Building a Semantic Role Labelling Toolkit for Vietnamese

Thesis · September 2015

CITATIONS
0

READS
88

2 authors:

Hoang Thai Pham
Alt Inc
13 PUBLICATIONS **41** CITATIONS
SEE PROFILE

Khoai Xuan Pham
FPT University
5 PUBLICATIONS **20** CITATIONS
SEE PROFILE

Building a Semantic Role Labelling Toolkit for Vietnamese

Pham Thai Hoang, Pham Xuan Khoai
Supervisor: Dr. Le Hong Phuong

August 25, 2015

Abstract

Semantic role labelling (SRL) is a task in natural language processing which detects and classifies the semantic arguments associated with the predicates of a sentence. It is an important step towards understanding the meaning of a natural language. There exists SRL systems for well-studied languages like English, Chinese or Japanese but there is not any such system for the Vietnamese language. In this thesis, we present the first SRL system for Vietnamese with encouraging accuracy. We first demonstrate that a simple application of SRL techniques developed for English could not give a good accuracy for Vietnamese. We then introduce a new algorithm for extracting candidate syntactic constituents, which is much more accurate than the common node-mapping algorithm usually used in the identification step. Finally, in the classification step, in addition to the common linguistic features, we propose novel and useful features for use in SRL. Our SRL system achieves an F_1 score of 73.53% on the Vietnamese PropBank corpus. This system, including software and corpus, is available as an open source project and we believe that it is a good baseline for the development of future Vietnamese SRL systems.

Keywords: Semantic Role labelling, Support Vector Machine, Maximum Entropy, Natural Language Processing, Vietnamese

Acknowledgements

Firstly, we would like to express our sincere gratitude to our advisor Dr. Le Hong Phuong for the continuous support of our BSc study and related research, for his patience, motivation, and immense knowledge. His guidance helped us in all the time of research and writing of this thesis. We could not have imagined having a better advisor and mentor for our BSc study.

Besides my advisor, I would like to thank Dr. Tran The Trung, director of FPT Technology Research Institute, for his help in providing data for our research.

My sincere thanks also goes to Dr. Dang Hoang Vu, who supported us to do our research. Without his precious support it would not be possible to conduct this research.

Last but not the least, we would like to thank our families: our parents and to our brothers and sister for supporting us spiritually throughout writing this thesis and my life in general.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

SRL is the task of identifying semantic roles of predicates in the sentence. In particular, it answers a question *Who did What to Whom, When, Where, Why?*. A simple Vietnamese sentence *Nam giúp Huy học bài vào hôm qua* (Nam helped Huy to do homework yesterday) is given in Figure 1.1.

$$\underbrace{\text{Nam}}_{Who}\ \textit{giúp}\ \underbrace{\text{Huy}}_{Whom}\ \underbrace{\text{học bài}}_{What}\ \underbrace{\text{vào hôm qua}}_{When}$$

Figure 1.1: An example sentence

To assign semantic roles for the sentence above, we must analyse and label the propositions concerning the predicate *giúp* (helped) of the sentence. Figure 1.2 shows a result of the SRL for this example, where meaning of the labels will be described in detail in Chapter 4.

$$\underbrace{\text{Nam}}_{Arg0}\ \textit{giúp}\ \underbrace{\text{Huy}}_{Arg1}\ \underbrace{\text{học bài}}_{Arg2}\ \underbrace{\text{vào hôm qua}}_{ArgM-TMP}$$

Figure 1.2: Semantic roles for the example sentence

SRL has been used in many natural language processing (NLP) applications such as question answering [20], machine translation [11], document summarization [1] and information extraction [7]. Therefore, SRL is an important task in NLP.

Examples of SRL Applications:

- Question Answering
  Question: *When was Napoleon defeated?*
  Looking for: *Arg0 – Napoleon*, *Predicate – defeat*, *ArgM-TMP – Answer*

- Machine Translation
  Translate sentence: *"He reads book"* to Vietnamese
  He - Arg0 -Anh ấy
  read - Predicate - đọc
  book - Arg1 - sách

The first SRL system was developed by Gildea and Jurafsky [8]. This system was performed on the FrameNet corpus and was used for English. After that, SRL task has been widely researched by the NLP community. In particular, there have been two shared-tasks, CoNLL-2004 [5] and CoNLL-2005 [6], focusing on SRL task for English. Most of the systems participating in these share-tasks treated this problem as a classification problem and applied some supervised machine learning techniques. In addition, there were some systems developed for other languages such as Chinese [25] or Japanese [23].

In this thesis, we present the first SRL system for Vietnamese with encouraging accuracy. We first demonstrate that a simple application of SRL techniques developed for English or other languages could not give a good accuracy for Vietnamese. In particular, in the constituent identification step, the widely used 1-1 node-mapping algorithm for extracting argument candidates performs poorly on the Vietnamese dataset, having F_1 score of 35.84%. We thus introduce a new algorithm for extracting candidates, which is much more accurate, achieving an F_1 score of 83.63%.

In the classification step, in addition to the common linguistic features, we propose novel and useful features for use in SRL, including function tags and word clusters obtained by performing a Gaussian mixture analysis on the distributed representations of Vietnamese words. These features are employed in two statistical classification models, Maximum Entropy and Support Vector Machines, which are proved to be good at many classification problems.

Our SRL system achieves an F_1 score of 73.53% on the Vietnamese PropBank corpus. This system, including software and corpus, is available as an open source project and we believe that it is a good baseline for the development of future Vietnamese SRL systems.

The thesis is structured as follows. Chapter 2 introduces briefly the SRL task and some well-known corpora for English and other languages. Chapter 3 describes the methodologies of some existing systems. Chapter 4 presents our method. Some difficulties of SRL for Vietnamese are also discussed. After that, we present the evaluation results and discussion. Finally, Chapter 5 concludes the thesis and suggests some directions for future work.

Chapter 2

Problem Description

2.1 Semantic Role Labelling task description

The SRL task is usually divided into two steps. The first step is argument identification. The goal of this step is to identify the syntactic constituents of a sentence which are the most likely to be semantic arguments of its predicates. This is a difficult problem since the number of constituent candidates is exponentially large, especially for long sentences.

$$\underline{\text{Nam}}\ \textit{giúp}\ \underline{\text{Huy}}\ \underline{\text{học bài}}\ \underline{\text{vào hôm qua}}$$

Figure 2.1: Example of identification task

The second step is argument classification which decides the exact semantic role for each constituent candidate identified in the first task. For example, the identification step of the sentence in the previous example *Nam giúp Huy học bài vào hôm qua* is described in Figure 2.1 and in the classification task, semantic roles are labelled as shown Figure 2.2.

$$\underbrace{\text{Nam}}_{Arg0}\ \textit{giúp}\ \underbrace{\text{Huy}}_{Arg1}\ \underbrace{\text{học bài}}_{Arg2}\ \underbrace{\text{vào hôm qua}}_{ArgM-TMP}$$

Figure 2.2: Semantic roles for the example sentence

2.2 Lexical Resource

2.2.1 Lexical Resource for English

Currently, there are two main lexical resources in English which annotate semantic roles in data: FrameNet and PropBank. While PropBank uses very generic labels such as *Ar0*, *Arg1* to annotate the semantic roles, FrameNet marks those semantic roles more detailed to provide several alternative syntactic frames and a set of semantic predicates.

FrameNet

The FrameNet project is a lexical database of English. It was built by annotating examples of how words are used in actual texts. It consists of more than 10,000 word senses, most of them with annotated examples that show the meaning and usage and more than 170,000 manually annotated sentences [3]. This is the most widely used dataset upon which SRL systems for English have been developed and tested.

FrameNet is based on the Frame Semantics theory [4]. The basic idea is that the meanings of most words can be best understood on the basis of a semantic frame: a description of a type of event, relation, or entity and the participants in it. All members in semantic frames are called frame elements. For example, a sentence in FrameNet is annotated in cooking concept as shown in Figure 2.3.

$$\underbrace{\text{The boy}}_{Cook}\ \textit{grills}\ \underbrace{\text{their catches}}_{Food}\ \underbrace{\text{on an open fire}}_{Heating-instrument}$$

Figure 2.3: An example sentence in the FrameNet corpus

The FrameNet database consists domains which are big concepts. In a particular domain, we have some semantic frames about some smaller concepts. Each of frames consists several frame elements. The Figure 2.4 is example about Communication domain.

