A Dataset for Semantic Role Labelling of Hindi-English Code-Mixed Tweets

by

riya.pal , Dipti Misra Sharma

in

57th Annual Meeting of the Association for Computational Linguistics (ACL)
(*ACL-2019*)

Florence, Italy

Report No: IIIT/TR/2019/-1

Centre for Language Technologies Research Centre
International Institute of Information Technology
Hyderabad - 500 032, INDIA
July 2019

A Dataset for Semantic Role Labelling of Hindi-English Code-Mixed Tweets

Riya Pal and **Dipti Misra Sharma**
Kohli Center on Intelligent Systems (KCIS)
International Institute of Information Technology, Hyderabad (IIIT-Hyderabad)
Gachibowli, Hyderabad, Telangana - 500032, India
riya.pal@research.iiit.ac.in
dipti@iiit.ac.in

Abstract

We present a data set of 1460 Hindi-English code-mixed tweets consisting of 20,949 tokens labelled with Proposition Bank labels marking their semantic roles. We created verb frames for complex predicates present in the corpus and formulated mappings from Paninian dependency labels to Proposition Bank labels. With the help of these mappings and the dependency tree, we propose a baseline rule based system for Semantic Role Labelling of Hindi-English code-mixed data. We obtain an accuracy of 96.74% for Argument Identification and are able to further classify 73.93% of the labels correctly. While there is relevant ongoing research on Semantic Role Labelling (SRL) and on building tools for code-mixed social media data, this is the first attempt at labelling semantic roles in Hindi-English code-mixed data, to the best of our knowledge.

1 Introduction

In recent times, social media has gained a lot of popularity and serves as a medium for people across the globe to communicate and express their opinions. Forums like Facebook and Twitter are used excessively for this purpose. Increasing availability of such resources online provide a large corpus and subsequently the need for linguistic analysis and tools for automated understanding of this data. Code-mixing is a phenomenon observed largely in social media text. It refers to *"the embedding of linguistic units such as phrases, words and morphemes of one language into an utterance of another language "*(Myers-Scotton, 1993). It is usually an intra-sentential phenomenon observed in multilingual societies in colloquial as well as online usage.

Benchmark NLP tools are majorly based on monolingual corpora which strictly follow the patterns and conform to the rules of the given language in terms of structure, syntax, morphology and so on. However, social media data deviate from these rules. Hence, numerous technologies perform poorly on social media data irrespective of it being monolingual or a mixture of languages (Solorio and Liu, 2008; Çetinoğlu et al., 2016; Bhat et al., 2018). Code-mixed data in particular introduces further variation in the morphology and syntax of the language which leads to poor performance of standard NLP tools. Following are a few instances of Hindi-English code-mixed tweets from the corpus:

T1: *"Lagta hai aaj Sri has not spoken to msd"*
Translation: "It looks like Sri has not spoken to MSD today"

T2: *"Lalu Yadav claimed that Yadav quota ke hisab se Umesh Yadav ko ye wkt mil jana chahiye tha"*
Translation: "Lalu Yadav claimed that according to the Yadav quota, Umesh Yadav should have taken a wicket"

In the above two examples we observe how the two languages are mixed in each utterance. Each tweet has tokens from both English and Hindi. T2 in particular shows a problem common to social media data. The token *'wkt'* doesn't correspond to any word. This may be a typo made by the user or simply a shorthand way of writing adopted by many users online. Here 'wkt' could mean "waqta" which means 'time' in Hindi, or "wicket" in the domain of cricket. As we have the context of the whole tweet and world knowledge about Umesh Yadav who is an Indian cricketer, we are able to disambiguate the usage of the token 'wkt', though this may not always be the case.

In this paper, we present a data set of Hindi-English code-mixed tweets labelled with semantic roles. These labels provide us with information of

Proceedings of the 13th Linguistic Annotation Workshop, pages 178–188
Florence, Italy, August 1, 2019. ©2019 Association for Computational Linguistics

the role played by an argument with respect to a verb in a given sentence. We seek to gain semantic information irrespective of the syntactic variation a sentence or an utterance may have. Semantic Role Labelling for code-mixed data will aid in better understanding of these texts and further the research of any understanding based tasks such as information retrieval (Surdeanu et al., 2003; Moschitti et al., 2003), document classification (Bastianelli et al., 2013), questioning answering systems (Shen and Lapata, 2007) and so on.

A Proposition Bank (Propbank) is a corpus of annotated semantic predicate-argument labels (Palmer et al., 2005). This is done with the help of verb frame files and the Proposition Bank tagset. The frame files contain the semantic roles needed for each verb and all the possible context variations of each verb (sense of the verb). To annotate, one must first identify the 'sense id' (Roleset id) of the verb present according to its usage, and then mark the corresponding labels present in its frame file. We follow exactly this process for the manual annotation of our corpus.

