Journal of Engineering and Applied Sciences 12 (24): 7508-7513, 2017
ISSN: 1816-949X

An Extended Model for Semantic Role Labeling Using Word Sense Disambiguation and Dependency Parsing

[1]G. Veena, [2]Lekshmi R. Pillai and [3]Deepa Gupta
[1]Department of Computer Science and Applications,
[2]Department of Computer Science and Engineering, Amrita School of Engineering,
Amrita University, Amrita Vishwa Vidyapeetham, Amritapuri, India
[3]Department of Mathematics, Amrita School of Engineering, Amrita University,
Amrita Vishwa Vidyapeetham, Banglore, India

Abstract: In NLP, the main two fundamental tasks are Word Sense Disambiguation (WSD) and Semantic Role Labeling (SRL). Semantic role labeling is used to label the arguments in the sentence with the help of predicates. Word sense disambiguation is the procedure of predicting the correct definition of a word in a given context. In the research, we improved SRL using WSD and dependency parsing. The dependency parser helps to improve the semantic relationship between the predicates and its arguments. A modified Conditional Random Field (CRF) is used to bind dependency parser with SRL. We have used SVM classifier for WSD and PractNLP tool is used for dependency parser. The model is evaluated and compared with an online WSD with SRL tool. From the results obtained with the aid of our proposals, the labeling performs much better than a tool.

Key words: Word sense disambiguation, dependency parsing, semantic role labeling, conditional random field, WordNet, SVM, PractNLP tool

INTRODUCTION

Semantic role labeling is used to represent sentence-level semantic information in the form of semantic roles. SRL with word sense features is widely used to identify these roles.

Word sense disambiguation: In Word Sense Disambiguation (WSD), the important factor is disambiguation (Soorya *et al*., 2015). Disambiguation serves to remove all ambiguity based on context information. For example, the word “bark” exists in two different meanings like the bark of the dog or the bark of the tree. We have to find the proper sense of a word based on the given context.

Dependency parsing: Dependency Parsing (DP) has been shown to improve NLP systems in certain languages and in many cases is considered the state of the art in the field. The basic idea behind dependency parsing is syntactic structure consists of lexical items, connected by binary asymmetric relations called dependencies. Dependency parsing has of two types, projectivity and non-projectivity. The projectivity means no cross-edges, regenerate the original sentence with the same word orders. The complexity of projectivity is O(n) or O(n^2). PractNLP tool is a kind of projectivity type. Dependencies are generated by pattern-matching rules.

Semantic role labeling: Semantic Role Labeling (SRL) is also called Shallow Semantic Parsing. Based on SRL, we can assign roles for each word in the sentence (Jurafsky and Martin, 2015). The core arguments of SRL are the agent(arg0), patient(arg1), instrument(arg2), source, destination, etc. The agent is the volitional causer of an event, the instrument is used in an event is instrument label and the patient is the experiencer of an event.

Literature review: Kim and Baldwin (2013) and Kolte and Bhirud (2008) reports an adaption of WSD in various algorithms. The algorithms are divided into supervised algorithm, unsupervised algorithm and semi-supervised algorithm. The essential objective is to disambiguate the word sense of component words in NCs, based on investigation of semantic collocation between them. The evaluation methods explain how WSD works in these resources and concluded with which one is efficient with accuracy. It has tested with small dataset. Che and Liu (2010) and Che *et al*. (2010) explains a joint approaching

Corresponding Author: G. Veena, Department of Computer Science and Applications, Amrita School of Engineering, Amrita University, Amrita Vishwa Vidyapeetham, Amritapuri, India

model of WSD with SRL. It uses a Markov logic model for combining WSD and SRL. This approach act as a tool. This approach will first identify the sense of the word then it identifies the relationship between the words and its arguments using Markov Model. In their approach, the WSD works with Lesk algorithm (Basile *et al*., 2014). For finding the relationship, it mainly focuses on words and its positions in the sentence. And it works in the form of predicate-argument structure form. It acts as a tool and works in fewer accuracy algorithms.