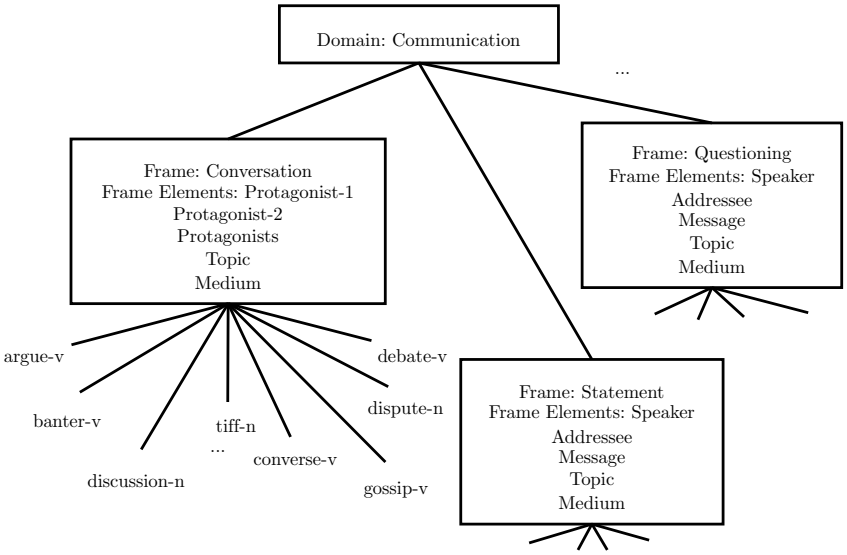


Figure 2.4: Sample domain and frames from the FrameNet lexicon.

Propbank

PropBank is a corpus that is annotated with verbal propositions and their arguments [2]. PropBank tries to supply a general purpose labelling of semantic roles for a large corpus to support the training of automatic semantic role labelling systems. However, defining such a universal set of semantic roles for all types of predicates is a difficult task; therefore, only Arg0 and Arg1 semantic roles can be generalized. In addition to the core roles, PropBank defines several adjunct roles that can apply to any verb. It is called Argument Modifier. The semantic roles covered by the PropBank are the following:

- **Core Arguments** (Arg0-Arg5, ArgA): Arguments define predicate specific roles. Their semantics depend on predicates in the sentence.
- **Adjunct Arguments** (ArgM-): General arguments that can belong to any predicate. There are 13 types of adjuncts.
- **Reference Arguments** (R-): Arguments represent arguments realized in other parts of the sentence.
- **Predicate** (V): Participant realizing the verb of the proposition.

Role Name	Description	Role Name	Description
ArgM-ADV	general-purpose	ArgM-CAU	cause
ArgM-DIS	discourse marker	ArgM-DIR	direction
ArgM-NEG	negation marker	ArgM-MNR	manner
ArgM-PRD	predication	ArgM-EXT	extent
ArgM-MOD	modal verb	ArgM-TMP	temporal
ArgM-REC	reciprocal	ArgM-PNC	purpose
ArgM-LOC	location		

Table 2.1: Adjunct arguments in English PropBank

2.2.2 Lexical Resource for other languages

Chinese PropBank

The Chinese PropBank is the first large-scale Chinese corpus annotated with semantic roles, and it is developed in close association with the English Propbank [24]. Like the English PropBank, there are two different kinds of semantic roles in this corpus *core arguments* and *adjunct arguments*. Core arguments consist semantic roles *Arg0* through *Arg5*. Adjunct arguments in the Chinese PropBank have some roles which are different from roles in the English PropBank. These adjunct roles in the Chinese PropBank are listed in table 2.2

Role Name	Description	Role Name	Description
ArgM-ADV	adverbial	ArgM-FRQ	freequency
ArgM-BNF	beneficiary	ArgM-LOC	locative
ArgM-CND	condition	ArgM-MNR	manner
ArgM-DIR	direction	ArgM-PRP	purpose or reason
ArgM-DIS	discourse marker	ArgM-TMP	temporal
ArgM-DGR	degree	ArgM-TPC	topic
ArgM-EXT	extent		

Table 2.2: Adjunct arguments in Chinese PropBank

NAIST

NAIST is a corpus for Japanese [10]. This corpus contains newswire stories from the 1995 corpus of Mainichi News, and was hand-annotated with labels for predicates and three argument types *GA*, *O*, and *NI*.

Chapter 3

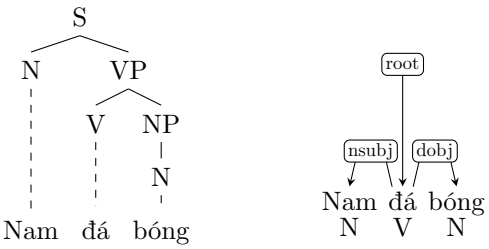
Review of Literature for Solution

3.1 Existing Approaches

This section summarizes existing approaches used by typical SRL systems for well-studied languages. We describe these systems by investigating two aspects, namely data type that the systems use and their strategies for labelling semantic roles, including model types, labelling strategies and degrees of granularity.

Data Types

There are some kinds of data used in the training of SRL systems. Some systems use bracketed trees as the input data. A bracketed tree of a sentence is the tree of nested constituents representing its constituency structure. Some systems use dependency trees of a sentence, which represents dependencies between individual words of a sentence. The syntactic dependency represents the fact that the presence of a word is licensed by another word which is its governor. In a typed dependency analysis, grammatical labels are added to the dependencies to mark their grammatical relations, for example *nominal subject* (nsubj) or *direct object* (dobj). Figure 3.1 shows the bracketed tree and the dependency tree of an example sentence.



(a) The bracketed tree (b) The dependency tree

Figure 3.1: Bracketed and dependency trees for sentence *Nam đá bóng* (Nam plays football)

SRL Strategy

Model Types There are two types of classification models: Independent Model and Joint Model. While independent model decides the label of each argument's candidate independently of other candidates, joint model finds the best overall labelling for all candidates in the sentence. Independent model runs fast but are prone to inconsistencies. For example, Figure 3.2 shows some typical inconsistencies, including overlapping arguments, repeating arguments and missing arguments of a sentence *Do học chăm, Nam đã đạt thành tích cao* (By studying hard, Nam got a high achievement). Figure 3.3 describes architecture of general SRL system.

Do học chăm, Nam đã *đạt* thành tích cao.
$Arg1$

Do học chăm, Nam đã *đạt* thành tích cao.
$Arg1$

(a) Overlapping argument

Do học chăm, Nam đã *đạt* thành tích cao.
$Arg1$ $Arg1$

(b) Repeating argument

Do học chăm, Nam đã *đạt* thành tích cao .
$Arg0$ $Arg0$

(c) Missing argument

Figure 3.2: An example of inconsistencies

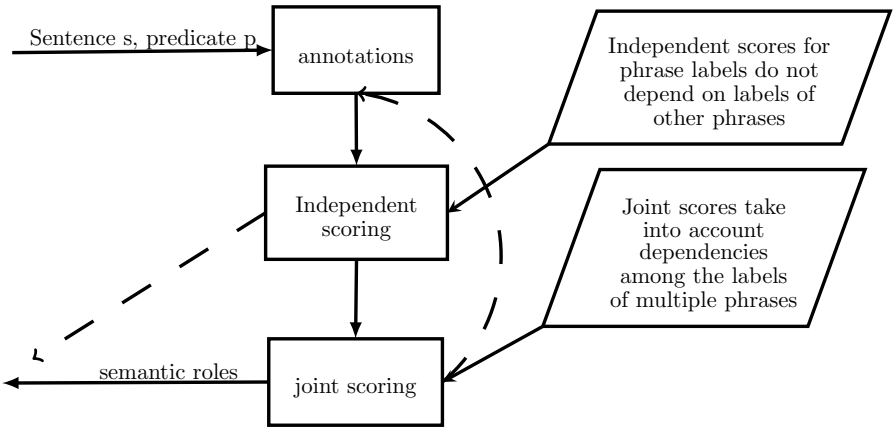


Figure 3.3: General SRL system

Labelling Strategies Strategies for labelling semantic roles are diverse, but we can summarize that there are three main strategies. Most of the systems use a two-step approach consisting of identification and classification [9, 12]. The first step identifies arguments from many candidates. It is essentially a binary classification problem. The second step classifies these arguments into particular semantic roles. Some systems use single classification step by adding a "null" label into semantic roles, denoting that this is not an argument [21]. Other systems consider SRL as a sequence tagging [14, 18].

Granularity Existing SRL systems use different degrees of granularity when considering constituents. Some systems use individual words as their input and perform sequence tagging to identify arguments. This method is called Word-by-Word (W-by-W) approach. Other systems directly take syntactic phrases as input constituents. This method is called Constituent-by-Constituent (C-by-C) approach.

Nam *giúp* Huy học bài vào hôm qua

(a) Example of C-by-C

Nam *giúp* Huy học bài vào hôm qua

(b) Example of W-by-W

Figure 3.4: C-by-C and W-by-W approaches

Compared to the W-by-W approach, C-by-C approach has several advantages. First, phrase boundaries are usually consistent with argument boundaries. Second, C-by-C approach allows us to work with larger contexts due to a smaller number of candidates in comparison to the W-by-W approach.