The structure of this paper is as follows. Section 2 talks about relevant work in the domains of Semantic Role Labelling and code-mixed data. We discuss our annotation scheme in section 3. In section 4, we propose a baseline rule based system for manual annotation of the data using dependency label information. Section 5 talks about the results and working of our baseline system. We analyse cases of high errors in classification and explore reasons for the same. In Section 6 we shed light on future scope and conclude the paper.

2 Background and Related Work

The release of large corpora with semantic annotations like the FrameNet (Lowe, 1997; Baker et al., 1998) and Propbank (Kingsbury and Palmer, 2002) have enabled the training and testing of classifiers for automated annotation models. Gildea and Jurafsky (2002) initiated the work on 2001 release of the English Propbank with statistical classifiers and linguistic features. Since then, Propbanks have been created for different languages (Xue and Palmer, 2009; Palmer et al., 2008; Bhatt et al., 2009; Duran and Aluísio, 2012) and several advances have been made towards automating the process of Semantic Role Labelling (Punyakanok et al., 2008; Kshirsagar et al., 2015) using neural networks (FitzGerald et al., 2015; Zhou and Xu, 2015), deep learning methods (He et al., 2018b; Tan et al., 2018), joint prediction of predicates and its arguments (Toutanova et al., 2008; He et al., 2018a; Swayamdipta et al., 2018).

Bali et. al (2014) analysed social media, Facebook in particular, and looking at the extent of Hindi-English code-mixed data available online, emphasise the need to develop NLP tools for code-mixed social media data. Vyas et al.(2014) worked on building a POS tagger for Hindi-English code-mixed data and noted the difficulty posed by transliteration of Hindi tokens onto roman script. Barman et al. (2014) addressed the problem of language identification on Bengali-Hindi-English Facebook comments. Sharma et al. (2016) built a shallow parsing pipeline for Hindi-English code-mixed data. Gupta et al. (2014) introduced the concept of Mixed-Script Information Retrieval and the problems posed by transliterated content such as spelling variations etc. There has been a surge of data set creation for code-mixed data (Bhat et al., 2017; Gupta et al., 2016) and application based tools such as question classification (Raghavi et al., 2015), named-entity recognition (Singh et al., 2018), sentiment analysis (Prabhu et al., 2016; Ghosh et al., 2017) and so on.

3 Data Creation

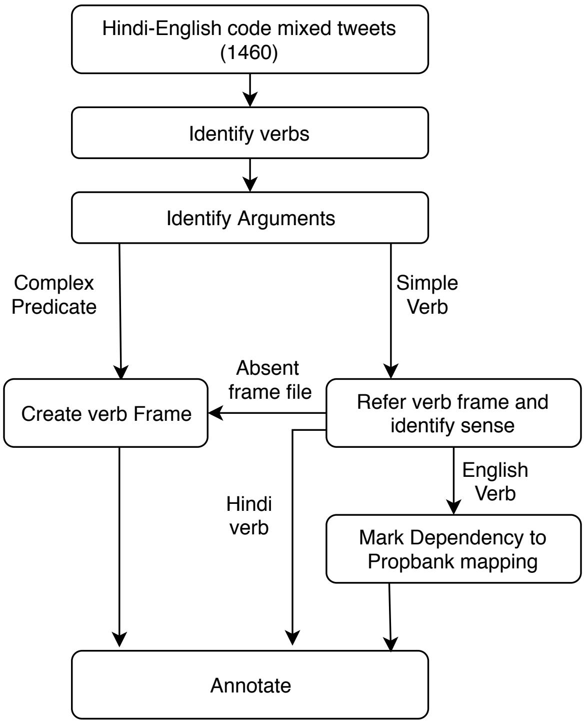

Figure 1: Data Creation workflow for gold annotation of the data

We built our corpus on syntactic information obtained from dependency labels. This allows us to annotate explicitly on the syntactic tree which enables consistency between Propbank structure and dependency structure. Dependency labels provide us with rich syntactic-semantic relations which facilitates mapping between dependency labels and Propbank labels. This would largely reduce annotation effort (Vaidya et al., 2011). We explore this in the working of our baseline model (Section 4). We present a Hindi-English code-mixed Twitter data set comprising 1460 tweets labelled with semantic roles according to the Hindi Propbank tagset. We use the corpus used by (Bhat et al., 2018) in which tweets are labelled with Paninian Dependency labels. Our corpus consists of simple verb constructions, in both Hindi and English, and also complex predicates which have been dealt with separately. These can be within the same language or across the two languages. Figure 1 shows the workflow for the gold annotation of the data.

