language by introducing CLVE strategy. The used of CLVE can put the significant effects to increase the performance. The combination of CLVE and multi-

Table 5. The comparison of different activation function in shared hidden layer

Methods	Accuracy		F1-score	
	Synonym	Hypernym	Synonym	Hypernym
Multi task (ReLU)	85.7	87.5	87.1	88.8
Single task (ReLU)	85.1	86.1	84.9	85.8
Multi task (Sigmoid)	76.5	80.0	80.0	83.3
Single task (Sigmoid)	72.7	78.6	71.7	78.5
Multi task (Linear)	78.4	82.4	81.1	84.3
Single task (Linear)	72.1	77.8	71.4	77.4

Table 6. The comparison with state of the art

Methods	Accuracy		F1-score	
	Synonym	Hypernym	Synonym	Hypernym
Logistic regression	72.7	79.1	72.1	78.8
SVM	79.2	83.1	78.7	82.9
Basic NN	72.2	78.6	71.8	78.1
Single NN	82.8	84.8	82.5	84.6
Multi task Balikas original [27]	77.5	80.7	80.3	83.1
Multi task Balikas w/ CLVE	80.0	83.4	82.4	85.3
Proposed	**83.7**	**85.8**	**85.7**	**87.6**

Table 7. The examples of miss translation from English data set to Indonesian

English Pair		Translation from English to Indonesian		Right translation
catch	collar	*menangkap*	*kerah* (translation error)	*menangkap*
rescue	delivery	*penyelamatan*	*pengiriman* (translation error)	*pembebasan*

task architecture is able to create better classifier compare with state of the art.

In this study, CLVE uses additional embedding of languages classified as rich resource languages, such as English, as an additional feature. This is based on the consideration that embedding from English comes from pre-trained word embedding that is trained on very large amounts of data. Thus the resulting embedding vector will be more representative. The addition of vectors that have a high level of representation is tantamount to adding important features that are able to describe a word better. Thus able to improve the performance of the classifier. This hypothesis is proven in Fig. 3, which shows the use of CLVE can increase the accuracy and F1-score for hypernym and synonym tasks. The use of CLVE also improves performance for both single neural network architectures (Single NN) and multi-task architecture (Balika's method and our proposed).

We have already developed Balika's method [27] using CLVE as input. The result in Table 6 shows that Balika's method gains higher accuracy and F1-scores after applying CLVE. These results indicate that in general CLVE can be applied in various architectural models to improve the predictive results of semantic relations in word pairs. The examples of pair relation that got a wrong label with Balika's method but can be classified correctly by proposed method are shown in Table 8 and Table 9. The analysis of improper word pairs classification shows that in the synonym task 98% that failed to be predicted by Balikas was a random relation, whereas for the hypernym task 95% that failed to predict was the hypernym relation. We also conducted the similar experiment to analyze the result of MVE and CLVE. The examples of relation that has been successfully classified by CLVE are shown in Table 10 and 11. The further analysis found that in synonym task, 80% relation that failed to recognize by MVE is synonym relation. In line with synonym task, hypernym tasks that cannot be classified correctly by MVE but can be classified by CLVE 80% are hypernym relations.

In the process of pseudo parallel corpora, the results of the translation are used directly by adding

Table 8. The example of pair word in synonym task that wrongly classified in Balikas's method but can be correctly classified by proposed method

Indonesian Pair		Translation from Indonesian to English		Relation
luminositas	*kecerahan*	luminosity	brightness	SYN
perhubungan	*perpautan*	nexus	linking	SYN
juri	*panel*	jury	panel	SYN
kembang	*menjernihkan*	flower	clear	RAND
kesimpulan	*lele*	conclusion	catfish	RAND
kalori	*mengamankan*	calories	secure	RAND

Table 9. The example of pair word in hypernym task that wrongly classified in Balikas's method but can be correctly classified by proposed method