3.2 Existing Systems

3.2.1 First Statistical Model

The first statistical system was developed by Gildea and Jurafsky in 2002 [8]. The system was based on statistical classifiers trained on roughly 50,000 sentences that were hand-annotated with semantic roles by the FrameNet semantic labelling project.

They extracted some features from full syntactic tree to use in the training stage. Their features are *phrase type*, *governing category*, *parse tree path*, *position*, *voice*, *head word* and *subcategorization*. Their feature set is the base feature set of future systems.

Performance of Gildea and Jurafsky system is presented in the Table 3.1.

Role	Number	known boundaries correct	unknown boundaries labeled recall	unknown boundaries unlabeled recall
Agent	2401	92.8	76.7	80.7
Experiencer	333	91.0	78.7	83.5
Source	503	87.3	67.4	74.2
Proposition	186	86.6	56.5	64.5
State	71	85.9	53.5	62.0
Patient	1161	83.3	63.1	69.1
Topic	244	82.4	64.3	72.1
Goal	694	82.1	60.2	69.6
Cause	424	76.2	61.6	73.8
Path	637	75.0	63.1	63.4
Manner	494	70.4	48.6	59.7
Percept	103	68.0	51.5	65.1
Degree	61	67.2	50.8	60.7
Null	55	65.5	70.9	85.5
Result	40	65.0	55.0	70.0
Location	275	63.3	47.6	63.6
Force	49	59.2	40.8	63.3
Instrument	30	43.3	30.0	73.3
(other)	406	57.9	40.9	63.1
Total	8167	82.1	63.6	72.1

Table 3.1: Performance of Gildea and Jurafsky system.

3.2.2 CoNLL-2004 Shared Task

The shared task of CoNLL-2004 concerns the recognition of semantic roles, for the English language [5]. They used the February 2004 release of PropBank to training and testing. The data consists of six sections of the Wall Street Journal part of the PropBank: training set (sections 15-18), development set (section 20) and test set (section 21). Ten systems have participated in the CoNLL-2004 shared task. They approached the task in several ways, using different learning components and labelling strategies. These systems in CoNLL-2004 shared task were based only on partial syntactic information. The data sample in CoNLL-2004 shared task is presented in Figure 3.5.

Most of teams used constituent-by-constituent approach, but there are some teams used word-by-word approach. Labelling strategies and granularity are described in the Table 3.2.

Table 3.3 shows features which teams used and performance of each team is described in Table 3.4

3.2.3 CoNLL-2005 Shared Task

Like the CoNLL-2004 shared task, the aim of the CoNLL-2005 shared task is to develop a machine learning system for SRL problem [6]. While systems in the CoNLL-2004 shared task were based only on partial syntactic information, systems in the CoNLL-2005 shared task evaluated the contribution of full parsing in SRL, the complete syntactic trees given by two alternative parsers have

The	DT	B-NP	(S*	O	-	(A0*	*
San	NNP	I-NP	*	B-ORG	-	*	*
Francisco	NNP	I-NP	*	I-ORG	-	*	*
Examiner	NNP	I-NP	*	I-ORG	-	*A0)	*
issued	VBD	B-VP	*	O	issue	(V*V)	*
a	DT	B-NP	*	O	-	(A1*	(A1*
special	JJ	I-NP	*	O	-	*	*
edition	NN	I-NP	*	O	-	*A1)	*A1)
around	IN	B-PP	*	O	-	(AM-TMP*	*
noon	NN	B-NP	*	O	-	*AM-TMP)	*
yesterday	NN	B-NP	*	O	-	(AM-TMP*AM-TMP)	*
that	WDT	B-NP	(S*	O	-	(C-A1*	(R-A1*R-A1)
was	VBD	B-VP	(S*	O	-	*	*
filled	VBN	I-VP	*	O	fill	*	(V*V)
entirely	RB	B-ADVP	*	O	-	*	(AM-MNR*AM-MNR)
with	IN	B-PP	*	O	-	*	*
earthquake	NN	B-NP	*	O	-	*	(A2*
news	NN	I-NP	*	O	-	*	*
and	CC	I-NP	*	O	-	*	*
information	NN	I-NP	*S)S)	O	-	*C-A1)	*A2)
.	.	O	*S)	O	-	*	*

Figure 3.5: Sentence in CoNLL-2004 shared task data

	prop.	lab.	gran.	glob.	post
hacioglu	s	t	P-by-P	no	no
punyakanok	s	fl	W-by-W	yes	no
carreras	j	fl	P-by-P	yes	no
lim	s	t	P-by-P	yes	no
park	s	rc	P-by-P	no	yes
higgins	s	t	W-by-W	no	yes
van den bosch	s	cj	P-by-P	part.	yes
kouchnir	s	rc	P-by-P	no	yes
baldewein	s	rc	P-by-P	yes	no
williams	s	t	mixed	no	no

Table 3.2: Main properties of the SRL strategies implemented by the ten participant teams. *prop.* stands for the treatment of all propositions of a sentence; possible values are: *s* (separate) and *j* (joint). *lab.* stands for labelling strategy; possible values are: *t* (one step tagging), *rc* (recognition + classification), *fl* (filtering + labelling), cj (classification + joining). *gran.* stands for granularity; *glob.* stands for global optimization. *post* stands for postprocessing.

been provided as input information for the task. There were nineteen systems in the CoNLL-2005 shared task. The data consists of sections of the Wall Street Journal part of the Penn TreeBank, with information on predicate-argument structures extracted from the PropBank corpus. Sections 02-21 were used for training, section 24 for development, and section 23 for test. In addition, the test set of the shared task includes three sections of the Brown corpus.

The Table 3.5 summarizes SRL strategies of each system.

Table 3.6 presents the overall results of all systems. The best performance was achieved by Punyakanok system. They reached an F_1 at 78% in the combined test set.

	sy	ne	al	at	as	aw	an	vv	vs	vf	vc	rp	di	pa	ex
hacioglu	+	+	+	-	-	+	-	+	+	-	+	+	+	+	+
punyakanok	+	+	+	+	+	+	-	+	-	+	+	+	-	+	+
carreras	+	-	-	-	+	+	-	+	-	-	-	+	-	+	+
lim	+	-	-	-	-	+	+	+	-	-	-	+	-	+	-
park	+	-	-	-	-	-	-	+	-	-	+	+	+	+	+
higgins	+	+	-	-	-	-	+	+	-	-	-	+	+	+	-
van den bosch	+	+	-	-	-	-	-	+	+	-	-	+	+	-	-
kouchnir	+	-	+	-	+	+	-	+	-	+	-	+	+	-	-
baldewein	+	+	+	+	+	+	-	+	+	-	-	+	+	-	-
williams	+	+	-	-	-	-	-	-	-	-	-	+	-	-	-

Table 3.3: Main feature types used by the 10 participating systems in the CoNLL-2004 shared task, sorted by performance on the test set. *sy*: use of partial syntax (all levels); *ne*: use of named entities; *al*: argument length; *at*: argument type; *as*: argument internal structure; *aw*: head word lexicalization of arguments; *an*: neighboring arguments; *vv*: verb voice; *vs*: verb statistics; *vf*: verb features derived from PropBank frames; *vc*: verb local context; *rp*: relative position; *di*: distance (horizontal or in the hierarchy); *pa*: path; *ex*: feature expansion.

development	Precision	Recall	F 1
hacioglu	74.18%	69.43%	71.72
punyakanok	71.96%	64.93%	68.26
carreras	73.40%	63.70%	68.21
lim	69.78%	62.57%	65.97
park	67.27%	64.36%	65.78
higgins	65.59%	60.16%	62.76
van den bosch	69.06%	57.84%	62.95
kouchnir	44.93%	63.12%	52.50
baldewein	64.90%	41.61%	50.71
williams	53.37%	32.43%	40.35
baseline	50.63%	30.30%	37.91

test	Precision	Recall	F 1
hacioglu	72.43%	66.77%	69.49
punyakanok	70.07%	63.07%	66.39
carreras	71.81%	61.11%	66.03
lim	68.42%	61.47%	64.76
park	65.63%	62.43%	63.99
higgins	64.17%	57.52%	60.66
van den bosch	67.12%	54.46%	60.13
kouchnir	56.86%	49.95%	53.18
baldewein	65.73%	42.60%	51.70
williams	58.08%	34.75%	43.48
baseline	54.60%	31.39%	39.87

