Knowledge Engineering and Data Science (KEDS)
Vol 5, No 2, December 2022, pp. 160–167

pISSN 2597-4602
eISSN 2597-4637

Indonesian Language Term Extraction using Multi-Task Neural Network

Joan Santoso [a,1,*], Esther Irawati Setiawan [a,2], Fransiskus Xaverius Ferdinandus [a,3], Gunawan [a,4], Leonel Hernandez Collantes [b,5]

[a] *Institut Sains dan Teknologi Terpadu Surabaya, Surabaya, Indonesia*
[b] *Institucion Universitaria de Barranquilla IUB, Columbia*
[1] *joan@istts.ac.id*;* [2] *esther@istts.ac.id;* [3] *ferdi@stts.edu;* [4] *gunawan@stts.edu;* [5] *lhernandezc@unibarranquilla.edu.co*
** corresponding author*

ARTICLE INFO

Article history:
Received 29 November 2022
Revised 10 December 2022
Accepted 19 December 2022
Published online 30 December 2022

Keywords:
Term Extraction
Multi-Task
Neural Network
Indonesian Language

ABSTRACT

The rapidly expanding size of data makes it difficult to extricate information and store it as computerized knowledge. Relation extraction and term extraction play a crucial role in resolving this issue. Automatically finding a concealed relationship between terms that appear in the text can help people build computer-based knowledge more quickly. Term extraction is required as one of the components because identifying terms that play a significant role in the text is the essential step before determining their relationship. We propose an end-to-end system capable of extracting terms from text to address this Indonesian language issue. Our method combines two multilayer perceptron neural networks to perform Part-of-Speech (PoS) labeling and Noun Phrase Chunking. Our models were trained as a joint model to solve this problem. Our proposed method, with an f-score of 86.80%, can be considered a state-of-the-art algorithm for performing term extraction in the Indonesian Language using noun phrase chunking.

I. Introduction

The rapid growth of Internet data, mainly text documents, has created a significant opportunity to acquire and store information as computer-based knowledge in our systems. The Internet plays a significant function in human life today. Daily, all information is obtained from the Internet using a computer or mobile device. A portion of knowledge representation is designed to represent data from domain-specific topics. A popular representation of storing information as computer-based knowledge is ontology. Ontology employs a concept and every relationship between concepts to represent knowledge. This computer-based knowledge can be utilized in various Natural Language Processing-related studies, including Question Answering and Dialogue Systems.

The task of term and relation extraction is one approach to addressing this opportunity for ontology development. Ontology has been used in query answering in [1], chatbots in [2], and many other Natural Language Processing topics research areas. Most ontology construction or building is conducted manually, as mentioned [3], and research in [4] describes the costly nature of the ontology building or construction process. Several numerous studies, [3][3][6], are conducted to automate the ontology building or construction process in response to these motivations.

Term extraction is the process of identifying essential terms within a document. Relation extraction is identifying semantic relationships between terms that appear in documents. Several methods have been developed for relation extraction in specific domains, such as the newswire domain in [7] and the biomedical domain in [8]. Most of the research has focused on the relation extraction domain. Meanwhile, our research focuses on the term extraction domain using Phrase Extraction, especially Noun Phrases. Numerous machine learning algorithms are currently employed for phrase extraction from documents. Ramshaw et al. [9] pioneered the noun phrase extraction technique. Maximum Entropy [10], SVM [11], and Memory-Based [12] are just a few of the

https://doi.org/10.17977/um018v5i22022p160-167

methods that have been used to extract English Phrases. In addition, some research has been conducted on extracting phrases from other languages, such as Indonesian [13] and Chinese [14][15].

We attempt to propose a machine learning model as Noun Phrase Chunking for the Indonesian Language based on the Machine Learning previous results. In recent years, joint models have become the algorithm used in many works. Several techniques for extracting the named entity and relation, such as the research in [16], are implemented using the joint model. With the increasing use of joint models as an algorithm in specific tasks, we also proposed combining two neural network models to extract noun phrases from documents.

