Towards Building Semantic Role Labeler for Indian Languages

Maaz Anwar Nomani[♣] and Dipti Misra Sharma[♣]

FC Kohli Center on Intelligent Systems (KCIS), IIIT-Hyderabad, India[♣]
maaz.anwar@research.iiit.ac.in, *dipti@iiit.ac.in*,

Abstract

We present a statistical system for identifying the *semantic relationships or semantic roles* for two major Indian Languages, *Hindi* and *Urdu*. Given an input sentence and a predicate/verb, the system first identifies the arguments pertaining to that verb and then classifies it into one of the semantic labels which can either be a DOER, THEME, LOCATIVE, CAUSE, PURPOSE etc. The system is based on 2 statistical classifiers trained on roughly 130,000 words for Urdu and 100,000 words for Hindi that were hand-annotated with semantic roles under the *PropBank* project for these two languages. Our system achieves an accuracy of *86%* in identifying the arguments of a verb for Hindi and *75%* for Urdu. At the subsequent task of classifying the constituents into their semantic roles, the Hindi system achieved *58%* precision and *42%* recall whereas Urdu system performed better and achieved *83%* precision and *80%* recall. Our study also allowed us to compare the usefulness of different linguistic features and feature combinations in the semantic role labeling task. We also examine the use of statistical syntactic parsing as feature in the role labeling task.

Keywords: PropBank, Semantic Role labeling, Treebank, Dependency relations

1. Introduction

We introduce a statistical *semantic role labeler* for *Hindi* and *Urdu*, two major Indian languages. A *Semantic Role labeler* (henceforth, SRL) automatically marks the arguments/valency of a predicate in a sentence. The proposed system is based on supervised machine learning approach on Hindi and Urdu PropBanks which are being built for these languages.

The approach is a 2-stage architecture in which, first the arguments pertaining to a predicate in a sentence are identified by the system and then those identified arguments are classified into one of the PropBank semantic labels. Our system uses a basic Logistic Regression machine learning algorithm (Pedregosa et al., 2011) for identifying the predicates and Support Vector Machines (Pedregosa et al., 2011) to classify the arguments of a predicate into semantic labels. We have used 10 linguistic features, 6 of which have been derived from literature and 4 are novel features. Among the new features, we have used the *karaka relations/dependency relations* (described later) which turns out to be the most discriminative feature among all. We have also experimented with the automatic parses which are used as features and derived from state-of-art Hindi and Urdu parsers. To the best of our knowledge, this is the first such attempt of building a semantic role labeler for any Indian language.

This paper is arranged as follows: Section 2 gives a brief introduction about *Semantic Parsing*. In Section 3, we give the data statistics of the language resources used. Section 4 talks about related work in semantic role labeling. In section 5, we briefly describe the Hindi and Urdu Propbanks and Treebanks which are being built for these two Indian languages. Section 6 presents the Semantic Role labeler in detail which includes our approach, its architecture, classifiers and features used. We also describe the baseline features, their impact on Indian language datasets and new features incorporated in the systems. In section 7, we describe in detail the different experiments done and show their results. Section 8 throws light on the impact of using automatic parses as features in the semantic role labeling task. In section 9, we thoroughly put the error analysis forward along with some corpus examples showing erroneous role labeling and discuss possible reasons for it. We conclude the paper in section 10.

2. Semantic Parsing

Semantic analysis of natural text and languages has always been an intriguing research area in NLP. Automatic semantic analysis of texts thus becomes a challenging as well as interesting task which includes tasks like Semantic Parsing, Shallow Semantic Parsing and Semantic Role labeling. Automatic and accurate tools that can predict naturally occurring arguments for a given predicate in a predicate-argument structure of a sentence can be efficiently used for Semantic Parsing, Summarization, Information Extraction tasks, wider NLP areas like Machine Translation and Question Answering and modern NLP challenges like parsing code-mixed data.

Semantic Parsing is essentially the research investigation of identifying WHO did WHAT to WHOM, WHERE, HOW, WHY and WHEN etc. in a sentence and adding a layer of semantic annotation in a sentence produces such a structure. Two major Indian languages, Hindi and Urdu are having such a resource in the form of *PropBank (also called Proposition Banks)* which establishes a layer of semantic representation in a Treebank already annotated with Dependency labels.