Table 3.4: Overall performances in CoNLL-2004 shared task

	ML-method	synt	pre	label	embed	glob	post	comb	type
punyakanok	SNoW	n-cha,col	x&p	i+c	defer	yes	no	n-cha+col	ac-ILP
haghighi	ME	n-cha	?	i+c	dp-prob	yes	no	n-cha	re-rank
marquez	AB	cha,upc	seq	bio	!need	no	no	cha+upc	s-join
pradhan	SVM	cha,col/chunk	?	c/bio	?	no	no	cha+col → chunk	stack
surdeanu	AB	cha	prun	c	g-top	no	yes	no	-
tsai	ME,SVM	cha	x&p	c	defer	yes	no	ME+SVM	ac-ILP
che	ME	cha	no	c	g-score	no	yes	no	-
moschitti	SVM	cha	prun	i+c	!need	no	no	no	-
tjongkimsang	ME,SVM,TBL	cha	prun	i+c	!need	no	yes	ME+SVM+TBL	s-join
yi	ME	cha,AN,AM	x&p	i+c	defer	no	no	cha+AN+AM	ac-join
ozgencil	SVM	cha	prun	i+c	g-score	no	no	no	-
johansson	RVM	cha	softp	i+c	?	no	no	no	-
cohn	T-CRF	col	x&	c	g-top	yes	no	no	-
park	ME	cha	prun	i+c	?	no	no	no	-
mitsumori	SVM	chunk	no	bio	!need	no	no	no	-
venkatapathy	ME	col	prun	i+c	frames	yes	no	no	-
ponzetto	DT	col	prun	c	g-top	no	yes	no	-
lin	CPM	cha	gt-para	i+c	!need	no	no	no	-
sutton	ME	n-bikel	x&p	i+c	dp-prob	yes	no	n-bikel	re-rank

Table 3.5: Main properties of the SRL strategies implemented by the participant teams, sorted by F_1 performance on the WSJ+Brown test set. *synt* stands for the syntactic structure explored; *pre* stands for pre-processing steps; *label* stands for the labelling strategy; *embed* stands for the technique to ensure nonembedding of arguments; *glob* stands for global optimization; *post* stands for postprocessing; *comb* stands for system output combination, and *type* stands for the type of combination.

	Development			Test WSJ			Test Brown			Test WSJ+Brown		
	P(%)	R(%)	F 1	P(%)	R(%)	F 1	P(%)	R(%)	F 1	P(%)	R(%)	F 1
punyakanok	80.05	74.83	77.35	82.28	76.78	79.44	73.38	62.93	67.75	81.18	74.92	77.92
haghighi	77.66	75.72	76.68	79.54	77.39	78.45	70.24	65.37	67.71	78.34	75.78	77.04
marquez	78.39	75.53	76.93	79.55	76.45	77.97	70.79	64.35	67.42	78.44	74.83	76.59
pradhan	80.90	75.38	78.04	81.97	73.27	77.37	73.73	61.51	67.07	80.93	71.69	76.03
surdeanu	79.14	71.57	75.17	80.32	72.95	76.46	72.41	59.67	65.42	79.35	71.17	75.04
tsai	81.13	72.42	76.53	82.77	70.90	76.38	73.21	59.49	65.64	81.55	69.37	74.97
che	79.65	71.34	75.27	80.48	72.79	76.44	71.13	59.99	65.09	79.30	71.08	74.97
moschitti	74.95	73.10	74.01	76.55	75.24	75.89	65.92	61.83	63.81	75.19	73.45	74.31
tjongkimsang	76.79	70.01	73.24	79.03	72.03	75.37	70.45	60.13	64.88	77.94	70.44	74.00
yi	75.70	69.99	72.73	77.51	72.97	75.17	67.88	59.03	63.14	76.31	71.10	73.61
ozgencil	73.57	71.87	72.71	74.66	74.21	74.44	65.52	62.93	64.20	73.48	72.70	73.09
johansson	73.40	70.85	72.10	75.46	73.18	74.30	65.17	60.59	62.79	74.13	71.50	72.79
cohn	73.51	68.98	71.17	75.81	70.58	73.10	67.63	60.08	63.63	74.76	69.17	71.86
park	72.68	69.16	70.87	74.69	70.78	72.68	64.58	60.31	62.38	73.35	69.37	71.31
mitsumori	71.68	64.93	68.14	74.15	68.25	71.08	63.24	54.20	58.37	72.77	66.37	69.43
venkatapathy	71.88	64.76	68.14	73.76	65.52	69.40	65.25	55.72	60.11	72.66	64.21	68.17
ponzetto	71.82	61.60	66.32	75.05	64.81	69.56	66.69	52.14	58.52	74.02	63.12	68.13
lin	70.11	61.96	65.78	71.49	64.67	67.91	65.75	52.82	58.58	70.80	63.09	66.72
sutton	64.43	63.11	63.76	68.57	64.99	66.73	62.91	54.85	58.60	67.86	63.63	65.68
baseline	50.00	28.98	36.70	51.13	29.16	37.14	62.66	33.07	43.30	52.58	29.69	37.95

Table 3.6: Overall performances in CoNLL-2005 shared task

Chapter 4

Solutions

4.1 Our Approach

The previous chapter has reviewed existing techniques for SRL which have been published so far for well-studied languages. In this section, we first show that these techniques per se cannot give a good result for Vietnamese SRL, due to some inherent difficulties, both in terms of language characteristics and of the available corpus. We then develop a new algorithm for extracting candidate constituents for use in the identification step.

Some difficulties of Vietnamese SRL are related to its SRL corpus. We use the Vietnamese PropBank [17] in the development of our SRL system.[1] This SRL corpus has 5,000 annotated sentences, which is much smaller than SRL corpora of other languages. For example, the English PropBank contains about 50,000 sentences, which is ten times larger. While smaller in size, the Vietnamese PropBank has more semantic roles than the English PropBank has – 25 roles compared to 21 roles. This makes the unavoidable data sparseness problem more severe for Vienamese SRL than for English SRL.

In addition, our extensive inspection and experiments on the Vietnamese PropBank have uncovered that this corpus has many annotation errors, largely due to encoding problems and inconsistencies in annotation. In many cases, we have to fix these annotation errors by ourselves. In other cases where only a proposition of a complex sentence is incorrectly annotated, we perform an automatic preprocessing procedure to drop it out, leave the correctly annotated propositions untouched. We finally come up with a corpus of 4,800 sentences which are semantic role annotated. This corpus will be released for free use for research purpose.

A major difficulty of Vietnamese SRL is due to the nature of the language, where its linguistic characteristics are different from occidental languages [13]. We first try to apply the common node-mapping algorithm which are widely

[1] To our knowledge, this is the first SRL corpus for Vietnamese which has been published for free research.

used in English SRL systems to the Vietnamese corpus. However, this application gives us a very poor performance. Therefore, in the identification step, we develop a new algorithm for extracting candidate constituents which is much more accurate for Vietnamese than the node-mapping algorithm. Details of experimental results will be provided in the rest of this Section.

In order to improve the accuracy of the classification step, and hence of our SRL system as a whole, we have integrated many useful features for use in two statistical classification models, namely Maximum Entropy (ME) and Support Vector Machines (SVM). On the one hand, we adapt the features which have been proved to be good for SRL of English. On the other hand, we propose some novel features, including function tags and word clusters.

In the next paragraph, we present our constituent extraction algorithm for the identification step. Details of the features for use in the classification step will be presented in Section 4.2.

Constituent Extraction Algorithm

This algorithm aims to extract constituents from a bracketed tree which are associated to their corresponding predicates of the sentence. If the sentence has multiple predicates, multiple constituent sets corresponding to the predicates are extracted. Pseudo code of the algorithm is described in Algorithm 1.

This algorithm uses several simple functions. The *root*() function gets the root of a tree. The *children*() function gets the children of a node. The *sibling*() function gets the sisters of a node. The *isPhrase*() function checks whether a node is of phrasal type or not. The *phraseType*() function and $functionTag()$ function extracts the phrase type and function tag of a node, respectively. Finally, the *collect*(*node*) function collects words from leaves of the subtree rooted at a node and creates a constituent.