Lim *et al.* (2013) explains how SRL works based on dependency relation, i.e., for each sense first they are identified predicate and corresponding arguments, then it performs predicate classification and argument classification. SRL can be built using constituent-based parsing and dependency based parsing. But constituency based parsing is not accurate compared to dependency based parsing. Dependency-based parsing gives inter-relationship between predicates and its arguments. The classification methods or sequence labeling are the main mechanisms of SRL. This architecture explains as dependency-based SRL system using sequence labeling can achieve state of the art performance. It faces many problems during testing time. Their future research is to resolve that problem by WSJ dataset. Chaplot *et al.* (2015) explains how an unsupervised WSD works with dependency parser for web page filtering. Basically, they explained dependency parsing with a tool called Minipar. Their focus is to maximize the joint probability of all the word senses in the sentence or text, given sense dependencies of each word. These sense dependencies are determined using a dependency parser while the required joint probability is maximized using MAP inference query on the Markov Random Field (MRF). This model works using graph structure. But sense dependencies are not always represented in the syntactic structure of the sentence and thus not identified by dependency parser. Pal and Saha (2015), Kumar *et al.* (2014) and Hosny (2003) describe how WSD works in different languages or simply how machine translation works in NLP using WSD. This study explained the recent works in different Indian languages. Basically, how WSD researches in Dravidian languages like Malayalam and morphologically rich language like Punjabi and Bengali language. How WSD algorithms like knowledge-based approaches, supervised algorithms and unsupervised algorithms works in these languages.

MATERIALS AND METHODS

Proposed model: The proposed model is depicted in Fig. 1 which is an extended model for improving the accuracy of labeling a sentence in SRL. The model is a hybrid approach in which the modules are Sense Identification (SI), Dependency Parsing (DP), CRF and Labeling module. The SI module is used for identifying the sense of a word with the supervised algorithm. DP module gives a relationship between predicates and its arguments. The CRF Model is used to combining DP with SRL. The output of labeling module is a semantic relationship between the predicates and its arguments. For example, the sentence shown in this study, we can find the predicate word “sat” has sense “be seated” and the sensed label is “sit.01”. Then the identification of the relationship between predicates and arguments using dependency parsing is done. The argument leaded by the token “The cat” with definition “feline animal” (cat.01) is mention to the agent and the argument leaded by the token “mat” with sense “a thick flat pad used as a floor covering” is referring to the object (patient) being sit.

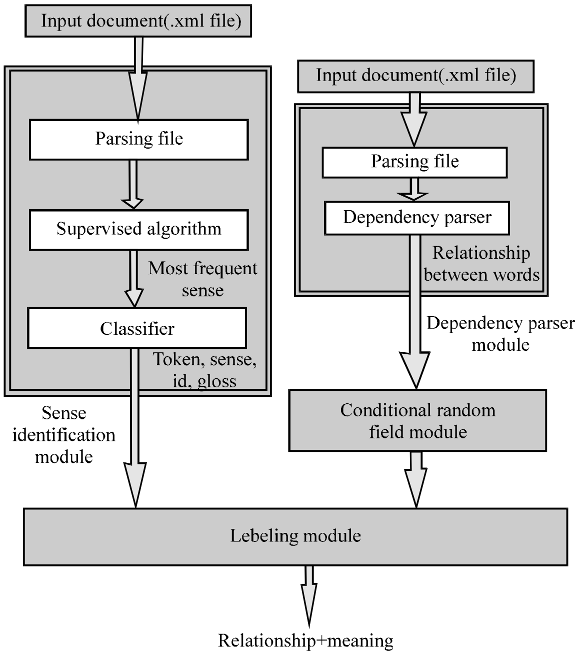

Fig. 1: Proposed model

Word sense disambiguation with semantic role Labeling:

	The	cat	sat	on	the	mat
sit.01	pgent (arg0)				patient (arg 1)	

Sense identification module: In sense identification module, the supervised algorithm is used for finding the most frequent sense of a word. The input is a text document and the final output is each word and its sense ids. The sense of the word in a sentence, i.e., the word "leave" has sense "leave behind unintentionally" and the sensed label is "leave.10". The steps listed as:

Word sense disambiguation:

She	left	her	bag
	leav.10		

Parsing the training file: The training document (Veena and Krishnan, 2015) contains instance id, context and sense id. At first parse, the training document will divide the context into three parts, i.e., left context, the headword and right context. The disambiguated word is called the headword. Then find the sense id of the headword based on the context iteratively. The output is left context, right context, the headword and its corresponding sense id for the further process.