Proposition Bank (Kingsbury and Palmer, 2003) is a corpus in which the arguments of each verb predicate (simple or complex) are marked with their semantic roles.

3. Data

Table 1 shows the data statistics of the language resources. We have used a part of these language resources for SRL task.

In the experiments reported here, we first performed the argument identification task which is a binary classification problem (either 'argument' or 'not an argument') by taking 130,000 tokens as training data and 30,000 as test data from Urdu PropBank. We took around 100,000 tokens as training data from Hindi PropBank and 20,000 as test data. Subsequently, we trained a multi-class classifier, SVM on the same training set, using a well defined feature-set described later.

Language Resource	Tokens	Sentences	Predicates	pbrel
Urdu Treebank	~200,000	~8,000	-	-
Hindi Treebank	~350,000	~14,000	-	-
Urdu Propbank	~180,000	~7,000	~2,200	~7,000
Hindi Propbank	~300,000	~11,000	~3,200	~12,000

Table 1: **Hindi and Urdu resources statistics**

4. Related Work

The last decade has seen an exhaustive research in semantic parsing for different languages. Gildea and Jurafsky (2002) started out the work on semantic role labeling on 2001 release of English Propbank. Surdeanu et al. (2003) build a decision tree classifier for predicting the semantic labels. Gildea and Hockenmaier (2003) uses features from Combinatory Categorial Grammar (CCG) which is a form of dependency grammar. Chen and Rambow (2003) used a decision tree classifier and additional syntactic and semantic representations extracted from Tree Adjoining Grammar (TAG). Swier and Stevenson (2004) talks of a novel bootstrapping algorithm for identifying semantic labels. (Cohn and Blunsom, 2005) applied conditional random fields (CRFs) to the semantic role labeling task. Xue and Palmer (2004) experimented with different linguistic features related to role labeling task.

Xue and Palmer (2005) built a role labeler for Chinese by showing that verb classes, induced from the predicate-argument information in the frame files helps in semantic role labeling. Johansson and Nugues (2006) came up with a FrameNet-based semantic role labeling system for Swedish text. Toutanova et al. (2005) used joint learning amd joint modeling of argument frames of verbs to improve the overall accuracy of semantic role labeler.

5. The PropBank and The Treebank

Indian Languages are very morphologically rich languages and *Treebanks* for Hindi and Urdu are available which include syntactico-semantic information in the form of dependency annotations as well as lexical semantic information in the form of predicate-argument structures, thus forming *PropBanks*. Hindi and Urdu PropBanks are part of a *multi-dimensional and multi-layered* resource creation effort for the Hindi-Urdu language (Bhatt et al., 2009).

The Hindi Dependency Treebank (HDT) and *Urdu Dependency Treebank* (UDT) are built following the CPG (Computational Paninian Grammar) framework (Begum et al., 2008). PropBank establishes a layer of *semantic* representation in the Treebank which is already annotated with Dependency labels or phrase structures (as in Penn Treebank for English). Capturing the *semantics* through predicate-argument structure involves quite a few challenges peculiar to each predicate type because the syntactic notions in which the verb's arguments and adjuncts are realized can vary based on the senses. Table 2 shows the PropBank labels for Hindi and Urdu and these labels are also used for building the Semantic Role labeler.

Propbank labels or semantic labels are closely associated with the karaka relations (Vaidya et al., 2011) (described later) in their structure though the former are defined on a verb-by-verb basis.

Label	Description
ARG0	Agent, Experiencer or doer
ARG1	Patient or Theme
ARG2	Beneficiary
ARG3	Instrument
ARG2-ATR	Attribute or Quality
ARG2-LOC	Physical Location
ARG2-GOL	Goal
ARG2-SOU	Source
ARGM-PRX	noun-verb construction
ARGM-ADV	Adverb
ARGM-DIR	Direction
ARGM-EXT	Extent or Comparision
ARGM-MNR	Manner
ARGM-PRP	Purpose
ARGM-DIS	Discourse
ARGM-LOC	Abstract Location
ARGM-MNS	Means
ARGM-NEG	Negation
ARGM-TMP	Time
ARGM-CAU	Cause or Reason

Table 2: **Hindi and Urdu PropBank labels with definitions**

6. Semantic Role labeler

This section gives a thorough analysis on our approach towards role labeling task. At first, one can comprehend the semantic role labeling task as a simple straight forward multi-class classification problem in which the semantic labels are classified directly without identifying them. But such a simple approach will not work due to various reasons.