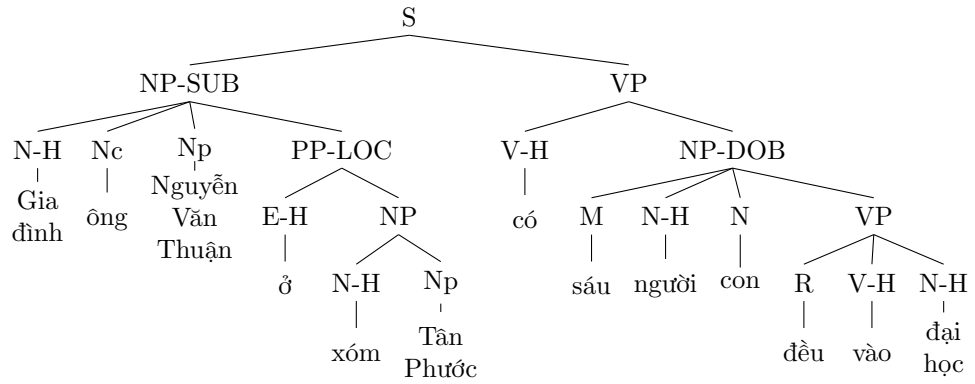


Figure 4.1: Example syntactic tree

Algorithm 1: Constituent Extraction Algorithm

```
input  : A bracketed tree T and its predicate
output: A tree with constituents for the predicate
begin
    currentNode ← predicateNode
    while currentNode ≠ T.root() do
        for S ∈ currentNode.sibling() do
            if |S.children()| > 1 and S.children().get(0).isPhrase() then
                sameType ← true
                diffTag ← true
                phraseType ← S.children().get(0).phraseType()
                funcTag ← S.children().get(0).functionTag()
                for i ← 1 to |S.children()| − 1 do
                    if S.children().get(i).phraseType() ≠ phraseType
                    then
                        sameType ← false
                        break
                    if S.children().get(i).functionTag() = funcTag then
                        diffTag ← false
                        break
                if sameType and diffTag then
                    for child ∈ S.children() do
                        T.collect(child)
            else
                T.collect(S)
        currentNode ← currentNode.parent()
    return T
```