3.1 Tagset

Label	Description
ARGA	Causer
ARG0	Agent or Experiencer or Doer
ARG1	Theme or Patient
ARG2	Benificiary
ARG2_ATTR	Attribute or Quality
ARG2_LOC	Physical Location
ARG2_GOL	Destination or Goal
ARG2_SOU	Source
ARG3	Instrument
ARGM_DIR	Direction
ARGM_LOC	Location
ARGM_MNR	Manner
ARGM_EXT	Extent or Comparison
ARGM_TMP	Temporal
ARGM_REC	Reciprocal
ARGM_PRP	Purpose
ARGM_CAU	Cause or Reason
ARGM_DIS	Discourse
ARGM_ADV	Adverb
ARGM_NEG	Negative
ARGM_PRX	Complex Predicate

Table 1: Hindi PropBank Tagset

The Propbank adds an additional layer of semantic information on top of the syntactic information present. The Hindi Propbank was built as a part of the "multi-representational and multi-layered" resource creation project for Hindi and Urdu (Bhatt et al., 2009) aimed at simultaneous development of the Propbank, Dependency Treebank and Phrase Structure Treebank. The Hindi Propbank is built on dependency structures unlike Propbanks for other languages such as English, Chinese, Arabic which are built on phrase structure trees (Kingsbury and Palmer, 2002). As we also use dependency structures to annotate Hindi-English code-mixed data, we use the Hindi Propbank tag set (see Table 1) (Palmer et al., 2005) to annotate our data and co-relate the dependency labels with semantic labels.

3.2 Frame File Creation

Frame files are used as guidance for Propbank annotation. Frame file creation is done in two steps:

1. A human expert builds a 'frame file' which marks all the arguments a verb may take across its syntactic variations, depending on the context of its usage.

2. This frame file is used to annotate roles for any occurrence of the said verb to maintain consistency.

Bonial et al (2014) present a lexicon of frame-sets for English Propbank annotation. Vaidya et al. (2013) present Hindi Propbank frame files for simple verb constructions as well as for nominal-verb constructions. As Hindi Propbank is built on syntactic information from Dependency Treebank and we build our model on dependency labelled Hindi-English code-mixed data, we use these frame files extensively for annotation of our corpus. We also refer to the English frame files to label the roles for simple English verbs in the corpus.

Frame file for *baca*	
Roleset id: *baca.01*: to remain	
ARG1	Thing left
Roleset id: *baca.02*: to avoid	
ARG0	person avoiding
ARG1	Thing avoided

Table 2: Frame file for the hindi verb *'baca'*. (Vaidya et al., 2013)

Table 2 shows a frame file for the Hindi verb *'baca'*. The rolesets in the frame file give us the senses of the predicate and the different arguments it may take depending on the context in which it is used. In certain cases, we had to create new frame files for novel occurrences of verbs and absence of the relevant frame file. We also created frame files for inter-language complex predicate formations and noted the dependency label to Propbank label mapping.

3.2.1 Absent Verbs

Existing frame files for both Propbanks - Hindi, and English - have been created keeping formal data sets in mind, such as news articles. Hence, the verbs and the senses of the verbs covered, don't necessarily represent all domains. Social media in particular allows its users to use colloquial terms and usage of predicates, some of which have not been taken care of by the existing frame files. To overcome this, we create the gold frame files for **14** such unique predicates in our corpus. (One such example is the verb 'born' shown in Table 3) Some of these include verbs for which a specific sense is not defined. For example, the English verb 'click' in the context of clicking pictures.

Frame file for *born*	
Roleset id: *born.01*: Brought to life by birth	
ARG1	Entity born

Table 3: Frame file created for the English verb *'born'*. There were 6 instances of this predicate in our corpus.

3.2.2 Complex Predicates

Complex Predicates (CP), also known as 'Light verb constructions' or 'Conjunct Verb Constructions' are seen in both Hindi and English (Butt, 2010). Ahmed et al.(2012) classified the complex predicates present in Hindi into 3 categories: noun-verb constructions, verb - verb constructions and causatives.

Hindi	209
English	21
Intra-language CP	230
Code-mixed CP	232
Total	462

Table 4: Distribution of unique Complex Predicates in the corpus

These constructions occur frequently in our corpus as well. There has been emphasis on the creation of lexical resources for annotation of complex predicates for English (Hwang et al., 2010) and Hindi (Vaidya et al., 2013) in the form of frame files. In our corpus, we observe complex predicate formations within the same language (intra-language) as well as between the two languages (inter-language or code-mixed). We have **462** unique complex predicates in our corpus. Table 4 gives the distribution of these in our data.

Most of these complex predicates are noun-verb constructions, also known as light verb constructions. Light verbs in Hindi are highly productive and can entirely change the meaning of the predicate. For instance, *'hona'* (to be) and *'karna'* (to do) are two Hindi light verbs. When used with an English noun, say 'save', they give rise to two different complex predicates with distinct meanings and structures: *'Save hona'* means to be saved and *'save karna'* would imply the act of saving something. Hence, we cannot leverage frame files from either language to obtain the argument structure for such constructions and thus built new frame-files for each unique combination encountered. An example from the corpus is as follows:

T3: *"Me in logon ko apny crush ki picture send tw kar dun but but but I cant trust them"*
Translation: "I can send my crush's picture to these people, but I can't trust them"

Frame file for *send_karna*	
Roleset id: *send_karna.01*: To Give	
ARG0	Entity sending (Sender)
ARG1	Entity sent
ARG2	Entity sent to

Table 5: Frame file created for the Complex Predicate *send karna*.