(Xue and Palmer, 2004) observes that for a given predicate, many constituents in a syntactic tree are not its semantic argument. So, non-argument count overwhelms the argument count for the given predicate and classifiers will not be efficient in predicting the right argument or in classifying them. This problem was encountered in both Hindi and Urdu data-sets as well.

Therefore, we adopted a 2-stage approach of first identifying the arguments and subsequently classifying the identified semantic arguments into one of the semantic labels. The second approach of directly classifyng the arguments was also experimented by us and we show by statistics that such a forthright approach will not work for Hindi and Urdu as well.

- **Approach I**
- *Argument Identification* - This is the sub-task of identifying a constituent in a dependency tree which represents the argument for the predicate in the argument-predicate structure in a Hindi or Urdu PropBank sentence. The identification is done by a binary classifier ('argument' and 'not an argument'). The classifier has to predict whether the constituent in the tree is an argument of that verb or not.
- *Argument Classification* - After identification of arguments is done, the identified arguments pertaining to that verb are assigned appropriate semantic labels by a multi-class classifier which is trained on multiple semantic labels present in the training data. We have used Gold Parses as well as Automatic Parses from Hindi and Urdu parsers as features in this task.
- **Approach II**
- *Argument Identification and Argument Classification* -

  We performed direct classification of arguments in Hindi and Urdu data and found that the results were not effective. We found that due to overwhelming number of non-arguments present in the corpus, the classifier failed to classify the arguments correctly. For example, in Urdu SRL system, the arguments for which gold parses were used as feature, were getting classified with a precision of only 20.98% and non-arguments were getting classified with a precision of 79.71%. For automatic parses, arguments got classified with a precision of 17.12% and non-arguments got classified with 82.87% precision.

6.1. SRL Architecture

Figure 1 is the flow diagram of Semantic Role Labeler for Hindi and Urdu. It is a 2-stage architecture in which at first the arguments of a predicate in a sentence are identified by a binary classifier and then in the subsequent stage only these identified arguments are classified as one of the semantic labels by a multi-class classifier.

6.2. Features: A critical look

Hindi and Urdu are free-word order languages i.e. the word order is somewhat flexible in these languages. For this reason, they are sometimes called as *SOV* languages (subject, object, verb). Hindi and Urdu employs postpositions and not prepositions (as in English), the resulting word order of postpositional phrases can be the reverse of prepositional phrases in English. We decided to use linguistic features in our Semantic Role Labeler based on our intuition that they might help the labeler in taking decisions to identify and classify the semantic arguments of a predicate in the sentence.

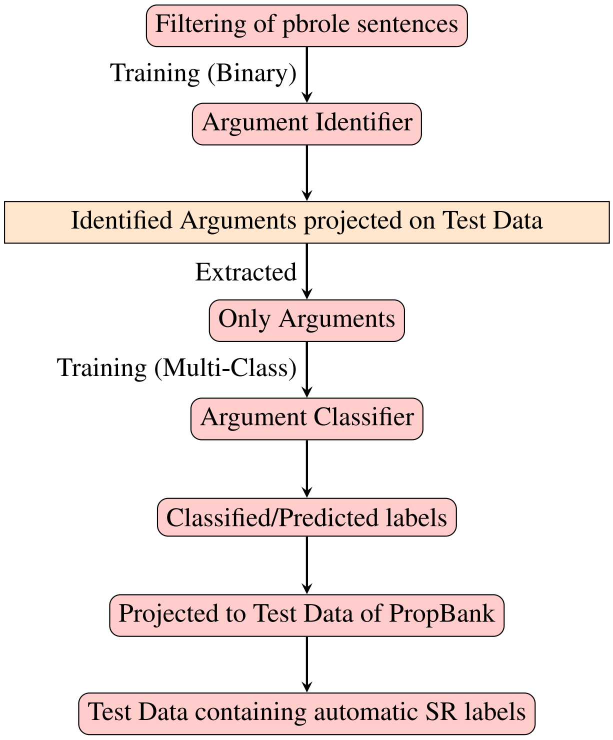

Figure 1: **SRL Architecture**

6.2.1. Baseline Features

The task of identifying and classifying the arguments of a verb is done at chunk/phrase level. Most of the previous works done in Semantic Parsing have shown that 'head' of the chunk is a very critical feature because chunks/phrases with certain head words are likely to be arguments of the verb and also certain types of arguments. This implies that head word of a chunk is a very discriminative feature for both the tasks.
For our Indian languages SRL, we have used a combination of baseline features introduced by (Gildea and Jurafsky, 2002) and (Xue and Palmer, 2004) for English because we wanted to see the impact of these standard features in context of the Indian languages and how these features perform w.r.t free word order languages.