Indonesian Pair		Translation from Indonesian to English		Relation
seseorang	*petugas*	somebody	officer	HYPER
bos	*atasan*	boss	boss	HYPER
mamalia	*kelinci*	mammal	rabbit	HYPER
memperbolehkan	*mati*	allow	die	RAND
keagungan	*mengaung*	majesty	roar	RAND
penyewa	*vista*	tenant	vista	RAND

Table 10. The example of pair word in synonym task that wrongly classified by using MVE but can be correctly classified by CLVE

Indonesian Pair		Translation from Indonesian to English		Relation
memeriksa	*pemeriksaan*	check	check	SYN
perusahaan	*usaha*	enterprise	endeavor	SYN
penyokong	*dukungan*	supporters	support	SYN
persahabatan	*rokok*	friendship	cigarette	RAND
kecelakaan	*menaungi*	accident	shade	RAND
kebersihan	*lembab*	courage	humidity	RAND

Table 11. The example of pair word in hypernym task that wrongly classified by using MVE but can be correctly classified by CLVE

Indonesian Pair		Translation from Indonesian to English		Relation
seseorang	*petugas*	somebody	officer	HYPER
kekayaan	*emas*	wealth	gold	HYPER
mamalia	*kelinci*	mammal	rabbit	HYPER
pendatang	*mesra*	newcomer	intimate	RAND
petinju	*banyak*	boxer	plenty	RAND
zaitun	*menusuk*	olives	piercing	RAND

them to the original data set. In the translation process, translation errors from the source language to the target language can occur due to different contexts. The context difference mainly occurs in words that have multiple meanings or commonly called polysemy. This is most likely occurred because we only translate a single word not a sentence, which causes machine translation not to know the exact context of the word. As a result, it can arise from wrong pair of words in the augmentation data set. However, this has not been considered in this study. The pseudo parallel corpora process is carried out without any filtering of the results of the translation before being combined with the original data set. Inaccurate data set results from augmentation can reduce the performance of the system. This can be seen in the results of the experiment in Table 4, where the use of pseudo parallel corpora in Indonesian and

Arabic produces a lower value than the use of the original data set even with a smaller amount. The large amount of data set does not guarantee an increase in the performance of the system if the augmentation is done incorrectly. Some example of translation error that leads to wrong pair can be seen in Table 7. Table 4 also shows that pseudo parallel corpora using rich resource language produce better results compared to low resource language. This is shown in the use of pseudo parallel corpora between Arabic + English and Arabic + Indonesian. In pseudo parallel corpora of Arabic + English CLVE is formed from embedding vectors in Arabic (source) and English (target). Likewise, in Arabic + Indonesian, CLVE is the result of concatenation between Arabic and Indonesian embedding vectors. The results show that the use of English is better than Indonesian both in terms of accuracy and F1-score. This is also shown in both the synonym task and the hypernym task. These results can show that the use of languages that are categorized as rich resource language in CLVE is better than using languages that are both derived from low resource languages.

In additional, Fig. 4 shows that the use of certain vector embedding also affect the overall performance of our proposed. Fasttext multi is one of the models of multi-lingual embedding that projects each mono embedding into general vector space. This algorithm can use to overcome the low representation of embedding that construct from low resource language. However, we could not achieve better results by using Fasttext multi. Our achievement using such method is similar with the one at Upadhyay et al. [34]. In that experiment, the result to measure the quality of vector embedding shows that mono-embedding got higher Qvec than cross-lingual embedding [34].

5. Conclusion

This research proposes multi-task neural network with CLVE to identify semantic relations. Besides that, the data set augmentation is performed using the pseudo parallel corpora method to produce more data sets that has two views, source view and target view. The experimental results show that the proposed framework was able to overcome the problem of low resource language and improve the performance of multi-task learning. This is shown by the results of the accuracy and F1-scores of the proposed method which reached 83.7% and 85.7% in the synonym task, and 85.8% and 87.6% in the hypernym task. Meanwhile the use of pseudo parallel corpora, if done in a naive manner without additional processes such as filtering can reduce the results of the model. In future work, we will perform filtering in pseudo parallel corpora process to reduce mistranslation that caused by context errors. In addition, we will do multi-task learning by utilizing other relationships such as meronym, holonym or co-hyponym.