The complex predicate construction observed here (T3) is 'send_karna', which is an inter-language, or code-mixed predicate. We created a frame file (Table 5) for the same which helps us to annotate this predicate for subsequent occurrences in the corpus. The given sentence would be labelled for *'send_karna'* as follows:

T4: *"(Me)*[ARG0]*(in logon ko)* [ARG2]*(apny crush ki picture)*[ARG1]*send tw kar dun but but but I cant trust them"*

Translation: "(I)[ARG0] can send (my crush's picture)[ARG1] to (these people)[ARG2], but I can't trust them"

3.3 Annotation

The annotation process is done in a series of steps as described in Figure 1. The first step is to identify all the verbs present in the sentence. We will use the following sentence as an example:

*"Yar **end karo** match I have to **sleep**"*

Translation: Hey, end the match, I have to sleep.

Here we can detect two verb constructions. One is a complex predicate *'end_karna'* and the other is a simple English verb construction for *'sleep'*. We refer to the frame files for both to identify the arguments in the given sentence. Since *'end_karna'* is a complex predicate containing an English nominal and a Hindi light verb, we create its frame file (Table 6). These constructions are easily detectable with the help of special label pof or "part-of" used in the Dependency Treebank. The second verb in the sentence is *'sleep'* for which the frame file is already present (Table 7 (Bonial et al., 2014)).

Frame file for *end_karna*	
Roleset id: *end_karna.01*: To Stop	
ARG0	Entity ending (Ender)
ARG1	Entity ended

Table 6: Frame file created for the Complex Predicate *'end_karna'* as discussed in Section 3.2.2.

Frame file for *sleep*	
Roleset id: *sleep.01*: To Sleep, Slumber	
ARG0	Sleeper
ARG1	Cognate entity
Roleset id: *sleep.02*: Engage in sexual relations	
ARG0	Agentive partner
ARG1	Prepositional Partner

Table 7: Frame file for the simple English verb *'sleep'*.

The token for complex predicate is marked with the label 'ARGM_PRX' according to the Propbank tagset. In the frame file for the verb 'sleep', given in Table 7, we can see possible rolesets or senses the predicate can take. Looking at the context in our sentence, we choose 'Roleset id: sleep.01'. With the help of frame files, we are able to identify and annotate the numbered arguments of the predicates. Next, we label the modifier arguments as described in Table 1.

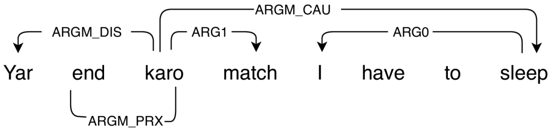

Figure 2: Sentence marked with Propbank labels

In the given sentence, the token *"Yar"* is a term used frequently in colloquial Hindi. It is used to refer to someone or call someone informally. The right label for it is ARGM_DIS (Discourse, according to Table 1). The reason for 'ending' the match was the action of 'sleeping'. Hence, we mark it with ARGM_CAU (Cause). Figure 2 shows the final sentence annotated with all the semantic roles. Since we are using code-mixed tweets which are annotated with Hindi dependency labels (Bhat et al., 2018), we also note the mappings from dependency labels to Propbank labels for all verb occurrences in the corpus. This mapping would help in automatic annotation of semantic roles of verbs from their syntactic dependents (Vaidya et al., 2011).

Total tokens	20, 949
Unique Hindi Simple Verbs	613
Unique English Simple Verbs	512
Complex Predicates	622

Table 8: Data Distribution 3.2.2

Table 8 shows the statistics of the corpus after annotation of 1460 tweets in the Hindi-English code-mixed tweets.

3.3.1 Pronoun Dropping

Pronoun dropping refers to the linguistic phenomenon of dropping or omitting pronouns wherein it is inferable from prior discourse context. It is observed widely across languages though the conditions may vary from language to language. Bhatia et al.(2010) emphasise the motivation and importance of introducing empty categories in the Hindi Dependency Treebank. This doesn't include empty categories for pronoun dropping but includes empty categories for dropped nouns, conjunctions, verbs etc. Empty categories were introduced in the Hindi Propbank to include core arguments missing from the predicate-argument structure after addition of the empty categories in the Hindi Dependency Treebank (Vaidya et al., 2012).

T5: *"Tore my calendar kyunki woh khana nai laya"*

Translation: "(I) tore my calendar because he/she didn't bring food."

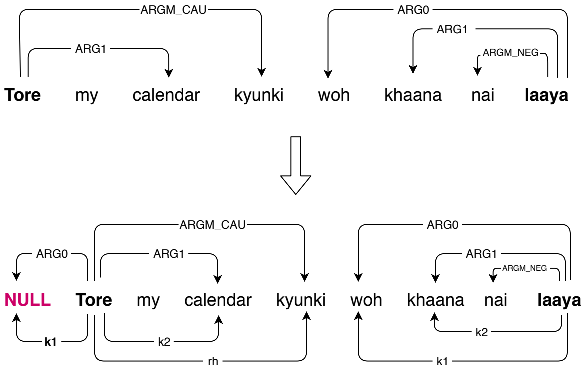

Figure 3: Tweet T5 marked with Propbank labels before and after 'NULL' insertion to account for pro-drop along with dependency relation labels.