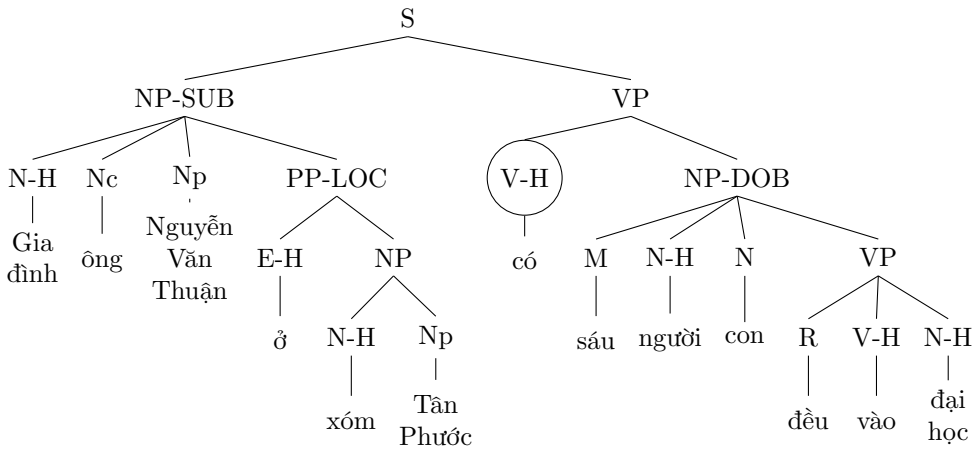


Figure 4.2: Step 1

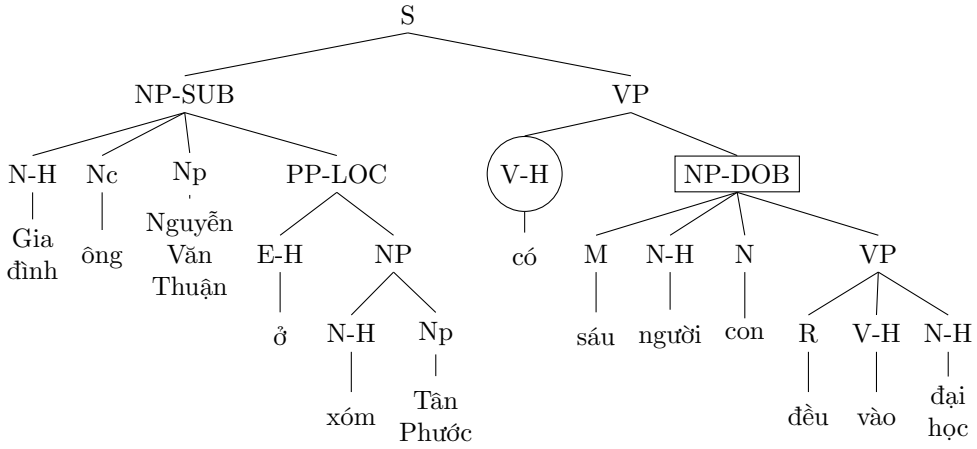


Figure 4.3: Step 2

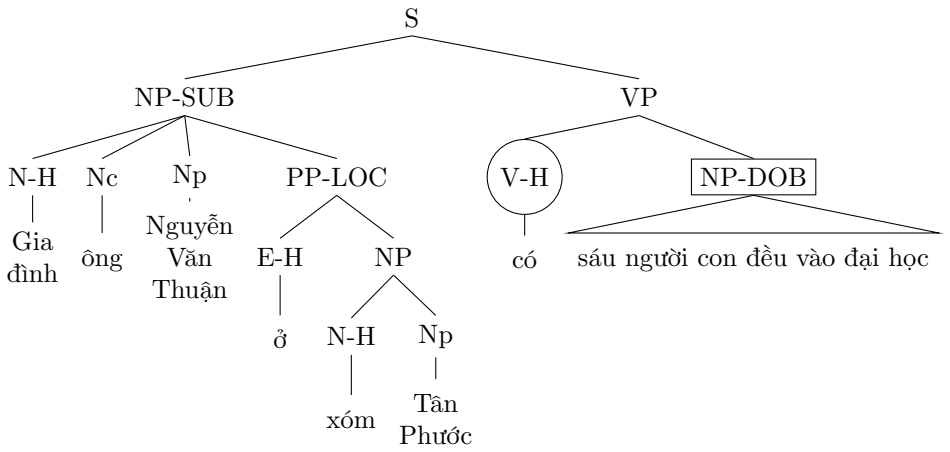


Figure 4.4: Step 3

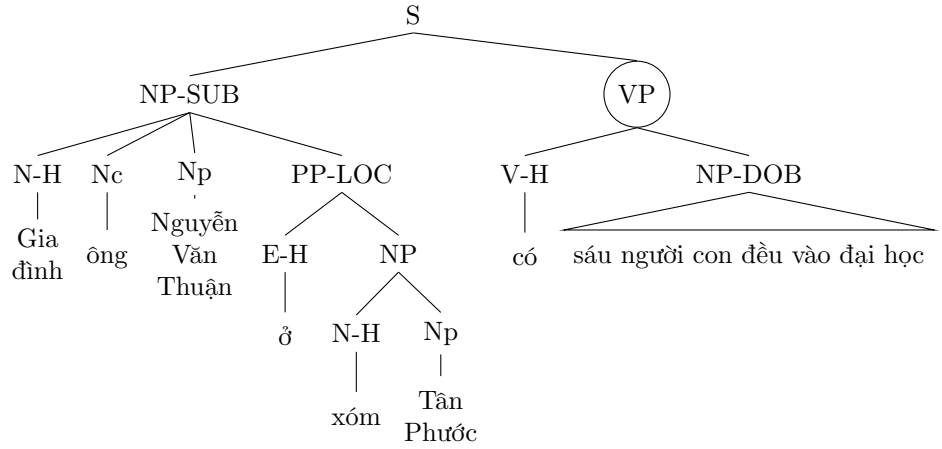


Figure 4.5: Step 4

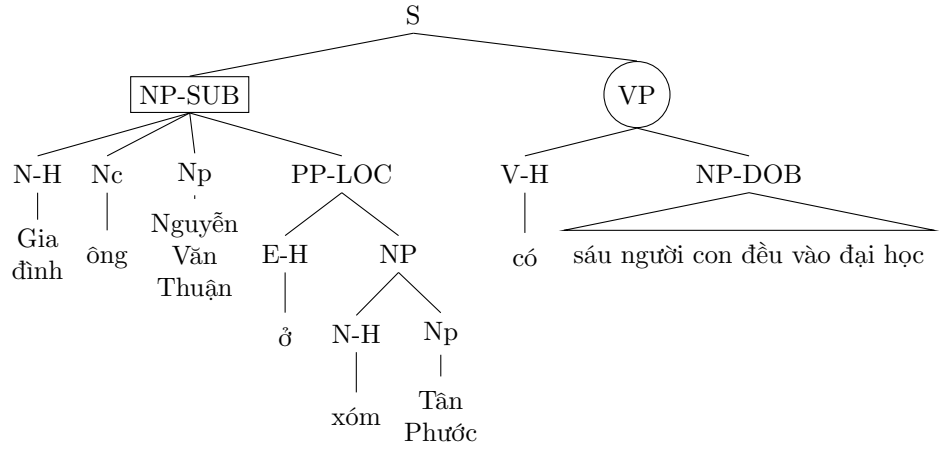


Figure 4.6: Step 5

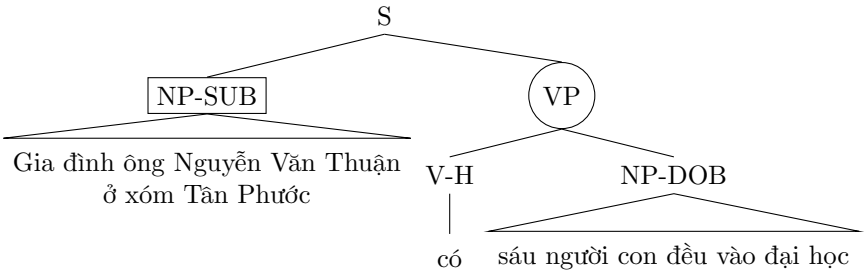


Figure 4.7: Step 6

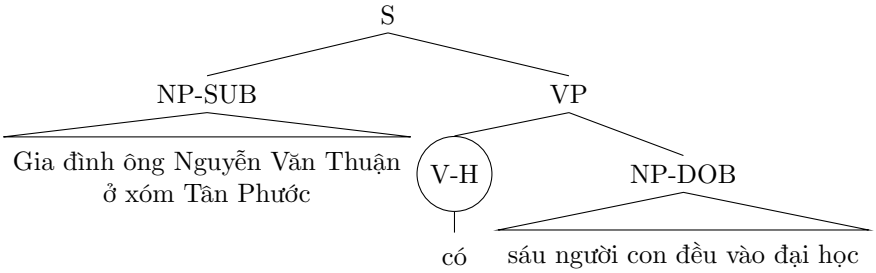


Figure 4.8: The final result

Figure 3.5 shows an example of running the algorithm on a sentence *Gia đình ông Nguyễn Văn Thuận ở xóm Tân Phước có sáu người con đều vào đại học* (Nguyen Van Thuan's family in Tan Phuoc has six children are in colleges). First, we find the current predicate node V-H *có* (has). The current node has only one sibling NP-DOB. This node has one child, so its associated words are collected. After that, we set current node to its parent and repeat the process until reaching the root of the tree. Finally, we obtain a tree with constituents: *Gia đình ông Nguyễn Văn Thuận ở xóm Tân Phước*, and *sáu người con đều vào đại học*.

Our SRL System

Our SRL system is developed on the Vietnamese PropBank. It thus operates on fully bracketed trees. We employ ME and SVM as classifiers. Its classification model is of type independent and its input are C-by-C.

Maximum Entropy Classifier In machine learning, maximum entropy is a classification method that generalizes logistic regression to multiclass problems. That is, it is a model that is used to predict the probabilities of the different possible outcomes of a categorically distributed dependent variable, given a set of independent variables.

The multinomial logistic regression model is defined as

$$P(y = k|\mathbf{x}; \theta_k) = \frac{1}{Z} \exp(\theta_k^T \mathbf{x}) \tag{4.1}$$

where Z is the normalization term:

$$Z = \sum_{k=1}^{K} P(y = k|\mathbf{x}; \theta_k) = \sum_{k=1}^{K} \exp(\theta_k^T \mathbf{x}) \tag{4.2}$$

Like other classifiers, we need to optimize a cost function. The cost function of maximum entropy is:

$$J(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \log P(y_i|\mathbf{x}_i; \theta) = -\frac{1}{N} \sum_{i=1}^{N} \left\{ \theta_{y_i}^T \mathbf{x}_i - \log \left(\sum_{k=1}^{K} \exp(\theta_k^T \mathbf{x}_i) \right) \right\} \tag{4.3}$$

If using L_2-regularization, the cost function is:

$$\begin{aligned} J(\theta) &= -\frac{1}{N}\sum_{i=1}^{N}\log P(y_i|\mathbf{x}_i;\theta) + \frac{\lambda}{2N}\sum_{k=1}^{K}\sum_{j=1}^{D}\theta_{kj}^1 \\ &= -\frac{1}{N}\sum_{i=1}^{N}\left\{\theta_{y_i}^T\mathbf{x}_i - \log\left(\sum_{k=1}^{K}\exp(\theta_k^T\mathbf{x}_i)\right)\right\} + \frac{\lambda}{2N}\sum_{k=1}^{K}\sum_{j=1}^{D}\theta_{kj}^1 \end{aligned} \tag{4.4}$$

Support Vector Machine

SVMs are among the best (and many believe are indeed the best) off-the-shelf supervised learning algorithm. Given a set of training examples, each marked for belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

SVM could be seen as an optimization problem:

$$\begin{aligned} &\min_{\gamma,\omega,b}\frac{1}{2}||\omega||^2 \\ s.t.\quad &y^i(\omega^T x^i + b) \geq 1, i = 1, ..., m \\ &with\quad \gamma = \min_{i=1,...m}\gamma^i \end{aligned} \tag{4.5}$$

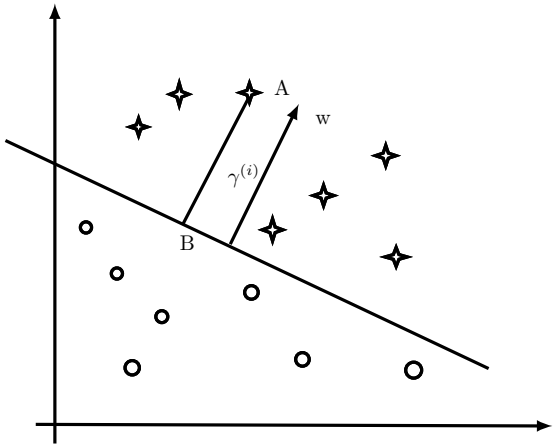


Figure 4.9: Geometric Margin

4.2 Experiments

In this section, we first introduce the Vietnamese PropBank upon which our SRL system has been trained and tested. We then propose two feature sets in use. Finally, we present and discuss experimental results.

4.2.1 Dataset

We conduct experiments on the Vietnamese PropBank [17] containing about 5,460 sentences which are manually annotated with semantic roles. This corpus has a similar annotation schema to the English PropBank. Due to some inconsistency annotation errors of the corpus, notably in many complex sentences, we were not able to use all the corpus in our experiments. We focus ourselves in simple sentences which have only one predicate rather than complex sentences with multiple predicates. After extracting sentences, we have a corpus of about 4,860 simple sentences which are annotated with semantic roles.

The semantic roles covered by the Vietnamese PropBank are the following:

- **Core Arguments** (Arg0-Arg4): Arguments define predicate specific roles. These core arguments are similar to those of the English PropBank, however, there are 5 roles instead of 7, compared to the English PropBank.
- **Adjunct Arguments** (ArgM-): There are 20 types of adjuncts, as listed in Table 4.1.
- **Predicate** (V): In Vietnamese, a predicate is not only a verb, but it could be also a noun, an adjective or a preposition.

Role Name	Description	Role Name	Description
ArgM-ADV	general-purpose	ArgM-CAU	cause
ArgM-DIS	discourse marker	ArgM-DIR	direction
ArgM-NEG	negation marker	ArgM-MNR	manner
ArgM-PRD	predication	ArgM-PRP	purpose
ArgM-MOD	modal verb	ArgM-TMP	temporal
ArgM-REC	reciprocal	ArgM-GOL	goal
ArgM-LVB	light verb	ArgM-EXT	extent
ArgM-COM	comitative	ArgM-I	interjection
ArgM-Partice	partice	ArgM-PNC	purpose
ArgM-ADJ	unknown	ArgM-RES	unknown

Table 4.1: Adjunct arguments in Vietnamese

4.2.2 Feature Sets

We use two feature sets in this study. The first one is composed of basic features which are commonly used in SRL system for English. This feature set is used in the SRL system of Gildea and Jurafsky on the FrameNet corpus [8].

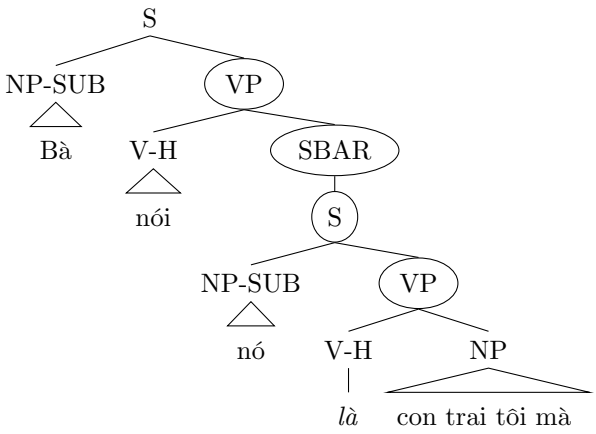


Figure 4.10: Example sentence with predicate *là*

Basic Feature Set

This feature set consists of 6 feature templates, as follows:

1. **Phrase Type**: This is very useful feature in classifying semantic roles because different roles tend to have different syntactic categories. For example, in the sentence in Figure 4.10 *Bà nói nó là con trai tôi mà*, the phrase type of constituent *nó* is *NP*.

2. **Parse Tree Path**: This feature captures the syntactic relation between a constituent and a predicate in a bracketed tree. This is the shortest path from a constituent node to a predicate node in the tree. We use either symbol ↑ or symbol ↓ to indicate the upward direction or the downward direction, respectively. For example, the parse tree path from constituent *nó* to the predicate *là* is *NP*↑*S*↓*VP*↓*V-H*.