Our model also incorporates the neural language model as an input representation. Numerous neural language models have been created, including Word2Vec [17] and GLOVE [18]. In addition, several current technological approaches use the neural language model as an input for their system, such as for named entity recognition [19], sentiment classification [20], and end-to-end relation extraction [16]. We will use the word2vec model as our representation because it is one of the most frequently used neural language models in Natural Language Processing research.

Using the joint model as our machine learning model and word2vec as our features, we believe our proposed method has become a novel method for Indonesian Language Noun Phrase Chunking. We advance our previous preliminary research by conducting these models and achieving a superior outcome. We are concentrating our research on term extraction because we believe term extractions play a significant role in this field of study. Numerous relation extraction tasks, such as [21] and [16], include the term extraction process in their procedures. Our research utilizes noun phrase chunking to extract the term because, as defined by Chen in [22], the entity or term in a document is typically described by noun phrases.

Research on noun phrase chunking in Indonesian has been conducted previously [23] and also in our preliminary research [13]. Extending our previous research, we develop an end-to-end system that autonomously extracts noun phrases using two jointly trained multilayer perceptrons. Furthermore, our proposed system integrates the pos tagging into the system as a joint model, as opposed to the conventional approach, which uses the pos tagging as a preprocessing step before the primary process. Details of our works can be explained in several sections. Section 2 will describe our proposed methodology. Section 3 will discuss our experiment scenario and result. Finally, section 5 will discuss our conclusion.

II. Method

This section will discuss the proposed methodology that is used in this study. There are two parts to this section. The first part will discuss the noun phrase in the Indonesian Language. The second part will discuss the Neural Network Multi-Task Models for Noun Phrase Chunking Models. The details of the architecture systems will be shown in Figure 1.

The process will be divided into three parts, first is about the data annotation. The second part is for the model training, and the third part will do the testing phase with some evaluation according to the previous state-of-the-art research. The data annotation process will divide these data into two parts. The first part containing 70% of the data, will be used as a training corpus. The second part of the data containing 30%, will be used for testing the corpus. Several preprocessing tasks are applied to each corpus before the feature extraction process in the training and testing phases.

Tokenization in this task is used to identify each token from sentence extraction results. This process is done by using some regular expressions. Sentence Extraction will be used in this preprocessing task to separate each sentence from the news paragraph. In this research, the sentence extraction process uses a rule created from the Indonesian Language sentence characteristic process. The training process will take the training corpora or dataset to the joint model in the Neural Networks. This process will produce models that will be tested in the testing phase using the testing dataset. Finally, we evaluate the model with the standard evaluation metrics used in the ConLL2000.