Baseline Features for Indian Languages
Predicate - predicate itself
Head word - syntactic head of the chunk/phrase
Head-word POS - Its POS category
Phrase Type - syntactic category (NP, VP etc. of the phrase)
Predicate + *Phrase Type* - combinational feature
Predicate + *Head word* - combinational feature

Table 3: **Baseline feature template**

6.2.2. New Features

After analyzing the PropBanks of both the languages, we came up with certain new features for which we had an intuition that they will contribute significantly towards this task. These are discussed below:

- *Dependency/karaka relation* - We experimented with the Dependency labels or karaka relations present in the Hindi and Urdu PropBanks. A sentence is composed of a primary modified which is the root in the dependency tree of the sentence and this modified is occasionally the main verb of the sentence. The words modifying this verb participate in the action specified by the verb. The participant relation of the verb with these words/elements are called *karaka relations*.

  There are in total 43 karaka relations for Hindi and Urdu as specified by (Begum et al., 2008). Table 4 shows some of the major karaka or dependency relations. These dependency relations helps to identify the syntactico-semantic relations in a sentence. Since, PropBank labels illustrates the semantic relations between the constituents, we had this intuition that using the karaka relations as features in our classifier to predict the arguments of a verb will have an impact on our system performance. Also, Vaidya et al. (2011) has shown that there is a correlation between dependency and predicate argument structures for Hindi. Since Hindi and Urdu are linguistically similar languages, we extracted the dependency/karaka relations of both the language Treebanks and used them as feature in isolation as well as in the system.

Dependency Labels	**Description**
k1	karta - doer/agent/subject
k2	karma - object/patient
k1s	noun complement
k4	sampradana - recipient
k3	karana - instrument
k4a	experiencer
k2p	goal, destination
k5	apadana - source
k7	location elsewhere
k7p	location in space
k7t	time
adv	adverbs
rh	reason
rd	direction
rt	purpose

Table 4: **Major karaka relations or Dependency labels of Hindi and Urdu Treebank**

From Tables 5 and 6, it is clear that dependency relations are indeed the most discriminative feature among all the features used by showing high precision and recall for both languages. The high percentage of precision and recall in the Argument Classification task can be accounted from the fact that dependency structure and predicate-argument structure are somewhat similar structures and dependency labels and propbank labels are similar in orientation. Therefore, using dependency structures as features to predict and classify the arguments of a predicate of Hindi and Urdu is very influential.

- *Named Entities (NEs)* - Named Entities also contribute towards the identification task but it doesn't contributes heavily towards the classification task. The reason being NEs are generally arguments of a predicate. This feature has been used in the past by (Pradhan et al., 2004) for English but we have used it differently by combining it with the POS category of the NE. As shown is Tables 5 and 6, NEs have a decent precision of 63% and 61% for Urdu and Hindi respectively when used in isolation for the argument identification task. But it is not able to discriminate between the class of semantic labels to which a identified argument belong.

- *Head + Phrase type* - A combination of head word of the chunk/phrase and syntactic category of the phrase.

- *Head POS + Phrase type* - A combination of the POS category of the head word of the chunk/phrase and syntactic category of the phrase.

  Abbreviations Used: 'P'=Precision, 'R'=Recall, 'PT'=Phrase Type, 'HW'=Head Word, 'POS'=Part-of-speech'

Feature	**Feature wise performance (Urdu)**					
	Argument Id.			**Argument Classification**		
	P	R	f-score	P	R	f-score
Predicate	51	72	60	00	01	00
Head-word	71	73	72	25	14	18
Head-POS	51	72	60	06	18	09
PhraseType	51	72	60	08	08	08
HeadPOS-PT	61	72	66	06	18	09
Head-PT	71	74	72	25	14	18
Predicate-HW	72	74	73	01	01	01
Predicate-PT	64	69	66	02	00	00
NEs	63	65	64	23	18	20
Dependency	**78**	**79**	**78**	**87**	**85**	**86**

Table 5: **Individual feature performance (Urdu)**

- We also experimented with certain other features like *chunk boundary* and *sentence boundary* of the sentences. We had an intuition that these boundaries will help the classifier learn more effectively and overlap of the 2 sentences will be minimized. But it turned out that both the features had a major negative impact on the overall results of the system, so we scrapped these two features.
  The reason for this dip can be accounted from the fact that chunk boundary and sentence boundary are merely integers for the classifier as there are 'n' numbers for 'n' sentences/chunks which lack any significant information.