3. **Position**: Position is a binary feature that describes whether the constituent occurs after or before the predicate. It takes value *0* if the constituent appears before the predicate in the sentence or value *1* otherwise. For example, the position of constituent *nó* in Figure 4.10 is *0* since it appears before predicate *là*.

4. **Voice**: Sometimes, the differentiation between active and passive voice is useful. For example, in an active sentence, the subject is usually an *Arg0* while in a passive sentence, it is often an *Arg1*. Voice feature is also binary feature, taking value *1* for active voice or *0* for passive voice. The sentence in Figure 4.10 is of active voice, thus its voice feature value is *1*.

5. **Head Word**: This is the first word of a phrase. For example, the head word for the phrase *con trai tôi mà* is *con trai*.

6. **Subcategorization**: Subcategorization feature captures the tree that has the concerned predicate as its child. For example, in Figure 4.10, the subcategorization of the predicate *là* is *VP(V-H, NP)*.

Modified Features and New Features

Preliminary investigations on the basic feature set give us a rather poor result. Therefore, we propose some modified features and novel features so as to improve the accuracy of the system. These features are as follows:

1. **Function Tag**: Function tag is a useful information, especially for classifying adjunct arguments. It determines a constituent's role, for example, the function tag of constituent *nó* is *SUB*, indicating that this has a subjective role.

2. **Partial Parse Tree Path**: Many sentences have complicated structure. It can make parse tree path very long and infrequent. We propose to cut a path from the lowest common ancestor to its predicate, instead of using the full path. For example, the partial path from the constituent *nó* to the predicate *là* in Figure 4.10 is *NP*↑*S*.

3. **Distance**: This feature records the length of the full parse tree path before pruning. This feature helps retaining some information that might be lost when a partial path, instead of a full path, is used. For example, the distance from constituent *nó* to the predicate *là* is *3*.

4. **Predicate Type**: Unlike in English, the type of predicates in Vietnamese is much more complicated. It is not only a verb, but is also a noun, an adjective, or a preposition. Therefore, we propose a new feature which captures predicate types. For example, the predicate type of the concerned predicate is *V-H*.

5. **Word Cluster**: Word clusters have been shown to help improve the performance of many NLP tasks because they alleviate the severity of the data sparseness problem. Thus, in this work we propose to use word cluster features. We first produce distributed word representations (or word embeddings) of Vietnamese words, where each word is represented by a dense, real-valued vector of 50 dimensions, by using a Skip-gram model described in [15, 16]. We then cluster these word vectors into 128 groups using a Gaussian mixture model.[2] A word cluster feature is defined as the cluster identifier of the concerned word.

4.2.3 Experiments and Results

Evaluation Method

We use a 10-fold cross-validation method to evaluate our system. The final accuracy scores is the average scores of the 10 runs.

The evaluation metrics are the precision, recall and F-measure. The precision (P) is the proportion of labelled arguments identified by the system which are correct; the recall (R) is the proportion of labelled arguments in the gold results which are correctly identified by the system; and the F-measure is the harmonic mean of P and R, that is $F_1 = 2PR/(P + R)$.

[2] Actually, there is an additional group for unknown words.

	1-1 Node Mapping	Our Extraction Alg.
Precision	29.53%	81.00%
Recall	45.60%	86.43%
F1	35.84%	83.63%

Table 4.2: Accuracy of two extraction algorithms

Baseline System

In the first experiment, we compare our constituent extraction algorithm to the 1-1 node mapping algorithm. Table 4.2 shows the performance of two extraction algorithms.

We see that our extraction algorithm outperforms significantly the 1-1 node mapping algorithm, in both of the precision and the recall ratios. In particular, the precision of the 1-1 node mapping algorithm is only 29.53%; this means that this method captures many candidates which are not arguments. In contrast, our algorithm is able to identify a large number of correct argument candidates, particularly with the recall ratio of 86.43%. This result clearly demonstrates that we cannot take for granted that a good algorithm for English could also work well for another language of different characteristics.

In the second experiment, we continue to compare the performance of the two extraction algorithms, this time at the final classification step and get the baseline for Vietnamese SRL. The classifier we use in this experiment is a Maximum Entropy classifier.[3] Table 4.3 shows the accuracy of the baseline system.

	1-1 node mapping	Our Extraction Alg.
Precision	52.80%	53.79%
Recall	3.30%	47.51%
F1	6.20%	50.45%

Table 4.3: Accuracy of baseline system

One again, this result confirms that our algorithm is much superior than the 1-1 node mapping algorithm. The F_1 of our baseline SRL system is 50.45%, compared to 6.20% of the 1-1 node mapping system. This result can be explained by the fact that the 1-1 node mapping algorithm has a very low recall ratio, because it identifies incorrectly many argument candidates.

Labelling Strategy

In the third experiment, we compare two labelling strategies for Vietnamese SRL. In addition to the ME classifier, we also try the Support Vector Machine

[3]We use the logistic regression classifier with L_2 regularization provided by the `scikit-learn` software package. The regularization term is fixed at 1.

(SVM) classifier, which usually gives good accuracy in a wide variety of classification problems.[4] Table 4.4 shows the F_1 scores of different labelling strategies.

	ME	SVM
1-step strategy	50.45%	68.91%
2-step strategy	49.76%	68.55%

Table 4.4: Accuracy of two labelling strategies

We see that the SVM classifier outperforms ME the classifier by a large margin. The best accuracy is obtained by using 1-step stragegy with SVM classifier. The current SRL system achieves an F_1 score of 68.91%.

Feature Analysis

In the fourth experiment, we analyse and evaluate the impact of each individual feature to the accuracy of our system so as to find the best feature set for our Vietnamese SRL system. We start with the basic feature set presented previously, denoted by Φ_0 and augment it with modified and new features as shown in Table 4.5. The accuracy of these feature sets are shown in Table 4.6.

Feature Set	Description
Φ_1	$\Phi_0 \cup$ {Function Tag}
Φ_2	$\Phi_0 \cup$ {Predicate Type}
Φ_3	$\Phi_0 \cup$ {Distance}

Table 4.5: Feature sets

Feature Set	Precision	Recall	F1
Φ_0	72.27%	65.84%	68.91%
Φ_1	**76.49%**	**69.65%**	**72.91%**
Φ_2	72.26%	65.87%	68.92%
Φ_3	72.35%	65.86%	68.95%

Table 4.6: Accuracy of feature sets in Table 4.5

We notice that amongst the three features, function tag is the most important feature which increases the accuracy of the baseline feature set by about 4% of F_1 score. The distance feature also helps increase slightly the accuracy. We thus consider the fourth feature set Φ_4 defined as

$$\Phi_4 = \Phi_0 \cup \{\text{Function Tag}\} \cup \{\text{Distance}\}.$$

[4] We use a linear SVM provided in the `scikit-learn` software package with default parameter values.

In the fifth experiment, we modify the feature set Φ_4 by replacing the predicate with its cluster and similarly, replacing the head word with its cluster, replacing the full path with its partial path, resulting in feature sets Φ_5, Φ_6, and Φ_7 respectively (see Table 4.7). The accuracy of these feature sets are shown in Table 4.8.

Feature Set	Description
Φ_5	$\Phi_4 \backslash \{\text{Predicate}\} \cup \{\text{Predicate Cluster}\}$
Φ_6	$\Phi_4 \backslash \{\text{Head Word}\} \cup \{\text{Head Word Cluster}\}$
Φ_7	$\Phi_4 \backslash \{\text{Full Path}\} \cup \{\text{Partial Path}\}$

Table 4.7: Feature sets (continued)

Feature Set	Precision	Recall	F1
Φ_4	76.60%	69.72%	73.00%
Φ_5	**76.86%**	**70.36%**	**73.47%**
Φ_6	72.50%	66.59%	69.41%
Φ_7	76.29%	69.58%	72.78%

Table 4.8: Accuracy of feature sets in Table 4.7

We observe that using the predicate cluster instead of the predicate itself helps improve the accuracy of the system by about 0.47% of F_1 score. For ease of later presentation, we rename the feature set Φ_5 as Φ_8.

In the sixth experiment, we investigate the significance of individual features to the system by removing them, one by one from the feature set Φ_8. By doing this, we can evaluate the importance of each feature to our overall system. The feature sets and their corresponding accuracy are presented in Table 4.9 and Table 4.10 respectively.