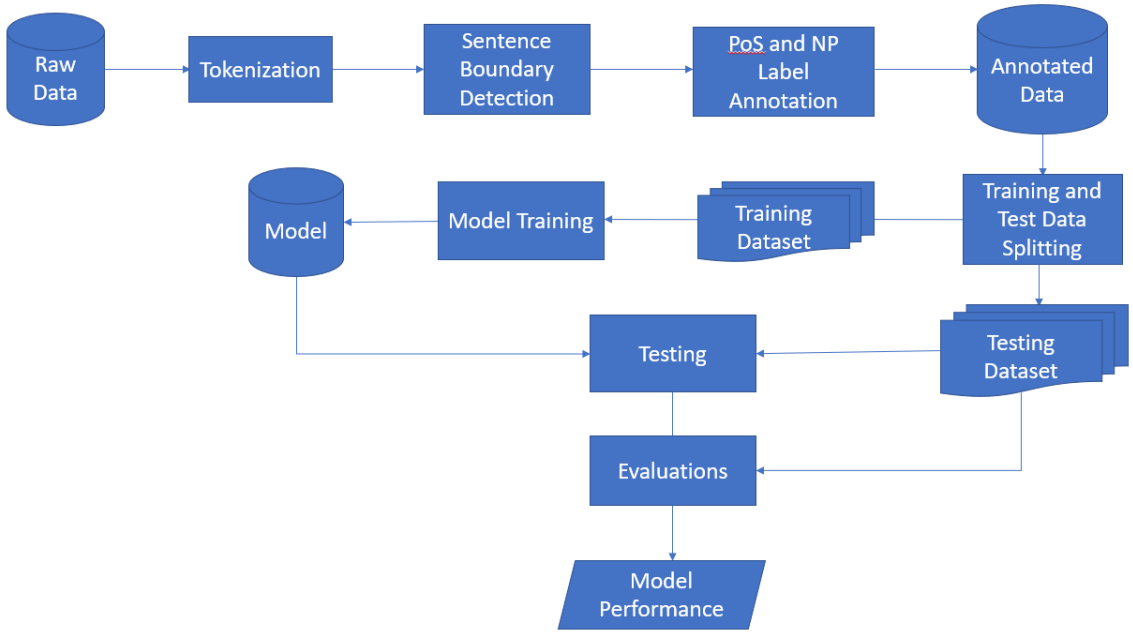

Fig. 1. The system architecture

A. Noun Phrase in the Indonesian Language

Phrases are a collection of words consisting of one word or a combination of two or more words that create a new meaning. A noun phrase is a phrase with a noun as the headword. Indonesian language noun phrase has the same function as English noun phrase.

Noun phrases usually describe a subject or object in a sentence. The difference between English phrases and Indonesian phrase is in grammatical structure according to each language structure. Several examples of Indonesian phrases can be seen in Table 1.

Table 1. Indonesian noun phrases example

No.	Indonesian Sentence	English Sentence
1.	*[Saya] memakan [Nasi Padang]*	[I] eat [Nasi Padang]
2.	*[Kompas] digunakan untuk menentukan [arah mata angin].*	[Compas] is used for determining the [point of compass]
3.	*[Surabaya] adalah ibu kota provinsi [Jawa Timur]*	[Surabaya] is a [East Java] Province capital
4.	*[Jakarta] adalah ibu kota [Indonesia]*	[Jakarta] is the capital of [Indonesia]
5.	*[Apel] dimakan oleh [Andi]*	[Apple] was eaten by [Andi]

We present our dataset in sequential classifier problems. We take each noun phrase's representation into IOB Tagset proposed in [24]. The illustration for IOB Tagging used in this research can be seen in Figure 2. The IOB tag set that we use consists of 3 parts that describe as follows: B-NP denotes the first word of a Noun Phrase, I-NP denotes a non-initial word in a Noun Phrase, and VBT O denotes a word outside of a Noun Phrase.

[NP Kota Surabaya] mengadakan .....

Kota NN B-NP
Surabaya NNP I-NP
mengadakan VBT O

Fig. 2. Dataset and annotation example

The example in Figure 2 consists of three words and one noun phrase. Each word will be labeled each PoS Tagset according to the corresponding part of speech of each word in the sentences. In addition, each word in the phrase will be labeled with the IOB Tag set discussed before. This label is used as an output by the models to identify each noun phrase in this study.

B. Noun Phrases Chunker Model

The model consists of two neural networks. We use a neural language model to represent our word input to the models. Mikolov proposes the neural language model used as a word embedding [17]. Word2Vec model consists of two models, i.e., Skip Gram and CBOW. We use the Skip Gram model because [25] shows that the result from Skip Gram with the negative sampling optimization gives a better result than the CBOW methods.

Our word embedding layer trained the Word2Vec using Indonesian Wikipedia Corpus with a default parameter from Word2Vec with a dimension size of 200. To represent the Part-of-Speech (PoS) in the second neural network embedding Layer, we did not use the pre-trained embedding, but it will be trained together with the model during the learning process. Details of our model illustration can be seen in Figure 3.