Although English is not a pro-drop language, pronoun dropping is observed largely in Hindi-English code-mixed data. The sentence above (T5) is such an example from the corpus. We incorporate this in our data by inserting 'NULL' arguments and labelling them with Propbank labels - ARG0, ARG1, ARG2, as appropriate. Table 9 shows the frame file for the verb 'tear'. Figure 3 shows the semantic roles associated with tweet T5 before and after the empty category insertion to account for pronoun dropping.

Frame file for *tear*	
Roleset id: *tear.01*: To pull apart	
ARG0	Tearer (**dmrel**: k1)
ARG1	Thing torn (drel: k2)

Table 9: Part of the Frame file for the simple English verb *'tear'*. This is the relevant roleset chosen according to the Tweet above (T5). We note the dependency role (drel) associated with the Propbank labels. In case of an empty category insertion, we assign a dummy dependency relation label ('dmrel') as appropriate.

3.3.2 Special Constructions

Code-mixed language refers to the usage of linguistic units of one language in a sentence of another language. One fairly common preliminary step while annotating code-mixed data is Language Identification (Vyas et al., 2014; Sharma et al., 2016). The tokens present in the corpus are marked 'hi', for Hindi, or 'en', for English, or 'ne' for Named Entities. This assumes that code-mixing doesn't occur at sub-lexical levels. However, in our corpus, we came across a few cases where new lexical items are formed by mixing the two languages and modifying the morphology of the individual languages. One way of doing this is to add affixes from one language to a word of the other language. These constructions are used widely in day to day usage. We treat these cases as 'Special Constructions'.

Frame file for *beztify*	
Roleset id: *beztify.01*: To insult	
ARG0	Entity insulting someone
ARG1	Entity insulted

Table 10: Frame file for the *'hinglish'* word *"beztify"*.

When these words of morphological modification play the role of predicates, we need to assign arguments and semantic roles accordingly. To deal with this, we create frame files for such cases. Table 10 shows the frame file for one such construction from our corpus - ***beztify***.

'bezti' is a Hindi noun which translates to 'insult' in English. The speaker here uses the English suffix "-fy" to use the word as a verb, thus making it *"beztify"* which translates to "to insult someone" in English.

4 Rule-based Approach

Semantic Role Labelling adds a layer of semantic information on top of the syntactic information. We use Paninian dependency labelled (karaka relations) Hindi-English code-mixed data (Bhat et al., 2018) for creating our corpus and labelling the data. Vaidya et al (2011) analysed the relation between dependency labels and Propbank labels for Hindi. They also proposed mappings between Hindi dependency labels to Propbank labels as shown in Table 11 and Table 12 for numbered arguments and some modifier arguments respectively.

Research shows that English Propbank data is similar to English Dependency Treebank labelled with Paninan dependency labels. (Vaidya et al., 2009). We use these mappings (Table 11, Table 12) to create a rule based model for automatic annotation of semantic roles.

Dependency label	Propbank label
k1 (karta); k4a (experiencer)	ARG0
k2 (karma)	ARG1
k4 (beneficiary)	ARG2
k1s (attribute)	ARG2_ATTR
k5 (source)	ARG2_SOU
k2p (goal)	ARG2_GOL
k3 (instrument)	ARG3

Table 11: Mappings from Dependency label to Propbank Numbered arguments

We first identify the predicates present in the sentence. Simple verb constructions are easily identified by their part of speech tag ('VM') and complex predicates are detected by the dependency label 'pof' as mentioned in Section 3.2.2.

The labelling is done in two steps. The first step is **Argument Identification**. Here, our model labels all the tokens in the sentence as "Argument" or "Not an Argument" with the help of the dependency tree structure. To achieve this, we mark all direct dependents of the identified predicates as their Arguments barring those tokens which are marked as auxiliary verbs, post-positions, symbols (emojis in social media text) or those which show coordination or subordination (drel: *'ccof'*). There can be certain cases in social media text where emojis may act as arguments of a predicate. However, we focus only on lexical items for the time being and plan to incorporate this as a part of our future work.

Dependency label	Propbank label
sent-adv (epistemic adv)	ARGM_ADV
rh (cause/reason)	ARGM_CAU
rd (direction)	ARGM_DIR
rad (discourse)	ARGM_DIS
k7p (location)	ARGM_LOC
adv (manner adv)	ARGM_MNR
rt (purpose)	ARGM_PRP
k7t (time)	ARGM_TMP

Table 12: Mappings from Dependency label to Propbank Modifier labels.

The second step is **Argument Classification** wherein we assign the identified arguments with Propbank labels according to the aforementioned mappings. We add more rules to the mappings for modifier labels as mentioned in Table 13. For the rare cases where no such mapping has been proposed, we train the model to label arguments as the most frequently occurring corresponding label in the gold data set.