Acknowledgments

This work was supported by the Ministry of Research, Technology and Higher Education of Republic Indonesia under PMDSU program that enable this joint-research with Hiroshima University.

References

[1] H. K. Azad and A. Deepak, “A New Approach for Query Expansion using Wikipedia and WordNet”, *Information Science*, Vol. 492, pp.147-163, 2019.

[2] O. A. L. Lemos, A. C. Paula, F. C. Zanichelli, and C.V. Lpoes, “Thesaurus-Based Automatic Query Expansion for Interface-Driven Code Search”, In: *Proc. of the 11th Working Conference on Mining Software Repositories*, pp. 212-221, 2014.

[3] S. A. Yousif, V. W. Samawi, I. Elkabani, and R. Zantout, “Enhancement of Arabic Text Classification Using Semantic Relations with Part of Speech Tagger”, *W transactions Advances in Electrical and Computer Engineering*, pp. 195-201, 2015.

[4] T. Vishnu and K. Himakireeti, “Automated Text Clustering and Labeling using Hypernyms”, *International Journal of Applied Engineering Research*, Vol. 14, No. 2, pp. 447-451, 2019.

[5] A. Gupta, R. Lebret, H. Harkous, and K. Aberer, “Taxonomy Induction using Hypernym Subsequences”, In: *Proc. of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 1329-1338, 2017.

[6] C. Wang, Y. Fan, and X. He, “Predicting Hypernym-Hyponym Relations for Chinese Taxonomy Learning”, *Knowledge and Information Systems*, Vol. 58, No. 3, pp. 585-610, 2019.

[7] M. Gambhir and V. Gupta, “Recent automatic text summarization techniques: a survey”, *Artificial Intelligence Review*, Vol. 47, No. 1, pp. 1-66, 2017

[8] M. Hearst, "Automatic Acquisition of Hyponyms from Large Text Corpora", In: *Proc. of the 14th conference on Computational linguistics*, Vol. 2, pp. 539-545, 1992.

[9] R. Snow, D. Jurafsky, and A.Y. Ng, "Learning Syntactic Patterns for Automatic Hypernym Discovery", In: *Proc. of the 17th International Conference on Neural Information Processing Systems*, pp. 1297-1304, 2004.

[10] A. Simanovsky and A. V. Ulanov, "Mining Text Patterns for Synonyms Extraction", In: *Proceedings of the 22nd International Workshop on Database and Expert Systems Applications*, pp. 473-477, 2011.

[11] M. N. Nityasya, R. Mahendra, and M. Adriani, "Hypernym-Hyponym Relation Extraction from Indonesian Wikipedia Text", In: *Proc. of International Conference on Asian Language Processing*, pp. 285-289, 2019.

[12] S. Roller, D. Kiela, and M. Nickel, "Hearst Pattern Re-visited: Automatic Hypernym Detection from Large Text Corpora", In: *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 358-363, 2018.

[13] A. Lenci and G. Benotto, "Identifying Hypernyms in Distributional Semantic Space", In: *Proc. of the first Joint Conference on Lexical and Computational Semantics*, pp. 75-79, 2012.

[14] L. Kotlerman, I. Dagan, I. Szpektor, and M. Z. Geffet, "Directional Distributional Similarity for Lexical Inference", *Natural Language Engineering*, Vol. 16, No. 4, pp. 359-389, 2010.

[15] M. Geffet and I. Dagan, "The Distributional Inclusion Hypotheses and Lexical Entailment", In: *Proc. of the 43rd Annual Meeting on Association for Computational Linguistics*, pp. 107-114, 2005.

[16] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representation in Vector Space", In: *Proc. of the International Conference on Learning Representations*, 2013.

[17] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global Vector for Word Representation", In: *Proc. of the conference on Empirical Methods on Natural Language Processing*, pp. 1532-1543, 2014.