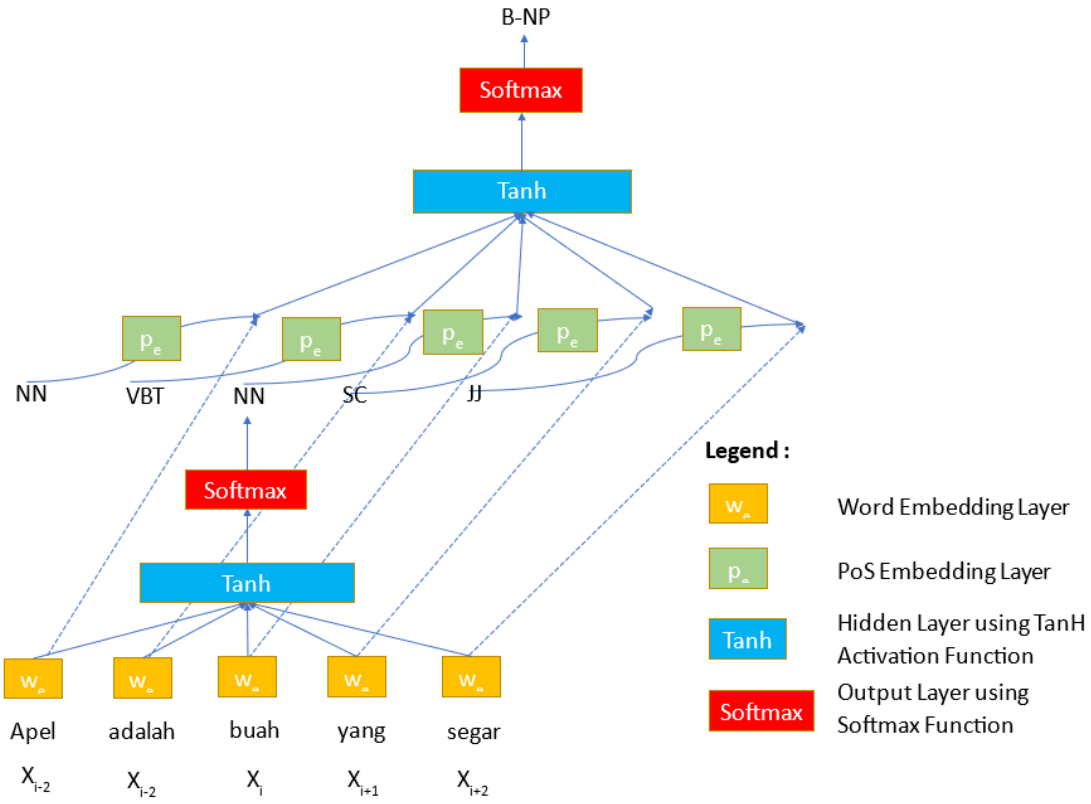

Fig. 3. Neural network architecture example

The output of these models consists of two parts: the part of speech and the phrase labels. For our phrase label representation, we will use 3 type target classes from IOB Tagging for representing the phrase as mentioned in the previous discussion. In addition, we use the Indonesian Language part-of-speech tagset proposed in [26]. We divide our proposed methods into two parts, the first is the pos tagging, and the second one is the noun phrase extraction. Both tasks were trained together as joint multi-task neural network models.

The first part of this model is Part-of Speech Neural Networks. We use this model as a PoS tagging. The feature of our models was using a contextual window with a window size of 2, like in [13]. The feature taken as an input in this model is using the word2vec representation of each word. We concatenate each vector into a large vector before passing it through the systems. Input representation can be seen as in (1).

$$X = [we_{i-2}, we_{i-1}, we_i, we_{i+1}, we_{i+2}] \quad (1)$$

where X defines the input of the neural networks used in the PoS Tagging model. The we_i defines the word embedding lookup from the Word2Vec models for each word input to the neural network. The models take an input of X with a vector size $(2 \text{ x } windows\ size + 1)\ x\ word\ vector\ dimension$. The mathematical model of our PoS Tagging models describes as follows in (2) and (3).

$$h(x)_{pos} = tanh\left(W_h^{pos} X + b_h^{pos}\right) \quad (2)$$

$$y_{pos} = softmax\left(W_{out}^{pos} * h(x)_{pos} + b_{out}^{pos}\right) \quad (3)$$

X is an input of Word Embedding with Contextual Features. Variable $h(x)_{pos}$ It is output from the hidden layer activation function. This activation layer uses the tanh function. W_h^{pos} define the weight of the hidden Layer from the PoS Tagging model. Variable b_{out}^{pos} defines bias from a hidden Layer in the PoS Tagging model. Meanwhile the W_{out}^{pos} defines the weight of the output layer of the PoS Tagging model and the b_{out}^{pos} defines the bias from the output layer in the PoS Tagging model. Output from this Layer will pass through a pos embedding Layer in the Noun Phrase Neural Networks and concatenate it with the word embedding of each word as an input to the Noun Phrase Neural Networks. The mathematical models of how each word represents an input in the Noun Phrase Neural Networks are defined as (4) and (5).