Feature Set	Description
Φ_9	$\Phi_8 \backslash \{\text{Function Tag}\}$
Φ_{10}	$\Phi_8 \backslash \{\text{Predicate Cluster}\}$
Φ_{11}	$\Phi_8 \backslash \{\text{Head Word}\}$
Φ_{12}	$\Phi_8 \backslash \{\text{Path}\}$
Φ_{13}	$\Phi_8 \backslash \{\text{Position}\}$
Φ_{14}	$\Phi_8 \backslash \{\text{Voice}\}$
Φ_{15}	$\Phi_8 \backslash \{\text{Subcategorization}\}$
Φ_{16}	$\Phi_{10} \cap \Phi_{15}$

Table 4.9: Feature sets (continued)

We see that the accuracy increases slightly when either the predicate cluster feature (Φ_{10}) or the subcategorization feature (Φ_{15}) is removed. However, removing both of the two features (Φ_{16}) makes the accuracy decrease. For this

Feature Set	Precision	Recall	F1
Φ_8	76.86%	70.36%	73.47%
Φ_9	72.27%	66.12%	69.06%
Φ_{10}	**76.87%**	**70.41%**	**73.50%**
Φ_{11}	72.91%	67.05%	69.86%
Φ_{12}	76.81%	70.36%	73.44%
Φ_{13}	76.41%	70.21%	73.18%
Φ_{14}	76.85%	70.36%	73.46%
Φ_{15}	**76.83%**	**70.51%**	**73.53%**
Φ_{16}	76.70%	70.31%	73.36%

Table 4.10: Accuracy of feature sets in Table 4.9

reason, we remove only the subcategorization feature. The best feature set includes the following features: predicate cluster, phrase type, function tag, parse tree path, distance, voice, position and head word. The best accuracy of our system is 73.53% of F_1 score.

Learning Curve

In the last experiment, we investigate the dependence of accuracy to the size of the training dataset. Figure 4.11 depicts the learning curve of our system when the data size is varied.

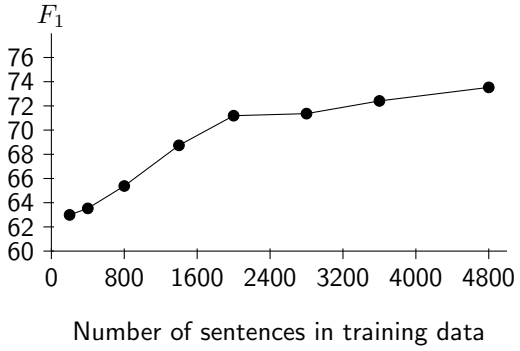


Figure 4.11: Learning Curve

It seems that the accuracy of our system improves only slightly starting from the dataset of about 3,000 sentences. Nevertheless, the curve has not converged, indicating that the system could achieve a better accuracy when a larger dataset is available.

Chapter 5

Conclusion and Future Works

5.1 Conclusion

In this thesis, we have presented the first system for Vietnamese semantic role labelling. Our system achieves a good accuracy of about 73.5% of F_1 score in the Vietnamese PropBank.

We have argued that one cannot assume a good applicability of existing methods and tools developed for English and other Western languages and that they may not offer a cross-language validity. For an isolating language such as Vietnamese, techniques developed for inflectional languages cannot be applied "as is". In particular, we have developed an algorithm for extracting argument candidates which has a better accuracy than the 1-1 node mapping algorithm. We have proposed some novel features which are proved to be useful for Vietnamese SRL, notably predicate clusters and function tags. Our SRL system, including software and corpus, is available as an open source project for free research purpose and we believe that it is a good baseline for the development and comparison of future Vietnamese SRL systems.

5.2 Future Works

In the future, we plan to improve further our system, in the one hand, by enlarging our corpus so as to provide more data for the system. On the other hand, we would like to investigate different models used in SRL, for example joint models [9] and recent inference techniques, such as integer linear programming [19, 22].

Bibliography

[1] C. Aksoy, A. Bugdayci, T. Gur, I. Uysal, and Fazli Can. Semantic argument frequency-based multi-document summarization. In *Proceedings of the 24th of the International Symposium on Computer and Information Sciences*, pages 460–464, Guzelyurt, Turkey, 2009.

[2] Olga Babko-Malaya. PropBank annotation guidelines. Technical report, Colorado University, 2005.

[3] Collin F. Baker, Charles J. Fillmore, and Beau Cronin. The structure of the FrameNet database. *International Journal of Lexicography*, 16(3):281–296, 2003.

[4] Hans C. Boas. From theory to practice: Frame semantics and the design of Framenet. In *Semantisches Wissen im Lexikon*, pages 129–160. Tübingen: Narr., 2005.

[5] Xavier Carreras and Lluís Màrquez. Introduction to the CoNLL-2004 shared task: semantic role labeling. In *Proceedings of the 8th Conference on Computational Natural Language Learning*, Boston, MA, USA, 2004.

[6] Xavier Carreras and Lluís Màrquez. Introduction to the CoNLL-2005 shared task: semantic role labeling. In *Proceedings of the 9th Conference on Computational Natural Language Learning*, pages 152–164, Ann Arbor, MI, USA, 2005.

[7] Janara Christensen, Stephen Soderland, Oren Etzioni, et al. Semantic role labeling for open information extraction. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies*, pages 52–60, Los Angeles, CA, USA, 2010.

[8] Daniel Gildea and Daniel Jurafsky. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288, 2002.

[9] Aria Haghighi, Kristina Toutanova, and Christopher D. Manning. A joint model for semantic role labeling. In *Proceedings of the 9th Conference on Computational Natural Language Learning*, pages 173–176, Ann Arbor, MI, USA, 2005.

[10] Ryu Iida, Mamoru Komachi, Kentaro Inui, and Yuji Matsumoto. Annotating a japanese text corpus with predicate-argument and coreference relations. In *Proceedings of the Linguistic Annotation Workshop*, pages 132–139. Association for Computational Linguistics, 2007.

[11] Chi kiu Lo and Dekai Wu. Evaluating machine translation utility via semantic role labels. In *Proceedings of The International Conference on Language Resources and Evaluation*, pages 2873–2877, Valletta, Malta, 2010.

[12] Peter Koomen, Vasin Punyakanok, Dan Roth, and Wen tau Yih. Generalized inference with multiple semantic role labeling systems. In *Proceedings of the 9th Conference on Computational Natural Language Learning*, pages 181–184, Ann Arbor, MI, USA, 2005.

[13] Phuong Le-Hong, Azim Roussanaly, and Thi-Minh-Huyen Nguyen. A syntactic component for Vietnamese language processing. *Journal of Language Modelling*, 3(1):145–184, 2015.

[14] Lluís Màrquez, Pere Comas, Jesús Giménez, and Neus Catala. Semantic role labeling as sequential tagging. In *Proceedings of the 9th Conference on Computational Natural Language Learning*, pages 193–196, Ann Arbor, MI, USA, 2005.

[15] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*, Scottsdale, Arizona, USA, 2013.

[16] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[17] Thi-Luong Nguyen My-Linh Ha, Viet-Hung Nguyen, Thi-Minh-Huyen Nguyen, Phuong Le-Hong, and Thi-Hue Phan. Building a semantic role annotated corpus for Vietnamese. In *Proceedings of the 17th National Symposium on Information and Communication Technology*, pages 409–414, Daklak, Vietnam, 2014.

[18] Sameer Pradhan, Kadri Hacioglu, Wayne Ward, James H. Martin, and Daniel Jurafsky. Semantic role chunking combining complementary syntactic views. In *Proceedings of the 9th Conference on Computational Natural Language Learning*, pages 217–220, Ann Arbor, MI, USA, 2005.

[19] Vasin Punyakanok, Dan Roth, Wen tau Yih, and Dav Zimak. Semantic role labeling via integer linear programming inference. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 1346–1352, University of Geneva, Switzerland, 2004.

[20] Dan Shen and Mirella Lapata. Using semantic roles to improve question answering. In *Proceedings of Conference on Empirical Methods on Natural Language Processing and Computational Natural Language Learning*, pages 12–21, Prague, Czech Republic, 2007.

[21] Mihai Surdeanu and Jordi Turmo. Semantic role labeling using complete syntactic analysis. In *Proceedings of the 9th Conference on Computational Natural Language Learning*, pages 221–224, Ann Arbor, MI, USA, 2005.

[22] Oscar Täckström, Kuzman Ganchev, and Dipanjan Das. Efficient inference and structured learning for semantic role labeling. *Transactions of the Association for Computational Linguistics*, 3:29–41, 2015.

[23] Hayato Tagami, Shinsuke Hizuka, and Hiroaki Saito. Automatic semantic role labeling based on Japanese FrameNet–A Progress Report. In *Proceedings of Conference of the Pacific Association for Computational Linguistics*, pages 181–186, Hokkaido University, Sapporo, Japan, 2009.

[24] Nianwen Xue. Annotation guidelines for the chinese proposition bank. Technical report, Brandeis University, 2007.

[25] Nianwen Xue and Martha Palmer. Automatic semantic role labeling for Chinese verbs. In *Proceedings of International Joint Conferences on Artificial Intelligence*, pages 1160–1165, Edinburgh, Scotland, UK, 2005.