Dependency label	Propbank label
k7a (according to)	ARGM_ADV
lwg__neg (negation)	ARGM_NEG
k*u (similarity/comparison)	ARGM_EXT

Table 13: Additional mappings from Dependency label to Propbank Modifier labels.

5 Results and Analysis

We obtain an overall accuracy of **96.74%** (overall F1 score of 95.41) for Argument Identification and **73.93%** for Argument Classification. The precision, recall and F1 scores for Argument Identification are given in Table 14. We also compute our scores separately for Numbered arguments and Modifier arguments.

	Dist.	P	R	F1
Overall	100.00	93.22	97.69	95.41
Numbered	61.09	98.81	90.22	94.32
Modifier	38.91	79.50	94.41	87.5

Table 14: Accuracy scores achieved for identification of Numbered and Modifier arguments by our rule based model along with their distribution in the data set.

Figure 4 shows us a sentence from the corpus where a token is labelled as [ARG0] by our model whereas the gold label is [ARG1]. This is a very common error seen across the corpus.

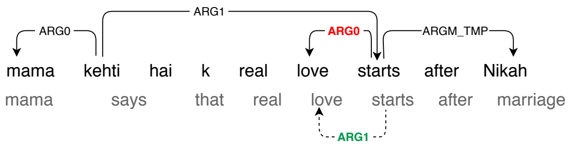

Figure 4: Tweet showing mis-classification between 'ARG0' (given by model, solid line) and 'ARG1' (Gold label, dotted line)

In the example shown, the dependency label given to the token *'love'* is 'k1'. Here, 'love' isn't really the agent of the verb 'start'. The Propbank label [ARG0] denotes the agent of the verb, the argument which causes the action, whereas [ARG1] denotes the argument which is affected or changed by the action. Paninian dependency labels don't account into unaccusativity and hence, k1 maps to both [ARG0] and [ARG1], subject to context (Vaidya et al., 2009, 2011).

Label	Dist.	P	R	F1
ARG0	15.65	81.79	93.83	65.21
ARG1	33.14	92.61	48.56	63.71
ARG2	4.62	75.91	31.04	44.06
ARG2_ATTR	5.63	76.95	86.76	81.56
ARG2_GOL	0.54	90.90	25.64	40.0
ARG2_SOU	0.57	80.00	68.29	73.68
ARG3	0.17	81.81	75.00	78.26
ARGM_DIR	0.07	50.0	80.0	61.53
ARGM_LOC	3.68	50.77	98.50	67.0
ARGM_MNR	7.83	51.52	89.26	65.33
ARGM_EXT	0.28	50.0	95.0	65.51
ARGM_TMP	8.19	97.61	89.73	93.51
ARGM_PRP	1.28	88.77	93.54	91.09
ARGM_CAU	1.71	96.19	81.45	88.21
ARGM_DIS	2.44	98.23	94.35	96.25
ARGM_ADV	0.43	72.31	82.92	77.25
ARGM_NEG	4.36	92.85	94.62	93.73
ARGM_PRX	8.58	97.47	99.35	98.41

Table 15: Precision, Recall and F-scores achieved for all labels with our rule based model. Also shows overall distribution of the labels in our data set.

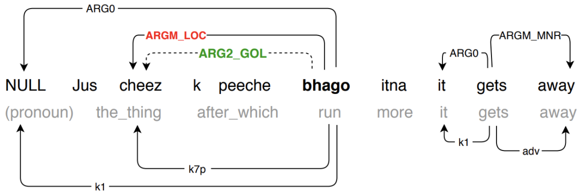

Figure 5: Tweet showing 'ARG2_GOL' (Gold label, dotted line) mis-labelled as 'ARGM_LOC' (given by model, solid line), and the dependency labels of the tokens.

The precision, recall and F1 scores for the various labels obtained in the Argument Classification step are given in Table 15. We see that [ARG2] and [ARG2_GOL] have a significantly low F1 score, although the precision values are decent. ARG2 is most commonly mis-labelled as ARG2_ATTR in our data which results in the low recall score.

Frame file for *BAga*	
Roleset id:*BAga.02*: To run towards something	
ARG0	entity running (drel: k1)
ARG1	destination (drel: k2p)

Table 16: Part of the Frame file for the simple Hindi verb *'BAga'*. This is the relevant roleset chosen according to the Tweet in figure 5.

Figure 5 shows an example where a token is labelled as [ARGM_LOC] because of the dependency label 'k7p' (Table 12). However, according to the frame file of the verb "Bhaaga" (to run) given in Table 16, the token must be given the label [ARG2_GOL]. We also do a NULL insertion for the dropped pronoun in this tweet as described in section 3.3.1. The mis-classification for [ARG2_GOL] occurs largely due to the ambiguity between the dependency labels 'k2p' and 'k7p' which then lowers the precision value of [ARGM_LOC] as well.