$$X_i^{NP} = [we_i,\ Pe_i] \quad (4)$$

$$X_{NP} = [X_{i-2}^{NP},\ X_{i-1}^{NP}, X_i^{NP}, X_{i+1}^{NP}, X_{i+2}^{NP}] \quad (5)$$

The Noun Phrase Neural Network is a second model to predict the correct phrase labels. The input of this model uses a concatenation between word embedding and PoS embedding. This PoS-embedding Layer will generate a new vector with some specific dimensions of d_{pos}. The dimension size for PoS Embedding in this study is set to 15. We combine all the word embedding vectors in the contextual feature windows with the pos embedding vector as an input in this Layer defined in Eq. (5). The Noun Phrase Neural Networks consist of three layers input layer, a hidden Layer, and an output layer. We will have a X_{NP} As an input to the input layer in the model. This input will have a vector with a length of d_{NP} That can be computed as in (6).

$$d_{NP} = (2 \text{ x } windows\ size\ + 1)\ x\ (word\ embedding\ vector\ dimension + pos\ embedding\ vector\ dimension) \quad (6)$$

The hidden model for the Noun Phrase Neural Network used a TanH activation function, and the output layer used a softmax function to get the correct phrase label in this study. Therefore, the model of our Noun Phrase Neural Networks can be computed as in (7) and (8).

$$h(x)_{NP} = tanh(W_h^{NP} X_{NP} + b_h^{NP}) \quad (7)$$

$$y_{NP} = softmax(W_{out}^{NP} * h(x)_{NP} + b_{out}^{NP}) \quad (8)$$

The variable X_{NP} defines the input of our models taken from Eq. (5), consisting of Word Embedding and PoS Embedding were concatenated together. The $h(x)_{NP}$ is the hidden Layer of Noun Phrase Neural Networks with the activation function of TanH. $W_h^{NP} X_{NP}$ is the weight of the hidden Layer and the b_h^{NP} is the bias from the hidden Layer. The output layer was defined in the variable y_{NP} The SoftMax functions help normalize the output and give the highest probability of the correct output. The W_{out}^{NP} defines the weight from the output layer and b_{out}^{NP} defines the bias for the output layer.

We train both models using the Adam optimizer with a cross-entropy cost function for both joined and trained models. We also use some optimization, such as dropout, introduced in [27] before outputting the hidden Layer pass to the output layer for both models. The dropout probability that is used is 0.5. The cost function that we used to train these models can be seen as in (9), (10), and (11).

$$J_{pos} = -\sum \log (t_{pos} \log (y_{pos})) \quad (9)$$

$$J_{NP} = -\sum \log (t_{NP} \log (y_{NP})) \quad (10)$$

$$J = J_{pos} + J_{NP} \quad (11)$$

variables t_{pos} and t_{NP} define the one hot vector representation of the correct answer of PoS and phrase label. Variables y_{pos} and y_{NP} are the result of the output layer from PoS and noun phrase models. We use this cost function J from (11) with the Adam optimizer to train the joint models.

III. Results and Discussion

We try used data from Indonesia's online News Website taken from [13] as previous research. These data include news from Detik, Vivanews, Surya, and Kompas. We crawled this dataset and manually annotated this dataset using two annotators. Statistics of these data can be seen in Table 2.