[18] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information", *Transactions of the Association for Computational Linguistics*, Vol. 5, pp. 135-146, 2017.

[19] M. Hagiwara, "A Supervised Learning Approach to Automatic Synonym Identification based on Distributional Features", In: *Proc. of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Student Research Workshop*, pp. 1-6, 2008.

[20] F. Hu, Z. Shao, and T. Ruan, "Self-Supervised Synonym Extraction from the Web", *Journal of Information Science and Engineering*, Vol. 31, pp. 1133-1148, 2015.

[21] K. A. Nguyen, S. S. Walde, and N. T. Vu, "Distinguishing Antonyms and Synonyms in a Pattern-based Neural Network", In: *Proc. of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pp. 76-85, 2005.

[22] J. Weeds, D. Clarke, J. Reffin, D. Weir, and B.Keller, "Learning to Distinguish Hypernym and Co-Hyponym", In: *Proc. of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 2249-2259, 2014.

[23] L. E. Anke, J. C. Collados, C.D. Bovi, and H. Saggion, "Supervised Distributional Hypernym Discovery via Domain Adaptation", In: *Proc. of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 424-435, 2016.

[24] E. Santus, F. Yung, A. Lenci, and C. R. Huang, "EVALution 1.0: An Evolving Semantic Dataset for Training and Evaluation of Distributional Semantic Models", In: *Proc. of the 4th Workshop on Linked Data in Linguistics (LDL) associated to Association for Computational Linguistics and Asian Federation of Natural Language Processing*, pp. 64-69, 2015.

[25] Z. Yu, H. Wang, X. Lin, and M. Wang, "Learning Term Embeddings for Hypernymy Identification", In: *Proc. of the 24th International Conference on Artificial Intelligence*, pp. 1390-1397, 2015.

[26] V. Shwartz, Y. Goldberg, and I. Dagan, "Improving Hypernymy Detection with an Integrated Path-Based and Distributional Method", In: *Proc. of the 54th Annual Meeting of the Association for Computational Linguistics*, Vol. 1, pp. 2389-2398, 2016.

[27] G. Balikas, G. Dias, R. Moraliyski, H. Akhmouch, and M. R. Amini, "Learning Lexical-Semantic Relations Using Intuitive Cognitive Links", In: *Proc. of European Conference on Information Retrieval ECIR 2019. Lecture Notes in Computer Science*, Vol. 11437, pp. 3-18, 2019.

[28] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning word Vectors for 157 Languages", In: *Proc. of the International Conference on Language Resources and Evaluation*, 2018.

[29] M. G. H. Al-Zamil and Q. Al-Radaideh, "Automatic Extraction of Ontological Relations from Arabic Text", *Journal of King Saud University-Computer and Information Sciences*, Vol. 26, pp. 462-472, 2014.

[30] G. Sahin, B. Diri, and T. Yildiz, "Pattern and semantic similarity based automatic extraction of hyponym-hypernym relation from Turkish corpus", In: *Proc. of the 23nd Signal Processing and Communications Applications Conference (SIU)*, pp. 674-677, 2015.

[31] M. Baroni, R. Bernardi, N. Q. Do, and C. C. Shan, "Entailment above the word level in distributional semantics", In: *Proc. of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pp.23-32, 2012.

[32] E. Vylomova, L. Rimell, T. Cohn, and T. Baldwin, "Take and took, gaggle and goose, book and read: Evaluating the utility of vector differences for lexical relation learning", In: *Proc. of the 54th Annual Meeting of the Association for Computational Linguistics*, pp.1671-1682, 2016.

[33] G. Lample, A. Conneau, L. Denoyer, and M. Ranzato, "Unsupervised Machine Translation Using Monolingual Corpora Only", In: *Proc. of the 6th International Conference on Learning Representations ICLR 2018*, 2018.

[34] S. Upadhyay, M. Faruqui, C. Dyer, and D. Roth, "Cross-lingual Models of Word Embeddings: An Empirical Comparison", In: *Proc. of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1661-1670, 2016.