6.2.3. Classifiers used

We used a basic Logistic Regression classifier and trained it on Hindi and Urdu data-sets thus building a binary classifier which identifies arguments as 'Arguments' or 'Not-an-Argument'. For multi-class classification in argument classification task, we implemented LinearSVC class of SVM (Pedregosa et al., 2011) for performing the 'one-vs-the-rest' multi-class strategy. All the parametres of Linear SVC were set to default at the time of training of Hindi and Urdu data-sets.

Feature	Feature wise performance (Hindi)					
	Argument Id.			Argument Classification		
	P	R	f-score	P	R	f-score
Predicate	41	64	50	41	61	49
Head-word	70	70	70	20	04	07
Head-POS	34	58	43	07	08	07
PhraseType	34	58	43	05	07	06
HeadPOS-PT	34	58	43	14	13	13
Head-PT	67	66	66	10	04	06
Predicate-HW	65	65	65	10	04	06
Predicate-PT	41	64	50	04	06	05
NEs	61	64	62	20	16	17
Dependency	**88**	**87**	**87**	**52**	**49**	**50**

Table 6: **Individual feature performance (Hindi)**

7. Experiments and Results

In this section, we show the results and extent of the baseline features (standard features) in Semantic Role Labeling in the experiments on the Indian data sets. We present results of baseline features for both the task and add new features subsequently in next step. We have laid out the experiments in a way that each experiment conveys the extent of usefulness and information gain of the feature in both the tasks. For each experiment, the settings of the SVM and features used in the previous experiment were retained. The baseline features performs well on the argument identification task of both languages with a precision of 71% and 67% for Hindi and Urdu respectively. The classification task of the identified arguments is not profited much by the baseline features for both the languages as can be seen from Tables 7 and 8.

Feature	Hindi system Feature Template					
	Argument Id.			Argument Classification		
	P	R	f-score	P	R	f-score
Baseline	71	71	66	29	14	15
+Dependency	**86**	**86**	**86**	**56**	**38**	**41**
+Head-PT	85	85	86	56	38	41
+HeadPOS-PT	84	84	84	57	40	42
+NEs	85	84	84	58	42	49

Table 7: **Hindi system performance**

As expected, the best result for both the tasks is obtained on incorporating dependency relations information, which further strengthens our conjecture on the importance of dependency labels in semantic role labeling. All 4 experiments shows the effectiveness of the features for Hindi and Urdu data sets. By incorporating NEs in the system, there is an increase in the identification task of Urdu and marginal increase in Hindi. The classification task is also benefitted by a marginal precision increment of 1% in both the data sets and recall of 2% in Hindi.

However, 'Head + Phrase Type' and 'HeadPOS + Phrase Type' features had a *negative* impact on the performance of classification task for both the sets. The identification task was benefitted by every new feature in Hindi with dependency relations contributing the most to the final result of the system.

Feature	Urdu system Feature Template					
	Argument Id.			Argument Classification		
	P	R	f-score	P	R	f-score
Baseline	67	70	68	16	14	15
+Head-PT	65	69	67	15	12	13
+HeadPOS-PT	66	69	67	15	12	13
+Dependency	**72**	**74**	**73**	**82**	**80**	**81**
+NEs	75	71	73	83	80	81

Table 8: **Urdu system performance**

8. Using Automatic Parses

So far, we have shown and discussed results using hand-corrected parses of the Hindi and Urdu PropBank. However, if we are building a NLP application, the SRL will have to extract features from automatic parses of the sentences. Also, after getting the insights of the influence of hand-corrected dependency relations, we wanted to analyse the same using automatic parses and evaluate our semantic role labeler on it. These automatic parses are generated by the parser of respective language.