Table 2. Corpus statistics

No.	Corpus Dataset	Total News	Total Tokens
1.	Detik Training	208	57374
	Detik Testing	104	26081
2.	Kompas Training	191	51322
	Kompas Testing	83	25489
3.	Surya Training	211	50244
	Surya Testing	91	22123
4.	Vivanews Training	152	66991
	Vivanews Testing	66	21131

To measure how good the model we conducted several experiments. For each dataset, we try several experiments and measure using CoNLL 2000 scoring system using F1-Score to show how robust our proposed system is. The F1-Score used in this study can be calculated as in (12), (13), and (14).

$$Precision = \frac{number\ of\ correct\ chunk\ given\ by\ system}{total\ number\ of\ chunk\ given\ by\ system} \quad (12)$$

$$Recall = \frac{number\ of\ correct\ chunk\ given\ by\ system}{total\ number\ of\ actual\ chunk\ in\ text} \quad (13)$$

$$F1 - score = \frac{2*Recall*Precision}{Precision+Recall} \quad (14)$$

The size of each hidden Layer that is was taken as half of the input size for each model. The first experiment was done to do the chunking task for each corpus. The second task was comparing the best performance the models can achieve with previous research from [13] using C4.5 and [11] SVM.

This section will describe our results. Our experiments are divided into two parts. The first part is experimenting with Indonesian Language Dataset that already describes before using our models. The second part compares our experiments with previous research [13]. The result of the first part can be seen in Table 3, and the second experiment can be seen in Table 4.

Table 3. Indonesian language experiment result

No.	Corpus	NP Chunking F-Score	PoS Tagging Accuracy
1.	Detik	86.80%	87.35%
2.	Kompas	85.59%	87.44%
3.	Surya	85.22%	87.17%
4.	Vivanews	84.72%	86.91%

Table 4. Indonesian language experiment result

No.	Models	F1-Score
1.	C4.5[13]	84.63%
2.	SVM[16]	87.98%
3.	Our Model	86.80%

From Table 3, the first experiments show that the best performance is achieved by Detik corpora with the highest F1-Score, about 86.80%. The views corpora show the lowest performance with F1-Score, about 84.72%. We also evaluate the accuracy of PoS Tagger in this study as one of the outputs from our joint models. The model's highest accuracy was given by Kompas corpora which is 87.44%, and the lowest performance was taken from Vivanews, with a PoS Tagging accuracy of about 86.91%. The second experiments compare our model with the previous state-of-the-art models. The result of the second experiment can be seen in Table 4.

Table 4 shows that our models can improve our previous experiment using C4.5 [13]. However, the result of the state-of-the-art model in Phrase Chunking proposed in [11] shows a better performance with only differences of about 1.18%. Although our model has a lower F1-Score than the state-of-the-art model in [11], we have more advantages to eliminate the need for external tools for acquiring PoS features. For example, our end-to-end models can label the PoS automatically without using external preprocessing tools.

IV. Conclusion

Our proposed methods show that they have an improvement from our preliminary result. Our methods can improve by about 2.17 from our previous research. However, if the model compares with the state-of-the-art model using SVM, our proposed model has a lower F1-Score with a difference of about 1.18%. Although our model has a lower F1-Score than the state-of-the-art models, it has more advantages in eliminating the need for external tools for acquiring PoS features. This research can be a new approach for Noun Phrases Extraction in the Indonesian Language. For further research, we will extend this model using transformer-based approaches and some large pre-trained models to help the noun extraction process. We also plan to integrate these models into the relation extraction system to detect a semantic relation between terms that extract using this system to construct computer knowledge based on ontology.

Declarations

Author contribution

All authors contributed equally as the main contributor of this paper. All authors read and approved the final paper.

Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of interest

The authors declare no known conflict of financial interest or personal relationships that could have appeared to influence the work reported in this paper.

Additional information

Reprints and permission information are available at http://journal2.um.ac.id/index.php/keds.

References

[1] D. S. Wang, "A domain-specific question answering system based on ontology and question templates," in *Software Engineering Artificial Intelligence Networking and Parallel/Distributed Computing (SNPD), 2010 11th ACIS International Conference on*, 2010, pp. 151–156.

[2] H. Al-Zubaide and A. A. Issa, "Ontbot: Ontology based chatbot," in *Innovation in Information & Communication Technology (ISIICT), 2011 Fourth International Symposium on*, 2011, pp. 7–12.

[3] A. D. S. Jayatilaka and G. Wimalarathne, "Knowledge extraction for Semantic Web using web mining," in *Advances in ICT for Emerging Regions (ICTer), 2011 International Conference on*, 2011, pp. 89–94.