Parsing the test file: The test document contains an instance id and context. For each sentence, we have to perform again left context, right context and headword division.

Dictionary building: After parsing the document, we have to build a dictionary based on the training file. On dictionary building, first, tokenize the leftcontext and right context. Then, we have to set a window size for performing k-nearest neighbor for finding nearest collocations. The output of building dictionary is the nearest collocations from the left tokens and right tokens (based on window size) and sense id of that sentence.

Vectorization of the sentence

Training data: After building the dictionary, the next step is to vectorize the data (especially left context and right context of each sentence). The vectorization of data stored in a (word, count) format. The output of vectorization is a vector form of each context and its corresponding sense id.

Test data: The test data vectorization is similar to training data vectorization. The output of vectorization of test data is the vector form of each context in test data.

Supervised algorithm: The supervised algorithm aims to identify the most frequent sense of a word in a context. The sense id of the most frequent vector is obtained by comparing the testing vector with training vector iteratively. The output of the algorithm is a word (token) of each context and its sense id of each test data.

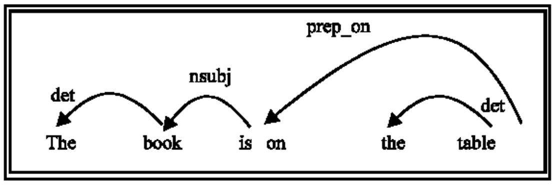

Fig. 2: Dependency parsing module

Classification: For finding the senses of the sentences, we used SENSEVAL-3 dataset. The dataset contains almost 1985 contexts. First, we trained the training set and found the senses of each context. Based on the training set, we used to perform SVM classifier for test data classification (Algorithm 1 and 2).

Algorithm 1; Main algorithm:

```
Input: Dictionary, test file
Output: Most frequency sense of the word (sence id)
For each sentence from the test file: do
  For each word in a sentence do: do
    sense id = get the sense (word dictionary)
      Print lexit and senid
  End
End
```

Algorithm 2; Get sense id:

```
Input: lexit (word), dictionary
Output: sense id
For each lexit or word do
  senid = dic[lexit]
  Return senid
End
```

Dependency parsing: PractNLP tool is used for finding the dependency parsing. Dependency parsing gives the relationship between the words in the sentence. Dependency structure independent of a word order means it is fit for free word order languages (like Indian languages). It consists of the syntactic structure consists of binary, asymmetrical relations between the words of a sentence. It is interested in grammatical relations between individual words, represented as a directed graph. The networks also have labels. Figure 2 illustrates the relationship between the words in the sentence. Dependency parsing is a straight-forward because parsing can be reduced to label each token w_i with w_j.

CRF module: The CRF Model helps to label a sentence properly. This model act as an intermediate step in between dependency parsing and SRL. It basically performs if-then-else rules. Whenever if condition is satisfied, then corresponding score will be calculated, otherwise else statements will be executed. After performing dependency parsing, we will get the relationship between words in the sentence. Then, we have to perform each and every relationship comparison and score finding. Assign each feature function f_j a weight λ_j. The p_l means previous label and c l means current label and sent means sentence.