To come up with such an evaluation, we used the Malt Parser (Nivre et al., 2006) on the same test data described above for both languages. Malt Parser requires data in CONLL format. By using state-of-art Urdu Parser (Bhat et al., 2012), we extracted the automatic parses from the Urdu test data and used them as feature for our identification and classification tasks.

Parsing accuracy is measured by LAS, UAS and LA which stands for *Labeled Attachment score*, *Unlabeled Attachment score* and *Label Accuracy* respectively. LAS is the percentage of words that are assigned the correct head and dependency label; UAS is the percentage of words with the correct head, and the label accuracy (LA) is the percentage of words with the correct dependency label.

Measure	Percentage (%)
LAS	76.61
UAS	88.45
LA	80.03

Table 9: **Parsing performance of the Urdu dependency parser**

By adding automatic parses to the baseline features of Urdu system, we observed a marginal increase in the argument identification task and comparable improvement in classification task in Table 10. The decent jump in the precision of the classification task while using automatic parses as compared to using gold parses can be attributed from the fact that gold parses (Urdu PropBank) have been manually validated by annotators whereas Urdu parser tends to give certain incorrect parses as can be seen from Table 9.

Feature	Automatic Parses (Urdu)					
	Argument Id.			Argument Classification		
	P	R	f-score	P	R	f-score
Baseline	67	70	68	16	14	15
+Automatic Parses	68	66	67	62	59	60

Table 10: **Urdu system with Automatic Parses**

The same process was repeated by using the state-of-art Hindi Parser (Kosaraju et al., 2010) on Hindi test data.

Measure	Percentage (%)
LAS	88.19
UAS	94.00
LA	89.77

Table 11: **Parsing performance of the Hindi dependency parser**

The usage of automatic parses of Hindi have similar impact on the Hindi SRL system. Table 12 shows the performance devaluation when automatically generated parses are used.

Feature	Automatic Parses (Hindi)					
	Argument Id.			Argument Classification		
	P	R	f-score	P	R	f-score
Baseline	71	71	66	29	14	15
+Automatic Parses	73	74	73	41	28	33

Table 12: **Hindi system with Automatic Parses**

9. Error Analysis and Discussion

In this section, we present a thorough error analysis of the Semantic Role Labeling task for Hindi and Urdu by taking a sub-set of test data for error analysis. Tables 14 and 15 are the *confusion matrices* for Hindi and Urdu SRL respectively. As Hindi and Urdu have postpositional phrases, the postpositions or case markers play a very significant role in deciding the semantic role for an argument. These postpositions may not be helpful cues in identifying the argument but they are when it comes to classifying the arguments.

There are 6 pre-dominant case markers in Hindu and Urdu viz. 'ne', 'ko', 'mein', 'se', 'ki', 'par'. Table 13 gives the definition of Hindi and Urdu case markers. These case markers provide important information about the token preceding it and the nature of the chunk/phrase in which they are present. For e.g. in both Hindi and Urdu, the case clitic '*ne*' is often indicative of an agent, and as 'ARG0' is mostly associated with 'agentivity', this can sometimes provide a clue about ARG0. Same is the case with '*mein*' which most of the times gives the 'locative' information in the chunk/phrase. Some post-positions are very ambigous in nature and here is when error in classifying the arguments starts. We show in the examples below how 'ko' and 'mein' post-position can take multiple types of semantic labels and therefore the classifier is not very discriminative during multi-class classification.

Urdu Case marker	Meaning
ne	Ergative
mein	Locative
par	Locative
ko	Dative/Accusative
se	Instrumental/Ablative
ki	Genetive

Table 13: **Hindi and Urdu Case markers**

As can be seen from confusion matrix of Hindi, the pair of labels in which the Hindi SRL is having the maximum confusion is 'ARG1' and 'ARG0'. 'ARG0' label is given to those arguments of verb which shows agentivity or volationality to do work whereas ARG1 is the theme or object arguments. Another frequent confusion which the Hindi SRL has is between 'ArgM-LOC' and 'ARG0'. The confusion between 'ARGM-LOC' and 'ARG2-LOC' can be attributed to the fact that both the labels depicts locative information though ARGM-LOC shows abstract location whereas ARG2-LOC has physical location information.