[4] B. Abdelbasset, K. Okba, and M. Sofiane, "Agent-based approach for building ontology from text," in *Computer Medical Applications (ICCMA), 2013 International Conference on*, 2013, pp. 1–6.

[5] H. Yang and J. Callan, "Metric-based ontology learning," in *Proceedings of the 2nd international workshop on Ontologies and information systems for the semantic web*, 2008, pp. 1–8.

[6] R. Snow, D. Jurafsky, and A. Y. Ng, "Semantic taxonomy induction from heterogenous evidence," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, 2006, pp. 801–808.

[7] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, "Open Information Extraction from the Web.," in *IJCAI*, 2007, pp. 2670–2676.

[8] C. Giuliano, A. Lavelli, and L. Romano, "Exploiting shallow linguistic information for relation extraction from biomedical literature.," in *EACL*, 2006, pp. 401–408.

[9] L. A. Ramshaw and M. P. Marcus, "Text chunking using transformation-based learning," in *Natural language processing using very large corpora*, Springer, 1999, pp. 157–176.

[10] W. Skut and T. Brants, "A maximum-entropy partial parser for unrestricted text," *arXiv preprint cmp-lg/9807006*, 1998.

[11] T. Kudoh and Y. Matsumoto, "Use of support vector learning for chunk identification," in *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning-Volume 7*, 2000, pp. 142–144.

[12] E. F. Sang, "Memory-based shallow parsing," *Journal of machine learning research*, vol. 2, no. Mar, pp. 559–594, 2002.

[13] J. Santoso, H. V. Gani, E. M. Yuniarno, M. Hariadi, M. H. Purnomo, and others, "Noun phrases extraction using shallow parsing with C4. 5 decision tree algorithm for Indonesian Language ontology building," in *Communications and Information Technologies (ISCIT), 2015 15th International Symposium on*, 2015, pp. 149–152.

[14] H. Li, J. J. Webster, C. Kit, and T. Yao, "Transductive HMM based chinese text chunking," in *Natural Language Processing and Knowledge Engineering, 2003. Proceedings. 2003 International Conference on*, 2003, pp. 257–262.

[15] G.-H. Fu, R.-F. Xu, K.-K. Luke, and Q. Lu, "Chinese text chunking using lexicalized HMMs," in *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, 2005, pp. 7–12.

[16] M. Miwa and M. Bansal, "End-to-end relation extraction using lstms on sequences and tree structures," *arXiv preprint arXiv:1601.00770*, 2016.

[17] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[18] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global Vectors for Word Representation.," in *EMNLP*, 2014, pp. 1532–1543.

[19] S. K. Sienčnik, "Adapting word2vec to named entity recognition," in *Proceedings of the 20th Nordic Conference of Computational Linguistics, NODALIDA 2015, May 11-13, 2015, Vilnius, Lithuania*, 2015, pp. 239–243.

[20] B. Xue, C. Fu, and Z. Shaobin, "A study on sentiment computing and classification of sina weibo with word2vec," in *Big Data (BigData Congress), 2014 IEEE International Congress on*, 2014, pp. 358–363.

[21] P. Pantel and M. Pennacchiotti, "Espresso: Leveraging generic patterns for automatically harvesting semantic relations," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, 2006, pp. 113–120.

[22] K. Chen and H.-H. Chen, "Extracting noun phrases from large-scale texts: A hybrid approach and its automatic evaluation," in *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, 1994, pp. 234–241.

[23] A. A. Arman, A. Purwarianti, and others, "Syntactic phrase chunking for indonesian language," *Procedia Technology*, vol. 11, pp. 635–640, 2013.

[24] E. F. Sang and J. Veenstra, "Representing text chunks," in *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, 1999, pp. 173–179.

[25] Y. Goldberg and O. Levy, "word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method," *arXiv preprint arXiv:1402.3722*, 2014.

[26] A. F. Wicaksono and A. Purwarianti, "HMM based part-of-speech tagger for Bahasa Indonesia," in *Fourth International MALINDO Workshop, Jakarta*, 2010.

[27] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting.," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.