Generally, there are three types of labels are used, i.e., agent(A0), instrument(verb) and patient(A1) as subject, verb and object, respectively. Basically, to label the agent, we used to define these examples: The sentence is: I went to bank to deposit some money; for finding labeling agent, then p_l_i or p_l of i is subject, i.e., if relationship(I, went) = subj then "score = 1"and c_l_i or c_l of i is verb, i.e., if relationship (went, bank) = verb, then "score=1". Then semantic relationship between (sentence, I, p_l_i, c_l_i) = Agent and total score = 2:

$$f(\text{Sent}, i, p\_l_i, c\_l_i) = 1, \text{if if}(p\_l = \text{sub}) \text{ and } (c\_l = \text{prep to}) \text{ and } 0 \text{ otherwise} \tag{1}$$

$$f(\text{Sent}, i, p\_l_i, c\_l_i) = 1, \text{if if}(p\_l = \text{sub}) \text{ and } (c\_l = \text{sub}) \text{ and } 0 \text{ otherwise} \tag{2}$$

$$f(\text{Sent}, i, p\_l_i, c\_l_i) = 1, \text{if if}(p\_l = \text{sub}) \text{ and } (c\_l = \text{verb}) \text{ and } 0 \text{ otherwise} \tag{3}$$

Basically, to label the patient, we used to define these examples:

$$f(\text{Sent}, i, p\_l_i, c\_l_i) = 1, \text{if if }(p\_l = \text{xcomp}) \text{ and } (c\_l = \text{obj}) \text{ and } 0 \text{ otherwise} \tag{4}$$

$$f(\text{Sent}, i, p\_li, c\_l_i) = 1, \text{if if }(p\_l = \text{verb}) \text{ and } (c\_l = \text{obj}) \text{ and } 0 \text{ otherwise} \tag{5}$$

From the above examples sub means subject, obj means object, prep to means preposition to, etc. For example relationship (i, went) = subject and relationship (went, bank) = prep to, so, the label is agent.

Labeling module: SRL assign roles to all words (arguments) in the sentence based on the predicate called predicate argument structure. The subject, object, verb as agent, patient, instrument, etc. If, we improve the labeling of the sentence in SRL, then it will lead to improve the application performance in SRL. The main application is a question answering section.

RESULTS AND DISCUSSION

Evaluation and test results: The WSD was evaluated using SENSEVAL-3 dataset. The senseval dataset means semantic evaluation dataset. The SENSEVA-2 explains about WSD of lexical sample words and translation. The SENSEVAL-3 explains about logic form trans formation, machine translation, SRL and WSD. SENSEVAL-3 is better because it considered more factors for each sentence. The F1 score of SENSEVAL-2 is less compared to SENSEVAL-3 because it has some limitations. This dataset file contains training and test dataset. The training dataset contains 1985 passages and test dataset contains 1985 passages. WordNet is used for finding the meaning of the token and it behaves like a thesaurus. The corpus has been annotated on with different levels of annotation including instance id or label id, lexe item, sense id and context. The dictionary contains instance id, sense id and gloss. The dictionary helps to determine the meaning of the word based on the sense id. Basic evaluation measures used for testing as precision and recall. Precision is defined as the ratio of number of true positives to number of true positives plus false positives. Recall is defined as the ratio of number of true positives to the number of true positives plus number of false negatives.

There is an SVM classifier is to classify the token and sense id. After training the data, then SVM classifier will predict the corresponding classes of each token in test data.

Sense identification module: Word sense disambiguation is the process of finding the correct gloss of a context. Table 1 explains how much precision and recall occur in two datasets. Figure 3 represents the graph for F1 score comparison of two dataset in WSD. The datasets are SENSEVAL-2 and 3. The baseline WSD system in the supervised algorithm is trained with all the training context in the SENSEVAL-2 and 3 dataset. Its performance on the basis of the F1 score is 80.53 and 82.35% (Fig. 4 and 5).