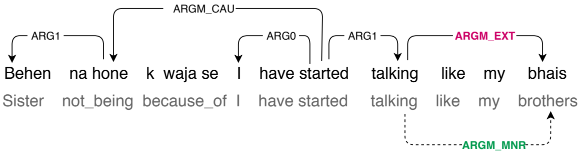

Figure 6: Tweet showing mis-classification between 'ARGM_EXT' (given by model, solid line) and 'ARGM_MNR' (Gold label, dotted line)

Another common error observed is between [ARGM_EXT] and [ARGM_MNR] as seen in Figure 6. The dependency label given to the token *'bhais'* (brothers) is 'k1u' which is used to mark similarities or comparisons. The Propbank label for comparisons is usually [ARGM_EXT]. However, here we are comparing the manner of talking of the speaker with his/her brother(s), and hence the appropriate Propbank label would be [ARGM_MNR]. A similar case can be seen for mis-classification between [ARGM_MNR] and [ARGM_ADV] labels. The former is meant for describing the manner in which the action is carried out and the latter describes the action. Sometimes, the model isn't able to distinguish between them. These cases explain the lower accuracy scores for the labels - 'ARGM_EXT' , 'ARGM_MNR' and 'ARGM_ADV'.

6 Conclusion and Future Work

We present a data set of Hindi-English code-mixed data marked with semantic roles. We take into account nuances of both languages such as complex predicate constructions, pronoun dropping and address issues specific to social media data such as typos, colloquial word usage, as well. We also present a baseline model which maps the correlation between dependency labels and Propbank labels as has been observed with both languages separately and note that the co-relation remains largely consistent. This will aid in faster annotation of such data henceforth. The data set is available online[1].

We plan to further expand this data set and try learning based approaches for code-mixed Semantic Role Labelling and also analyse and compare them with models for monolingual data sets.

References

Tafseer Ahmed, Miriam Butt, Annette Hautli, and Sebastian Sulger. 2012. A reference dependency bank for analyzing complex predicates. In *LREC*.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics.

Kalika Bali, Jatin Sharma, Monojit Choudhury, and Yogarshi Vyas. 2014. " i am borrowing ya mixing?" an analysis of english-hindi code mixing in facebook. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 116–126.

Utsab Barman, Amitava Das, Joachim Wagner, and Jennifer Foster. 2014. Code mixing: A challenge for language identification in the language of social media. In *Proceedings of the first workshop on computational approaches to code switching*, pages 13–23.

Emanuele Bastianelli, Giuseppe Castellucci, Danilo Croce, and Roberto Basili. 2013. Textual inference and meaning representation in human robot interaction. In *Proceedings of the Joint Symposium on Semantic Processing. Textual Inference and Structures in Corpora*, pages 65–69.

Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Manish Shrivastava, and Dipti Misra Sharma. 2017. Joining hands: Exploiting monolingual treebanks for parsing of code-mixing data. *arXiv preprint arXiv:1703.10772*.

Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Manish Shrivastava, and Dipti Misra Sharma. 2018. Universal dependency parsing for hindi-english codeswitching. *arXiv preprint arXiv:1804.05868*.

Archna Bhatia, Rajesh Bhatt, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Misra Sharma, Michael Tepper, Ashwini Vaidya, and Fei Xia. 2010. Empty categories in a hindi treebank. In *LREC*.

Rajesh Bhatt, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Sharma, and Fei Xia. 2009. A multi-representational and multi-layered treebank for hindi/urdu. In *Proceedings of the Third Linguistic Annotation Workshop (LAW III)*, pages 186–189.

Claire Bonial, Julia Bonn, Kathryn Conger, Jena D Hwang, and Martha Palmer. 2014. Propbank: Semantics of new predicate types. In *LREC*, pages 3013–3019.

Miriam Butt. 2010. The light verb jungle: Still hacking away. *Complex predicates in cross-linguistic perspective*, pages 48–78.

Özlem Çetinoğlu, Sarah Schulz, and Ngoc Thang Vu. 2016. Challenges of computational processing of code-switching. *arXiv preprint arXiv:1610.02213*.

Magali Sanches Duran and Sandra Maria Aluísio. 2012. Propbank-br: a brazilian treebank annotated with semantic role labels. In *LREC*, pages 1862–1867.

Nicholas FitzGerald, Oscar Täckström, Kuzman Ganchev, and Dipanjan Das. 2015. Semantic role labeling with neural network factors. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 960–970.

Souvick Ghosh, Satanu Ghosh, and Dipankar Das. 2017. Sentiment identification in code-mixed social media text. *arXiv preprint arXiv:1707.01184*.

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

Parth Gupta, Kalika Bali, Rafael E Banchs, Monojit Choudhury, and Paolo Rosso. 2014. Query expansion for mixed-script information retrieval. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 677–686. ACM.

Sakshi Gupta, Piyush Bansal, and Radhika Mamidi. 2016. Resource creation for hindi-english code mixed social media text. In *The 4th International Workshop on Natural Language Processing for Social Media in the 25th International Joint Conference on Artificial Intelligence*.

Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018a. Jointly predicting predicates and arguments in neural semantic role labeling. *arXiv preprint arXiv:1805.04787*.

[1] https://github.com/riyapal/Hi-En-SRL

Shexia He, Zuchao Li, Hai Zhao, and Hongxiao Bai. 2018b. Syntax for semantic role labeling, to be, or not to be. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2061–2071.

Jena D Hwang, Archna Bhatia, Clare Bonial, Aous Mansouri, Ashwini Vaidya, Nianwen Xue, and Martha Palmer. 2010. Propbank annotation of multilingual light verb constructions. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 82–90. Association for Computational Linguistics.

Paul Kingsbury and Martha Palmer. 2002. From treebank to propbank. In *LREC*, pages 1989–1993. Citeseer.

Meghana Kshirsagar, Sam Thomson, Nathan Schneider, Jaime Carbonell, Noah A Smith, and Chris Dyer. 2015. Frame-semantic role labeling with heterogeneous annotations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 218–224.

John B Lowe. 1997. A frame-semantic approach to semantic annotation. *Tagging Text with Lexical Semantics: Why, What, and How?*

Alessandro Moschitti, Paul Morarescu, Sanda M Harabagiu, et al. 2003. Open domain information extraction via automatic semantic labeling. In *FLAIRS conference*, volume 3, pages 397–401.

Carol Myers-Scotton. 1993. Dueling languages: Grammatical structure in code-switching. claredon.

Martha Palmer, Olga Babko-Malaya, Ann Bies, Mona T Diab, Mohamed Maamouri, Aous Mansouri, and Wajdi Zaghouani. 2008. A pilot arabic propbank. In *LREC*.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.

Ameya Prabhu, Aditya Joshi, Manish Shrivastava, and Vasudeva Varma. 2016. Towards sub-word level compositions for sentiment analysis of hindi-english code mixed text. *arXiv preprint arXiv:1611.00472*.

Vasin Punyakanok, Dan Roth, and Wen-tau Yih. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2):257–287.

Khyathi Chandu Raghavi, Manoj Kumar Chinnakotla, and Manish Shrivastava. 2015. Answer ka type kya he?: Learning to classify questions in code-mixed language. In *Proceedings of the 24th International Conference on World Wide Web*, pages 853–858. ACM.

Arnav Sharma, Sakshi Gupta, Raveesh Motlani, Piyush Bansal, Manish Srivastava, Radhika Mamidi, and Dipti M Sharma. 2016. Shallow parsing pipeline for hindi-english code-mixed social media text. *arXiv preprint arXiv:1604.03136*.

Dan Shen and Mirella Lapata. 2007. Using semantic roles to improve question answering. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*.

Vinay Singh, Deepanshu Vijay, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. Named entity recognition for hindi-english code-mixed social media text. In *Proceedings of the Seventh Named Entities Workshop*, pages 27–35.

Thamar Solorio and Yang Liu. 2008. Part-of-speech tagging for english-spanish code-switched text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1051–1060. Association for Computational Linguistics.

Mihai Surdeanu, Sanda Harabagiu, John Williams, and Paul Aarseth. 2003. Using predicate-argument structures for information extraction. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*.

Swabha Swayamdipta, Sam Thomson, Kenton Lee, Luke Zettlemoyer, Chris Dyer, and Noah A Smith. 2018. Syntactic scaffolds for semantic structures. *arXiv preprint arXiv:1808.10485*.

Zhixing Tan, Mingxuan Wang, Jun Xie, Yidong Chen, and Xiaodong Shi. 2018. Deep semantic role labeling with self-attention. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Kristina Toutanova, Aria Haghighi, and Christopher D Manning. 2008. A global joint model for semantic role labeling. *Computational Linguistics*, 34(2):161–191.

Ashwini Vaidya, Jinho D Choi, Martha Palmer, and Bhuvana Narasimhan. 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.

Ashwini Vaidya, Jinho D Choi, Martha Palmer, and Bhuvana Narasimhan. 2012. Empty argument insertion in the hindi propbank. In *LREC*, pages 1522–1526.

Ashwini Vaidya, Samar Husain, Prashanth Mannem, and Dipti Misra Sharma. 2009. A karaka based annotation scheme for english. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 41–52. Springer.

Ashwini Vaidya, Martha Palmer, and Bhuvana Narasimhan. 2013. Semantic roles for nominal predicates: Building a lexical resource. In *Proceedings of the 9th Workshop on Multiword Expressions*, pages 126–131.

Yogarshi Vyas, Spandana Gella, Jatin Sharma, Kalika Bali, and Monojit Choudhury. 2014. Pos tagging of english-hindi code-mixed social media content. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 974–979.

Nianwen Xue and Martha Palmer. 2009. Adding semantic roles to the chinese treebank. *Natural Language Engineering*, 15(1):143–172.

Jie Zhou and Wei Xu. 2015. End-to-end learning of semantic role labeling using recurrent neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1127–1137.