	Arg0	Arg1	Arg2	Arg2LOC	Arg2SOU	Arg2ATR	ArgADV	ArgMLOC	ArgDIR
Arg0	23	03	02						
Arg1	21	21	04			05	02	03	
Arg2			01	01				01	
Arg2LOC		02						02	
Arg2SOU					02				
Arg2ATR	03	01				07			
ArgMADV		01							
ArgMLOC	05	01		04	01			16	
ArgMDIR									02

Table 14: **Confusion Matrix of Propbank labels for Hindi SRL**

For Urdu, similar confusion is observed between 'ARG0'-'ARG1' and 'ARGM-LOC'-'ARG2-LOC' pairs. Such confusions can be reduced by coming up with linguistic features which are aimed at solving ARG0 and ARG1 confusion because it is the most frequent erroneous pair for Indian languages SRL. These errors also occur due to the presence of ambigous case markers with the arguments of a verb and therefore the classifier is unable to discriminate between the two labels.

Another interesting confusion pair is ARG1:ARG2-ATR. There are instances in the training set in which the arguments of a verb are annotated with 'ARG1' or 'ARG2-ATR' based on the context and sentence structure.

	Arg0	Arg1	Arg2	Arg2LOC	Arg2SOU	Arg2ATR	ArgADV	ArgMLOC	ArgDIR
Arg0	45	02	02	01					
Arg1	08	48	03			05	02	04	
Arg2	01		07					01	
Arg2LOC				10	01		01	03	
Arg2SOU		04	02	01	01	04			
Arg2ATR		01							
ArgMADV		01					12		
ArgMLOC			01					36	
ArgMDIR									03

Table 15: **Confusion Matrix of Propbank labels for Urdu SRL**

We extracted certain sentences from the test set of Hindi and Urdu and analyzed how our SRL is performing in comparison with the gold test data. These sentences are shown below:

(1) *(iss dauran)*
this period
[gold:ARGM-TMP; SRL: ARGM-TMP]
his
(uski) *(putri ko)*
daughter DAT man
[gold: ARG1; SRL: ARG1] *(yuwak)*
roam live
[gold: ARG0; SRL: ARG0] **(ghumata)**
be-PAST.
[predicate] *(rehta tha).*

'During this period, the man used to roam around his daughter.' ***HINDI TEST***

(2) *(idhar)* **[gold: ARGM-DIR; SRL: ARGM-DIS]**
here Assam
(Assam mein)
LOC BJP
[gold: ARGM-LOC; SRL: ARGM-LOC] *(BJP)*
main party
[gold: ARG1; SRL: ARG0] *(mukhya party*
manner LOC emerge
ke) *(roop mein)*
live be-PRS.
[gold: ARGM-MNR; SRL: ARGM-MNR]

(ubhar) *[predicate]* *(rahi)* *(hai).* ***URDU TEST***

'Here, in Assam, BJP is emerging as the main party.'

(3) *(Jwala ne)* *(bataya)* *(ki)* *(central market*
Jwala ERG said that central market
ke muqaable mein)
comparison LOC here
[gold: ARGM-EXT; SRL: ARGM-LOC]
rate
(yahan)
economical
[gold: ARG2-LOC; SRL: ARG2:LOC] *(rate)*
live be-PRS
[gold: ARG1; SRL: ARG1] *(munaasib)*

[gold: ARGM-PRX; SRL: ARGM:PRX] *(rehta*

hai) *[predicate].* ***HINDI TEST***

'Jwala said that rate here is economical as compared to central market.'

(4) *(Barack Obama ko)*
Barack Obama DAT
[gold: ARG0-GOL; SRL: ARG2] *(ehsaas)*
feel be-PRS
[gold: ARG1; SRL: ARG2-ATR] *(hai)* *(ki)*
thathis party election
(unki) *(party)* **[gold: ARG0; SRL: ARG1]**
LOC 2009 of
(intekhabaat mein)
miracle not
[gold: ARGM-LOC; SRL: ARGM-LOC] *(2009*
see get.
ka) *(karishma)* *(nahi)* *(dikha)* *[predicate]*

(paaegi). ***URDU TEST***

'Barack Obama feels that his party will not be able to repeat the miracle of 2009 in the election.'

In sentence 3, *'central market ke muqaable mein'* is 'ARGM-EXT' in gold test data as it conveys the meaning of 'comparison' with some entity. But Hindi SRL gives it ARGM-LOC label. One possible reason for this may be that 'mein' case marker generally occurs more frequently with ARGM-LOC or ARG2-LOC label in the *locative* sense in the training data as compared to *comparison*. In sentence 4, *'Barack Obama ko'* gets ARG2 label by Urdu SRL instead of ARG0-GOL because of the same reasons cited in the above example. Here, 'ko' case marker more frequently occurs with ARG2 label i.e. *beneficiary* in the training data as compared to *experiencer* here.