Dependency parsing: This module find the relationship between all words in the context or sentence. Dependency parser gives the relationship between predicates and arguments in each sentence. The PractNLP tool is used for performing dependency parser. The PractNLP tool is a pythonic library over SENNA and Stanford Dependency Extractor. The main feature of SENNA is it is written in C. So, it is fast. The benefit is that it uses only dependency extractor component of stanford parser which takes in syntactic parse from SENNA and applies dependency

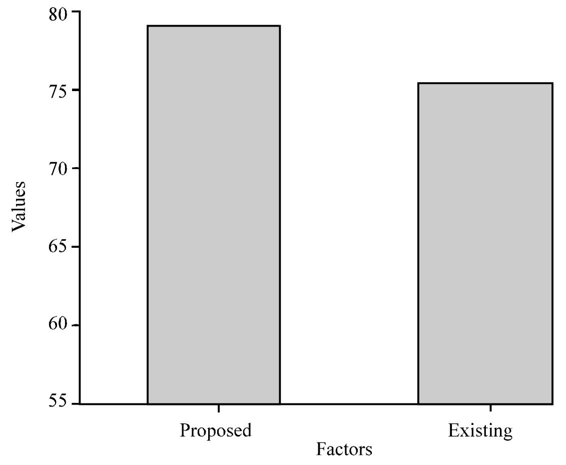

Fig. 3: Word sense disambiguation

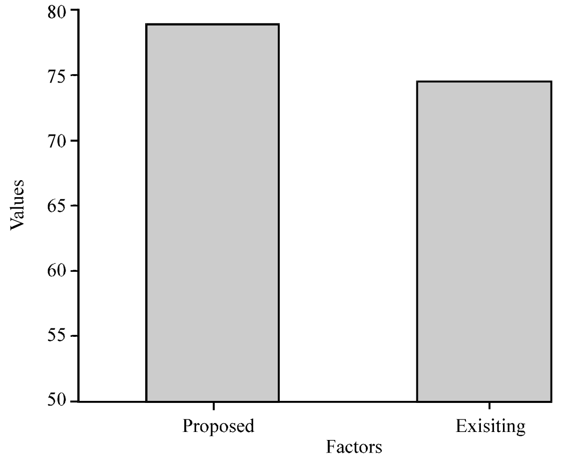

Fig. 4: Comparison of model using F1 score with SENSEVAL-3 dataset

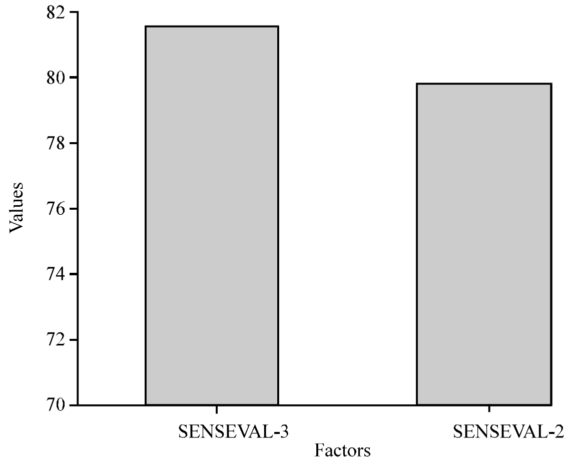

Fig. 5: Comparison of model using F1 score with SENSEVAL-2 dataset

Table 1: Word sense disambiguation

Dataset	Precision	Recall	F1 score
SENSEVAL-3	81.6	83.12	82.35
SENSEVAL-2	80.0	81.08	80.53

Table 2: Dependency parsing with

Dataset	Precision	Recall	F1 score
SENSEVAL-3	88.5	89.8	89.14

Table 3: Comparison of models using SENSEVAL-3 and 2 dataset

	SENSEVAL-3		SENSEVAL-2	
WSD with SRL	Proposed	Existing	Proposed	Existing
Precision	81.12	70.80	77.40	74.41
Recall	81.52	72.70	83.84	78.64
F1-score	81.31	71.73	80.49	76.46

extraction. This tool has no need to load parsing models for stanford parser which takes time. And it is easy to use.

Conditional random field module: Based on the output of the dependency parser, the model will write some functions for connecting dependency parsing to SRL. It is a mapping function. So, we have to explain all the loopholes of a sentence. For example: "I went to bank to deposit some money".

After getting input as output of dependency parser, then the functions will be come if relationship (I, went) = subject and relationship(went, bank) = preposition to, then output becomes agent or A0. Table 2 illustrates how much precision and recall occurred in dependency parser with SRL using SENSEVAL-3 dataset (Table 3).

CONCLUSION

In this study, we presented a model for improving the accuracy of labeling a sentence with WSD and dependency parsing. It can observably conclude that the labeling as well as sense of the sentence performed better. We can conclude that F1 score for the proposed method was better compared to the existing WSD with SRL online tool. We could identify that the algorithm used in tool is Lesk algorithm which performed less as compared to our proposed method.

RECOMMENDATION

As a future research, the question answering application can be implemented through proposed method.

ACKNOWLEDGEMENT

We are thankful to Dr M. R. Kaimal, Chairman Department of Computer Science, Amrita University for his valuable feedback and suggestions.

REFERENCES

Basile, P., A. Caputo and G. Semeraro, 2014. An enhanced lesk word sense disambiguation algorithm through a distributional semantic model. Proceedings of the International Conference on Computational Linguistics COLING'14, August 23-29, 2014, ACL Publisher, Dublin, Ireland, pp: 1591-1600.

Chaplot, D.S., P. Bhattacharyya and A. Paranjape, 2015. Unsupervised word sense disambiguation using Markov random field and dependency parser. Proceedings of the 29th AAAI Conference on Artificial Intelligence, January 25-30, 2015, AAAI, Austin, Texas, ISBN:0-262-51129-0, pp: 2217-2223.

Che, W. and T. Liu, 2010. Using word sense disambiguation for semantic role labeling. Proceedings of the 4th International Symposium on Universal Communication (IUCS), October 18-19, 2010, IEEE, Beijing, China, ISBN:978-1-4244-7821-7, pp: 167-174.

Che, W., T. Liu and Y. Li, 2010. Improving semantic role labeling with word sense. Proceedings of the Annual Conference on Human Language Technologies: North American Chapter of the Association for Computational Linguistics, June 02-04, 2010, ACM, Los Angeles, California, ISBN:1-932432-65-5, pp: 246-249.

Hosny, S.S., 2003. Shape grammars: Style generators in computer- aided architectural design. Intl. J. Eng. Appl. Sci., 50: 37-55.

Jurafsky, D. and J.H. Martin, 2015. Semantic role labeling. Master Thesis, Stanford University, Stanford, California.

Kim, S.N. and T. Baldwin, 2013. Word sense and semantic relations in noun compounds. ACM. Trans. Speech Lang. Process., Vol. 10, 10.1145/2483969.2483971.

Kolte, S.G. and S.G. Bhirud, 2008. Word sense disambiguation using wordnet domains. Proceedings of the 1st International Conference on Emerging Trends in Engineering and Technology ICETET'08, July 16-18, 2008, IEEE, Nagpur, Maharashtra, ISBN:978-0-7695-3267-7, pp: 1187-1191.

Kumar, M.A., S. Rajendran and K.P. Soman, 2014. Tamil word sense disambiguation using support vector machines with rich features. Intl. J. Appl. Eng. Res., 9: 7609-7609.

Lim, S., C. Lee and D. Ra, 2013. Dependency-based semantic role labeling using sequence labeling with a structural SVM. Pattern Recognit. Lett., 34: 696-702.

Pal, A.R. and D. Saha, 2015. Word sense disambiguation: A survey. Intl. J. Control Theor. Comput. Model., 5: 1-16.

Soorya, P., G. Aiswarya and N.A. Kumar, 2015. A rule based approach on word sense disambiguation. Intl. J. Appl. Eng. Res., 10: 2001-2004.

Veena, G. and S. Krishnan, 2015. A Concept Based Graph Model for Document Representation using Conference Resolution. In: Intelligent Systems Technologies and Applications, Berretti, S., S. Thampi and P. Srivastava (Eds.). Springer, Kochi, Kerala, India, ISBN:978-3-319-23035-1, pp: 367-379.