10. Conclusion

In this work, we have presented a supervised semantic role labeler for Hindi and Urdu. We have used linguistic features like predicate, head, head-POS, phrase type etc. and combinations of certain features. In all, 10 features have been used to guide the classifiers in predicting, identifying and classifiyng the arguments of a verb.
We further used the gold parses (dependency parses) as a feature which had a solid impact on the overall accuracy of the systems. Subsequently, we extracted the automatic parses by using state-of-art Hindi and Urdu parsers and used them as features in our SRL. We believe that dependency relations are the best features for classifying the semantic arguments of a predicate because of their close proximity with semantic labels.

11. Acknowledgements

We would like to thank Mr. Mujadia Vandan for the discussions which we had with him to get a better insight of the task.

12. Bibliographical References

Begum, R., Husain, S., Dhwaj, A., Sharma, D. M., Bai, L., and Sangal, R. (2008). Dependency annotation scheme for indian languages. In *IJCNLP*, pages 721–726. Citeseer.

Bhat, R. A., Jain, S., and Sharma, D. M. (2012). Experiments on dependency parsing of urdu. In *The 11th International Workshop on Treebanks and Linguistic Theories*.

Bhatt, R., Narasimhan, B., Palmer, M., Rambow, O., Sharma, D. M., and Xia, F. (2009). A multi-representational and multi-layered treebank for hindi/urdu. In *Proceedings of the Third Linguistic Annotation Workshop*, pages 186–189. Association for Computational Linguistics.

Chen, J. and Rambow, O. (2003). Use of deep linguistic features for the recognition and labeling of semantic arguments. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 41–48. Association for Computational Linguistics.

Cohn, T. and Blunsom, P. (2005). Semantic role labelling with tree conditional random fields. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, pages 169–172. Association for Computational Linguistics.

Gildea, D. and Hockenmaier, J. (2003). Identifying semantic roles using combinatory categorial grammar. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 57–64. Association for Computational Linguistics.

Gildea, D. and Jurafsky, D. (2002). Automatic labeling of semantic roles. *Computational linguistics*, 28(3):245–288.

Johansson, R. and Nugues, P. (2006). A framenet-based semantic role labeler for swedish. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 436–443. Association for Computational Linguistics.

Kingsbury, P. and Palmer, M. (2003). Propbank: the next level of treebank. In *Proceedings of Treebanks and lexical Theories*, volume 3. Citeseer.

Kosaraju, P., Kesidi, S. R., Ainavolu, V. B. R., and Kukkadapu, P. (2010). Experiments on indian language dependency parsing. *Proceedings of the ICON10 NLP Tools Contest: Indian Language Dependency Parsing*.

Nivre, J., Hall, J., and Nilsson, J. (2006). Maltparser: A data-driven parser-generator for dependency parsing. In *Proceedings of LREC*, volume 6, pages 2216–2219.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830.

Pradhan, S. S., Ward, W., Hacioglu, K., Martin, J. H., and Jurafsky, D. (2004). Shallow semantic parsing using support vector machines. In *HLT-NAACL*, pages 233–240.

Surdeanu, M., Harabagiu, S., Williams, J., and Aarseth, P. (2003). Using predicate-argument structures for information extraction. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 8–15. Association for Computational Linguistics.

Swier, R. S. and Stevenson, S. (2004). Unsupervised semantic role labelling. In *Proceedings of EMNLP*, volume 95, page 102.

Toutanova, K., Haghighi, A., and Manning, C. D. (2005). Joint learning improves semantic role labeling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 589–596. Association for Computational Linguistics.

Vaidya, A., Choi, J. D., Palmer, M., and Narasimhan, B. (2011). Analysis of the hindi proposition bank using dependency structure. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 21–29. Association for Computational Linguistics.

Xue, N. and Palmer, M. (2004). Calibrating features for semantic role labeling. In *EMNLP*, pages 88–94.

Xue, N. and Palmer, M. (2005). Automatic semantic role labeling for chinese verbs. In *IJCAI*, volume 5, pages 1160–1165. Citeseer.